一种基于角色的数据库安全访问控制方案的设计与实现

王秀坤, 杨南海, 张志勇
(大连理工大学 计算机科学与工程系, 辽宁 大连 116023)

摘　要: 在管理信息系统和决策支持系统的研究与广泛应用中,后台数据库的安全性问题至关重要。采用了基于角色访问控制模型的数据库安全性设计方案。该方案利用后台数据库的安全管理机制,结合具体角色实现了权限所及的安全访问控制,并成功地运用于"黑龙江防洪决策支持系统"等多个实际的工程项目中,有效地解决了安全访问控制问题。

关键词: 信息系统; 数据库安全; 角色; 访问控制
中图法分类号: TP311　　文献标识码: A　　文章编号: 1001-3695(2003)03-0087-02

A Scheme Based on RBAC of Database Secure Access Control Designing and Realizing

WANG Xiu-kun, YANG Nan-hai, ZHANG Zhi-yong
(*Dept. of Computer Science & Engineering, Dalian University of Technology, Dalian Liaoning* 116023, *China*)

Abstract: In research and application of Management Information System and Decision Support System, database security is very important. The paper uses a database security scheme based on RBAC. The scheme integrates material roles, realizes secure access control using database security mechanism, and applys successfully in some practical projects including "Heilongjiang Preventing Flood Decision Support System", resolves effectively the question of secure access control.

Key words: Information System; Database Security; Role; Access Control

1　引言

目前,各行各业的企事业单位普遍建立了适合于本单位的管理信息系统(MIS)和决策支持系统(DSS)。MIS 偏重于日常事务处理,DSS 着重于为决策者提供分析和决策支持,但两者的共同点是在系统的后台存放了大量的数据信息。这些信息是整个系统的灵魂,其安全性至关重要。保证数据库中的信息被安全存储、恰当地使用,使得授权的合法用户在权利所及范围内访问数据库,以及跟踪监视数据库中的数据信息的使用过程,是 MIS, DSS 必须具备的数据安全保护功能。本文在实际的工程项目"黑龙江防洪决策支持系统"研发过程中,针对数据库安全问题,采用了一种基于角色访问控制模型的安全访问控制设计方案。

2　基于角色的访问控制模型

关系型数据库的访问控制模型主要有三种:①自主访控模型(DAC);②强制访控模型(MAC);③基于角色的访控模型(RBAC)。RBAC 是美国 Ravi Sandhu 提出的,它解决了具有大量用户、数据客体和各种访问权限的系统中的授权管理问题。其中主要涉及用户、角色、访问权限、会话等概念。用户、角色、访问权限三者之间是多对多的关系。角色和会话的设置带来的好处是容易实施最小特权原则。在 RBAC 模型中,将若干特定的用户集合和某种授权连接在一起。这样的授权管理与个体授权相比较,具有强大的可操作性和可管理性,因为角色的变动远远少于个体的变动。通过引入 RBAC 模型,系统的最终用户并没有与数据对象有直接的联系,而是通过角色这个中间层来访问后台数据信息。在应用层次上角色的逻辑意义和划分更为明显和直接,因此 RBAC 通常使用于应用层的安全模型。鉴于 MIS, DSS 在授权管理和访问控制方面的特点,本文的设计方案中采用了 RBAC 模型。

3　方案的设计与实现

3.1　设计目标

在现有的应用信息系统中主要存在以下问题:①面对大型数据库系统,如 Oracle, Sybase 等,应用系统的用户并非专业的数据库管理人员。由于大型数据库安全管理的复杂和专业化,使得普通用户在使用过程中很容

收稿日期: 2002-06-25; 修返日期: 2002-07-18

易发生误操作,影响了 DBS 本身的安全性能。②现有的一些数据库应用系统中虽然实现了数据库安全访问控制,但设计缺乏灵活性,在实际应用中,随时间和情况的变化,程序的适应能力差。为了解决上述问题,设计目标力求在保证后台数据库安全的前提下,提高软件的可用性,以及操作界面的友好性。

3.2 设计思想

首先,在解决后台数据库安全,减少用户使用过程中误操作的问题上,方案采用了将应用程序和后台数据库管理系统紧密结合。通过在应用程序和 DBMS 之间建立相应的接口,使得用户能够通过简便、友好的界面对相关的数据信息进行安全的访问控制,保障用户操作的安全性。这样,就将后台数据库系统强大的安全管理机制引入了应用系统中。其次,采用动态管理机制提高软件的灵活性。RBAC 很好地解决了大量授权问题,但用户根据本单位的具体情况和实际需求,按不同的职务划分角色,随时间和应用的变化,会发生角色的增加、删除和权限的变化。如果在开发设计过程中事先规划好角色,不能再改变,则影响了程序的适应性和通用性。此外,应用系统功能模块的划分是通过菜单项来体现的。通过严格地对功能模块的授权访问,也保证了后台数据库中数据信息的安全。由于在实际应用中功能项还会发生增删改的变化,因此对功能项实行动态管理。

3.3 方案的具体实现

该方案的实现采用流行的数据库前台应用开发工具 PowerBuilder7.0 和 Oracle 8.1.6。其功能如图 1 所示。

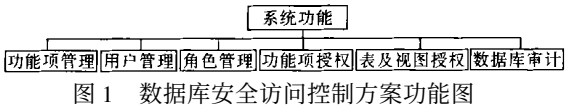

图 1 数据库安全访问控制方案功能图

3.3.1 功能项的动态管理

在数据库中建立功能项(菜单项)信息表 Menutable,其中主要的属性有 Menuitem_id,Menuitem_name,Menuitem_description 和 Menuitem_flag。在应用系统启动时,将系统菜单信息存入 Menutable 中。当菜单项发生增、删、改时,只需调用两个自定义函数,便可对 Menutable 中的菜单项信息进行维护,将新增加的菜单项信息加入 Menutable,且将已不存在的菜单项信息从表中删除,而不需要直接操作这个菜单项信息表。由于菜单的数据结构属于树形结构,因此对菜单项的访问(菜单的动态生成、增、删、改)属于树的遍历。此方案中的功能项管理解决了功能项变化时处理的灵活和简便。而且,最终用户能够访问到的功能项集合是根据用户当前的活动角色的权限进行动态生成的。在功能项管理的具体实现中,定义了一个用户对象的实例 Uo_menu_manage 和它的两个内部函数:Fbrowsemenu,Fbrowsemenu_deal。Fbrowsemenu 利用递归算法实现功能项的按深度优先遍历,Fbrowsemenu_deal 实现对当前节点(功能项)的访问处理。按深度优先遍历的递归算法描述如下:

```
fbrowsemenu(string curstr; ref menu submenu)
li_count= upperbound(submenu.item[])
for i= 1 to li_count
    if curstr="" then
      if fbrowsemenu_deal(submenu.item[i].text, submenu.item[i])= 1
then
        this.fbrowsemenu(submenu.item[i].text, submenu.item[i])
      else
        if fbrowsemenu_deal(curstr+"."+ submenu.item[i].text, sub-
menu.item[i])=1 then
          fbrowsemenu(curstr+"."+submenu.item[i].text, submenu.item
[i])
    end if
```

在 Fbrowsemenu_deal 函数中,进行当前功能项的处理。主要的处理方式有:(1)将该功能项的信息添加到 Menutable 中。(2)如果该功能项是最新加入的,则需加入到 Menutable 中。(3)如果该功能项已不存在,则从数据库中删除。当该功能项是一棵子树的根时,则需要删除整个子树。(4)如果当前用户被赋予了多个角色,为保证最小特权原则,则取其当前选定的活动角色的权限,决定该功能项是否有权被访问(可视或不可视)。(5)将该功能项的默认权限(不可视)赋予新增角色。(6)其它操作。在整个遍历过程中,功能项的标志取其在树中的绝对路径,这样就不再考虑功能项所处层次的问题。

在功能项的生成过程中,要达到完全的动态,即在应用程序中直接调用菜单库,组合成整个菜单,而不需要在模板上定制,还需做进一步的改进。

3.3.2 动态的用户、角色管理和角色授权

用户管理包括对用户信息的增、删、改。用户信息包含用户名、用户具体描述、加密后口令、创建时间及当前状态。在用户登录应用系统时,除对用户名和用户口令进行相符验证外,还需判断用户的当前状态。在建立用户信息时,将用户口令这样敏感的数据进行加密处理(加密算法采用单向函数加密算法),然后存放于用户表 usertable 中。在应用程序中建立一个用户后,在后台数据库系统中也创建相应的数据库用户。同样,应用程序角色的建立也和后台数据库中角色的创建一一对应。

在实际应用程序中,除了一个默认用户(应用系统管理员)事先赋予特权(DBA 角色)和功能模块的全部可视,其他普通用户创建、角色的授权都在应用程序中完成。对角色授予功能项的权限为:功能项可视或不可视。对于普通用户,经常使用的是数据库中的表及视图,因此,在安全设计中仅对系统角色进行表级授权。首先,为角色授予 Oracle 系统权限(Connect 权限),然后再授予所需对象权限(Select 权、Update 权、Insert 权、Delete 权)。角色所需权限分配好后,便可以为用户分配角色。在应用程序中进行授权,方便了数据库管理员的操作,提高了安全性。用户登录通过系统身份验证,连接数据库通过 Oracle DBMS 来验证,实现了应用程序级和 DBMS 级两级的安全管理机制。为管理用户和角色信息及功能项授权,在具体实现中,创建以下主要应用系统表:

(下转第 107 页)

4　可视化算法和框图

程序功能:利用轴线跟踪法围网孔,将屏幕分成两个窗口:一部分图形窗口,另一部分为数据窗口。在轴线跟踪时,查找到一根后继轴时,同时依据后继轴的两端点坐标及计算结果在图形显示窗口绘制当前轴线。取出相关的计算参数,进行计算处理,将计算结果显示在数据窗口中。建筑图形和计算结果也可以利用绘图仪或打印机输出(图 2)。

5　结束语

由讨论可以看出此数学模型计算工程量,其通用性强、条理清晰,在处理过程中能惟一地围每一个网孔,采用轴线跟踪法来计算工程量的各项目必要参数,使工程量的自动计算成为可能。由于我们采用了可视化模型的计算,所显示的图形是与工程量计算同步进行,使用户感到直观,当网孔不封闭时及时报告准确的出错位置,避免了人为造成丢项计算的问题。非正交双向网格图处理是一种独特的数学模型及处理模式,它提高了工程量计算的准确性,具有速度快、可靠性好的特点,解决了建筑工程量计算的主要问题。目前我们正在研究工程量计算与 CAD 接口问题,如果研究成功,那么建筑工程预算全部自动化将成为可能。

本研究课题获建设部一等奖。

参考文献:

[1] 第三届全国建工系统计算机应用学术年会论文集[C]. 1986. 61-72.

[2] 张海藩. 软件工程[M]. 北京:清华大学出版社, 1998. 57-65.

作者简介:

孙静波(1956-),女,副教授,本科,主要研究方向为建筑业计算机应用研究;侯秀萍(1964-),女,副教授,研究生,主要研究方向为软件工程、智能管理;应红霞(1966-),女,实验师,研究生,主要研究方向为计算机应用与软件开发;索东梅(1974-),女,助理工程师,本科,主要研究方向为计算机软件开发与测试。

(上接第 88 页)

(1)用户信息表 create table usertablec

用户标志　char(10) not null primary key
用户名　varchar2(20) not null
用户描述　varchar2(40)
用户加密口令　varchar2(40)
用户创建时间　date
用户状态　char(1)　正常或禁止,默认为正常)

(2)角色信息表 create table roletable

角色标志　char(5) not null primary key
角色名(20) not null
角色描述　varchar2(30))

(3)用户-角色表 create table user _ roletable(

用户标志　char(10) not null references
角色标志　char(5) not null references
当前活动角色标志　char(5) not null)

(4)角色-功能项权限表 create table role _ menu _ priv(

角色标志　char(10) not null references
功能项标志　varchar(40) not null references
功能项权限　char(1) 可视或不可视,默认为不可视)

(5)角色-表对象权限表 create table role _ table _ priv(

角色标志　char(10) not null references
表名　varchar2(20) not null
表权限　char(1)select: S, insert: I, update: U, delete: D)

由于在 PowerBuilder 脚本中,无法直接使用有关创建、删除用户、角色,以及分配、回收角色、对象权限的 SQL 语句或带有变量的动态 SQL 语句,因此在实现过程中,采用了 Execute 语句完成 PowerBuilder 和 Oracle DBMS 的接口。

3.3.3　数据库审计

为了便于数据库管理员查看数据库的使用情况,而又避开 Oracle 8i 复杂的审计管理,通过对后台数据库重要表建立相应的 Delete, Insert, Update 触发器,随时将用户对表进行上述操作时,把相关的操作时间(Opertime)、操作人员(Operuser)、操作方式(Opermode)、操作表名(Opertable)或视图名(Operview)记录在一个审计表(Auditable)中。系统管理人员便可以通过应用程序的界面跟踪监视后台数据库中数据对象的使用情况,及时发现问题,确保数据库安全。

4　结束语

本方案完成了在信息系统中基于角色的授权管理和自主访问控制(DAC),要达到强制访问控制(MAC),防止信息流从高密级流向低密级和实现多级安全访问控制,还需要做进一步的研究。由于目前主流的商用 DBMS 都采用了较强的安全管理机制(部分已基本达到 B1 级安全),将 DBMS 安全管理和应用系统有机地结合,则会提高数据库的安全性能。因此,本方案在实际的工程项目中是一种通用的关于解决数据库安全访问控制问题的方案。

参考文献:

[1] 宋志敏,南相浩,等. 数据库安全的研究与进展[J]. 计算机工程与应用, 2001, (1): 85-87.

[2] 刘启原,刘怡. 数据库与信息系统的安全[M]. 北京:科学出版社, 2000.

[3] Kevin Loney. Oracle8 数据库管理员手册[M]. 李晓军,等. 北京:机械工业出版社, 2000.

作者简介:

王秀坤(1945-),女,教授,研究方向为数据库系统,决策支持系统;杨南海(1970-),男,讲师,硕士,研究方向为数据库系统;张志勇(1975-),男,硕士生,研究方向为数据库系统。