文章编号:1672 - 058X(2012)05 - 0073 - 05

# 一种基于 C#的 SQL 服务器安全的监控方法*

陈 建 华
(广东松山职业技术学院 计算机系,广东 韶关 512126)

**摘　要**:为保障 SQL 数据库服务器的安全,提出了一种基于 C#的 SQL 服务器安全的监控方法,实现对 SQL SERVER 数据库服务器的动态监控,通过及时监控 SQL 数据库服务器的变化,了解 SQL 数据库服务器的安全状态,分析变化原因,及时采取措施,从而保障 SQL 数据库服务器的安全;如何保障数据库的安全,提出了一个新的思路,用程序实现对数据库的监控和管理,这是数据库服务的安全管理方面和安全防范策略的一种补充和延伸,在实际工作中具有重要意义。

**关键词**:SQL 服务器;数据库权限控制;数据表监控
**中图分类号**:TP311.1　　**文献标志码**:A

## 1　SQL 的安全体系

### 1.1　SQL Server 的安全性体系四层结构

SQL Server 的安全性体系由 4 层构成:第 1 层操作系统级的安全性、第 2 层服务器级的安全性、第 3 层数据库级的安全性以及第 4 层表和列级的安全性[1]。SQL SERVER 的安全控制策略是一个层次结构系统的集合。只有满足上一层系统的安全性要求之后,才可以进入下一层[2]。图 1 给出了 SQL SERVER 安全控制策略示意图。

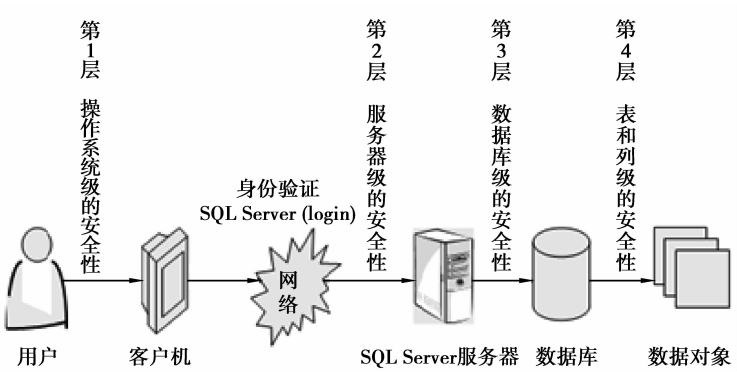


图 1　SQL Server 安全性控制策略示意图


图 2　获取登录帐号

**收稿日期**:2011 - 10 - 10;**修回日期**:2011 - 11 - 07.
* **基金项目**:中国教育学会教育机制研究分会"十一五"科研项目(2010KC102).
**作者简介**:陈建华(1979-),女,湖南汉寿人,讲师,从事计算机及应用研究.

### 1.2 操作系统级安全性

在用户使用客户计算机通过网络实现对 SQL Server 服务器的访问时,用户首先要获得客户计算机操作系统的使用权。一般来说,在能够实现网络互连的前提下,用户没有必要向 SQL Server 服务器的主机进行登录,除非 SQL Server 服务器就运行在本地计算机上。SQL Server 可以直接访问网络端口,所以可以实现对 Windows NT 或 Windows 2000 Server 安全体系以外的服务器及其数据库的访问。操作系统的安全性是操作系统管理员或网络管理员的任务,所以主要考虑从余下 3 层考虑去监控 SQL Server 数据库服务器的安全。

## 2 监控服务器级的安全性

SQL Server 服务器的安全性是建立在控制服务器登录帐号和口令的基础上的。SQL Server 采用了标准的 SQL Server 登录和集成 Windows 登录两种方法。无论是哪种登录方式,用户在登录时提供的登录帐号和口令决定了用户能否获得对 SQL Server 服务器的访问权,以及在获得访问权后用户可以利用的资源[3]。设计和管理合理的登录方式是 SQL Server 数据库管理员 DBA 的重要任务,在 SQL Server 的安全体系中,DBA 是发挥主动性的第一道防线。SQL Server 事先设计了许多固定的服务器角色,可供具有服务器管理员资格的用户分配和使用,拥有固定服务器角色的用户可以拥有服务器级的管理权限。

### 2.1 监控登录帐号

当 SQL SERVER 创建注册用户后,用户就能合法进 SQL SERVER,注册用户信息会放在 master 数据库中的 syslogins 表中。但只有注册用户成为某一数据库用户,并且对用户赋予某些权限时,注册用户才能在限制条件下使用数据库中的表。身份验证是指当用户访问系统时,系统对该用户的账号和口令的确认过程。身份验证的内容包括确认用户的账号是否有效、能否访问系统、能访问系统的哪些数据等。SQL SERVER 的验证是把一组帐户、密码与 Master 数据库 Sysxlogins 表中的一个清单进行匹配,Syslogins 是 Master 数据库中的一个系统视图,用于提供该 SQL 实例上的帐号相关信息,其 Base Table 为 Sysxlogins,此表记录了系统的所有登录帐户的有关信息[4]。使用如下方法可以监控 SQL SERVER 中的登录账号,如图 2 所示。

```
private void button1_Click_1(object sender,EventArgs)
{
    sqlConnection1.ConnectionString = "Data Source = hkm;Initial Catalog = master;Integrated Security = True";
    sqlConnection1.Open();
    String sqlstr = "select name from sysxlogins";
    sqlDataAdapter1.SelectCommand.CommandText = sqlstr;
    sqlDataAdapter1.SelectCommand.Connection = sqlConnection1;
    sqlDataAdapter1.Fill(dataSet1,"sysxlogins");
    dataGridView1.DataSource = dataSet1.Tables["sysxlogins"];
    sqlConnection1.Close();
}
```

### 2.2 监控数据库权限控制

SQL SERVER 数据库服务器拥有所有权限的帐号是 Sa,其他还有 Sysadmin,Db_owner 等不同权限帐号。但是,SQL SERVER 数据库最高权限帐号 Sa 的默认密码是空,就会给数据带来毁灭性的灾难。恶意攻击者可以修改,删除所有数据,更重要的是 SQL SERVER 帐号可以利用扩展执行系统命令。可以利用如下 SQL 查询语句来检查所有帐号是否密码为空,检查对存储过程和扩展存储过程的执行权,提防不必要的执行权限扩散,尽可能的删除存储扩展,防止本地用户利用存储扩展执行恶意命令。

(1) 检查所有帐号,是否密码为空或者过于简单,在连接数据库 master 后,将下面 SQL 语句传递到程序中,就可以检查用户密码是否为空:

Select name,Password from syslogins where password is null。

(2) 用下面语句对所有帐号,检查对存储过程和扩展存储过程的执行权,提防不必要的执行权限扩散。

Select sysobjects. name From sysobjects,sysprotects Where sysprotects. uid = 0 AND xtype IN ('X','P') AND sysobjects. id = sysprotects. id。

(3) 检查除 SA 以外的系统帐号及建立更新时间。

Select name,createdate,updatedate from sysusers where issqluser = 1。

## 3　监控数据库级的安全性

在用户通过 SQL Server 服务器的安全性检查以后,将直接面对不同的数据库入口。这是用户接受的第 3 次安全性检查。默认情况下,只有数据库的所有者才可以访问该数据库内的对象,数据库的所有者可以给其他用户分配访问权限,以便让其他用户也拥有针对该数据库的访问权。一个用户在取得合法的登录帐号,只表明该帐号可以通过 Windows 认证或 SQL Server 认证,其在当前服务器上可以访问哪些数据库,以及对数据库内的数据及数据对象进行哪些操作,与该帐号对应的数据库用户所有的权限有关[5]。SQL Server 提供了许多固定的数据库角色,可以用来在当前数据库内向用户分配部分权限。同时,还可以创建用户自定义的角色,实现特定权限的授予。

### 3.1　监控能访问该数据库的所有用户

在此可以获取到能访问该数据库的所有用户,通过动态监控所有用户的变化,了解 SQL SERVER 数据库服务器的安全状态。在对应数据库中,Sysusers 表中存放着能访问该数据库中的所有用户信息,在上面获取 SQL SERVER 数据库服务器中的所有数据库后,选择指定数据库,通过 Sysusers 表可以获取到能访问该数据库的所有用户,可以通过如下 SQL 语句来访问对应数据库中的 Sysusers 表。

Select name,createdate,updatedate from sysusers where issqluser = 1

其中 name,createdate,updatedate 分别表示能访问该数据库的用户名、建立时间、更新时间;issqluser 如果该帐户是 SQL SERVER 用户,则为 1。

利用上面 SQL 语句,再使用如下方法,将访问该数据库的所有用户列举出来,如图 3 所示。

```
sqlConnection1. ConnectionString = " Data Source = hkm;Initial Catalog = 学生宿舍管理;Integrated Security = True" ;
sqlConnection1. Open( ) ;
String sqlstr = " Select name,createdate,updatedate from sysusers where issqluser = 1" ;
sqlDataAdapter1. SelectCommand. CommandText = sqlstr;
sqlDataAdapter1. SelectCommand. Connection = sqlConnection1 ;
sqlDataAdapter1. Fill( dataSet1," sysusers" ) ;
dataGridView1. DataSource = dataSet1. Tables[ " sysusers" ] ;
sqlConnection1. Close( ) 。
```

### 3.2　监控 SQL SERVER 所有数据库的变化

可以通过获取 SQL SERVER 数据库服务器中所有数据库,监控其是否发生变化,从而了解数据库服务器的安全状态。只要有权限,用户就可以操作指定内容,如建立、删除、修改数据库,建立数据表更新数据等。SQL SERVER 安装完毕后默认安装了 master、model、msdb、northwind、pubs、tempdb 6 个系统数据库,用户可以建立自己的数据库,用户数据库和系统数据库的相关信息均保存在系统数据库 Master 的 Sysdatabases

表中[6]。因此,使用下面的SQL语句可以查询SQL SERVER服务器中用户数据库。


图3 获取能访问对应数据库的所有用户


图4 获取SQL SERVER的所有数据库

```
Select name,createdate,filename from sysdatabases where status =16。
```

其中name、crdate、filename分别表示数据库名、建立时间、数据库对应文件的存放路径;status =16表示是用户数据库,status <>16表示系统数据库。

使用如下方法将SQL SERVER服务器中所有数据库信息列表显示,如图4所示。

```
sqlConnection1. ConnectionString = " Data Source = hkm;Initial Catalog = master;Integrated Security = True" ;
sqlConnection1. Open( ) ;
String sqlstr = " select name,crdate,filename from sysdatabases" ;
sqlDataAdapter1. SelectCommand. CommandText = sqlstr;
sqlDataAdapter1. SelectCommand. Connection = sqlConnection1 ;
sqlDataAdapter1. Fill( dataSet1 ," sysdatabases" ) ;
dataGridView1. DataSource = dataSet1. Tables[ " sysdatabases" ] ;
sqlConnection1. Close( ) 。
```

## 4 监控表和列级的安全性

数据库对象的安全性是核查用户权限的最后一个安全等级。在创建数据库对象时,SQL Server自动将该数据库对象的所有权赋予该对象的创建者。对象的所有者可以实现以该对象的完全控制。

默认情况下,只有数据库的所有者可以在该数据库下进行操作。当一个普通用户想访问数据库内的对象时,必须事先由数据库的所有者赋予该用户关于某指定对象的指定操作权限。例如,一个用户想访问某数据库表的信息,则必须在成为数据库的合法用户的前提下,获得由数据库所有者分配的针对该表的访问许可。

### 4.1 监控对应数据库的相关表的变化

在对应数据库中,Sysobjects中存放着该数据库中的所有数据表的信息,获取SQL SERVER数据库服务器中的所有数据库后,选择指定数据库,只要随时动态监控系统数据库的相关表的变化,就可以随时了解SQL SERVER数据库服务器的安全状态。下面的查询语句可以得到该数据库中的用户表。strSQL = "SELECT name FROM sysobjects WHERE xtype ='U',下面的查询语句可以得到该数据库中的系统表。strSQL = "SELECT name FROM sysobjects WHERE xtype ='S'。如果要得到所有表,去掉WHERE语句即可。再使用如下方法显示对应数据库中的所有的数据表列表,如图5所示。

```
sqlConnection1. ConnectionString = " Data Source = hkm;Initial Catalog = 学生宿舍管理;Integrated Security = True" ;
sqlConnection1. Open( ) ;
String sqlstr = " select name FROM sysobjects where xtype = 'U'" ;
sqlDataAdapter1. SelectCommand. CommandText = sqlstr;
sqlDataAdapter1. SelectCommand. Connection = sqlConnection1 ;
sqlDataAdapter1. Fill( dataSet1 ," sysobjects" ) ;
dataGridView1. DataSource = dataSet1. Tables[ " sysobjects" ] ;
sqlConnection1. Close( ) 。
```


图 5　获取对应数据库中的所有数据表


图 6　显示对应数据表中的内容

## 4.2　监控数据表的相关内容的变化

选择相应数据表,动态显示数据表中的内容。以了解服务器数据是否发生变化,分析变化原因,及时采取措施。使用如下方法显示对应数据表中的内容,如图 6 所示。

```
sqlConnection1. ConnectionString = " Data Source = hkm;Initial Catalog = 学生宿舍管理;Integrated Security = True" ;
sqlConnection1. Open( ) ;
String sqlstr = " select * FROM 宿舍表" ;
sqlDataAdapter1. SelectCommand. CommandText = sqlstr;
sqlDataAdapter1. SelectCommand. Connection = sqlConnection1 ;
sqlDataAdapter1. Fill( dataSet1 ) ;
dataGridView1. DataSource = dataSet1. Tables[ 0 ] ;
sqlConnection1. Close( ) 。
```

**参考文献:**

[1] MICHAEL O. ADO. NET 技术参考大全[M]. 北京:清华大学出版社,2003
[2] 李志中. Visual C# 2008 数据库编程实训教程[M]. 北京:清华大学出版社,2010
[3] 闪四清. SQL Server 实用简明教程(第二版) [M]. 北京:清华大学出版社,2005
[4] 耿肇英. C#应用程序设计教程[M]. 北京:人民邮电出版社,2007
[5] 李岩. SQL Server2005 实用教程[M]. 北京:清华大学出版社,2008
[6] 宋先斌. C#应用于开发[M]. 北京:清华大学出版社,2010

(下转第 84 页)