3 JWT 和 JWS

上一篇我们提到了token支持JWT格式，那这一篇来盘一下JWT到底是个啥。

JSON Web Token（JWT）

https://jwt.io

JSON Web Token (JWT) (RFC 7519)

JSON Web Token (JWT) 是一个开放标准 (RFC 7519)，它定义了一种紧凑且自包含的方式，用于在各方之间传输信息（Claims）。这些信息以JSON格式定义。

JWT 可以理解为是一个规范。实际上，具体的实现方案有 JWS(JSON Web Signature) 和 JWE(JSON Web Encryption)。

JWT 本质就是一个字符串，不同的实现方案的差异其实就是这个字符串的构成有所不同。

（网上大多数介绍JWT的文章实际介绍的都是JWS，只是JWS比较常用，所以提起JWT经常指的是JWS）

JWT vs 传统token

传统token认证方式，服务端生成token并保存对应的认证信息，将token返回给前端，前端携带token调用服务端接口时，服务端会检查是否有对应的认证信息，若有，则通过验证，反之亦然。

对于JWT，先要介绍下它的的自包含特性，就是指将认证信息直接编码放到JWT中，服务端就不需要存放认证信息。

基于JWT的认证方式，服务端生成JWT，直接返回给前端。

这里就出现了另一个问题，服务端接收到一个带着JWT的请求，怎么验证JWT是合法的呢？也就是判断这个JWT就是自己生成的呢？当然得是自己生成的JWT才能请求自己的接口，随便来个JWT就糊弄过去，那不乱套了。

这里就要介绍最最常用的JWT实现方案 JWS，JWS 使用签名算法来实现JWT的验证。（加签、验签）

JSON Web Signature（JWS）

JSON Web Signature (JWS) (RFC 7515)

JWS 由三部分组成：Header + Payload + Signature，各部分以 . 分隔，结构如下：

Header.Payload.Signature

Header

Header 用于描述JWS的基本的信息，例如其签名算法。

Header 的原文为JSON格式，规定了一些特殊的key，比如 alg(algorithm，签名算法)、typ(type，类型) 等。

经过 Base64 编码生成字符串，作为JWS第一部分。

Payload

Payload 其中包含声明（Claims）。声明是关于实体（通常是用户）和附加数据的陈述。

Payload 的原文为JSON格式，规定了一些特殊的key，比如 iss(issuer，发布者)、iat(issue at，发布时间)、exp(expiration，过期时间)、aud(audience，使用者) 等。（开发者可以自由添加key用于传输数据，比如用户名、权限等等）

经过 Base64 编码生成字符串，作为JWS第二部分。

注意，Payload 只是进行了编码，而非加密，其过程是可逆的！因此，绝对不能存放敏感信息，如密码。

Signature

Signature，即签名，它由编码后的header和payload，使用用户指定的密钥(JWK)，采用header中指定的签名算法（JWA）生成。

Signature 可用于验证数据是否篡改（保证数据完整性）。

接收方收到JWS时，可以采用同样的签名算法和密钥验证签名signature，如果正确，说明数据没有被篡改。（验签）

案例

下面以一个实际的 JWS 讲解各个部分：demo

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV\_adQssw5c

Header

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

使用Base64解码后，得到以下JSON对象：

{

"alg": "HS256",

"typ": "JWT"

}

Payload

eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ

使用Base64解码后，得到以下JSON对象：

{

"sub": "1234567890",

"name": "John Doe",

"iat": 1516239022

}

Signature

SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV\_adQssw5c

密钥为：your-256-bit-secret，使用Header指定的签名算法对该签名进行验证（验签）。

其他

JWA(JSON Web Algorithm)：JWT的加密算法

JWK(JSON Web Key)：JWT的密钥。在对称加密算法中指的是密钥，在非对称加密算法中则指的是公/私钥

JWKS(JSON Web Key Set)：指多个JWK组合在一起的一种格式

reference:

Introduction to JWT (Also JWS, JWE, JWA, JWK)

JWT,JWS与JWE

JWT详解

end