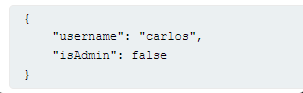
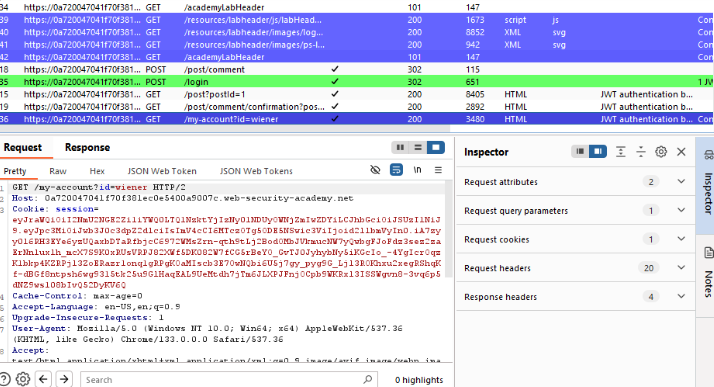
\*\*JWT \*\*

\*\*Lab1: JWT authentication bypass via unverified signature\*\*

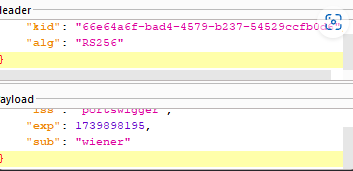
Do lỗi triển khai, máy chủ không xác minh chữ ký của bất kỳ JWT nào mà nó nhận được.



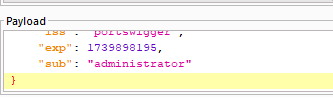
Đăng nhập trước vào tk được cấp



Bạn có thể đọc cookie bằng jwt.io hoặc có thể cài extention trong burp



Ta thấy phần sub là tên người dùng và trang yêu cầu ta vào trang /admin nên ta sẽ thử thay đổi giá trị của nó



Ta đã vào được trang /admin



------------

\*\*LAB2: JWT authentication bypass via flawed signature verification\*\*

The server is insecurely configured to accept unsigned JWTs.

Lab 2 cũng tương tự lab1.Ta login vào tk

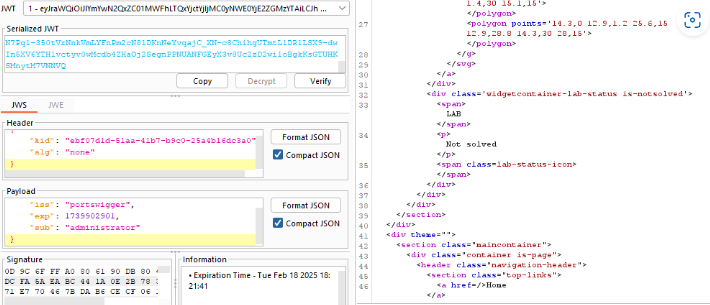
Ta bắt được gói tin như trên



-Tiêu đề JWT chứa một alg tham số. Điều này cho máy chủ biết thuật toán nào đã được sử dụng để ký mã thông báo và do đó, thuật toán nào cần sử dụng khi xác minh chữ ký.

-Điều này về bản chất là sai sót vì máy chủ không có lựa chọn nào khác ngoài việc tin tưởng ngầm vào dữ liệu đầu vào do người dùng kiểm soát từ mã thông báo mà tại thời điểm này vẫn chưa được xác minh

Nên ta sẽ đổi alg thành "none" và thay sub bằng "administrator":



Nhưng sao vẫn ko đc 😢

Khi bạn thay đổi alg thành none thì phần Signature(Chữ ký số dùng để xác thực tính toàn vẹn của token.) không cần thiết và sẽ được bỏ trống

Done!!!!!

------------

LAB3:



Một số thuật toán ký, chẳng hạn như HS256 (HMAC + SHA-256), sử dụng một chuỗi độc lập tùy ý làm khóa bí mật. Giống như mật khẩu, điều quan trọng là bí mật này không dễ đoán hoặc bị kẻ tấn công dùng phương pháp brute-force. Nếu không, chúng có thể tạo JWT với bất kỳ giá trị tiêu đề và tải trọng nào chúng thích, sau đó sử dụng khóa để ký lại mã thông báo bằng chữ ký hợp lệ.

Ta sẽ dùng hashcat để brute-force key:



-a 0: Chỉ định chế độ tấn công (attack mode) là dictionary attack (tấn công bằng từ điển).

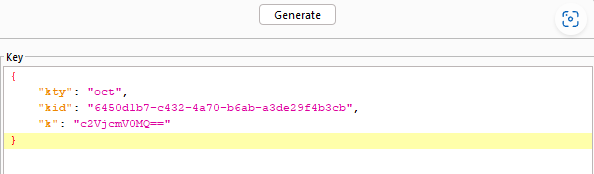
-m 16500: Chỉ định mã băm (hash mode) là JWT (JSON Web Token) – chính xác là phần chữ ký

Sau khi hash ta có được secretkey

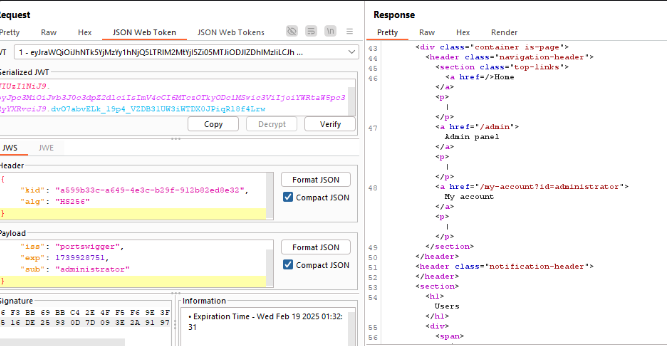
eyJraWQiOiIwZTFiODMwOS0zNjJlLTRhNWYtYWEwZS1hZWExMjcyM2UxZGMiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJwb3J0c3dpZ2dlciIsImV4cCI6MTczOTkwNDg4NCwic3ViIjoid2llbmVyIn0.guUy8s91HcoeoEv7pduM6I2siWllCgeNdvh-1UQl6RI:secret1

Ta mã hóa secret-key bằng base64

Ta vào JWT editor thay giá trị k bằng secret-key vừa được mã hóa



Ta vào lại repeater thay sub và nhấn sign về key vừa được tạo



DONE!!!!!

--------

LAB4:



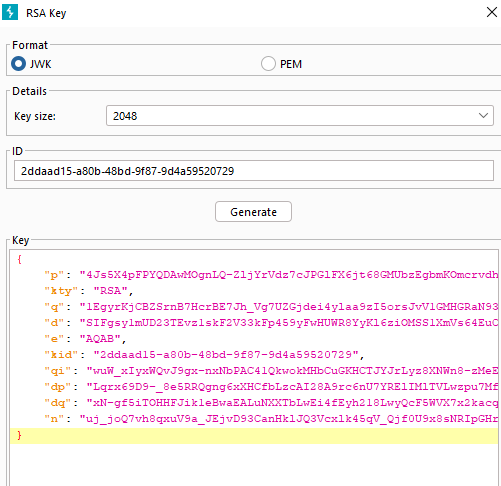
Máy chủ hỗ trợ jwk tham số trong tiêu đề JWT. Đôi khi, điều này được sử dụng để nhúng khóa xác minh chính xác trực tiếp vào mã thông báo. Tuy nhiên, nó không kiểm tra xem khóa được cung cấp có đến từ nguồn đáng tin cậy hay không.

Lý tưởng nhất là máy chủ chỉ nên sử dụng danh sách trắng giới hạn các khóa công khai để xác minh chữ ký JWT. Tuy nhiên, máy chủ cấu hình sai đôi khi sử dụng bất kỳ khóa nào được nhúng trong jwktham số.

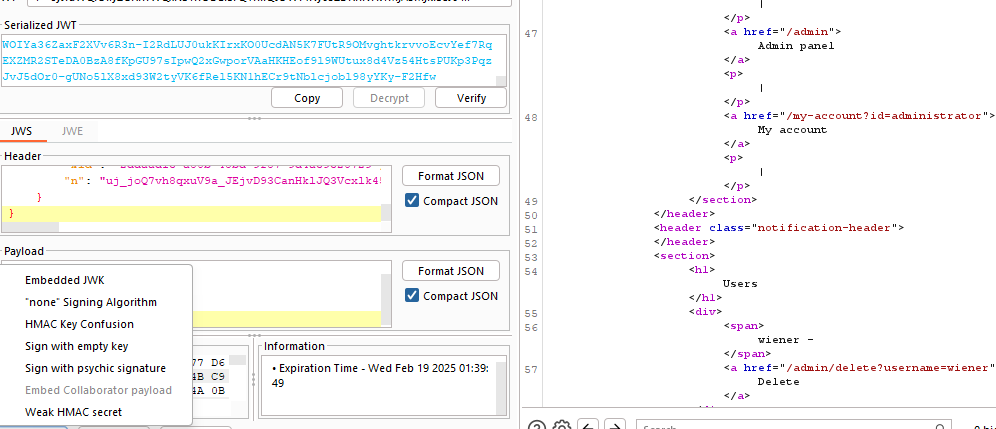
Bạn có thể khai thác hành vi này bằng cách ký JWT đã sửa đổi bằng khóa riêng RSA của riêng bạn, sau đó nhúng khóa công khai phù hợp vào jwk tiêu đề.

Ta thử lại các cách ở trên nhưng không được

Theo hướng dẫn về jwk



Ta thay đổi sub thành “administrator” và attach về key RSA vừa tạo

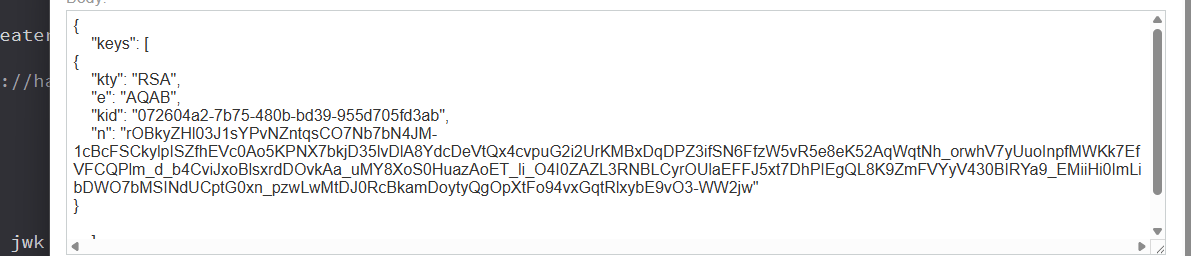


DONE!!!!

LAB5:



Đầu tiên ta tạo khóa RSA mới, sau đó ta copy publickey as jwk và dán theo format key trong trang exploit



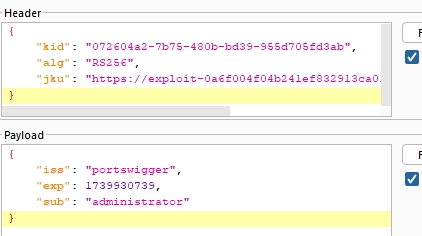
Về lại burp thay kid cũ bằng kid trên trang exploit

Key ID (kid) – dùng để xác định khóa bí mật để giải mã.

Khi thực hiện tấn công CSRF, bạn thường không thể thay đổi JWT trực tiếp, nhưng có thể cố gắng làm giả hoặc gửi request hợp lệ. Nếu kid khác nhau, JWT không thể giải mã đúng.

`

Thêm trong tải “jku”:link exploit



Và



DONE!!

LAB6:



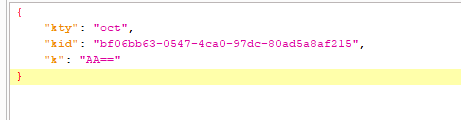
Máy chủ có thể sử dụng nhiều khóa mật mã để ký các loại dữ liệu khác nhau, không chỉ JWT. Vì lý do này, tiêu đề của JWT có thể chứa tham kidsố (ID khóa), giúp máy chủ xác định khóa nào sẽ sử dụng khi xác minh chữ ký.

Máy chủ có thể sử dụng nhiều khóa mật mã để ký các loại dữ liệu khác nhau, không chỉ JWT. Vì lý do này, tiêu đề của JWT có thể chứa tham kidsố (ID khóa), giúp máy chủ xác định khóa nào sẽ sử dụng khi xác minh chữ ký.

Mà theo hưỡng dẫn chúng ta phải vào được /dev/null

Nếu bạn đang sử dụng tiện ích mở rộng JWT Editor, hãy lưu ý rằng tiện ích này không cho phép bạn ký mã thông báo bằng chuỗi rỗng. Tuy nhiên, do lỗi trong tiện ích mở rộng, bạn có thể khắc phục bằng cách sử dụng byte null được mã hóa Base64.

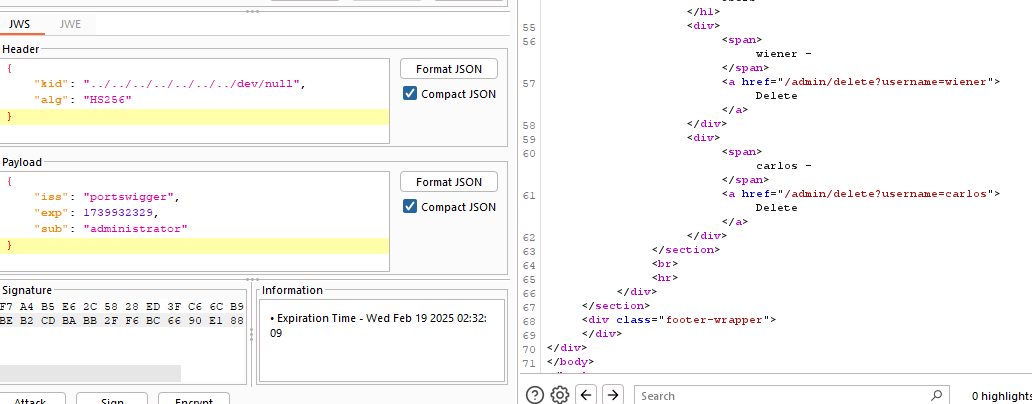
Ta thấy thuật toán HS256 (đối xứng) nên ta tạo một khóa symmestric để kiểm tra xem có bypass được hay không



Khi đặt k = null, một số ứng dụng không kiểm tra kỹ và vẫn cho phép JWT được xác thực.

Điều này có nghĩa là có thể kiểm soát kid mà không bị ràng buộc bởi bất kỳ khóa bí mật nào.

Khi này ta đổi đường dẫn của kid bằng pathtraversal và sub



DONE!!!