文章编号:1000 - 582X(2005)02 - 0085 - 04

基于目标的安全管理体系在 MIS 中的应用*

胡小兵[1],袁　锐[2]
(重庆大学 1. 数理学院; 2. 软件学院,重庆　400030)

摘　要:为实现大型 MIS 系统的安全管理,提出了一种基于目标的安全管理体系。对 MIS 系统的用户而言,只能通过用户界面与该系统进行交互,因此,将 MIS 系统中用户界面上需要管理的对象(目标)进行注册,并定义目标各种可能的状态(如可用、不可用、可见、不可见和只读等)。对于某个用户,可为其分配不同的对象集,并定义该集合中对象的状态,使系统对用户权限的控制粒度达到任意程度,从而为 MIS 系统提供了一种灵活的安全管理策略。

关键词:目标;用户界面;访问控制;管理信息系统
中图分类号:TP39　　**文献标识码**:A

MIS 在大、中型企业中的广泛应用,不仅极大地提高了企业的管理效率,同时也为企业的管理者提供决策依据。随着越来越多的企业数据存入 MIS 系统,数据的安全性已成为企业越来越关注的问题,这为 MIS 系统的安全设计提出了一个新的课题。

在 MIS 系统中,安全管理主要针对系统中的资源进行控制与管理,使用户能够访问和修改已授权的资源,但不能访问未授权的资源。在系统安全管理设计时,通常是将数据库中的资源,如表、视图、表中的字段、存储过程等进行登记(也可以直接使用数据库管理系统提供的安全方案,但将会牺牲很多灵活性),并对这些资源的访问进一步细分为增加、删除、更新、查询等操作。当用户登录系统时,首先获得和自己相关的数据库资源的访问控制信息,通过该控制信息实现对后台数据库的访问控制。这种访问模式由于将后台服务器上的资源作为访问控制对象,存在如下的缺点[1-3]:

1) 在 MIS 中,用户所能访问的资源,有些并不能简单地用前面提到的那几种操作来表示,比如,用户界面上某一按钮是否可用、菜单项是否可用等;

2) 对用户的权限管理缺乏灵活性,仅限制对数据增加、删除、修改和查询进行控制,大多数情况下不能满足系统的要求。

笔者在参与"葛洲坝水力发电厂运行与维护管理子系统"的开发中,对 MIS 系统中的安全管理进行了较为深入的思考和探索,提出了基于目标的安全管理体系[4],该安全体系在系统设计中得到了很好的实施,收到了非常满意的效果。

1　基本概念

1.1　目标

定义 1:在 MIS 系统中,需要被管理的命名资源成为目标,目标可以是数据库中的表、视图、存储过程,也可以是用户界面上的某个按钮、文本框。

对于目标,用下面的三元组来标记,即{TargetID, TargetName, Description},其中 TargetID 是目标标识符,TargetName 为目标名称,Description 为目标描述符。为了管理系统中的所有目标,需要建立如表 1 所示的表结构。

表 1　目标表结构

字段名	类型	长度	能否为空
目标标识	字符	6	否
目标名称	字符	30	否
描述	字符	250	能

在一个大型 MIS 系统中,有成千上万个目标,需要将控制的目标存入 Target 表中。而对于绝大多数不

* 收稿日期:2004 - 09 - 12
基金项目:重庆大学基础及应用基础研究资助项目(717411061)
作者简介:胡小兵(1975 -),男,湖北京山人,重庆大学讲师,博士,主要研究方向:计算智能、机器人控制技术、软件工程。

需要控制的目标,作为公共资源,即任何用户都能访问该资源。

1.2 访问控制表

目标只能用来标识某个受控的对象,为了能够记录某个用户及其所能访问的目标集,定义如下的访问控制表。访问控制表用来存储所有用户及其对目标的访问信息。其表结构如表 2 所示。

表 2 访问控制表结构

字段名	类型	长度	能否为空
ID	数值	8	否
用户(UserID)	字符	20	否
目标标识(TargetID)	字符	6	否
控制类型标识(ControlTypeID)	字符	4	否
描述(Description)	字符	250	能

其中 ID 为表的关键字;UserID 为用户帐号;TargetID 为目标标识;ControlTypeID 为控制类型标识;Description 为描述。

1.3 访问类型表

用户对目标的访问有许多种类型,如对某一个命令按钮是否可用,对某一文本框的值是否能做修改操作(即文本框是否为只读);对某标签上的字符串是否能看见等等。为了适应系统对各种不同目标的访问控制,用一张表记录所有能访问的类型,其表的结构如表 3所示。

表 3 访问类型表结构

字段名	类型	长度	能否为空
访问类型标识(AccessTypeID)	字符	4	否
访问类型名称(AccessTypeName)	字符	20	否
访问类型标识(AccessTypeValue)	字符	20	否
描述(Description)	字符	250	能

其中 AccessTypeID 为访问类型标识;AccessTypeName 为访问类型名称,其可能的取值为 Enabled,Visible,ReadOnly 等,根据系统对访问要求的不同,可以相应地增加类型;AccessTypeValue 为访问类型的取值,如对于访问类型名为 Enabled,其取值可以为 true 或者 false; Description 为描述。

2 安全管理的实现

2.1 获得用户权限集

用户登录过程如图 1 所示。当用户登录系统时,首先验证用户的口令,如果口令不正确,要求用户重新输入或者退出系统,否则根据用户帐号从访问控制表

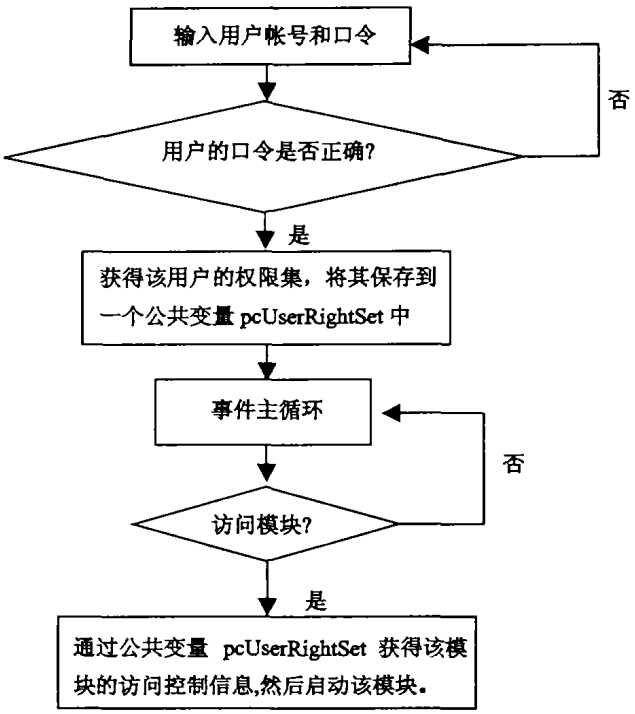

图 1 用户登录过程

中获得用户的访问控制信息,并将其存放到公共变量 pcUserRightSet 之中。函数 GetUserRightSet( ) 实现该功能,其实现代码如下[5-6]。

```
'---参数 sUserID 为登录系统时获得用户的帐号。
Public Function GetUserRightSet( sUserID ) As String
  Dim sTemp As String
  Dim rs As New ADODB. RecordSet
  '定义记录集 rs
  Dim sTemp = ""
  If sUserID < > "" Then
        rs. ActiveConnection = Cnn  'Cnn 为连接串
        '设置连接
        rs. CursorLocation = adUseClient
        rs. CursorType = adOpenStatic
        rs. LockType = adLockReadOnly
        Rs. Source = "SELECT tAccessControl. TargetID, tAccessType. AccessTypeName"&_
        "tAccessType. tAccessTypeValue"&_
        "FROM tAccessControl,tAccessType "& _
        "WHERE tAccessControl. UserID = sUserID "
    '该 SELECT 语句根据用户标识 sUserID 选出所有相关的目标及其访问类型
    rs. Open                    '打开记录集
    IF rs. RecordCount > 0 Then
        '如果记录数大于零,则将相应的目标及其
```

```
访问类型写入变量 sTemp 中
        rs. MoveFirst
        While Not rs. Eof
        Temp = sTemp + rs("TargetID") & "=" & rs("AccessTypeName") & _
                rs("AccessTypeValue") & "|"
            rs. MoveNext
        Wend
    Else
        sTemp = ""
    End If
End If
GetUserRightSet = Left(sTemp,len(sTemp) - 1)
'返回用户 sUserID 的权限字符串
End Function
```

当用户登录成功时，系统调用函数 GetUserRightSet( )获得用户的权限集信息，并把该权限集存放在公共变量 pcUserRightSet 中，以便在访问某一模块时作为用户的访问控制信息。通过函数 GetUserRightSet( ) 的实现可以看出，在公共变量 pcUserRightSet 中存放的信息具有如下的形式：

"TargetID1 = AccessTypeName1 = AccessTypeValue1 | TargetID2 = AccessTypeName2 = AccessTypeValue2 | TargetID3 = AccessTypeName3 = AccessTypeValue3 | ……"

其中 TargetID1，TargetID2…为目标标识，与之相应的 AccessTypeName1，AccessTypeName2…为该目标的访问类型，AccessTypeValue1，AccessTypeValue2…为其类型的取值。在具体实现中，如在公共变量 pcUserRightSet 中有如下的信息："…|000003 = Visible = True|…"则表示该用户对编号为"000003"的目标具有可见的权限，即该目标对该用户是可见的。这样就实现了对该目标的一种访问控制。

2.2　设置用户访问控制

利用系统启动时所获得的权限集（存放在公共变量 pUserRightSet 中），很容易实现用户的权限控制。当用户访问某一个模块时，在该模块的初始化事件中来控制用户的权限。采用下面的过程 SetUserAccessControl 来实现该功能[5-6]。

```
Public Sub SetUserAccessControl (frm AS Object)    '此处的 frm 为某一表单
    Dim sAccessTypeName As String    '定义访问类型名变量
    Dim sAccessTypeValue As String    '定义访问类型值变量

    IF NOT IsObject(frm)
      Exit Sub
    End If
    For Each Ctrl In frm. Controls    '循环搜索表单中的控件
        IF Ctrl. Tag < > ""
            sAccessTypeName = FindAccessTypeName(Ctrl. Tag)
            sAccessTypeValue = FindAccessTypeValue(Ctrl. Tag)
            SELECT CASE sAccessTypeName
                CASE "Visible"
                    If sAccessTypeValue = "true"
                        Ctrl. Visible = true
                    Else
                        Ctrl. Visible = false
                    End If
                CASE "Enabled"
                    If sAccessTypeValue = "true"
                        Ctrl. Enabled = true
                    Else
                        Ctrl. Enabled = false
                    End If
                CASE
                    ……
            END SELECT
        End If
    Next
End Sub
```

在上面的过程中，用到了控件的 Tag 属性。将需要控制的每一个控件的目标编号 TargetID 存放在控件的 Tag 属性中，通过在表单中搜索每一个控件，并对它的 Tag 属进行判断，如果 Tag 属性不为空，则说明该控件的访问需要被控制。由该控件的 Tag 属性（即该对象的目标编号 TargetID），利用函数 FindAccessTypeName(sTargetID) 从公共变量 pcUserRightSet 中找到控制的访问类型名，同时利用函数 FindAccessTypeValue(sTargetID) 找到控制的访问类型值，从而对该控件的访问控制进行设置。

3 结 语

针对大型 MIS 系统中的安全访问进行了较为深入的探讨,提出了一种基于目标的安全管理方案。其基本思想是将用户界面上的对象(主要指各种控件)进行登记管理,使这些目标与每个用户相联系,从而确定该目标对用户的使用权限。该模型由于将用户界面上的每一个元素作为控制对象,可以实现各种粒度要求的访问控制。实践证明该方法具有高度的灵活性和可操作性。

参考文献:

[1] 张晓辉, 王培康. 大型信息系统用户权限管理[J]. 计算机应用, 2000, 20(11):35 - 37.

[2] 潘德锋, 徐少平, 梁庆中, 等. 基于操作的 MIS 多级授权模型的实现[J]. 计算机应用, 2003, 23(11): 100 - 103.

[3] 叶劲风,杨路明. 信息管理系统安全性规划[J]. 长沙电力学院学报(自然科学版),2003, 18(2):28 - 31.

[4] 胡小兵. MIS 系统中安全方案的分析与设计[D]. 重庆: 重庆大学数理学院, 2000.

[5] JOHN PAPA, MATTHEW SHEPKER. SQL Server 7 编程技术内幕[M]. 前导工作室译. 北京:机械工业出版社,2000.

[6] DIANNE SIEBOLD. Visual Basic 开发指南——SQL Server 篇[M]. 丘仲潘 译. 北京:电子工业出版社,2000.

Target-based Security Management Architecture Applied to Management Information System

HU Xiao-bing[1], *YUAN Rui*[2]

(1. College of Mathematics and Science; 2. College of Software, Chongqing University, Chongqing 400030, China)

Abstract: In order to implement the security management of management information system (MIS), a target-based security management architecture is proposed. For MIS users, the user interface (UI) is the only way to interact with the MIS system, so all the objects (targets) which need to be managed on UI is registered first, thus for every target the state such as enabled, not enabled, visible, not visible, readonly etc., is assigned. For some users, a set of targets with some state is assigned so that the management granularity in MIS can reach any degree. This approach provides a very flexible way to the security management in MIS.

Key words: target; user interface; access control; MIS

(编辑 张 苹)