JWT Authentication and Authorization

Ilya Averyanov

2022

JWT
JSON Web Token

JSON Web Token (**JWT**) is a compact, URL-safe means of representing claims to be transferred between two parties. The **claims** in a JWT are encoded as a **JSON object** that is used as the payload of a JSON Web Signature (JWS) structure or as the plaintext of a JSON Web Encryption (JWE) structure, enabling the claims to be digitally signed or integrity protected with a Message Authentication Code (MAC) and/or encrypted.

JWT
JSON Web Token

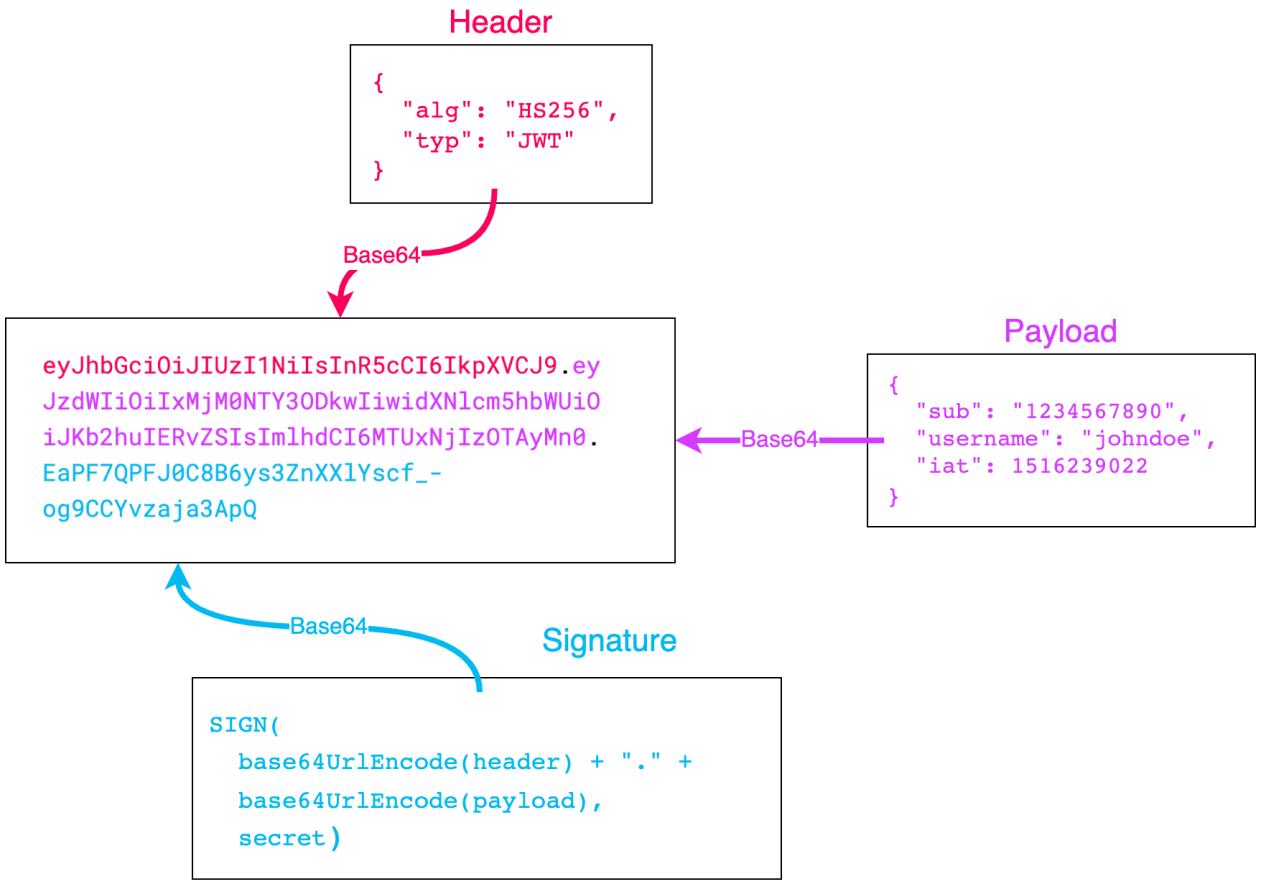
- https://datatracker.ietf.org/doc/html/rfc7519.
- *Claims* are actually predefined or custom fields.
- Generally, JSON Web Signature (JWS) structure is implied for JWTs.

JWT
JWT Structure

- Base64 encoded JSON header with signature algorithm and other signature params.
- Base64 encoded JSON payload with claims.
- Base64 signature of header and payload.

The parts are conactenated with the period

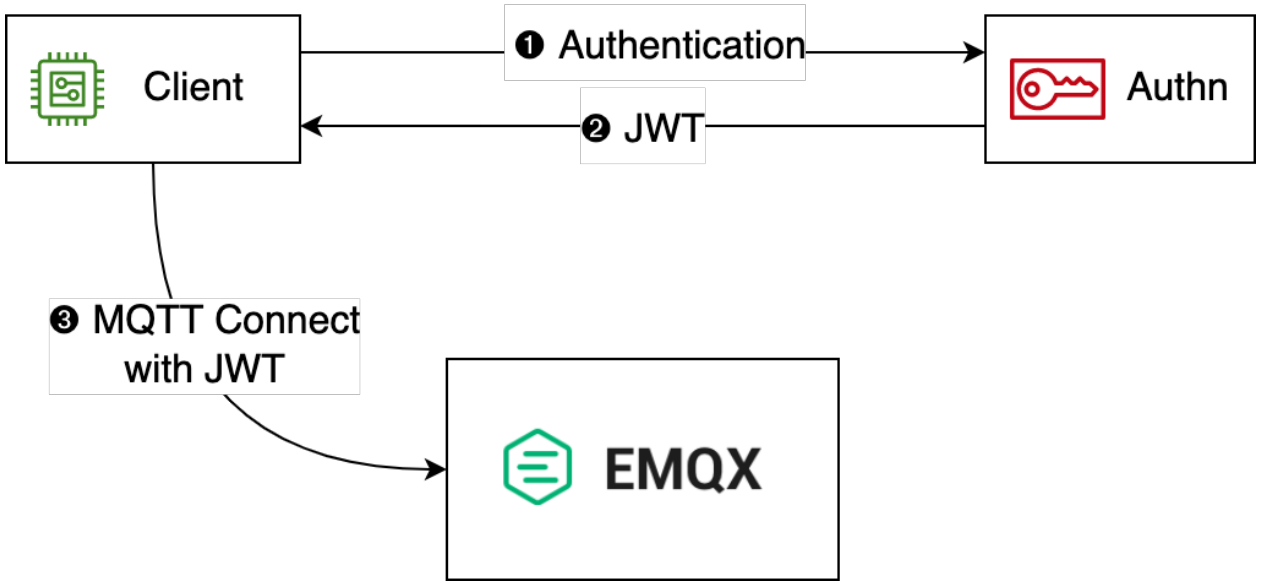
JWT
JWT Structure



JWT Authentication
EMQX

- A client submits its credentials to an authentication server and exchanges them to a JWT.
- Connects to EMQX carrying JWT as username or password.
- EMQX verifies the signature and, optionally, JWT claims. If the checks pass, the authentication succeeds.

JWT Authentication
EMQX



JWT Authentication
Advantages

- JWTs are a standard for service communication, there are many production-ready authentication servers supporting JWT exchange.
- JWTs are small and self-contained, so they are easy to manipulate.
- Authentication with JWT decouples credential management from EMQX. Users may choose their own mechanisms of provisioning clients with JWTs.
- Accessing EMQX with JWTs is lightweight. EMQX does not have to access any external resources to verify a JWT.

JWT Authentication
Caveats

- JWTs are signed, not encrypted. They should not contain sensitive information.
- JWTs cannot be revoked easily. So if stolen, JWTs can be used by an intruder until expiration. Therefore, JWT lifetimes should be chosen wisely.
- Use of SSL connection is strongly recommended to avoid JWT compromisation.
- If HMAC signatures are used, the secret must be stored securely by the accessed service (EMQX).

JWT Authentication
Enabling in EMQX

Enable `emqx_auth_jwt` plugin in `data/loaded_plugins`:

```
...
{emqx_auth_jwt, true}.
```

Configure `emqx_auth_jwt` in `etc/plugins/emqx_auth_jwt.conf`:

```
auth.jwt.secret = emqxsecret
#auth.jwt.pubkey = etc/certs/jwt_public_key.pem
#auth.jwt.jwks = https://auth.server/keys.json

auth.jwt.from = password
```

JWT Authentication
Identity Verification

Authentication server issues JWTs with username or clientid claims:

```
{
    "exp": 1656425413,
    "iat": 1656424813,
    "username": "someuser"
}
```

Configure `emqx_auth_jwt` to verify claims:

```
...
auth.jwt.verify_claims = on
auth.jwt.verify_claims.username = %u
```

JWT Authorization
ACL rules

Additionally, the authentication provides `acl` claim with topic patterns available for publish(`pub`), subscribe(`sub`), and publish-subscribe(`all`) operations:

```
{
    ...
    "username": "someuser",
    "acl": {
        "sub": ["foo/#"],
        "pub": ["bar/#", "baz/#"]
    }
}
```

Thank you!