# 基于JWT的安全认证

1. 什么是JWT

Json web token (JWT), 是为了在网络应用环境间传递声明而执行的一种基于JSON的开放标准。JWT的声明一般被用来在身份提供者和服务提供者间传递被认证的用户身份信息，以便于从资源服务器获取资源，也可以增加一些额外的其它业务逻辑所必须的声明信息，该token也可直接被用于认证，也可被加密。

基于token的鉴权机制类似于http协议也是无状态的，它不需要在服务端去保留用户的认证信息或者会话信息。这就意味着基于token认证机制的应用不需要去考虑用户在哪一台服务器登录了，这就为应用的扩展提供了便利。

二、流程

1、用户使用用户名密码来请求服务器

2、服务器进行验证用户的信息

3、服务器通过验证发送给用户一个token

4、客户端存储token，并在每次请求时附送上这个token值

5、服务端验证token值，并返回数据

这个token必须要在每次请求时传递给服务端，它应该保存在请求头里。

三、JWT构成

JWT的构成实例：

eyJhbGciOiJIUzUxMiIsImlhdCI6MTYwODE5Nzk2OCwiZXhwIjoxNjA4MTk3OTc0fQ.eyJ1c2VyX25hbWUiOiJseWYiLCJ1c2VyX3Bhc3N3ZCI6IjEyMzQ1IiwidXNlcl9hZGRyZXNzIjoiMTI3LjAuMC4xIiwidGltZW91dCI6NiwiaWF0IjoxNjA4MTk3OTY4LjI5NjcwMzh9.1WJAKiQAKcn-9aRiYPqUq5M7cVy59HtnOjhMYLh4U2tXpMAZlTjU5SEBk5h1myzSDpyepb470ZRpZLHKQFAQfw

1、header（蓝色部分）

装载了两部分信息。

①声明类型，这里是jwt

②声明加密的算法 通常直接使用 HMAC SHA256

JSON表示如下：

|  |
| --- |
| {  'typ': 'JWT',  'alg': 'HS256'  } |

然后将头部进行base64加密，生成了蓝色部分的字符串。

2、payload（红色部分）

装载的是有效信息，包括三部分。

①标准中注册的声明

②公共的声明

③私有的声明

（1）标准中注册的声明 (建议但不强制使用)

iss: jwt签发者

sub: jwt所面向的用户

aud: 接收jwt的一方

exp: jwt的过期时间，这个过期时间必须要大于签发时间

nbf: 定义在什么时间之前，该jwt都是不可用的.

iat: jwt的签发时间

jti: jwt的唯一身份标识，主要用来作为一次性token,从而回避重放攻击。

（2）公共的声明   
 公共的声明可以添加任何的信息，一般添加用户的相关信息或其他业务需要的必要信息.但不建议添加敏感信息，因为该部分在客户端可解密.

（3）私有的声明   
私有声明是提供者和消费者所共同定义的声明，一般不建议存放敏感信息，因为base64是对称解密的，意味着该部分信息可以归类为明文信息。

一个payload的JSON版如下。

|  |
| --- |
| {  "sub": "1234567890",  "name": "John Doe",  "admin": true  } |

然后进行base64加密，得到第二部分红色字符串。

1. signature

签证信息，由三部分组成：

①base64加密后的header

②base64加密后的payload

③secret（盐）

将加密后的header与payload连接组成字符串然后通过header中声明的方式进行加盐secret组合加密，得到了JWT的第三部分黑色字符串。

**注：secret是保存在服务端用于jwt签发和验证的，即服务器私钥。**

四、优点

①因为json的通用性，所以JWT是可以进行跨语言支持的，像JAVA,JavaScript,NodeJS,PHP等很多语言都可以使用。

②因为有了payload部分，所以JWT可以在自身存储一些其他业务逻辑所必要的非敏感信息。

③便于传输，jwt的构成非常简单，字节占用很小，所以它是非常便于传输的。

④它不需要在服务端保存会话信息, 所以它易于应用的扩展。

五、代码实现

基于itsdangerous库实现。见JWT.py。