Web中CSRF跨站请求伪造与CORS跨域请求获取资源

原创

忘记以前 于 2019-08-18 23:17:29 发布 350 收藏

分类专栏： Web 文章标签： CORS CSRF Web

本文链接：https://blog.csdn.net/itpedestrian/article/details/99709505

CORS跨域获取资源

```
1. 浏览器的同源策略: 协议, 主机和端口port都相同的两个地址是同源地址, 否则是非同源地址
2. 当发起请求的页面地址和被请求的地址不是同源, 那么这个请求就是跨域请求
3. 在发起跨域请求时, 如果浏览器发现你的请求时跨域请求, 就会给请求报文中添加印记--Origin: 标志你的这个源请求地址, 接下来在服务器返回响应的过程中, 如果响应头中包含Access-Control-Allow-Origin: 源请求IP地址(浏览器会自动的判断被访问的服务器是否支持跨域, 如果认为支持, 就会Allow)
```

举例:

项目中, 一般会有后端API服务器和静态文件服务器, 客户端向静态文件服务器发送一个请求, 静态文件服务器返回一个静态网页, 用户通过点击静态页面中的超链接去向后端API服务器去访问动态数据, 这个过程中用户通过点击静态文件服务器返回的页面跨域访问了后端API服务器, 此时后端的API服务器就要设置一个白名单(里面包含静态服务器的ip和端口), 有了这个白名单, 浏览器访问后端API服务器的时候, 浏览器会自动判断被访问服务器是否支持跨域, 并且API服务器也会判断请求的源地址是否在白名单中.

CSRF 跨站请求伪造

举例:

```
1. 条件: 存在漏洞的网站WebA, 危险攻击网站WebB, 受害者User和WebA
2. 浏览并信任网站A
3. 验证通过, 网站A服务器设置了cookie并返回给了浏览器
4. 此时在没有退出网站A的情况下, 访问危险网站B
5. 网站B要求访问网站A, 并给网站A发送了一个请求
6. 根据网站B的请求, 浏览器带着A网站服务器返回的cookie去访问A网站
7. A网站并不知道请求是用发送的, 还是网站B发送的
```

要完成CSRF攻击受害者要完成两个必要条件:

登录受信任网站A，并在本地生成Cookie。

在不登出A的情况下，访问危险网站B。

防御CSRF攻击

1. 浏览器中Referer表示的是发起请求的网站的开头域名，这样网站防御可以通过验证Referer字段来判断发起请求的网站.
2. Referer是由浏览器提供的，一些小的浏览器可以篡改Referer的值，所以如果防御网站但靠Referer来判定身份，会出大问题，所以有些高级浏览器处于安全考虑不再提供Referer，防御网站反而会误认为是CSRF攻击，拒绝访问
3. 添加token，http请求中随机生成token，发送给服务器，服务器储存在session中，每次请求会把请求中的token和session中的token比对，攻击者因为浏览器的同源策略是不能够解析到数据信息的，只能发送请求而已，所以只要token不保存在cookie中，是无法成功的发送请求.
4. 但是每个请求都要携带token是非常麻烦的，只能通过遍历对所有dom标签中的每个a，form标签进行加上token，如果攻击者在一些可以上传自己网站超链接的网站上，在发送http请求到攻击者的网站(超链接a标签会携带产生的token到攻击者的网站)，这样的话，就能拿到token，发起CSRF攻击，
5. 在http请求头中自定义属性，也是不完美的解决方案，局限很大.....(感兴趣的话自己谷歌)

关于CSRF漏洞检测

1. 检测CSRF漏洞是一项比较繁琐的工作，最简单的方法就是抓取一个正常请求的数据包，去掉Referer字段后再重新提交，如果该提交还有效，那么基本上可以确定存在CSRF漏洞。
2. 随着对CSRF漏洞研究的不断深入，不断涌现出一些专门针对CSRF漏洞进行检测的工具，如CSRFTester，CSRF Request Builder等。
3. 以CSRFTester工具为例，CSRF漏洞检测工具的测试原理如下：使用CSRFTester进行测试时，首先需要抓取我们在浏览器中访问过的所有链接以及所有的表单等信息，然后通过在CSRFTester中修改相应的表单等信息，重新提交，这相当于一次伪造客户端请求。如果修改后的测试请求成功被网站服务器接受，则说明存在CSRF漏洞，当然此款工具也可以被用来进行CSRF攻击。

总结

很难杜绝，只能控制住自己的手，如果身处上文所述的CSRF攻击条件下，少点陌生的链接