第34卷 第17期 计 算 机 工 程 2008年9月
Vol.34 No.17 Computer Engineering September 2008
·安全技术· 文章编号：1000—3428(2008)17—0183—03 文献标识码：A 中图分类号：TP309

针对基于多父角色 RBAC 模型的研究与应用

史永昌，鲁书喜
(平顶山学院计算机科学与技术学院，平顶山 467000)

摘 要：针对基于角色的访问控制(RBAC)模型中由于继承关系产生的子角色不能拥有私有权限问题进行了研究。当前的解决方案在表示同一机构或相同业务性质的角色共有特定权限方面存在不足，也不能满足多父角色权限继承的要求。对 RBAC 模型进行了扩展，给出一种基于域和域权限的解决方案，并结合实际项目具体分析系统实现权限管理的方法，提出多父角色权限继承的算法，解决了多父角色权限继承问题，在系统的安全管理中实现了基于角色和域的访问控制。

关键词：RBAC 模型；角色；权限；访问控制；域

Research and Application on Multi Father Role Based RBAC Model

SHI Yong-chang, LU Shu-xi
(Institute of Computer Science and Technology, Pingdingshan University, Pingdingshan 467000)

【Abstract】 A problem that child role cannot obtain private permissions because of inherited relation in the Role-Based Access Control(RBAC) model is researched. The specific permission of the roles in same department or similar business, is not discussed in the past solutions, and the permission cannot be inherited by multi father role. Thus a new solution with domain and domain's permission is presented. The method of permission management is analyzed, an algorithm to inherit permissions from one child for multi father roles is provided, and the question of inheritance is solved. The access control theory based on role and domain in the application system is realized.

【Key words】 Role-Based Access Control(RBAC) model; role; permission; access control; domain

访问控制是安全技术的重要部分，而基于角色的访问控制(Role-Based Access Control, RBAC)作为目前主流的访问控制模型，成为研究的热点。1992 年，美国国家标准与技术研究所(NIST)的David Ferraiolo和Rick Kuhn提出基于角色的访问控制模型框架，并给出了RBAC模型的一种形式化定 义[1]。Ravi Sandhu等人于 1996 年提出了著名的RBAC96 模型，奠定了RBAC控制模型在安全控制领域的重要地位。

1 RBAC 模型的局限性及改进模型

1.1 RBAC96 的局限性

为方便权限管理，在RBAC96 模型中引入角色层次，但该层次关系中的权限处理有缺陷，即低级角色的全部权限都被高级角色继承，不能拥有自己的私有权限。对此，RBAC96 模型通过引入私有角色这一概念来解决这个问题[2]。

为使角色能够保留部分不被继承的权限，为每个需要保留部分不被继承权限的角色增加私有角色。此时这些角色已不是完整意义上的角色，只是为了保留角色的私有权限而存在。同时，模型不能全面地反映实际应用中复杂的角色层次关系，属于同一机构或处于不同角色层次的同一业务性质的角色也不能拥有本机构或具体业务共有的特定权限。

1.2 改进的模型

为解决 RBAC96 模型中的私有权限问题，研究人员提出了不同的解决方案，文献[3-6]都从理论角度对 RBAC96 模型中权限的继承问题进行了改进，但增加了模型的复杂程度，也没有完全解决权限继承问题。如在一个高校教务管理系统中，教务处教务科长是院系教务科长的父角色，继承院系教务科长的部分权限。同时，院系领导也是院系教务科长的父角色，也要继承其部分权限。这时，子角色的某一权限可以被哪一父角色继承，继承被分别传递的深度怎样控制，都得不到很好的解决。针对以上问题，放弃私有角色思路，在 RBAC 中引入域和域权限的概念，提出基于域的 DRBAC 模型。其特点是能够用更简单的方式表达复杂的继承关系，而且具有更大的适应性和可伸缩性。

在 DRBAC 模型中，把系统中的角色、权限等按特定方式划分至不同的域，再对这些域进行管理。组织中具有相同性质业务的垂直系统定义为一个域，称为垂直域(或域面)。如院系教务科长与教务处教务科长为同一域面。组织中的每一个机构也定义为一个域，再将处于同一级别的、具有相同或相似组织结构的域放在一起组成一个域层。角色根据业务内容和工作部门划分到域。扩展后的 RBAC 模型如图 1 所示。

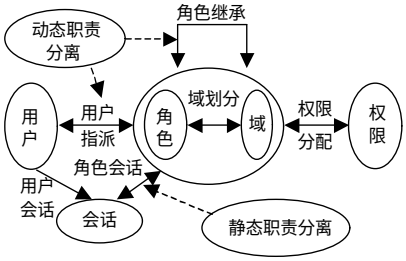

图 1 DRBAC 模型

这时，同一角色可以属于多个域，一个角色可以有多个父角色，而各个父角色存在的域可能不同。如高校院系领导

作者简介：史永昌(1971－)，男，讲师、硕士，主研方向：信息安全，计算机应用；鲁书喜，副教授、硕士

收稿日期：2008-03-17 **E-mail：**tsjksyc@163.com

与院系教务科长为同一个域，院系教务科长与校教务处教务科长也为同一垂直域(或域面)，而院系领导和校教务处教务科长不为同一域，但院系领导和校教务处教务科长都继承院系教务科长部分权限。域的关系示例如图 2 所示。

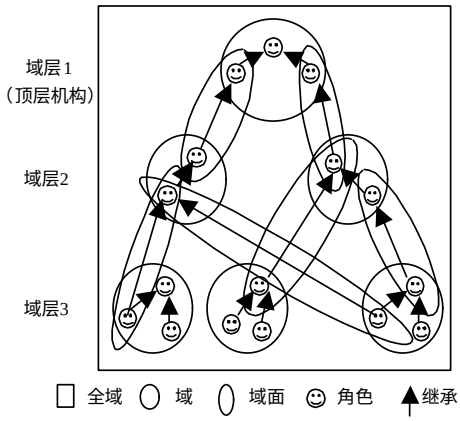

图 2　DRBAC 模型组织结构示例

引入域权限和域传播深度的概念。将角色的权限分为 3 类：私有权限，公有权限和域权限。

私有权限PRP(Private Permissions)。$PRP:R\rightarrow 2^{P}$，则$P_{\mathrm{PRP}}(r)$为角色r所具有的私有权限的集合。私有权限具有不能被继承的性质。

公有权限PBP(Public Permissions)。$PBP:R\rightarrow 2^{P}$，则$P_{\mathrm{PBP}}(r)$为角色r所具有的公有权限的集合。公有权限具有永远被所有角色继承的性质。

域权限DMP(Domain Permissions)。$DMP:R\rightarrow 2^{P}$，则$P_{\mathrm{DMP}}(r)$为角色r所具有的域权限的集合。域权限在域内具有继承性质，能否被继承依据是否为同一域而定。

域权限的形式为 $DMP(X, M, DN)$。其中，X 是客体或客体标志符，也就是被保护的对象；M 是 X 的非空访问模式集；DN 是集合$\{(Di,Ni)|(D1,N1),(D2,N2),\cdots,(Dj,Nj), j\geq 1\}$。$Di\in D$，$Ni>0$ 为权限 DMP 在域 Di 中的传播深度，同时该权限只能向上级传播。(Di, Ni)表明权限 DMP 在域 Di 中的传播深度为 Ni。每个域权限都有与域相对应的一个域传播深度，从而控制域权限可以被属于哪些域的父角色继承、继承到何种程度。

增设角色基本单元 Base Role(BR)，即拥有最基本权限的最小角色单位，是公有权限的集合。该角色中权限可以被所有的域、所有的上级角色所继承。BR 在角色层次图中处于最底层。把一些最基本的权限授予 BR，对其他角色进行授权时，就不必对这些权限进行重复授权，只需直接或简单地继承 BR 的权限即可。角色 R 的形式为 $R(rn, rp, rd)$。其中，rn 为角色的名称；rp 为该角色拥有的权限；rd 为该角色所在的域集。$rp=rp1\cup rp2\cup rp3$。其中，$rp1$ 为公有权限集；$rp2$ 为该角色的私有权限集；$rp3$ 为该角色的域权限，当 $Ni>0$ 时，在域 Di 内被父角色继承。

2　应用实例

教学质量评估系统(Education Quality Evaluation System, EQES)是一个架构在 B/S 结构上的应用系统。该系统共有问卷管理，院系、班级、课程及选课管理，质量测评，结果查询统计等多个子模块。每个模块又包含了若干功能项。系统用户包括学校领导、教务处管理人员、各院系领导、各院系教务管理人员和全体教师、学生及教学督导等。各用户通过浏览器访问系统，完成信息查询、数据管理及在线测评等工作。处于不同机构、部门或班级的用户对系统的各个功能有着不同的访问权限。因此，采用基于域的 RBAC 模型，对系统访问权限实施管理，简化和规范授权操作，以确保系统资源的正确和安全访问。

2.1　相关数据库表的设计

(1)系统功能表(functable)。以学生信息表为例，包含查询、更新、编辑学生信息 3 种功能(或权限)，如表 1 所示。

表 1　系统功能表

功能编号	功能名	功能描述
“0000”	查询学生信息	查询
“0001”	更新学生信息	更新
“0002”	编辑学生信息	编辑
…	……	……
“0961”	查询测评信息	测评结果
“0962”	测评	测评
…	……	……

(2)角色表(roletable)。角色表包括角色编号、角色名称、角色所在的域集、子角色集和角色描述字段。具体结构定义如表 2 所示。角色表的设计关键在于角色的域集和子角色集的定义。角色的域集字段每 3 个字符为 1 节，1 节为 1 个十进制数的字符形式。子角色集字段每 4 个字符为 1 节，每节为 1 个角色编号。

表 2　系统角色表结构

字段名	类型	描述
RoleId	Nvarchar(4)	角色编号
RoleName	Nvarchar(20)	角色名称
RoleDomain	Nvarchar(30)	角色的域集
RoleSon	Nvarchar(60)	子角色集
RoleDesc	Nvarchar(30)	角色描述

(3)授权表(rolepermission)。授权表包括角色号、功能号、可传播的域和在域内传播的深度。具体结构定义如表 3 所示。

表 3　系统授权表结构

字段名	类型	描述
RoleId	Nvarchar(4)	角色编号
FuncId	Nvarchar(4)	功能号
DomainId	Nvarchar(30)	继承域
Deepnees	Integer	传播深度

2.2　角色权限的指派

2.2.1　角色、域和域面的设计

根据以上对系统的分析和 DRBAC 模型，将系统用户根据岗位和在系统中承担的任务划分出若干个角色，再将角色按组织机构与层级确定出由角色组成的不同的域或域层，即校级域、管理部门域和院系域，然后根据业务性质划分域面。具体划分示例如下：

校级域：系统管理员，校领导等；

管理部门域(以教务处为例)：教务处管理员，教务处领导，教务处教务科长等；

院系级域(以计算机科学与技术学院为例)：院系领导，院系教务科长，教员，学生等；

教务管理域面：教务处教务科长，院系教务科长、院系教务管理员；

行政管理域面：院系领导，教务处领导，校领导等。

2.2.2　权限管理

完成角色设计工作之后，可对角色授权。授权算法如下：

(1)继承子角色的域权限。1)调用算法求子角色的域权限集。2)判断该子角色本身是否具有自己的域权限。如是，则对该子角色中的每个域权限进行运算：传播深度值 $Ni=Ni-1$，判断权限属性值 Ni。若 $Ni\geq 0$，转 3)，否则，舍弃该权限。3)判断 Di，若 $Di\in rd$，即子角色与父角色在同一域中，则将

该权限放到 *rp*3 中；否则，舍弃该权限。

(2)对该角色所有从子角色继承的域权限，作以下运算：传播深度值 *Ni*=*Ni*-1。

(3)继承公有权限。

(4)指派私有权限。

(5)指派新的域权限。

系统的功能都可以从功能表中读出。如在给角色学院教务管理员指派权限时，算法完成域权限和公有权限的继承后，再对角色指派私有权限和新的域权限。新的域权限可以在指定的域内继承。

2.2.3 授权检查代码实现

由于各模块对授权访问的检查机制是相同的，因此在系统中设计一个 Web Service 来实现对访问用户角色的权限检查，具体代码如下：

```
<System.Web.Services.WebService(Namespace:="http://localhost/wsjp/roleperschk")>_
Public Class rolepersionchk
Inherits System.Web.Services.WebService
<WebMethod()> Public Function roleperschked(ByVal Role_id As String,_
ByVal func_id As String) As String
Dim strconn As String="Data Source=localhost;uid=sa;pwd=;Initial Catalog=wsjp"
Dim myConn As New SqlConnection(strconn)
myConn.Open()
Dim strsql As String
strsql="select * from RolePermission where roleid=role_id and funcid=func_id"
Dim objcmd As New SqlCommand(strsql, myConn)
Dim Rolepersdr As SqlDataReader=objcmd.ExecuteReader()
'判断是否有授权记录
If Rolepersdr.Read () Then
   flag = flag & "1"   '为"1"，有访问当前功能的授权记录
Else
   flag = flag & "0"   '为"0"，无访问当前功能的授权记录
End If
Rolepersdr.Close()
myConn.close()
Return flag
End Function
End Class
```

在该 Web Service 中给出了一个方法 *roleperschked*()，其参数 *Role_id* 为角色值，*Func_id* 是功能号。返回值 *flag* 是字符“0”或“1”，相当于令牌。若 *flag* 为“1”，说明该用户具有该功能号所标识的功能。每个模块可以根据返回的 *flag* 值决定用户可以访问的功能(页面、菜单、控件和数据元素)。

各功能模块在用户要求访问时调用该 Web 服务。编程时首先要求在模块中用 <%@import namespace=" roleperschk"%>导入服务类 *roleperschk*，具体调用的 VB.NET 代码如下：

```
dim rolepers as roleperschk=new roleperschk
dim permission as string=rolepers.roleperschked(role_id,func_id)
```

3 结束语

本文基于 NIST RBAC 模型讨论了 RBAC 存在的局限性，对原有的模型进行扩展并提出了一种新的基于域的 DRBAC 模型。该模型将域和域权限的概念引入 RBAC，将每个角色的权限集分为公有权限集、私有权限集和域权限集，给出一种新的基于域的多父角色权限继承问题解决方法，并在系统 EQES 中实现了对用户访问权限的管理，避免了角色权限的重复指派，简化了权限继承。

参考文献

[1] Ferraiolo D, Kuhn R. Role-based Access Controls[C]//Proc. of the 15th NIST-NCSC National Computer Security Conference. Baltimore, Maryland, USA: NIST-NCSC, 1992: 554-563.

[2] Sandhu R, Coyne E J. Role Based Access Control Models[J]. IEEE Computer, 1996, 29(2): 38-47.

[3] 吕宜洪, 宋瀚涛, 龚元明. 基于 RBAC 改进模型的角色权限及层次关系分析[J]. 北京理工大学学报, 2002, 22(5): 611-614.

[4] 余文森, 张正秋, 章志明, 等. 基于角色的访问控制模型中私有权限问题的研究[J]. 计算机应用研究, 2004, 21(4): 50-51.

[5] 鞠成东, 廖明宏. 基于 RBAC 模型的角色权限及层次关系研究[J]. 哈尔滨理工大学学报, 2005, 10(4): 95-99.

[6] 章志明, 张正球, 余 敏. 一种基于 RBAC 的多个域之间安全访问控制[J]. 计算机工程, 2005, 31(15): 135-139.

（上接第 175 页）

5 结束语

本文针对传统信息安全领域的一些现状，提出了一种安全事件综合关联分析框架，并对各个模块的功能给予定义，最后对本框架的关键技术点压制聚合技术和攻击重构技术进行了概要性的介绍。经过分析比较，发现这种事件分析处理方案，不仅在很大程度上降低了误报，而且解决了一定的漏报问题。但是该方案在实现过程中也存在一些难点，如：各类告警到触发事件的映射，本文所研究的告警类型主要是针对入侵检测系统 Snort 的告警；在攻击重构中，很大程度上依赖于告警的准确性。

参考文献

[1] Curry D, Debar H. Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition[EB/OL]. (2001-02-24). http://www.silicondefense com/idwg/draft-ietf-idwg-idmefxml-07txtdraft-ietfidwg-idmef-xml-03.txt.

[2] Debar H, Wespi A. Aggregration and Correlation of Intrusion Ddetection Alerts[C]//Proc. of the 4th International Symposium on Recent Advances in Intrusion Detection. London, UK: Springer-Verlag, 2001: 85-103.

[3] Peng Ning , Xu Dingbang. Learning Attack Strategies from Intrusion Alerts[C]//Proc. of the 10th ACM Conference on Computer and Communications Security. Washington D.C., USA: [s. n.], 2003: 200-209.

[4] Peng Ning, Cui Yun, Reeves D S, et al. Tools and Techniques for Analyzing Intrusion Alerts[J]. ACM Transactions on Information and System Security, 2004, 7(2): 273-318.

[5] Barford P, Kline J, Plonka D, et al. A Signal Analysis of Network Traffic Anomalies[C]//Proc. of ACM SIGCOMM Internet Measurement Workshop. New York, USA: ACM Press, 2002: 71-82.

[6] CERTAdvisoryCA-2003-20: W32/Blasterworm[EB/OL]. (2003-04-13). http://www.cert.org/advisories/CA-2003-20.html.