2024/8/26 13:12 JWT基础知识

# JWT 基础知识

# 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 (RFC 7519。

该 token 被设计为紧凑且安全的,特别适用于分布式站点的单点登录(SSO)场景,是目前最流行的跨域认证解决方案。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。

### JWT 的原理

JWT 的原理是,服务器认证以后,生成一个 JSON 对象,发回给用户,就像下面这样。

JSON 🗗 复制

```
{
    "姓名": "张三",
    "角色": "管理员",
    "到期时间": "2018年7月1日0点0分"
}
```

以后,用户与服务端通信的时候,都要发回这个 JSON 对象。服务器完全只靠这个对象认定用户身份。为了防止用户篡改数据,服务器在生成这个对象的时候,会加上签名(详见后文)。

服务器就不保存任何 session 数据了,也就是说,服务器变成无状态了,从而比较容易实现扩展。

### JWT 的数据结构

实际当中 JWT 长这个样子:

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
JzdWIiOiIxMjM0NTY30DkwIiwibmFtZSI6IkNUR
kh1YiIsImlhdCI6MTUxNjIzOTAyMn0.Y2PuCD6SfCRpsPN19\_1Sb4WPJNkJr7lhG6YzA8-90Q

纯文本 🗗 复制

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjMONTY3ODkwIiwibmFtZSI6IkNURkh1YiIsImlhdCI6MTUxNjIzOTAyMnO.Y2PuC-D6S fCRpsPN19\_1Sb4WPJNkJr7lhG6YzA8-90Q

它是一个很长的字符串,中间用点(.)分隔成三个部分。注意,JWT 内部是没有换行的

JWT 的三个部分依次如下:

- Header (头部)
- Payload (负载)
- Signature (签名)

写成一行,就是下面的样子。

纯文本 🗗 复制

Header.Payload.Signature

每个部分最后都会使用 base64URLEncode 方式进行编码

Python 🗗 复制

#!/usr/bin/env python

function base64url\_encode(\$data) {

2024/8/26 13:12 JWT基础知识

```
Header

eyJhbGciOiJlUzl1NilslnR5cCl6lkpXVCJ9.eyJzdWliOlixMjM0NTY3ODkwliwibmFtZSl6lkNURkh1YilslmlhdCl6MTUxNjlzOTAyMn0,Y2PuC-D6SfCRpsPN19_1Sb4WPJNklr7lhG6YzA8-9OQ

Payload
```

#### Header

Header 部分是一个 JSON 对象,描述 JWT 的元数据,以上面的例子,使用 base64decode 之后:

header 部分最常用的两个字段是 alg 和 typ。

alg 属性表示 token 签名的算法(algorithm),最常用的为 HMAC 和 RSA 算法

typ 属性表示这个 token 的类型(type), JWT 令牌统一写为 JWT。

### **Payload**

Payload 部分也是一个 JSON 对象,用来存放实际需要传递的数据。JWT 规定了 7 个官方字段,供选用。

• iss (issuer): 签发人

• exp (expiration time): 过期时间

• sub (subject): 主题

• aud (audience): 受众

• nbf (Not Before): 生效时间

• iat (Issued At): 签发时间

• jti (JWT ID):编号

除了官方字段,还可以在这个部分定义私有字段,以上面的例子为例,将 payload 部分解 base64 之后:

```
#文本 | の复制 eyJzdWIiOiIxMjMONTY30DkwIiwibmFtZSI6IkNURkh1YiIsImlhdCI6MTUxNjIzOTAyMn0

JSON | の复制 {
    "sub": "1234567890",
    "name": "CTFHub",
    "iat": 1516239022
}
```

注意: JWT 默认是不会对 Payload 加密的,也就意味着任何人都可以读到这部分 JSON 的内容,所以不要将私密的信息放在这个部分

### **Signature**

Signature 部分是对前两部分的签名,防止数据篡改

首先,需要指定一个密钥(secret)。这个密钥只有服务器才知道,不能泄露给用户。然后,**使用 Header 里面指定的签名算法(默认是 HMAC SHA256)**,按照下面的公式产生签名。

2024/8/26 13:12 JWT基础知识

```
HMACSHA256(
base64UrlEncode(header) + "." +
base64UrlEncode(payload),
secret)
```

算出签名以后,把 Header、Payload、Signature 三个部分拼成一个字符串,每个部分之间用"点"(.)分隔,就可以返回给用户。

### **FLAG**

ctfhub{cfd61b8a7397fa7c10b2ae548f5bfaef}

纯文本 🗗 复制

# 参考链接

- https://jwt.io/introduction/
- https://en.wikipedia.org/wiki/JSON\_Web\_Token
- https://www.ruanyifeng.com/blog/2018/07/json\_web\_token-tutorial.html