

## 相关主题

RECOMMEND ARTICLE

- ▶ 在ASP.NET 2.0中建立站点导航层次
- ▶ JSP和JSF双剑合并 打造完美Web应用
- ▶ 用ASP.NET2.0在数据库中存储二进制文件
- ▶ ASP.NET中上传文件到数据库
- ▶ ASP.NET定制简单的错误处理页面
- ▶ 基于JSF技术的WEB应用开发研究
- ▶ ASP.NET实现投票结果的图片进度条显示
- ▶ 圣殿骑士PHP 2007年Web开发技术预言

[MORE](#)

## 推荐文章

RECOMMEND ARTICLE

- ▶ 数据广播方案的优化
- ▶ 网络游戏的位置同步
- ▶ 游戏音乐制作案例之《战火 红色警戒》音效制作揭秘
- ▶ 英雄连Online 原画
- ▶ 游戏音乐制作案例之《乱武天下》
- ▶ 游戏音乐制作案例之《诛仙》
- ▶ 《鹿鼎记》最新原画
- ▶ MIDP2.1规范的新特性

[MORE](#)

## 热门文章

HOT ARTICLE

- ▶ [电子书下载]游戏设计 — 原理与实践
- ▶ [电子书下载]网络游戏开发
- ▶ 游戏设计全过程
- ▶ [电子书下载]游戏设计技术
- ▶ [电子书下载]游戏设计理论
- ▶ CS游戏人物模型制作教程
- ▶ CG人物插画基本流程
- ▶ [转贴]MAX高级人头教程

[MORE](#)

您的位置: WEB技术



文章标题	ASP.NET中如何防范SQL注入式攻击		
来源:	[ ogdev ]	浏览:	[434]

### 一、什么是SQL注入式攻击?

所谓SQL注入式攻击,就是攻击者把SQL命令插入到Web表单的输入域或页面请求的查询字符串,欺骗服务器执行恶意的SQL命令。在某些表单中,用户输入的内容直接用来构造(或者影响)动态SQL命令,或作为存储过程的输入参数,这类表单特别容易受到SQL注入式攻击。常见的SQL注入式攻击过程类如:

(1) 某个ASP.NET Web应用有一个登录页面,这个登录页面控制着用户是否有权访问应用,它要求用户输入一个名称和密码。

(2) 登录页面中输入的内容将直接用来构造动态的SQL命令,或者直接用作存储过程的参数。下面是ASP.NET应用构造查询的一个例子:

```
System.Text.StringBuilder query = new System.Text.StringBuilder(
"SELECT * from Users WHERE login = '"
.Append(txtLogin.Text).Append("' AND password=' ")
.Append(txtPassword.Text).Append("'");
```

(3) 攻击者在用户名字和密码输入框中输入"或'1'='1"之类的内容。

(4) 用户输入的内容提交给服务器之后,服务器运行上面的ASP.NET代码构造出查询用户的SQL命令,但由于攻击者输入的内容非常特殊,所以最后得到的SQL命令变成:SELECT \* from Users WHERE login = '' or '1'='1' AND password = '' or '1'='1'。

(5) 服务器执行查询或存储过程,将用户输入的身份信息和服务器中保存的身份信息进行对比。

(6) 由于SQL命令实际上已被注入式攻击修改,已经不能真正验证用户身份,所以系统会错误地授权给攻击者。

如果攻击者知道应用会将表单中输入的内容直接用于验证身份的查询,他就会尝试输入某些特殊的SQL字符串篡改查询改变其原来的功能,欺骗系统授予访问权限。

系统环境不同,攻击者可能造成的损害也不同,这主要由应用访问数据库的安全权限决定。如果用户的帐户具有管理员或其他比较高级的权限,攻击者就可能对数据库的表执行各种他想要做的操作,包括添加、删除或更新数据,甚至可能直接删除表。

### 二、如何防范?

好在要防止ASP.NET应用被SQL注入式攻击闯入并不是一件特别困难的事情,只要在利用表单输入的内容构造SQL命令之前,把所有输入内容过滤一番就可以了。过滤输入内容可以按多种方式进行。

(1) 对于动态构造SQL查询的场合,可以使用下面的技术:

第一:替换单引号,即把所有单独出现的单引号改成两个单引号,防止攻击者修改SQL命令的含义。再来看前面的例子,"SELECT \* from Users WHERE login = '' or '1'='1' AND password = '' or '1'='1'"显然会得到与"SELECT \* from Users WHERE login = '' or '1'='1' AND password = '' or '1'='1'"不同的结果。

第二:删除用户输入内容中的所有连字符,防止攻击者构造出类如"SELECT \* from Users WHERE login = 'mas' -- AND password = ''"之类的查询,因为这类查询的后半部分已经被注释掉,不再有效,攻击者只要知道一个合法的用户登录名称,根本不需要知道用户的密码就可以顺利获得访问权限。

第三:对于用来执行查询的数据库帐户,限制其权限。用不同的用户帐户执行查询、插入、更新、删除操作。由于隔离了不同帐户可执行的操作,因而也就防止了原本用于执行SELECT命令的地方却被用于执行INSERT、UPDATE或DELETE命令。

(2) 用存储过程来执行所有的查询。SQL参数的传递方式将防止攻击者利用单引号和连字符实施攻击。此外,它还使得数据库权限可以限制到只允许特定的存储过程执行,所有的用户输入必须遵从被调用的存储过程的安全上下文,这样就很难再发生注入式攻击了。

(3) 限制表单或查询字符串输入的长度。如果用户的登录名字最多只有10个字符,那么不要认可表单中输入的10个以上的字符,这将大大增加攻击者在SQL命令中插入有害代码的难度。

(4) 检查用户输入的合法性,确信输入的内容只包含合法的数据。数据检查应当在客户端和服务端都执行——之所以要执行服务器端验证,是为了弥补客户端验证机制脆弱的安全性。

在客户端,攻击者完全有可能获得网页的源代码,修改验证合法性的脚本(或者直接删除脚本),然后将非法内容通过修改后的表单提交给服务器。因此,要保证验证操作确实已经执行,唯一的办法就是在服务器端也执行验证。你可以使用许多内建的验证对象,例如RegularExpressionValidator,它们能够自动生成验证用的客户端脚本,当然你也可以插入服务器端的方法调用。如果找不到现成的验证对象,你可以通过CustomValidator自己创建一个。






(5) 将用户登录名称、密码等数据加密保存。加密用户输入的数据,然后再将它与数据库中保存的数据比较,这相当于对用户输入的数据进行了"消毒"处理,用户输入的数据不再对数据库有任何特殊的意义,从而也就防止了攻击者注入SQL

命令。System.Web.Security.FormsAuthentication类有一个HashPasswordForStoringInConfigFile，非常适合于对输入数据进行消毒处理。

(6) 检查提取数据的查询所返回的记录数量。如果程序只要求返回一个记录，但实际返回的记录却超过一行，那就当作出错处理。

本栏目登载此文出于传递信息之目的，如有任何的问题请及时和我们联系！

无任何评论!

<p><b>请您注意:</b></p> <ul style="list-style-type: none"><li><input checked="" type="checkbox"/> 尊重网上道德，遵守《全国人大常委会关于维护互联网安全的决定》及中华人民共和国其他各项有关法律法规</li><li><input checked="" type="checkbox"/> 尊重网上道德，遵守中华人民共和国的各项有关法律法规</li><li><input checked="" type="checkbox"/> 承担一切因您的行为而直接或间接导致的民事或刑事法律责任</li><li><input checked="" type="checkbox"/> 中国网游研发中心新闻留言板管理人员有权保留或删除其管辖留言中的任意内容</li><li><input checked="" type="checkbox"/> 您在中国网游研发中心留言板发表的作品，中国网游研发中心有权在网站内转载或引用</li><li><input checked="" type="checkbox"/> 参与本留言即表明您已经阅读并接受上述条款</li></ul>	<p><b>发表评论:</b></p> <p>昵称: <input type="text"/> <input type="button" value="GO"/></p> <p>联系EMAIL: <input type="text"/></p> <p>    </p> <p>j&lt; j&lt; j&lt; j&lt; j&lt; j&lt;</p> <p><input type="text"/></p>
--	--

[关于我们](#) - [免责声明](#) - [联络热线](#) - [申请链接](#) - [站点地图](#) - [网站帮助](#)

Copyright © 2004-2007 盛趣信息技术(上海)有限公司 All rights reserved.  
OGDEV.NET -- 网络游戏研发网 最佳分辨率 1024×768