为什么很多人不推荐你用JWT?

为什么很多人不推荐你用JWT?

如果你经常看一些网上的带你做项目的教程，你就会发现 有很多的项目都用到了JWT。那么他到底安全吗？为什么那么多人不推荐你去使用。这个文章将会从全方面的带你了解JWT 以及他的优缺点。

什么是JWT?

这个是他的官网[JSON Web Tokens - jwt.io](http://cxyroad.com/"https://jwt.io/")

这个就是JWT

![img](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/ca70c0c14c55488fb4c8e6cecf4afe47~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2381&h=1096&s=183073&e=png&b=fefefe)

JWT 全称`JSON Web Token`

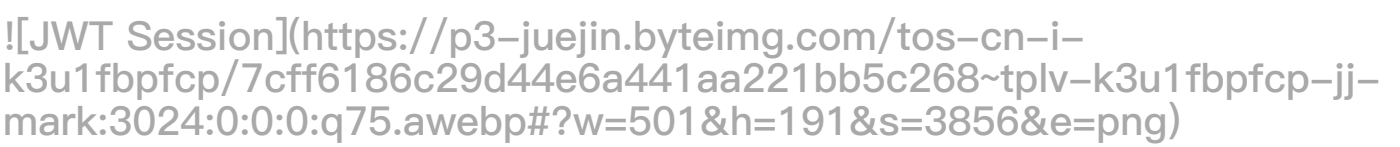
如果你还不熟悉JWT，不要惊慌！它们并不那么复杂！

你可以把JWT想象成一些JSON数据，你可以验证**这些数据**是来自你认识的人。

当然如何实现我们在这里不讲，有兴趣的可以去自己了解。

下面我们来说一下他的流程：

1. 当你登录到一个网站，网站会生成一个**JWT**并将其发送给你。
2. 这个JWT就像是一个包裹，里面装着一些关于你**身份的信息**，比如你的用户名、角色、权限等。
3. 然后，**你在每次与该网站进行通信时都会携带这个JWT**。
4. 每当你访问一个需要验证身份的页面时，**你都会把这个JWT带给网站**。
5. 网站收到JWT后，会验证它的签名以确保它是由网站签发的，并且检查其中的信息来确认你的身份和权限。
6. 如果一切都通过了验证，你就可以继续访问受保护的页面了。



为什么说JWT很烂?

首先我们用JWT应该就是去做这些事情:

* 用户注册网站
* 用户登录网站
* 用户点击并执行操作
* 本网站使用用户信息进行创建、更新和删除 信息

这些事情对于数据库的操作经常是这些方面的

* 记录用户正在执行的操作
* 将用户的一些数据添加到数据库中
* 检查用户的权限，看看他们是否可以执行某些操作

之后我们来逐步说出他的一些缺点

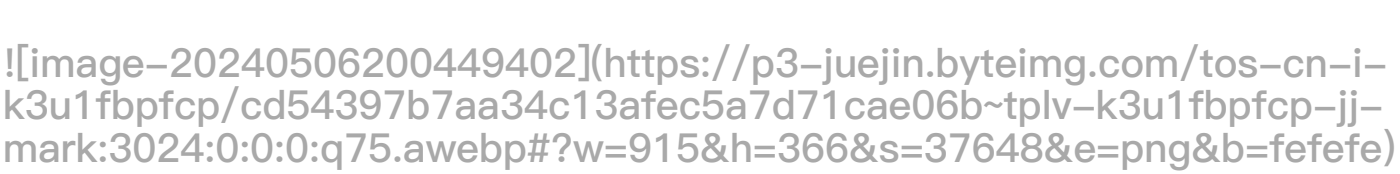
大小

这个方面毋庸置疑。

比如我们需要存储一个用户ID 为xiaou

如果存储到cookie里面，我们的总大小只有5个字节。

如果我们将 ID 存储在 一个 JWT 里。他的大小就会增加大概51倍



这无疑就增大了我们的宽带负担。

冗余签名

JWT的主要卖点之一就是其加密签名。因为JWT被加密签名，接收方可以验证JWT是否有效且可信。

但是，在过去20年里几乎每一个网络框架都可以在使用普通的会话cookie时获得加密签名的好处。

事实上，大多数网络框架会自动为你加密签名（甚至加密！）你的cookie。这意味着你可以获得与使用JWT签名相同的好处，而无需使用JWT本身。

实际上，在大多数网络身份验证情况下，JWT数据都是存储在会话cookie中的，这意味着现在有两个级别的签名。一个在cookie本身上，一个在JWT上。

令牌撤销问题

由于令牌在到期之前一直有效，服务器没有简单的方法来撤销它。

以下是一些可能导致这种情况危险的用例。

注销并不能真正使你注销！

想象一下你在推特上发送推文后注销了登录。你可能会认为自己已经从服务器注销了，但事实并非如此。因为JWT是自包含的，将在到期之前一直有效。这可能是5分钟、30分钟或任何作为令牌一部分设置的持续时间。因此，如果有人

在此期间获取了该令牌，他们可以继续访问直到它过期。

可能存在陈旧数据

想象一下用户是管理员，被降级为权限较低的普通用户。同样，这不会立即生效，用户将继续保持管理员身份，直到令牌过期。

JWT通常不加密

因此任何能够执行中间人攻击并嗅探JWT的人都拥有你的身份验证凭据。这变得更容易，因为中间人攻击只需要在服务器和客户端之间的连接上完成

安全问题

对于JWT是否安全。我们可以参考这个文章

[JWT （JSON Web Token） （in）security - research.securitum.com](http://cxyroad.com/”https://research.securitum.com/jwt-json-web-token-security/”)

同时我们也可以看到是有专门的如何攻击JWT的教程的

[高级漏洞篇之JWT攻击专题 - FreeBuf网络安全行业门户](http://cxyroad.com/”https://www.freebuf.com/articles/web/375465.html”)

总结
--

总的来说，JWT适合作为**单次授权令牌**，用于在两个实体之间传输声明信息。

但是，JWT不适合作为**长期持久数据的存储机制**，特别是用于**管理用户会话**。使用JWT作为会话机制可能会引入一系列严重的安全和实现上的问题，相反，对于长期持久数据的存储，更适合使用传统的会话机制，如会话cookie，以及建立在其上的成熟的实现。

但是写了这么多，我还是想说，如果你作为自己开发学习使用，不考虑安全，不考虑性能的情况下，用JWT是完全没有问题的，但是一旦用到生产环境中，我们就需要避免这些可能存在的问题。

原文链接: https://juejin.cn/post/7365533351451672612