RBAC模型的细粒度扩充及应用

司 炜[1] 曾广周[1] 盛 琦[1] 李英俊[2]
(山东大学计算机科学与技术学院 济南 250061)[1] (济南大学信息科学与工程学院 济南 250022)[2]

摘 要 基于角色的访问控制是一种高效安全的访问控制机制,但是RBAC2001建议标准中没有提出根据单位特征、功能特征和数据特征来细化控制角色指派的详细方法。本文结合RBAC模型思想和大型企业信息系统的实际需求,对核心RBAC模型进行细粒度的扩充,在单位、功能、数据等维度对模型进行了细化,并给出了实例应用,有效地解决了大型企业信息系统的安全访问控制难以细化的问题。

关键词 细粒度,角色,基于角色的访问控制,大型企业信息系统

Fine Grain Extension and Application of the RBAC Model

SI Wei[1] ZENG Guang-Zhou[1] SHENG Qi[1] LI Ying-Jun[2]
(School of Computer Science and Technology, Shandong University, Jinan 250061)[1]
(School of Information Science and Engineering, Jinan University, Jinan 250022)[2]

Abstract Role-Based Access Control(RBAC) is a kind of access control mechanism which is secure and high performance. But the standard for RBAC2001 model does not give the method which based the character of department, function and data to control the role assignment. Combining with the idea of RBAC model and the requirement of large-scale enterprise information system, the article make fine grain extension on Core RBAC model, and thinning the role assignment in the dimensionality of department, function and data. At last, this article gives practical application of the model, and resolves effectively the question that secure access control difficult to thin in large-scale enterprise information system.

Keywords Fine grain, Role, RBAC, Large scale enterprise information system

近年来,基于角色的访问控制(RBAC)在信息系统安全控制方面得到广泛的应用。该技术主要研究将用户划分成与其在组织结构体系相一致的角色,以减少授权管理的复杂性,降低管理开销。本文在RBAC2001建议标准的参考模型的基础上,结合大型企业信息系统的要求,对RBAC模型进行了扩充,在角色、功能及数据等维度对系统权限控制进行细化,有效地实现了多级管理员体系,增强了系统的安全管理性能。

1 RBAC2001建议标准的核心RBAC模型

2001年8月,RBAC提出机构NIST在RBAC96模型[1,2]及其后的一系列成果[3,4]的基础上,提出了一个关于RBAC技术的建议标准[5](下称RBAC2001建议标准)。该标准中的参考模型包括核心RBAC、层次RBAC、静态职责分离和动态职责分离四个模型构件,分别描述RBAC模型某一方面的特性。在构造实际RBAC系统中,核心RBAC构件是必选的,其它构件都是可选的。

1.1 核心RBAC(Core RBAC)

如图1,核心RBAC主要包括如下基本要素集:用户USERS、角色ROLES、客体OBS、操作OPS、权限PERM。用户USERS指对数据对象进行操作的主体。角色ROLES主要反映用户的岗位和职责。操作OPS代表系统中所有功能操作的集合。客体OBS代表系统中所有功能操作的客体对象的集合。权限PERM $=2^{(OPS\times OBS)}$代表对系统中的数据或者用数据表示的其它资源进行访问操作的许可集合。核心RBAC还引入会话SESSION概念,SESSIONS代表系统中所有的会话集。会话是一个用户与角色子集的映射,用户激活角色时,可以打开多个会话。用户是一个静态的概念,会话则是一个动态的概念,它代表用户与系统进行交互。各集合之间的关系定义为:

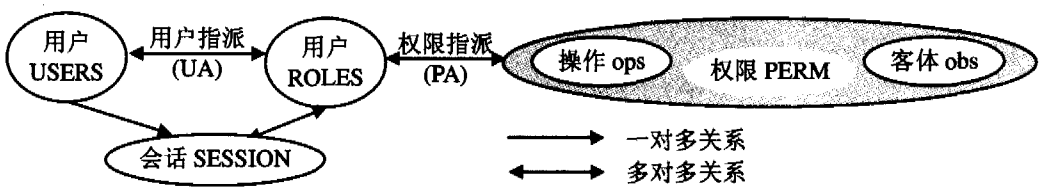

图1 核心RBAC模型

UA:USERS×ROLES表示用户与角色之间的委派关系。

PA:PERMS×ROLES表示权限与角色之间的授予关系。

司 炜 工程师,硕士,主要研究方向:软件工程、计算机网络安全。曾广周 教授,博士生导师,主要研究方向为CSCW、智能计算、移动计算。
盛 琦 工程师,硕士,主要研究方向:软件工程、软件过程改进。李英俊 讲师,硕士,主要研究方向:数据仓库,软件工程。

US:用户 USERS 与会话 SESSION 是一对多的关系。一个用户可以进行多个会话,一个会话只允许一个用户参与。

SR:会话 SESSION×角色 ROLES 表示会话与角色是多对多的关系。一个会话中可以存在多个角色,一个角色可以被多个会话使用。

1.2 层次 RBAC 模型与带约束的 RBAC

层次 RBAC 模型支持角色间的层次关系,这种层次关系是角色的任意半序关系,层次角色之间存在访问权限和用户的继承关系。由于实际工作中角色的上下级关系有很多限制,RBAC2001 建议标准将角色层次区分为通用角色层次和限制角色层次。

有约束的 RBAC 规定在 RBAC 模型上实行职责分离机制。RBAC2001 建议标准引入两种职责分离模型:静态职责分离(SSD)和动态职责分离(DSD)。静态职责分离要求用户/角色委派时实施约束。它仅限于角色集上的约束关系,特别是用户和角色(UA)的关系。动态职责分离对用户会话中可激活的角色进行约束,它解决的是用户角色委派时潜在的冲突问题。DSD 是会话与角色集之间的约束机制。

2 大型企业信息系统对 RBAC 模型的要求

随着市场竞争日趋激烈和计算机技术高速发展,企业信息化做为提升企业核心竞争力的重要手段越来越受到大型企业的重视。如何统一合理的管理大型企业信息系统中各级用户及复杂多变的数据信息,安全高效地实现信息处理和数据共享就成为一个非常重要的问题。

• 大型企业信息系统要求 RBAC 模型能够支持设立省一地市一区县的多级管理员体系,自上而下的逐级授权。例如某省级电信运营商综合业务信息系统中,省公司下辖十几个地市公司,每个地市公司下辖十几个区县公司,每个区县公司在系统中须设置十几个用户,这样整个系统有各级单位的上千个用户。由省公司系统管理员给所有用户都逐一定义角色,分配权限,这在实际工作中是不现实的。实际工作中必须设立省一地市一区县的多级管理员体系,一级一级地自上而下进行授权。

• RBAC 模型应该细化对功能权限的控制粒度,提供为用户设置不同功能的不同级别的权限的方法。为了与实际工作相对应,每个角色的操作权限需要分得较细,不同功能菜单应该有查询、修改、删除、禁用等等不同的级别,这样才能真正体现用户在系统中可用功能的差异性。

• RBAC 模型应该细化对操作对象的控制粒度,体现数据类型和数据时间周期的特征。实际工作中,不同角色在系统中操作的数据是不一样的,例如财务预算员关注的是每月财务预算计划的实际执行情况,而市场营销人员则关心每天的客户数增减和收入变化情况,RBAC 模型应该支持为用户设置如此细致的操作数据属性的方法。

3 RBAC 模型的细粒度扩充及应用

针对大型信息系统的要求,本文在核心 RBAC 模型的基础上,对 RBAC 模型进行了细粒度的扩充,从单位、功能、数据等多个维度对该模型进行了细化和加强,并实现了多级管理员体系,增强了模型的安全控制能力。

3.1 带有单位特征的细粒度的角色划分

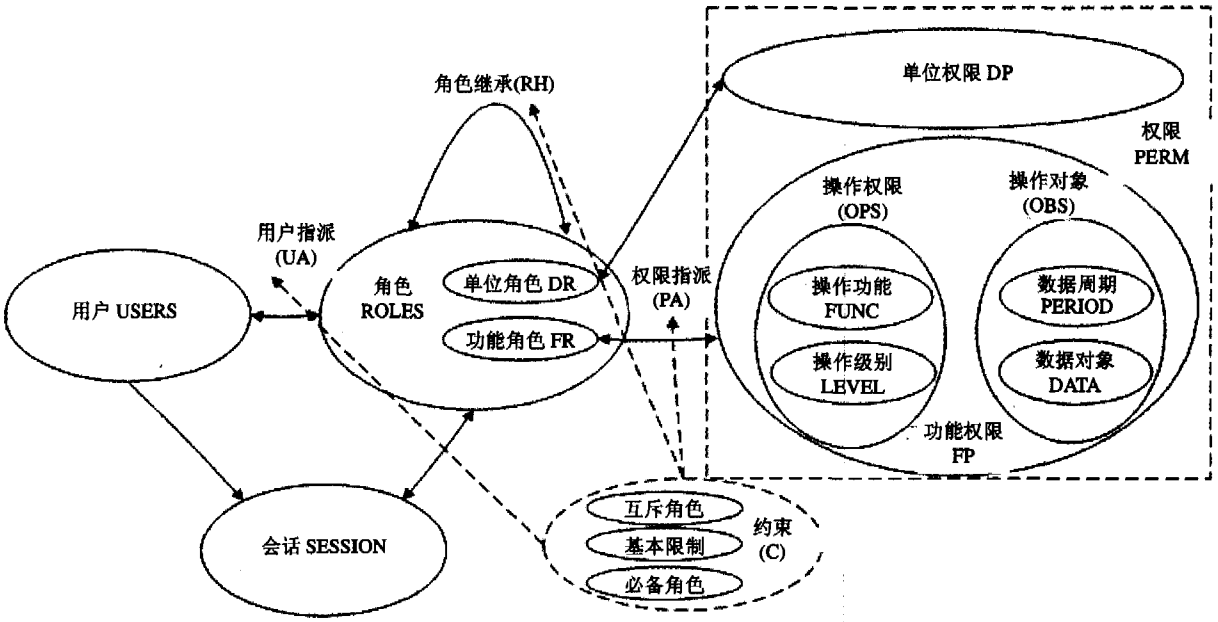

图 2 细粒度扩充后的 RBAC 模型

如图 2,细粒度扩充后的 RBAC 模型将角色集 ROLES 细分为单位角色集(DR)和功能角色集(FR)。一个完整的角色必须同时具备这两种角色。另外,将权限集 PERM 分为单位权限集(DP)和功能权限集(FP)。单位角色赋权时对应单位权限集,功能角色赋权时对应功能权限集。

• 单位角色集(DR)=$(dr_1, dr_2, \cdots, dr_n)$主要是反映在大型企业信息系统中的单位角色集合。

• 功能角色集(FR)=$(fr_1, fr_2, \cdots, fr_n)$主要是反映在大型企业信息系统中的功能角色集合。

• 角色集(R)=$(r_1, r_2, \cdots, r_n)$=DR×FR 主要是指用户在企业中的工作岗位,反映用户的职责。一个完整的角色由单位角色和功能角色组合而成,缺一不可。

• 用户集(U)=$(u_1, u_2, \cdots, u_n)=2^{(R)}=2^{(DR\times FR)}$,用户是指信息系统中对数据对象进行操作的主体。

• UA:U×R=U×(DR×FR)表示用户与角色之间多对多的关系。一个用户可以担任多个角色,一个角色也可以被分配给多个用户。该关系反映用户可以在大型信息系统中担任多个单位的多种角色。

·单位权限集(DP)=(dp_1,dp_2,…,dp_n)主要是反映信息系统中对单位的访问许可集合。

·单位角色集(DR)和单位权限集(DP)的关系是多对多的关系,即一个单位角色可以拥有对多个单位的访问许可,一个单位的访问许可也可以被赋予多个单位角色。

利用此模型的角色划分,可以使角色设置更加细化灵活,管理更为简化方便,例如,在某省电信运营商综合系统的实际应用中,可以设置多种功能角色,如系统管理员、报账员、预算员等,设置多种单位角色,如北京、山东、济南、朝阳区等,用户张三的工作岗位是山东公司系统管理员,只需在系统中将功能角色(系统管理员)和单位角色(山东)赋予他即可。用户李四是北京市朝阳区公司的预算员兼报账员,则在系统中将功能角色(预算员、报账员)和单位角色(朝阳区)赋予他即可。如上所述,这种角色划分和赋权方式相当于设置各级单位的具体工作岗位,然后将人员安排在具体工作岗位上,可以一人一岗或一人多岗。

利用此改进模型可以实现多级管理员体制。在上面的例子中,张三是山东公司系统管理员用户,他建立一个功能角色(地市级系统功能管理员),然后给每个地市建立一个单位角色(如济南管理员,该单位角色的单位权限包括本单位及下辖区县单位)。每个地市管理员角色就由功能角色(地市级系统功能管理员)和该地市的单位角色(如济南管理员)组合而成。张三将上面定义的地市管理员角色赋予相应地市的管理员用户,这样地市管理员用户的实际权限为:拥有地市级通用功能权限和该地市单位本身及所有下辖区县的单位权限。同理,每个地市系统管理员用户可以依此建立一个功能角色(区县级系统功能管理员),然后各自给自己地市所辖的每个区县建立一个该区县的单位角色(如历城区管理员,),每个区县管理员角色就由功能角色和该区县的单位角色组合而成。地市管理员用户将每个区县管理员角色赋予相应的区县系统管理员用户。这样每个区县系统管理员用户的实际权限为:拥有所有的区县级通用功能权限和该区县单位本身及所有下辖单位(乡镇或城区营业厅)的单位权限。

大型系统中权限管理的复杂性和分配的巨大工作量,通过多级管理员体制,被分解到省、地市、区县各级管理员处,每个单位的管理员只负责设置本单位角色和用户及直属下级单位系统管理员用户的功能权限和单位权限。这样就大大减轻了各级管理员的工作量,使得各级管理员有精力来重视操作权限和操作对象的设置,从而使功能权限的细化成为可能。

3.2 功能权限的细粒度扩充

如图2所示,细粒度RBAC模型将功能权限集(FP)分为两部分,一部分为操作权限集(OPS),另一部分为操作客体集(OBS)。其中操作权限集(OPS)又细分为操作功能集(FUNC)和操作级别集(LEVEL),操作客体集(OBS)细分为数据对象集(DATA)和数据周期集(PERIOD)。

·操作功能集(FUNC)=($func_1$,$func_2$,…,$func_n$)主要是指在信息系统中的功能菜单,如报表审批、收支分析、预算编制、用户管理等等。

·操作级别集(LEVEL)=($level_1$, $level_2$,…,$level_n$)主要是指在信息系统中的功能的操作级别,如禁用、查询、修改、删除等。

·操作权限集(OPS)=(ops_1, ops_2, …, ops_n)=$2^{(FUNC\times LEVEL)}$主要是指在信息系统中的功能的访问许可集合。例如,查询分析报表,修改预算数据,增删用户,禁用单位管理功能等等。

·数据对象集(DATA)=($data_1$,$data_2$,…,$data_n$)主要是指在信息系统中的具体数据对象,如各种报表、项目、文档、分析图形等。

·数据周期集(PERIOD)=($period_1$, $period_2$,…,$period_n$)主要是指在信息系统中的具体数据对象的周期,如日期(1号、2号……)、月度(一月、二月、…、十二月)、季度(一季度、…、四季度),年度(2004、2005)等等。($period_1$,$period_2$,…,$period_n$)代表系统中的具体数据对象的数据周期。在信息系统中每个数据对象的数据周期是不一样的,每个数据对象的数据周期在系统中都有相应的规定。

·操作客体集(OBS)=(obs_1, obs_2, …, obs_n)=$2^{(DATA\times PERIOD)}$代表系统中所有功能操作的客体对象的集合。例如,2月15日客户情况变化分析图,1月份财务收支表,二季度促销费用预算表,2005年度传输网络建设文档等等。

·功能权限集(FP)=(fp_1,fp_2,…,fp_n)=$2^{(OPS\times OBS)}$主要是指在信息系统中的功能和数据对象的访问许可集合。fp_1,fp_2,…,fp_n代表系统中的具体功能和数据的操作许可。每个明细功能权限fp_i实际上是一个四元组($func_j$,$level_k$,$data_l$,$period_m$),$1\leqslant i,j,k,l,m\leqslant n$。例如,利用分析报表功能查询2月15日客户情况变化图表,利用预算编制功能修改二季度促销费用预算表等。

·功能角色集(FR)=(fr_1,fr_2,…,fr_n)=$2^{(FP)}$主要是指信息系统中功能角色的集合。fr_1,fr_2,…,fr_n是具体的功能角色。一个功能角色fr_i本身具备一个或多个具体功能权限fp_1,fp_2,…,fp_j,$1\leqslant i,j\leqslant n$。

·功能权限指派(PA)=$f(fp_i)$=f($func_j$,$level_k$,$data_l$,$period_m$),$1\leqslant i,j,k,l,m\leqslant n$,是定义功能角色集FR和功能权限集FP之间映射关系的表达式,其中f为功能权限指派PA的函数表达式。

与核心RBAC模型相比,该改进模型在权限设置上更加细致,每一级管理员可以按照实际需要,对角色的功能权限从数据周期、数据对象、操作功能、操作级别等四个维度上进行明细设置,并灵活深入地控制角色的功能和其访问的数据内容,从而达到精确管理的目的。

3.3 其他方面的补充细化

3.3.1 继承的细化

继承(RH-role hierarchy):允许将某一角色定义为另一角色的子角色,通过角色间的继承关系,间接地拥有其子角色所定义的权限。

与层次RBAC模型相比,细粒度扩充后的模型允许对单位角色和功能角色分别继承,对子单位角色和子功能角色各自补充权限后,形成新的单位角色和功能角色。

3.3.2 约束的细化

约束(C-constraint):不同角色之间的相互制约。通过设置约束条件,可以对角色授权、用户授权进行各种必要的限制。

角色互斥约束:指一个用户只能担任两个互斥角色中的一个。例如每个单位的预算表的审批角色与录入角色不允许为同一个用户。

基本限制约束:规定了一个角色可被分配的最大用户数,例如某单位总经理角色只能分配给一个用户。

必备角色约束:描述执行某些功能必须具有的资格。如具有审批功能的角色必须为部门经理级以上人员,具有增减

用户功能的必须是管理员角色等。

4 细粒度扩充后的 RBAC 模型的应用实例

某大型企业的综合业务管理系统采用 B/S 结构，采用细化后的 RBAC 模型实现其安全访问控制体系，取得了良好效果。系统主要数据表及处理过程如下：

4.1 基础表

单位表、功能菜单表、操作级别表、数据周期表、数据对象表。这些表主要是存储系统的各项基础数据。

4.2 角色类表

功能角色表(功能角色 ID，功能角色名称，所属单位 ID，父功能角色 ID，互斥角色 ID，必备角色 ID，基本限制数)

功能角色权限表(功能角色 ID，功能菜单 ID，操作级别 ID，数据周期 ID，数据对象 ID)

单位角色表(单位角色 ID，单位角色名称，所属单位 ID，父单位角色 ID，互斥角色 ID，必备角色 ID，基本限制数)

单位角色权限表(单位角色 ID，允许访问单位 ID)

角色表(角色 ID，角色名称，所属单位 ID，父角色 ID，单位角色 ID，功能角色 ID，互斥角色 ID，必备角色 ID，基本限制数)

管理员在系统中分别定义单位角色(存入单位角色表)和功能角色(存入功能角色表)，然后给这些角色设置相应单位权限和功能权限(包括功能菜单 ID、操作级别 ID、数据周期 ID、数据对象 ID 等)，分别存入单位角色权限表和功能角色权限表。

管理员定义角色基本信息，然后给该角色指定单位角色和功能角色，存入角色表。

管理员还可以分别给角色、单位角色和功能角色设置该角色的互斥角色、必备角色和基本限制数，用来对其进行相应的约束。另外角色、单位角色和功能角色支持角色继承，设置的相应父角色 ID 分别存入角色表，单位角色表和功能角色表的父角色 ID 列。

4.3 用户类表：

用户表(用户 ID，用户姓名，所属部门 ID，用户级别 ID)

用户角色表(用户 ID，角色 ID)

用户权限表(用户 ID，允许访问单位 ID，功能菜单 ID，操作级别 ID，数据周期 ID，数据对象 ID)

管理员首先录入用户基本信息(存入用户表)，然后给用户指定一个或多个角色(这些信息存入用户角色表)，系统会根据角色表中的互斥角色和必备角色约束来判断这些指定是否正确；如果正确，那么系统根据角色 ID 及父角色 ID 在角色表中找出对应的单位角色 ID 和功能角色 ID，然后根据单位角色 ID 和功能角色 ID 分别在单位角色权限表和功能角色权限表找出相应的明细权限，组合后存储到用户权限表中。这样就完成了每个用户的明细权限设置。

用户在登录系统后，系统自动在页面上根据用户权限表中的用户 ID 和功能 ID 列出该用户有权访问的功能菜单；用户点击某个功能菜单后，系统根据部门 ID(从单位角度)、数据周期 ID(从时间周期角度)、数据对象 ID 来决定该用户能够访问的哪些单位哪些时间的表、文档、项目等等。最后，系统会根据操作级别 ID 来决定用户在具体功能页面上能够对这些数据进行怎样的操作(增删、修改、查询等等)。

结论 通过对 RBAC 模型进行细粒度的扩充，在单位、功能、数据等维度对模型进行了细化，实现了大型信息系统的多级管理员体系，使得系统的功能及数据的控制更加灵活，用户和角色的管理更加方便。该模型在某省级电信运营商的综合信息系统中进行了实践应用并取得了很好的效果，受到用户广泛的好评。

参 考 文 献

1 Sandhu R S, Samarati P. Access control: principles and practice [J]. IEEE communications, 1994, 32(9): 40～48

2 Sandhu R S, Coyne E J, Feinstein H, et al. Role-Based Access Control Models. IEEE Computer, 1996, 29(2): 38～47

3 Sandhu R, Bhamidipati V, Coyne E, et al. The ARBAC97 model for role-based administration of roles: Preliminary description and outline [A]. In: Proceedings of the Second ACM Workshop on Role-Based Access Control [C], Fairfax, Virginia, USA: ACM, 1997. 41～50

4 Sandhu R S, Ferraiolo D F, Kuhn D R. The NIST model for role based access control: Towards a unified standard. In: Proc. of the 5th ACM Workshop on Role Based Access Control, New York, NY: ACM Press, 2000. 47～63

5 Ferraiolo D F, Sandhu R S, Gavrile S, et al. Proposed NIST standard for role-based access control. ACM Transactions on Information and Systems Security, 2001, 4(3): 224～274

(上接第 276 页)

件表示采用的规范函数，在原有规约匹配的相关理论基础上，对函数名做了一定的规范化处理，大大提高了构件的查准率。以它为基础，已经搭建了包括 ERP、OA 等多个企业应用系统。实践表明，该构件库的检索效率令人满意。今后将在两个方面继续进行优化，一方面减少构件的入库代价，逐步实现 Java 构件的自动化分类和表示，另一方面提供构架构件，通过用户修改构架配置，系统自动检索匹配满足构架的构件，实现企业应用系统的快速成型。

参 考 文 献

1 杨芙清，梅宏，李克勤. 软件复用与软件构件技术[J]. 电子学报，1999(2)：68～75

2 Mili R, Mili A, Mittermeir R T. A Survey of SoftwareStorage and Retrieval [J], Ann. Software Eng., 1998, 5(2): 349～414

3 Morel B, Alexander P, Member S. SPARTACAS: Automating Component Reuse and Adaptation [J]. IEEE Trans. Software Eng., 2004, 30(9): 587～600

4 张世琨，张文娟，常欣，等. 基于软件体系结构的可复用构件制作和组装[J]. 软件学报，2001，12(9)：1352～1359

5 马亮，等. 基于规约匹配的构件检索[J]. 小型微型计算机系统，2002，23(17)：1153～1157