CSRF攻防汇总梳理

lshack / 2017-03-12 15:53:00 / 浏览数 3838 安全技术 技术讨论 顶(0) 踩(0)

---

Author■lshack

网文很多，这里总结整理一篇，此敬系列作者们，引用如下：

最火的《浅谈CSRF攻击方式》

圈内科普文1《CSRF简单介绍及利用方法 | WooYun知识库》

圈内科普文2《邪恶的CSRF | WooYun知识库》

圈内科普文3《从零开始学CSRF》

CSRF（Cross-Site Request
Forgery，跨站点伪造请求）是一种网络攻击方式，该攻击可以在受害者毫不知情的情况下以受害者名义伪造请求发送给受攻击站点，从而在未授权的情况下执行在权限保护

我们同样按照传播度最广的上面那片文章来梳理。

## 1.CSRF是什么？

CSRF（Cross-site request forgery），中文名称：跨站请求伪造，也被称为：one click attack/session riding，缩写为：CSRF/XSRF。

## 2.CSRF可以做什么？

攻击者可以根据浏览器或者网站程序漏洞盗用你的身份去进行一切你的权限可以执行的操作包括刷粉丝，关注大v，发论坛帖子，发邮件，以及早年的银行账户转账。所以由

## 3.CSRF的原理是什么？

用户在左侧浏览器端使用用户名密码登录了a网站，a网站匹配校验了用户名密码返回一个cookie给左侧浏览器用户从此访问a网站的时候就带着第一次校验生效的cookie进行

## 4.CSRF的分类？

如xss一般，xss有反射和存储。csrf固然有两种，因为他的温床是上图中的a和b和cookie，基于网站给予的温床才可以"施展"，所以根据现实适用性和广度来划分，大体分为

*存在即合理。我们来看下几个yy的场景。*

### 4.1.GET型：

这种类型的CSRF一般是由于程序员安全意识不强造成的。GET类型的CSRF利用非常简单，只需要一个HTTP请求，所以，一般会这样利用：

```
<img src=http://xxxx.org/csrf.php?xx=11 />
```

是的，csrf叫做跨站伪造请求，基于跨站的，所以他是xss的表兄弟。

在访问b站里含有这个img的页面后，成功向http://aaaaaa.org/csrf.php?xx=11发出了一次HTTP请求。所以，如果将该网址替换为存在GET型CSRF的地址，就能完成攻击

如图这个地址固然不存在所以请求返回状态码是404，那么如果我们换成是a站里头真实存在的一个请求地址就完成了一次get请求的csrf攻击。

在具体些我们走入群众的视线去视察这样一种猥琐。

在一个bbs社区里，用户在发言的时候会发出一个这样的GET请求：

```
GET /talk.php?msg=hello HTTP/1.1
Host: www.bbs.com
…
Cookie: PHPSESSID=ee2cb583e0b94bad4782ea
```

这是用户发言内容为"hello"时发出的请求，当然，用户在请求的同时带上了该域下的cookie，于是攻击者构造了下面的csrf.html页面：

```
<html>
<img src=http://www.bbs.com/talk.php?msg=goodbye />
</html>
```

可以看到，攻击者在自己的页面中构造了一个发言的GET请求，然后把这个页面放在自己的网站上，链接为`http://www.bbbbb.com/csrf.html`。之后攻击者通过某种方

### 4.2.POST型:

这种类型的CSRF危害没有GET型的大，利用起来通常使用的是一个自动提交的表单，如：

```
<form action=http://aaaaaa.org/csrf.php method=POST>
<input type="text" name="xx" value="11" />
</form>
<script> document.forms[0].submit(); </script>
```

访问该页面后，表单会自动提交，相当于模拟用户完成了一次POST操作。

我们同样走入群众。

在一个CMS系统的后台，发出下面的POST请求可以执行添加管理员的操作：

```
POST /manage.php?act=add HTTP/1.1
Host: www.cms.com
…
Cookie: PHPSESSID=ee2cb583e0b94bad4782ea;
is_admin=234mn9guqgpi3434f9r3msd8dkekwel

uname=test&pword=test
```

在这里，攻击者构造了的csrf2.html页面如下：

```
<html>
<form action="/manage.php?act=add" method="post">
<input type="text" name="uname" value="evil" />
<input type="password" name="pword" value="123456" />
</form>
<script>
document.forms[0].submit();
</script>
</html>
```

该页面的链接为`http://www.bbbbb.com/csrf2.html`，攻击者诱骗已经登录后台的网站管理员访问该链接（比如通过给管理员留言等方式）会发生什么呢？当然是网站

4.3. 其他猥琐流CSRF:

过基础认证的CSRF(常用于路由器):

```
<img src=http://admin:admin@192.168.1.1 />
```

加载该图片后，路由器会给用户一个合法的SESSION，就可以进行下一步操作了。

*综上*

CSRF攻击会根据场景的不同而危害迥异。小到诱使用户留言，大到垂直越权进行操作。这些攻击的请求都是跨域发出,并且至关重要的一点，都是在受害者的身份得到认证以

## 5.CSRF如何防御？

从上面我们看出几个思路：

需要完成攻击的话？1、a生成cookie存于本地缓存b。2、不关闭a访问b

然而问题来了1、你不一定每次保证退出登录后cookie清除干净。2、不保证退出a，清除a，关闭a，再访问b。

所以要防御CSRF攻击，我们就要牢牢抓住CSRF攻击的几个特点。

首先是“跨域”，我们发现CSRF攻击的请求基本都是跨域的，针对这一特点，我们可以在服务端对HTTP请求头部的Referer字段进行检查。一般情况下，用户提交的都是站内的

第二点是“伪造”，这也是CSRF攻击的核心点，即伪造的请求。我们来想一下，攻击者为什么能够伪造请求呢？换句话说，攻击者能够伪造请求的条件是什么呢？纵观之前我们

*在服务器端：*

5.1.添加验证码：

CSRF攻击的过程，往往是在用户不知情的情况下构造网络请求。所以如果使用验证码，那么每次操作都需要用户进行互动，从而简单有效的防御了CSRF攻击。但是如果你在

5.2.验证referer：

根据HTTP协议，在HTTP头中有一个字段叫Referer，它记录了该HTTP请求的来源地址。在通常情况下，访问一个安全受限页面的请求必须来自于同一个网站。比如某银行的
test，然后通过点击页面上的按钮来触发转账事件。当用户提交请求时，该转账请求的Referer值就会是转账按钮所在页面的URL（本例中，通常是以bank.
test域名开头的地址）。而如果攻击者要对银行网站实施CSRF攻击，他只能在自己的网站构造请求，当用户通过攻击者的网站发送请求到银行时，该请求的Referer是指向攻

test开头的域名，则说明该请求是来自银行网站自己的请求，是合法的。如果Referer是其他网站的话，就有可能是CSRF攻击，则拒绝该请求。

5.3.使用一次性token：

所谓token是一段字母数字随机值，我们可以把它理解为一个服务端帮我们填好的验证码！每当我们访问该页面时，服务端会根据时间戳、用户ID、随机串等因子生成一个随

5.4.限制Session生命周期：

顾名思义，缩短Session的有效时间。



上一篇：SQL注入之获取指定数据库数据My... 下一篇：绕过CDN的一些小套路（漫画版）

1. 1 条回复

zenf4rcing 2017-10-21 01:11:41

谢谢

0 回复Ta

---