



[CSRF攻击原理及防御](http://www.cnblogs.com/shytong/p/5308667.html)

一、CSRF攻击原理



　　CSRF是什么呢？CSRF全名是Cross-site request forgery，是一种对网站的恶意利用，CSRF比XSS更具危险性。想要深入理解CSRF的攻击特性我们有必要了解一下网站session的工作原理。   
  
　　session我想大家都不陌生，无论你用.net或PHP开发过网站的都肯定用过session对象，然而session它是如何工作的呢？如果你不清楚请往下看。   
先问个小问题：如果我把浏览器的cookie禁用了，大家认为session还能正常工作吗？   
  
　　答案是否定的，我在这边举个简单的例子帮助大家理解session。   
比如我买了一张高尔夫俱乐部的会员卡，俱乐部给了我一张带有卡号的会员卡。我能享受哪些权利（比如我是高级会员卡可以打19洞和后付费喝饮料，而初级会员卡只能在练习场挥杆）以及我的个人资料都是保存在高尔夫俱乐部的数据库里的。我每次去高尔夫俱乐部只需要出示这张高级会员卡，俱乐部就知道我是谁了，并且为我服务了。  
  
　　这里我们的高级会员卡卡号 = 保存在cookie的sessionid；   
而我的高级会员卡权利和个人信息就是服务端的session对象了。   
  
　　我们知道http请求是无状态的，也就是说每次http请求都是独立的无关之前的操作的，但是每次http请求都会将本域下的所有cookie作为http请求头的一部分发送给服务端，所以服务端就根据请求中的cookie存放的sessionid去session对象中找到该会员资料了。   
当然session的保存方法多种多样，可以保存在文件中，也可以内存里，考虑到分布式的横向扩展我们还是建议把它保存在第三方媒介中，比如redis或者mongodb。   
  
　　我们理解了session的工作机制后，CSRF也就很容易理解了。CSRF攻击就相当于恶意用户A复制了我的高级会员卡，哪天恶意用户A也可以拿着这张假冒的高级会员卡去高尔夫俱乐部打19洞，享受美味的饮料了，而我在月底就会收到高尔夫俱乐部的账单！   
  
　　了解CSRF的机制之后，危害性我相信大家已经不言而喻了，我可以伪造某一个用户的身份给其好友发送垃圾信息，这些垃圾信息的超链接可能带有木马程序或者一些欺骗信息（比如借钱之类的），如果CSRF发送的垃圾信息还带有蠕虫链接的话，那些接收到这些有害信息的好友万一打开私信中的连接就也成为了有害信息的散播着，这样数以万计的用户被窃取了资料种植了木马。整个网站的应用就可能在瞬间奔溃，用户投诉，用户流失，公司声誉一落千丈甚至面临倒闭。曾经在MSN上，一个美国的19岁的小伙子Samy利用css的background漏洞几小时内让100多万用户成功的感染了他的蠕虫，虽然这个蠕虫并没有破坏整个应用，只是在每一个用户的签名后面都增加了一句“Samy 是我的偶像”，但是一旦这些漏洞被恶意用户利用，后果将不堪设想，同样的事情也曾经发生在新浪微博上面。   
  
　　举例：   
  
　　CSRF攻击的主要目的是让用户在不知情的情况下攻击自己已登录的一个系统，类似于钓鱼。如用户当前已经登录了邮箱，或bbs，同时用户又在使用另外一个，已经被你控制的站点，我们姑且叫它钓鱼网站。这个网站上面可能因为某个图片吸引你，你去点击一下，此时可能就会触发一个js的点击事件，构造一个bbs发帖的请求，去往你的bbs发帖，由于当前你的浏览器状态已经是登陆状态，所以session登陆cookie信息都会跟正常的请求一样，纯天然的利用当前的登陆状态，让用户在不知情的情况下，帮你发帖或干其他事情。

二、CSRF防御

* 通过 referer、token 或者 验证码 来检测用户提交。
* 尽量不要在页面的链接中暴露用户隐私信息。
* 对于用户修改删除等操作最好都使用post 操作 。
* 避免全站通用的cookie，严格设置cookie的域。