[**web安全之token**](http://www.cnblogs.com/bukudekong/p/3829875.html)

参考：http://blog.csdn.net/sum\_rain/article/details/37085771

Token，就是令牌，最大的特点就是随机性，不可预测。一般黑客或软件无法猜测出来。

那么，Token有什么作用？又是什么原理呢？

Token一般用在两个地方:

* 1)**防止表单重复提交**、
* 2)anti csrf攻击（跨站点请求伪造）。

两者在原理上都是通过session token来实现的。当客户端请求页面时，服务器会生成一个随机数Token，并且将Token放置到session当中，然后将Token发给客户端（一般通过构造hidden表单）。下次客户端提交请求时，Token会随着表单一起提交到服务器端。  
然后，如果应用于“anti csrf攻击”，则服务器端会对Token值进行验证，判断是否和session中的Token值相等，若相等，则可以证明请求有效，不是伪造的。  
不过，如果应用于“防止表单重复提交”，服务器端第一次验证相同过后，会将session中的Token值更新下，若用户重复提交，第二次的验证判断将失败，因为用户提交的表单中的Token没变，但服务器端session中Token已经改变了。

上面的session应用相对安全，但也叫繁琐，同时当多页面多请求时，必须采用多Token同时生成的方法，这样占用更多资源，执行效率会降低。因此，也可用cookie存储验证信息的方法来代替session Token。比如，应对“重复提交”时，当第一次提交后便把已经提交的信息写到cookie中，当第二次提交时，由于cookie已经有提交记录，因此第二次提交会失败。  
不过，cookie存储有个致命弱点，如果cookie被劫持（xss攻击很容易得到用户cookie），那么又一次gameover。黑客将直接实现csrf攻击。

所以，安全和高效相对的。具体问题具体对待吧。

此外，要避免“加token但不进行校验”的情况，在session中增加了token，但服务端没有对token进行验证，根本起不到防范的作用。

还需注意的是，对数据库有改动的增删改操作，需要加token验证，对于查询操作，一定不要加token，防止攻击者通过查询操作获取token进行csrf攻击。但并不是这样攻击者就无法获得token，只是增大攻击成本而已。