

一种基于信息安全的 Web 系统 数据库设计与访问策略

——以安徽开放大学网上教学活动安排系统为例

杨建华, 李 斌

(安徽开放大学, 合肥 230022)

摘要:针对当前 Web 系统数据库在设计时存在表名与列名命名简单、易被猜出,数据访问操作使用用户输入内容拼接构建 SQL 查询语句存在的安全隐患问题,提出一种对数据库中数据表、字段统一命名,用户输入内容参数化,数据访问操作封装化的策略。实践结果表明:应用这一策略能有效防止 SQL 注入攻击,规范了数据库访问操作,提升了代码的重用率。

关键词:信息安全;Web 系统;数据库设计;SQL 注入攻击

中图分类号:TP311

文献标识码:A

文章编号:2097-0625(2023)03-0087-05

一、引言

Web 系统无须安装客户端,用户通过浏览器,在设计好的界面上即可与系统进行交互,其本质是对后台数据进行一系列的访问操作。在此过程中,若数据库设计人员缺乏信息安全意识,数据库设计过于简单,容易猜测出数据库中的表名、字段名等敏感信息;若程序开发人员应用程序编写不严谨,直接将用户提交的内容与 SQL 命令的关键词拼接成 SQL 语句^[1],这些都给 SQL 注入攻击打开了方便之门。为防止这种攻击,本文提出一种数据库访问操作封装化策略,在数据库设计时,对表名、列名使用统一的“假名”策略,做好“假名”与“真名”的文档对照;在数据访问时,对数据表中单条记录的增加、修改、查找、删除操作,进行统一封装,对用户提交的内容检测后进行参数化传入 SQL 语句,杜绝显式的 SQL 构造语句出现,从软件开发设计环节减少 Web 数据库系统的 SQL 注入攻击风险。

二、Web 系统简介

Web 系统也叫 Web 应用程序,是一种利用网络

浏览器和网络技术在互联网上执行特定任务的计算机程序。

(一)Web 系统的结构

目前,主流的 Web 系统结构由三部分组成,客户端、Web 服务程序、数据库服务程序^[2],如图 1 所示。

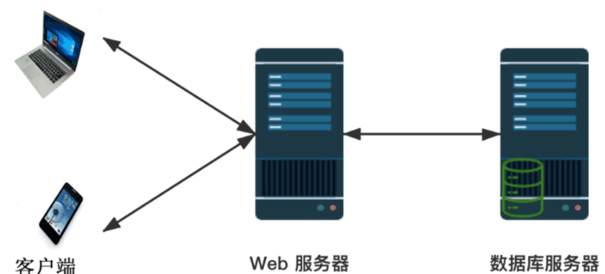


图 1 Web 系统结构图

(二)Web 系统工作流程

Web 系统工作时,客户端通过浏览器发出 http 请求,由 Web 服务器上部署好的对应 Web 应用程序进行相应处理,如果在处理的过程中需要调用数据库,Web 服务器会把请求发给数据库服务器,数据库服务器通常根据收到的 SQL 语句对指定数据库中的

收稿日期:2023-02-13

基金项目:安徽省高校自然科学重点项目(项目编号:KJ2020A1221);安徽省质量工程项目(项目编号:2020jyxm0264)

作者简介:杨建华(1979—),男,安徽无为,人,讲师。研究方向:软件开发、计算机视觉。

数据进行处理,并将处理的结果反馈给 Web 服务器,由 Web 服务器把最终的处理结果以网页的形式显示在客户端的浏览器上。

(三) Web 系统的安全问题

Web 系统是开放式的平台,互联网上的用户都可以访问它,在带给用户方便的同时,其自身的安全性问题也越来越突出,针对 Web 系统漏洞的网络攻击也一直没有停止过。从攻击对象来分,可以分为前端漏洞攻击与后端漏洞攻击。前端攻击主要是通过运行在客户端上的脚本语言 JavaScript 代码来进行,如跨站脚本攻击(XSS)、跨站伪造请求(CSRF)、点击劫持(Clickjacking)、网络钓鱼(Phishing)^[3];Web 系统的后端漏洞攻击,不再针对 Web 系统的用户,而是 Web 服务器,其中最常见的后端漏洞为 SQL 注入,另外还有 Web 系统的文件上传漏洞、远程访问控制漏洞等。

SQL 注入,即结构化查询语言(Structured Query Language,SQL)注入,就是把恶意 SQL 命令插入到统一资源定位符(UniformResource Locator,URL)中或 Web 表单的输入域中,欺骗服务器执行恶意 SQL 命令以达到攻击目的^[4]。若 Web 系统的数据库设计人员缺少安全意识,使用简单的英文单词或汉语拼音作为表名或列名,这些容易被攻击者猜出,为 SQL 注入提供了方便。此外,Web 系统的开发人员若没有对用户提交的数据进行充分验证,直接将其应用到 SQL 语句中,也会增加 SQL 注入风险。如在常见的用户登录操作时,通常构建的 SQL 语句为“Select * from users where name='* * *' and password='* * *'”,若用户在输入密码的后面加上“or '2'='2'”,则无论用户名与密码是否正确,查询条件都将会为真,从而返回用户表信息,导致用户表信息泄漏,造成数据被修改等严重后果。如在 2018 年,某高校学生利用学校教务系统的漏洞,通过所掌握的网络技能侵入数据库,对自己不及格的科目成绩进行修改,使其全部成绩都在合格线以上。

三、安徽开放大学网上教学活动安排系统

(一) 系统简介

2019 年新冠疫情发生后,安徽开放大学将传统面授教学转为线上教学,开发了网上教学活动安排系统,供教师填报直播教学课的时间、直播平台、直播网址等信息,学院管理人员审核通过后,及时向全省学

员发布。系统数据库使用 SQL Server2005 进行存储,开发工具为 Visual Studio,编程语言选择 C# 语言。

(二) 系统主要功能

系统主要面向学员、教师、审核员、系统管理员四类角色,实现了查询、填报、审核等主要功能,系统功能模块如图 2 所示。

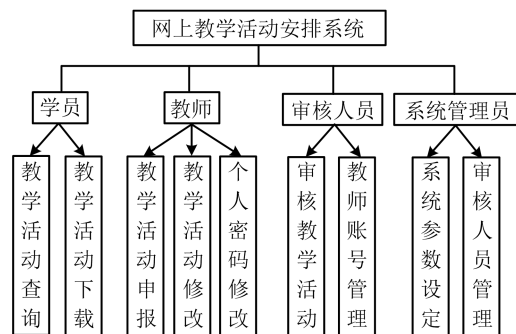


图 2 网上教学活动安排系统模块结构

四、系统数据库设计

(一) 数据表的命名策略

在 Web 系统的数据库设计中,新手很喜欢根据现实场景给数据表命名,比如,将存储用户账号信息的表命名为 users 或 yhb,此类名称很容易被别有用心的猜中。在安徽开放大学教学活动安排系统中,采用了一种给数据表取别名或“假名”策略,如表 1 所示,列举了系统中的一些数据表实际命名。

表 1 教学活动安排系统部分表名

| 数据库中 表名(别名) | 现实中表名 | 含义 |
|----------------|---------|--------------------|
| tbl_01 | 系统参数表 | 存储系统开放时间,当前学期 |
| tbl_02 | 系统用户表 | 存储当前系统中的账号信息 |
| tbl_03 | 教学活动记录表 | 存储系统中所有的 教学活动记录 |
| tbl_04 | 学院代码表 | 存储学校各学院信息 |
| ... | ... | ... |

在实现 Web 系统库设计时,表的命名可以按上述规则编号命名,做好“假名”与真实意义表名的对照文档维护,没有文档参考,外人将很难理解每个表所对应的实际含义,从表名进行猜测攻击将很难实现。

(二) 数据表的列命名策略

对上述数据库中的表,所有列的命名按如下规

则, c_1, c_2, c_3, \dots , 为后继操作的方便, 可将所有表中的第一列, 即 c_1 所在列设置为整型, 标识增量与标识种子设置为 1, 由系统自动分配值, 作为数据表的关键字。如在安徽开放大学教学活动安排系统中, 教学活动记录表, 即 tbl_03 表中部分列的命名如表 2 所示。

表 2 教学活动记录表中部分列的命名

| 列名 | 含义 | 数据类型 | 数据长度 |
|-------|----------------|----------|------|
| c_1 | 数据行 id, 系统自增量 | int | 4 |
| c_2 | 用户 id, 表示活动提交者 | int | 4 |
| c_3 | 专业名称 | nvarchar | 100 |
| c_4 | 课程名称 | nvarchar | 50 |
| ... | ... | ... | ... |

为系统中所有表编写上述定义文档, 为后续数据封装操作做好准备工作, 采取这样的统一命名策略, 可有效防范字段名被猜测出而引起的 SQL 注入风险。同时, 也为设计系统的数据访问封装策略做好了前期准备。

五、数据库的访问

(一) 数据访问的总体封装策略

计算机软件产品的规范化设计是软件能否有效运作的前提^[5]。受开发人员经验、素质及系统规模等因素影响, 很多 Web 系统在开发过程中存在一定的随意性。特别是在对数据库进行访问操作时, 没有编写统一的函数接口, 各人随意发挥, 在需要的地方根据自己的喜好构造 SQL 语句, 普遍存在将用户的输入值直接拼接到 SQL 语句中或跳转链接中的情况。这将给系统的安全留下隐患, 也不利于系统测试与维护。

在安徽开放大学教学活动安排系统的代码实现阶段, 对 Web 系统数据库的访问操作, 采用了面向对象的封装技术, 在代码中对所有的数据表实现了相应的实体类, 编写了根据列名得到对应的字段在数据库系统中的数据类型与数据长度的成员函数。在此基础上, 实现了对所有数据表都通用的单条数据的增、改、删、查功能接口函数。调用时, 对数据的增加与修改操作, 只需要在内存中构造相应的数据表表头与数据行内容; 对数据的删除与查找操作, 只需要提供表名与 c_1 列的值即可。构造 SQL 语句过程由接口自动生成, 无须手工一次次编写 SQL 查询语句, 代码中

将不显式的出现 SQL 语句。采用封装的数据访问策略, 从软件开发自身减少 SQL 注入漏洞, 有利于软件开发的规范化, 有利于代码的重用与维护。

(二) 建立数据库表对应的实体类

为系统中需要访问的数据表建立对应的实体类, 当数据表数量少时可以手工编码, 当系统数据表数量多时, 可下载数据库表转换实体类工具^[6]指定数据表, 选择列名, 生成器能自动生成相应的 C# 实体类的代码, 如图 3 所示。

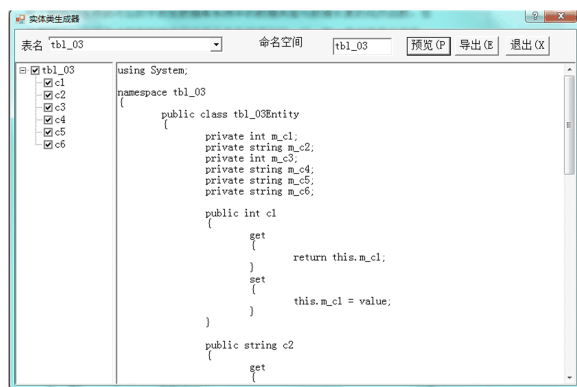


图 3 使用 C# 数据库表转实体类工具生成教学活动记录表实体类

给生成好的数据库表实体类添加获取字段类型与字段长度接口函数, 为后续生成数据访问的统一接口函数服务, 如实现获取表字段在数据库中定义的类型函数, 其实现的伪代码如算法 1 所示, 获取字段在数据库中的长度算法与算法 1 类似, 此处省略。

算法 1: 获取字段在数据库中定义的类型

input: col_name 列名称

output: $type$ 字段类型

function GET_DATECOLUMNTYPE(col_name) /*

根据数据库表中的列名, 返回对应的数据类型 */

```

 $type \leftarrow SqlDbType.Int$ 
switch ( $col\_name$ )
case "c1":
     $type \leftarrow SqlDbType.Int$ 
    break
case "c2":
     $type \leftarrow SqlDbType.Int$ 
    break
.../* 处理其他字段 */
default:
     $type \leftarrow SqlDbType.Int$ 
    break

```

```

end switch
return type
end function

```

(三)封装数据库表中对单条数据的访问操作

对 Web 系统的数据访问,本质上是对数据库服务器中数据表内容的访问,最频繁的操作是对数据表进行以行为单位的操作,如对数据的增、删、改、查这些共性的操作,可以写成数据表的基类函数,供所有数据表实体类继承使用。

1. 数据行的增加与修改操作策略

表 4 增加、修改教学活动记录时生成的临时数据行

| 操作 | C1 | C2 | C3 | C4 | C5 | C6 | ... |
|----|-------|-----|----|----------|------|-----------------------------|-----|
| 增加 | | 378 | 1 | 计算机科学与技术 | 软件工程 | 2021 年 4 月 19 日 19:00—20:00 | ... |
| 修改 | 26937 | | | | 数据结构 | 2021 年 4 月 25 日 19:00—20:00 | ... |

表 4 中,增加表示在教学活动记录表中增加一行数据,修改表示对某行数据中课程名称与直播课时间字段值进行修改。临时数据行只需提供表名、列名及相应的值即可,接口函数通过动态构造 SQL 语句,再调用 ADO.NET 中 SqlCommand^[8]对象的 Parameters.Add()函数,将列值作为参数传给 SqlCommand 对象的参数集,实现增加与修改功能。其实现的伪代码如算法 2 所示。

算法 2:增加数据行操作

```

input: row 要增加的数据行
output: j 增加操作是否执行成功
function INSERT_ONEDATA(row)
    query1 ← string.Format("insert into {0}(", row,
Table.TableName)
    query2 ← " values("
    first ← true /* 处理首列,其他列前面加上逗号 */
    for i = 0 → row.Table.Columns.Count-1 do /* 逐
列处理,构造 */
        if (! first) then /* insert into 表名(列) values(值)
SQL 语句 */
            query1 ← query1 + string.Format(",")
            query2 ← query2 + string.Format(",")
        end if
        first ← false;
        query1 ← query1 + string.Format("{0}", row.Table.Columns[i].ColumnName)
        query2 ← query2 + string.Format("@p" + i.ToString()

```

当用户与 Web 系统交互,进行数据信息的提交或修改操作时,系统会对用户填写的信息进行验证,最终得到一行数据,增加操作就是将这行数据插入到数据库表中;修改操作时,就是用这行数据更新数据表中某行对应位置的列值。为此,在实际 Web 系统开发时可以用 C# 代码动态地构造一条 DataRow^[7]类型的临时数据行,其表结构与要进行操作的数据库中表结构一致。如在安徽开放大学教学活动安排系统中,教师进行教学活动填报与修改时,系统会构造出如表 4 的临时数据行。

```

myCommand.Parameters.Add("@p" + i.ToString()
(),获取列类型,获取列长度),列名称)
myCommand.Parameters["@p" + i.ToString()].
Value ← row[i]
end for
query1 ← query1 + string.Format(")")
query2 ← query2 + string.Format(")")
query ← query1 + query2
myCommand.CommandText ← query
j ← myCommand.ExecuteNonQuery()
return Convert.ToBoolean(j)

```

算法 2 中获取列的类型,可根据表的列名,调用算法 1 即可得到,获取列长度同理也可得到。算法 2 的优点在于无论参数 row 中的列数有多少,都能动态构造出向数据表中添加一行数据的 SQL 语句,增强了接口的泛化能力,提高了代码的通用性。如程序执行增加一条教学活动记录操作时,生成动态 SQL 语句为:insert into tbl_03(c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14,c14,c16) values(@p0,@p1,@p2,@p3,@p4,@p5,@p6,@p7,@p8,@p9,@p10,@p11,@p12,@p13,@p14)。对数据的修改操作,动态 SQL 语句的构造与增加操作类似,可参照算法 2 实现,此处不再赘述。

2. 数据行的查找与删除操作策略

相对于数据行增加与修改,数据行的查找与删除操作相对容易实现。由于设计表时,第一列统一设置为系统自增变量,所以只需要向接口函数提供表名与 c1 列的值,就能构造出相应的 SQL 语句。对多行数

据的查找与删除,只需要修改 SQL 语句的条件部分即可。

3. SQL 参数化查询策略

在以上对单条数据的操作中,虽然都动态构造了 SQL 语句,但对用户的输入内容,都通过 SqlCommand 对象的 Parameters 参数集形式进行处理。首先,用户输入的内容要符合程序中定义的数据类型与数据长度要求,这将过滤掉大部分的非法输入;其次,在使用参数化查询的情况下,数据库服务器不会将参数的内容视为 SQL 指令的一部分来直接进行处理,而是在等数据库完成 SQL 指令的编译后,才代入参数值运行,所以,就算参数中含有 SQL 注入的非法指令,也不会被数据库所执行,从而能有效防范 SQL 的注入攻击。而传统的做法都是对用户输入的内容直接拼接构造 SQL 语句,如插入一条课程活动记录时的 SQL 语句为:insert into kcb(用户 id,活动

类型,专业名称,课程名称...)values(378,1,'计算机科学与技','软件工程',...),用户输入的内容将显式地呈现出来,在向服务器提交的过程中容易泄漏信息,且有可能会被 SQL 注入。

六、结语

对数据库的表名、列名采用别名的策略,能有效降低表名、列名被猜测出来的风险,对数据行的基本访问操作,通过构建动态 SQL 语句与用户输入信息参数化相结合的策略,实现封装各类操作的统一接口,程序执行时生成的 SQL 语句将隐藏 Web 系统实际业务中的表名、列名、列值所代表的具体真实含义,能有效防范 SQL 的注入攻击,也有利于编码的规范化。在采用此策略的基础上,对一些敏感、机密的数据库表,可对表内容数据进行加密,将能进一步提升数据库系统的信息安全。

参考文献:

- [1] 翟宝峰. SQL 注入攻击的分析与防范[J]. 辽宁工业大学学报(自然科学版),2021,41(3):141-143.
- [2] 李明. Web 应用 SQL 注入漏洞分析及防御研究[J]. 福建电脑,2020,36(5):25-27.
- [3] 潘志岗. 互联网企业 Web 系统易忽视漏洞分析[J]. 信息安全研究,2020,6(2):181-187.
- [4] 陈刚,逯柳. Web 系统安全问题与防护机制研究[J]. 无线互联科技,2019,16(15):108-109.
- [5] 陈妍. 计算机软件开发的规范化探析[J]. 软件,2013,34(7):33-34.
- [6] liyizhi_tb. C# 数据库表转实体类[CP/DK]. (2015-11-10)[2022-02-18]. <https://download.csdn.net/download/lyzgored/9258559>.
- [7] 马丽艳,郭子平,程慧芬. 数据库英文字段的中文显示研究[J]. 计算机应用与软件,2007(4):168-170.
- [8] 于磊. 基于 C# 的 WinForm 开发中存储过程应用研究[J]. 软件导刊,2018,17(4):178-179.

Design and Access Strategy of Web System Database Based on Information Security:

Taking Anhui Open University Online Teaching Activity Arrangement System as an Example

YANG Jianhua, LI Bin

(Anhui Open University, Hefei 230022, China)

Abstract: In view of the existing security problems in the design of the current Web system database, such as the simple naming of table names and column names, which are easy to guess, and the use of user input content splicing to construct SQL query statements in data access operations, a strategy of unified naming of data tables and fields in the database, parameterization of user input content, and encapsulation of data access operations is proposed. By applying this strategy in the online teaching activity arrangement system of Anhui Open University, it is concluded that it can effectively prevent SQL injection attacks, standardize database access operations, and improve the code reuse rate.

Keywords: Information security, Web system, database design, SQL injection attacks

[责任编辑 李潜生]