第 30卷 第 3期　　昆 明 理 工 大 学 学 报 (理 工 版)　　Vol 30 No. 3
2005年 6月　　Journal of Kunming University of Science and Technology (Science and Technology)　　Jun. 2005

校园网 SQL注入攻击与防范技术研究

戴诗发
(昆明理工大学 财务处,云南 昆明　650093)

摘要: 针对 Web网站的安全问题,着重分析了在校园网建设中所遇到的 SQL注入攻击技术,提出了有效的解决方法,作为应用层的解决方案,所提出的技术已在昆明理工大学财务信息门户中得以采用,取得了较好的效果.

关键词: SQL;校园网;网络财务;网络安全
中图分类号: TP393. 08; TP309. 2　**文献标识码**: A　**文章编号**: 1007 - 855X(2005)03 - 0072 - 04

SQL Injection Attack and Defending Technology

DAI Shi-fa
(Financial Department, Kunming University of Science and Technology, Kunming 650093, China)

Abstract: To solve Web security problems, SQL injection attack techniques encountered in constructing campus Web are analyzed in detail. An effective solution is then put forward and succesfully adopted in the Wed site of Financial Department of Kunming University of Science and Technology.

Key words: SQL; campus network; network finance; network security

0 引 言

经过近 10年来的迅速发展,中国高校信息化由起步阶段已经进入全面发展阶段. 教育信息化建设从初期的以基础设施建设为重点,逐步过渡到以支持教学应用、资源共享和电子校务为主要目标[1]. 现在全国高校 90%以上建设了校园网,大多数高校具有成熟的应用,就是中西部高校都有超过 37%的学校有比较成熟的应用[2]. 随着成熟应用的不断推广,高校的教学、科研、管理与服务等日常运转已经开始依赖于信息系统的支持,信息系统的稳定运行直接影响着高校的工作,因此,处于发展阶段的高校信息化对于安全稳定运行提出了越来越严格的要求.

我校自 2002年以来,着重推进了校园信息化建设,财务处作为学校管理的重要环节之一,如何积极推进"电子校务",利用当今的计算机技术、网络技术与数据库技术等更好地为广大师生员工提供服务,如何为学校评建提供更好的保障与支持,已经成为校财务处信息化建设的一个重要课题.

最近几年,数字校园建设理论日趋成熟,软硬件建设均衡发展,重视管理、运行与服务得到了广泛的认可. 财务处经过近两年的努力,利用当今流行的技术,如 Web、DOT NET以及三层应用架构等,开发完成了昆明理工大学财务信息门户 (http: //money. kmust. edu. cn),门户的建设,为广大师员提供了教职工工资、所得税、项目经费查询,学生学费查询,部门经费到帐查询,以及职工编号、部门编号查询等,有效地解决了很多困扰教职工关于工资、税收、经费等方面的咨询问题,提高了财务处的工作效率,取得了较好的效果.

但是,随着网站的运行,网站同样遭受了来自互联网的大量的黑客攻击,即使在采用当前常见的访问控制列表 (ACL)技术、防火墙技术等基础上,网站仍受到了巨大的安全威胁,其中,最大的麻烦就来自于 SQL注入攻击,因为攻击者利用网站开发者设计开发的漏洞,防火墙技术对这类攻击毫无作用.

根据网络安全的 P2DR模型,从整体安全策略来分析, SQL注入攻击属于应用层攻击,利用程序设计中所存在的漏洞进行攻击,以往的基本包过滤算法,基于状态的检测方法对应用层攻击是完全无效的. 要

收稿日期: 2004 - 03 - 12.
作者简介:戴诗发 (1963～),男,会计师. 主要研究方向:财务管理. **E - mail:** dsf@cnlab. net

解决 SQL注入攻击,就必须在应用层进行充分的防范,确保系统的安全.

1 SQL注入攻击分析

所谓 SQL注入式攻击,源于英文"SQL Injection Attack".目前尚未见对该攻击手段的确切定义,微软技术中心从两个方面进行了描述:(1)脚本注入式的攻击;(2)恶意用户输入用来影响被执行的 SQL脚本.基本的方法就是攻击者把 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.

由于 SQL注入攻击利用的是 SQL语法,使得这种攻击具有广泛性.理论上说,对于所有基于 SQL语言标准的数据库软件都是有效的,包括 MS SQL Server、Oracle、DB2、Sybase、MySQL等.当然,各种软件有自身的特点,最终的攻击代码可能不尽相同. SQL注入攻击的原理相对简单,易于掌握和实施,并且整个 Internet上连接有数目惊人的数据库系统,在过去的几年里, SQL攻击的数量一直在增长.

在网站设计中,后台利用数据库是十分必要的.当前的系统主流模式均为数据库服务、应用服务器与客户端三级,也就是所谓的 B/S三层模式,校财务信息门户采用 Microsoft SQL Server 2000作为数据库,采用 IIS做为应用服务器,利用 IE浏览器做为客户端,这样的应用构架,非常适于推广与建设.但随之也就遇到了 SQL注入攻击.例如,在早期的门户建设中,为了判断来访者是否是一个真实用户,需要在主页上进行"登录".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.当用户填写后,点击"提交",然后由后台的 ASP代码来判断是否为合法用户.

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 username = '")
.Append(txtLogin.Text).Append("' AND password = '")
.Append(txtPassword.Text).Append("'");
```

在正常情况下,这一段代码是没有任何问题的,但是如果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.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,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 username = ''or 1'='1' AND password = ''or 1'='1.'

可以明显看出,这一条 SQL命令无论如果都是正确的, 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.

从上述攻击描述来看, SQL注入攻击是目前所有的防火墙技术都无法防范的.造成 SQL注入攻击容易成功的原因,主要是由于编程时缺乏经验,对安全性缺少必要的认识和防范所致.在财务门户建设的初期,由于编程时部分代码不尽完善,给攻击者带来了众多的可乘之机,这些从整个攻击过程来分析:

1) 即使严格按照 P2DR模型实现的安全体系,利用了多种网络安全设备,如基于包过滤的防火墙,也是完全无法抵御 SQL注入式攻击的.由于提供 HTTP访问服务,就必须要求包过滤防火墙开放 80号端口,允许外部用户对服务器进行 Web访问,那么 SQL注入就可能成功.

2) 攻击是基于应用层的,除非有基于内容的防火墙与防御策略,否则没有办法防止 SQL注入攻击.

3) 只要攻击者能够构造出合理的 SQL命令,所造成的危害是巨大的,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想做的操作,包括添加、删除或更新数据,甚至可能直接删除表.

2 SQL注入的防范

从上面的分析可见,SQL注入是纯应用层攻击,因此,对SQL注入式攻击的防范主要在应用层来解决.

1) 对提交内容进行有效的过滤与判断

SQL攻击时,主要通过表单提交来完成,因此,对用户提交的内容进行有效的检查,可以避免大量的攻击行为的发生.根据SQL攻击的分析,当出现以下符号或字符串时,发生SQL攻击的概率非常高.字符列表如下:

net user
xp_cmdshell
/add
exec master. dbo. xp_cmdshell
net localgroup administrators
select
count
Asc
char
mid
'
:
"
insert
delete from
drop table
update
truncate
from
%

为了彻底克服SQL注入攻击的影响,在昆明理工大学财务信息门户建设中,最终采用了客户端和服务端双重检查的方法,这样的方法与国内其他文献提出的方法相比,具有明显的优点,客户端检查的主要目的是减少网络流量,降低服务器负荷,将一般误操作、低等级攻击与高等级攻击行为区分开来.客户端检查主要采用客户端的脚本语言,如JavaScript,这样可以在第一时间将错误信息反馈给用户.由于在技术上,客户端的检查可能被有经验的攻击者绕开,而将提交的数据直接发往服务端,因此,在服务器端设定二级检查就显得十分必要.为了保证系统安全,最终的程序代码在所有涉及到数据提交的场合,均有如下ASP代码:

```
If Instr(strTemp, "select%20") or Instr(strTemp, "insert%20") or Instr(strTemp, "delete%20from") or Instr(strTemp, "count(") or Instr(strTemp, "drop%20table") or Instr(strTemp, "update%20") or Instr(strTemp, "truncate%20") or Instr(strTemp, "asc(") or Instr(strTemp, "mid(") or Instr(strTemp, "char(") or Instr(strTemp, "xp_cmdshell") or Instr(strTemp, "exec%20master") or Instr(strTemp, "net%20localgroup%20administrators") or Instr(strTemp, ":") or Instr(strTemp, "net%20user") or Instr(strTemp, "'") or Instr(strTemp, "%20or%20") then
Response. Redirect "Error. aspx" //重定向到错误页.显示"不要攻击网站"信息.
```

这样处理后,只要用户提供的内容包含上述信息,就自动地转向报警,从而避免了SQL攻击的可能.

2) 其他防范技术

为了预防万一,在昆明理工大学财务信息门户中,除了在用户提交信息方面进行了有效的过滤判断,还采用了多种方式结合,取得了较好的效果.具体技术如下:

(1)限制控制:充分地利用 SQL SERVER对权限的控制机制.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.

由于 ASP代码的问题,如果在代码中嵌入 SQL命令,SQL注入攻击成功的概率要高得多.在实际编程时,应尽可能地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规范,这样就很难再发生注入式攻击了.

(2)严格编程:必须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.

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.

(3)结果检查: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.

3 结束语

经过上述技术的综合实施,昆明理工大学财务信息门户较好地达到了预期效果,结合现在的 P2DR网络安全模型,在前端利用防火墙进行包过滤与状态检测,关闭除 80端口以外的应用端口,在应用上,严格实施上述技术,结果表明,这样的处理是成功的.昆明理工大学财务信息门户经过了近 1年的稳定运行,根据对 IIS系统日志的分析可以看出,系统抵御了大量的 SQL注入攻击试探,整个系统仍然稳定可靠运行,充分说明本文所述的技术是可行的,虽然在程序设计中复杂性有所增加,但系统的安全性得到了较大的提高,取得了较好的应用效果.

致谢　衷心感谢昆明理工大学信息安全研究所与昆明理工大学信息中心相关同志对整个系统维护的支持!

参考文献:

[1] 新浪科技. 2004年 9大行业信息化趋势报告 [OL/DB]. http://tech.sina.com.cn/it/, 2003-12-24.

[2] 中国计算机报.中西部教育信息化现状及未来需求调查报告 [OL/DB]. http://industry.ccidnet.com/, 2004-2-5

[3] David Litchfield. Web Application Disassembly with ODBC Error Messages [OL/DB]. http://www.nextgenss.com/papers/webappdis.doc

[4] SQL Server Security Checklist[OL/DB]. http://www.sqlsecurity.com/checklist.asp