XSS（跨站脚本攻击）:它是通过对网页注入可执行代码且成功地被浏览器执行，达到攻击的目的，形成了一次有效XSS攻击，一旦攻击成功，它可以获取用户的联系人列表，然后向联系人发送虚假诈骗信息，可以删除用户的日志等等，有时候还和其他攻击方式同时实施比如SQL注入攻击服务器和数据库、Click劫持、相对链接劫持等实施钓鱼，它带来的危害是巨大的，是web安全的头号大敌。

执行条件：需要向web页面注入恶意代码；这些恶意代码能够被浏览器成功的执行。

XSS攻击方式：XSS存储型攻击：恶意代码被保存到目标网站的服务器中，这种攻击具有较强的稳定性和持久性，比较常见场景是在博客，论坛等社交网站上；XSS反射型攻击：意代码并没有保存在目标网站，通过引诱用户点击一个链接到目标网站的恶意链接来实施攻击的；DOM型攻击。

xss存储型攻击示例：用户可以一html的方式上传一次文章，如果黑客用户在上传文章时，在html里加入了一些违规的js代码，那么其他用户在访问这篇文章时，浏览器会默认执行黑客上传的js代码，从而完成了对其他用户攻击

xss反射型攻击方式：有时前端html页面需要把get方式传回来的字段显示到页面上，这时如果黑客在传的字段后加上一段js代码，则也是有可能被执行的。

攻击类型很多，例如：

盗取用户账号cookie

流量劫持（通过访问某段具有window.location.href定位到其他页面）

防御方式：

将前端提交的数据编码转义：一般表单提交的时候完全可以用strip\_tags函数去除html标签，如果涉及到富文本编辑器需要保留html标签，可以用htmlspecialchars对提交数据进行过滤，从而转换（& < > ’ ” ）这几个字符；还有可以利用htmlentities函数转换所有的html标记

对cookie的保护：对重要的cookie设置httpOnly, 防止客户端通过document.cookie读取cookie。服务端可以设置此字段。

过滤掉所有可执行的js代码：例如利用lexer方式

防范方法：

当恶意代码值被作为某一标签的内容显示：在不需要html输入的地方对html 标签及一些特殊字符( ” < > & 等等 )做过滤，将其转化为不被浏览器解释执行的字符。

当恶意代码被作为某一标签的属性显示，通过用 “将属性截断来开辟新的属性或恶意方法：属性本身存在的 单引号和双引号都需要进行转码；对用户输入的html 标签及标签属性做白名单过滤，也可以对一些存在漏洞的标签和属性进行专门过滤。