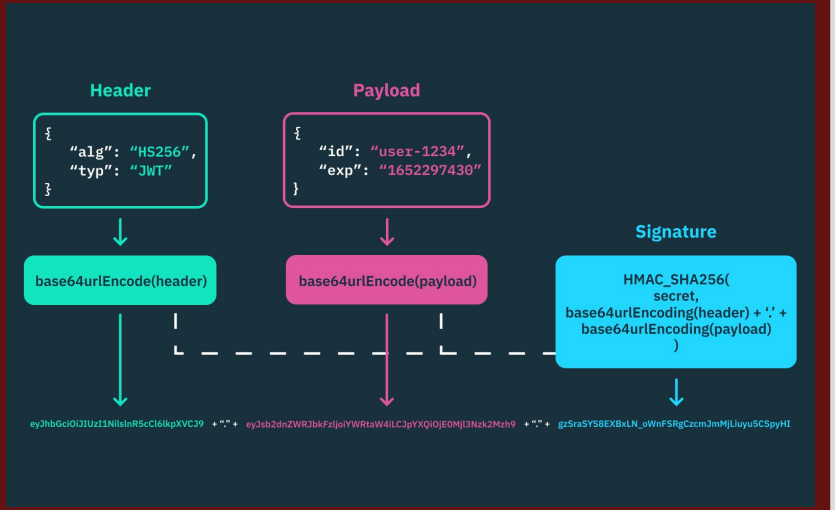
Broken Authentication

Andres Cardoso, Nimreet Gill, James Mata

Challenge Present:

- JWT (JSON Web Token) misconfiguration
- JWT encodes data with a signature that allows a receiver to verify that the data was not modified in transit



The Vulnerability:

- The algorithm used to encode the payload is in the header of the JWT
- Servers may accept the algorithm in the header as valid
- An attacker can forge a token with a more convenient algorithm to use

The Algorithms:

- The most secure JWT algorithm is RS256 which uses a two key system to verify the data is authentic
- This would mean an attacker would have to have access to the servers private key to forge a token
- HS256 only requires one key usually the servers public key.

How to Attack:

- If the server doesn’t mandate the algorithm used for encoding and decoding JWT
- Discover the public key for the server which would be used to decode the JWT
- Create your own JWT signing it with the servers public key
- Intercept a JWT heading to the server replacing it with the forged token and a HS256 header
- The server should accept it as a valid payload

Demonstration:

How to Attack:

- If the server doesn't mandate the algorithm used for encoding and decoding JWT
- Discover the public key for the server which would be used to decode the JWT
- Create your own JWT signing it with the servers public key
- Intercept a JWT heading to the server replacing it with the forged token and a HS256 header
- The server should accept it as a valid payload

Who's been affected?

How to prevent:

- Never assume the algorithm provided in the header hasn't been tampered with
- Always force the use of RS256 for critical data transfers
- Never leave the key used to sign HS256 JWTs easily available if it will be used for anything
- Update to the newest versions of JWT authentication as many libraries now force developers to follow these rules