**Web安全浅析**

龙红平 16130130293

**摘要：**为了加强Web项目的安全性,保证用户数据安全，初步了解Web相关的安全知识，从世界观安全出发，探究客户端脚本安全、服务器端应用安全，最后回归互联网安全，其中涉及浏览器安全、XSS、CSRF、SQL注入、DDOS等概念，加强自己对Web安全的理解，培养信息安全意识。

**关键词：**Web安全；XSS；CSRF；SQL注入；DDOS；加密算法

**Web Security Analysis**

LONG Hongping 16130130293

**Abstract:** In order to strengthen the security of the Web project, to ensure the safety of user data, a preliminary understanding of Web related safety knowledge, starting from the world security, explore the client script, server applications, finally return to Internet security, including the browser security, SQL injection, XSS, CSRF, DDOS and other concepts, strengthen their understanding of Web security, train their consciousness of information security.

**Key Words:** Web security;XSS;CSRF;SQL injection;DDOS;The encryption algorithm

# 一、前言

Web是互联网的核心，是云计算和移动互联网的最佳载体，因此Web安全也是互联网安全业务中最重要的组成部分。然而在当今的互联网行业中，对安全的重视程度普遍不高。有统计显示，互联网公司对安全的投入不足收入的百分之一。最近几年“脱裤”事件时有发生，用户信息泄露的消息也层出不穷，这让人们意识到中国互联网的安全现状有多么糟糕，可这些事件的发生只是那些网站为他们曾经的不作为买单。而要解决“拖库”的问题，不仅仅是保证一个数据库的安全，而是要解决整个互联网的信息安全问题。

# 二、什么是Web安全

相信不少人都看过“黑客帝国”系列电影，其中的黑衣人利用网络穿梭于世界之间，让人啧啧惊奇，梦想着自己能成为一名同样强大的“Hacker”，而现实世界中的“Hacker”，也就是黑客，虽然没有那么神奇，但也以他们的方式在这个世界穿梭着。在现实中，黑客对计算机有着深入的理解，他们是一群破坏规则、不喜欢被拘束的人，因此总想着去发现系统的漏洞，以获得一些权限之外的权力，从而获得满足感。

我们可以将中国黑客发展分为三个阶段：

（1启蒙时代。20世纪90年代，热爱技术的青少年出于好奇心和求知欲走上黑客这条路，和利益完全没有关系，纯粹爱好和满足感；

（2黄金时代。中美黑客大战，一战成名，黑客文化和黑客技术吸引了更多青少年加入其中，各种黑客组织和论坛越来越多；

（3黑暗时代。黑客精神Open、Free、Share消亡，黑灰产阴云密布，黑客功利性越来越强，破坏力也越来越强，这也导致了Web安全问题的产生。

Web攻击技术的发展也可以分为几个阶段。Web1.0时代，人们更多的是关注服务器端动态脚本，比如将一个可执行脚本（俗称webshell）上传到服务器上，从而获得权限；之后SQL注入的出现是WEB安全史上的一个里程碑，黑客以此来获取一些敏感数据；XSS的出现则是WEB安全史上的另一个里程碑，XSS蠕虫事件让人们意识到了Web安全的重要性。到现在Web2.0的时代，各种攻击技术都在变强，攻击方式也变得多样化，Web安全的研究必须跟紧时代的步伐。攻击和防御在不断的对抗和碰撞中得到发展，两者是相互促进的辩证关系，任何一劳永逸的想法都是不现实的而且是很危险的；

要知道安全的本质就是信任的问题，一切的安全设计的基础，都是建立在信任的关系上的。我们必须相信一些东西，必须有一些最基本的假设，安全方案才能得以建立。安全三要素机密性、完整性、可用性要求我们要保护数据内容不能泄露，保护数据内容是完整的、没有篡改的，保护资源“随取而得”，所以我们需要加密、数字签名这些技术手段来帮助我们构筑一个完善的Web安全保护伞。

下表是微软提出的STRIDE模型，借此我们可以去简单地分析一下Web安全到底要面对哪些威胁。

表2-1 STRIDE模型

|  |  |  |
| --- | --- | --- |
| 威胁 | 定义 | 对应的安全属性 |
| Spoofing | 冒充他人身份 | 认证 |
| Tampering | 修改数据或代码 | 完整性 |
| Repudiation | 否认做过的事情 | 不可抵赖性 |
| InformationDisclosure | 机密信息泄露 | 机密性 |
| Denial of Service | 拒绝服务 | 可用性 |
| Elevation of Privilege | 未经授权获得许可 | 授权 |

# 三、客户端安全

客户端安全，一般是脚本安全，指的就是用户在使用Web应用时可能造成的安全隐患，又可以分成浏览器安全、跨站脚本攻击（XSS）、跨站点请求伪造（CSRF）、点击劫持（clickJacking）、HTML5安全，下面会分别说明。

## 3.1 XSS攻击

XSS攻击，通常指用户通过Html链接注入，插入恶意的脚本，篡改了网页，从而在其他用户浏览网页的时候，控制用户浏览器的一种攻击形式，所以我们平时要注意不要随意点击奇怪的链接，典型就是发到你邮箱的不明链接，这些都是可能存在的安全隐患。而从一开始这种攻击的案例是跨域的，所以叫做跨站脚本。

XSS根据效果分为三个类型

反射型。只是简单的把用户输入的数据“反射”给浏览器.黑客往往需要诱使用户点击一个恶意链接，才能攻击成功；

储存型XSS。通过把用户输入的数据储存到服务器端，这种XSS具有很强的稳定性。常见的就是黑客写一篇包含恶意的javaScript的代码文章，然后访问该文章的用户就会被动的在他们的浏览器中执行这段恶意的JavaScript代码；

DOM Based XSS。通过修改DOM节点形成的XSS,从效果上也是反射性XSS，但是因为是基于DOM树形成的，有所区别。

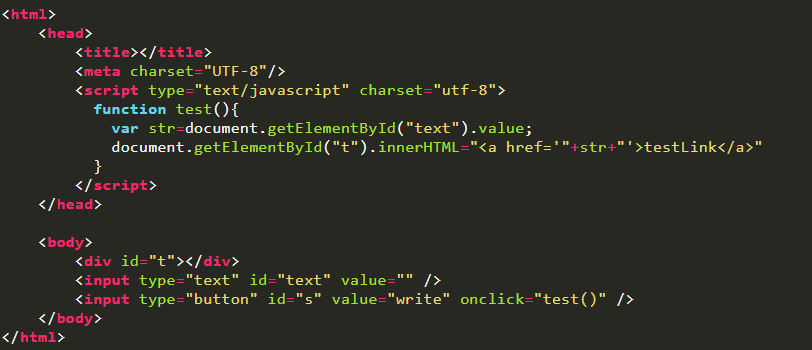
举个例子：

图3-1

这里点击提交之后，就会在网页中嵌入testLink，从而获取想要的信息。

利用XSS攻击，黑客就可以钓鱼、识别用户的浏览器识别用户的安装软件、获取用户IP，从而造成了信息的泄露。

关于XSS的防御，XSS的本质是一种HTML注入，用户的数据被当做HTML代码一部分执行，已混淆了原本的语义，产生了新的语义，我们要做的就是检查用户的输入输出，对一些特殊的字符进行转义，进行一些简单的防御。

## 3.2 CSRF攻击

跨站请求伪造，它与XSS不同，XSS是利用站点内的信任用户，而CSRF则是伪装成受信任用户的请求来利用受信任的网站。CSRF攻击是源于Web的隐式身份验证机制！Web的身份验证机制虽然可以保证一个请求是来自于某个用户的浏览器，但却无法保证该请求是用户批准发送的。另外这里面就会涉及到Cookie的问题。

 举个例子。

图3-2

假如你在写的博客里加入这么一段代码，那么每一个点开你这篇文章的人就会自动关注你了（当然这只是个例子）。

关于CSRF防御，我们要尽量使用POST，限制GET，降低攻击风险，这是因为GET和POST特性的不同；或者用验证码来强制用户与应用交互；也可以自己验证Referer Check和TOKEN。

# 四、服务器端安全

服务器端安全，应用安全，主要涉及注入攻击（SQL注入）、文件上传漏洞、应用层拒绝服务攻击(DDOS)等服务器端的行为已经前后端交互时的漏洞。

## 4.1 SQL注入

SQL注入就是通过把SQL语句插入到网页表单递交、输入域名或页面请求的查询字符串，达到欺骗服务器执行恶意的SQL命令。具体来说，它是利用现有应用程序，将恶意的SQL命令注入到后台数据库执行的能力，它可以通过在网页表单中输入恶意SQL语句得到一个存在安全漏洞的网站上的数据库，而不是按照设计者意图去执行SQL语句。

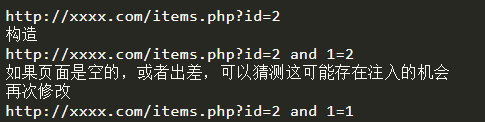
**** 在我们什么都不知道时，一般都会选择盲注，就像下面这样：

图4-1

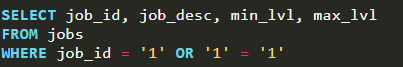
假如我们有一个URL是<http://localhost:3452/ExcelUsingXSLT/Default.aspx>，通过试探推出可能有的参数id，再以1=1这种永真的式子拼凑出<http://localhost:3452/ExcelUsingXSLT/Default.aspx?id=1'or'1'='1>，获取权限。这个URL等效于SQL语句：

图4-2

要想防御SQL注入的话，我们可以预编译语句，绑定变量，比如Java里的PreparedStatement；可以使用存储过程，调用存在数据库里的函数，避免直接接触数据的机会；可以检查数据类型，如果输入是数字，就用integer做检验，如果是邮箱电话之类的就用正则表达式去做校验；使用定义好的安全函数来转义等，可以根据开发用的语言来决定防御措施，现在很多开发框架也帮我们考虑好了这一部分。

## 4.2 DDOS攻击

DDOS又叫分布式拒绝服务，它利用合理的请求造成资源过载，导致服务不可用，可以说是流氓式攻击。

我们可以把DDOS攻击分成SYN flood攻击和应用层DDOS。

### 4.2.1 SYN flood攻击

SYN flood攻击，我把它叫做洪流冲击。它模拟一个请求建立连接的过程，首先会伪造大量的源IP地址，分别向服务器端发送大量的SYN包，此时服务器端会返回SYN/ACK包，因为源地址是伪造的，所以伪造的IP并不会应答，服务器端就会因为等待回应导致资源的浪费。攻击者大量发送这种伪造源地址的SYN请求，服务器会消耗非常多的资源来处理这种半连接。这主要发生在TCP三次握手时。

对抗SYN flood的主要策略有SYN Cookie，SYN Proxy，Safereset等算法,具体的话我也不是太了解，简单来说就是设立一个请求异常标准，超过这个标准就认为这个请求是恶意的，丢弃之。

### 4.2.2 应用层DDOS

由于发生在应用层，TCP三次握手已经完成，发起的IP都是真实的，这种攻击会比网络层DDOS更可怕，耗费资源更多。细分下来就是CC攻击和资源耗尽攻击。

CC攻击原理非常简单，就是对一些消耗资源较大的应用页面不断发起正常的请求，以达到消耗服务器资源的目的。在Web应用中，读写数据库，读写文件等需要IO的操作都会消耗成比较多的资源，也是重灾区。

资源耗尽攻击，Slowloris攻击，其原理是以极低的速度往服务器发送HTTP请求。由于服务器对于并发的连接数有一定的上限，因此若是连接长时间被恶意地占有且不释放，那么服务器就无法接收到正常的请求，正常用户也无法获取服务，由此导致极大的损失。

一个简单的实现是构造一个畸形的HTTP请求：

图4-3

由于WebServer只收到一个\r\n,因此将认为Http Headers部分没有结束，并保持连接，造成资源的浪费。

对于DDOS攻击我们除了识别恶意的请求之外，还可以做的就是优化自身服务器的性能，通过使用内存数据库、实现负载均衡、CDN等方式，缓解服务器压力，提高性能。

## 4.3 文件上传漏洞

文件上传后导致的常见安全问题一般有：

（1上传文件是Web脚本语言，服务器的Web容器解释并执行用户上传的脚本，导致代码执行；

（2上传文件是Flash的策略文件crossdomain.xml,黑客用以控制Flash在该域下的行为(其他通过类似行为控制策略文件的情况类似)；

（3上传文件是病毒，木马文件，用以诱惑用户或者管理员下载执行；

（4上传文件是钓鱼图片或为包含脚本的图片，在某些版本浏览器中会被作为脚本执行，被用于钓鱼和欺诈。

可见文件上传的漏洞主要在于用户上传文件类型的未知性，一旦用户上传恶意文件，就有可能对我们的站点造成伤害。所以我们要在有文件上传功能的页面，将文件上传的目录设置为不可执行，只要Web容器不解析该目录下的文件，即使上传了脚本文件不会有影响，不过也不能一直不处理；还有一种简单的就是判断文件类型，可以结合MIME Type,后缀检查等方式，文件格式不正确就不接受不处理它。

# 五、结论

一般来说，安全是系统的一个特性，我们在开发系统时都会被要求考虑系统安全性，保证数据安全。安全本身可以作为系统的一个重要组成部分，一个好的系统，在设计的时候就应该考虑是否会存在安全隐患，从而提前准备好对策，也便于解决安全和系统功能之间的矛盾。对于密码是否明文存储，验证码的使用，用户身份的认证等，都是需要考虑的安全问题。

其次，随着计算机技术的进步，Web安全的威胁越来越多，各种钓鱼网站、木马病毒换着花样的出现，这要求我们提升自己的辨识能力，减少被欺骗事件的发生。

**参考文献：**

[1] 吴翰清. 白帽子讲Web安全[M]. 北京：电子工业出版社，2014.6

WU Hanqing. White hat on Web security[M].Beijing: Electronic industry press,2014.6