第 33 卷 第 5 期 计 算 机 工 程 2007 年 3 月
Vol.33 No.5 Computer Engineering March 2007
·安全技术· 文章编号：1000—3428(2007)05—0120—03 文献标识码：A 中图分类号：TP308

角色管理自动化的访问控制

李 佳[1]，徐向阳[2]
（1. 湖南大学校长办公室，长沙 410082；2. 湖南大学计算机与通信学院，长沙 410082）

摘 要：基于角色的访问控制是简化企业信息系统访问控制的一个有效策略。近年来规则已经被用于支持用户角色的自动管理。该文引入职能控制集的概念，结合角色和规则的优点，提出了一种新的适合于大型企业的安全访问控制方案，实现角色分解和权限细粒度控制的目的，根据企业的安全管理策略和用户的属性，自动管理用户-角色的分配，还引入否定授权策略，增强了客体权限分配的灵活性和安全性。
关键词：访问控制；角色；规则；职能控制集

Role Auto-assignment for Access Control

LI Jia[1], XU Xiangyang[2]
（1. School Master Office, Hunan University, Changsha 410082; 2. School of Computer and Communication, Hunan University, Changsha 410082）

【Abstract】 Role-based access control (RBAC) is a useful policy for simplifying access control on enterprise information system. Recently, rule concept is used to support role assignment automatically. By introducing the concept of function control sets and combining the virtue of role and rule, this paper proposes a new security access control scheme suitable for large organizations, which can enhance the flexibility and security on object permission assignment, assign role for user automatically based on user attributes.
【Key words】 Access control; Role; Rule; Function control sets

1 概述

随着信息技术的迅速发展和广泛应用，网络安全威胁日趋严重。在网络环境下用户对服务器资源的需求是动态变化的，用户、资源和权限的关系较为复杂，研究系统资源访问控制技术是信息安全领域的热点问题。目前最流行的访问控制模型有自主访问控制模型(Discretionary Access Control，DAC)、强制访问控制模型(Mandatory Access Control，MAC)和基于角色的访问控制模型(Role-based Access Control，RBAC)。

自主访问控制安全模型是非中心化的、基于授权规则的模型，在操作系统和数据库系统中得到了广泛的应用。在DAC模型中，访问权限的授予是可以传递的，一旦访问权被传递出去将难以控制并可能带来严重的安全问题。MAC是一般在军事环境下使用的中心化的多层安全模型，源于对信息机密性的要求以及防止特洛伊木马之类的攻击，虽然MAC增强了信息的机密性但不能实施完整性控制。随着计算机网络的发展，使信息具有完整性和可用性的要求超过了对机密性的要求，同时，对授权管理和策略配置的便捷性提出了更高的要求，而传统的DAC和MAC策略把访问权限直接授予用户难以提供这方面的支持。

基于角色的访问控制(RBAC)是David Ferraiolo和Richard Kuhn在1992年引入的一种新的访问控制技术[2]，RBAC96模型[3]以及后来的NIST建议[4]为现代的Web安全管理奠定了坚实的基础，并被广泛地应用于数据库系统管理和操作系统软件。该模型提供基于角色的灵活管理策略，是一种公认的、较适合在大型企业计算机网络中实施的访问控制技术。然而，对于大型企业来说，无论是采用user-pull还是server-pull结构[3]，角色作为安全管理的基础都存在一些 缺点：

(1)尽管使用角色可以减少管理大批量用户 ID 的开销，但还是需要做大量给用户分配角色的工作。

(2)随着组织规模的增加，成员类别和工作级别都会变得更加复杂，导致角色的层次更加复杂。

(3)角色权限的管理不够灵活且过于复杂，将难以适应用户工作和任务的变化。

本文针对上述问题引入职能控制集的概念和规则管理的思想，结合角色和规则的优点，提出一种新的适合于大型企业的安全访问控制方案。该方案不但增强客体权限分配的灵活性和安全性，而且可根据用户属性自动管理角色分配。

2 基于规则支持的扩展 RBAC 模型

2.1 模型的引入

为满足角色自动管理和细化权限管理的要求，在文献[1]提出的模型基础上，增加一个权限控制实体，提出了RBP-ERBAC（Rule-based Provisioning of Extended RBAC）模型，主要由扩展RBAC系统和基于规则的用户-角色分配系统组成，如图1所示。

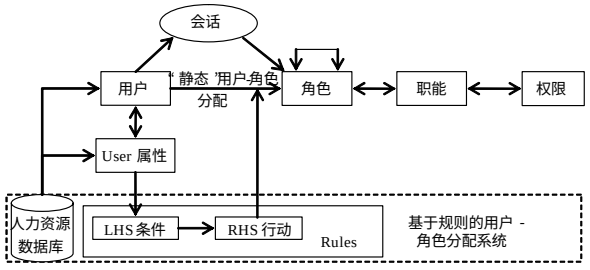

图 1 基于规则供应的扩展 RBAC 模型

(1)扩展RBAC系统

作者简介：李 佳(1980－)，女，硕士生，主研方向：信息安全和计算机技术；徐向阳，副教授、博士
收稿日期：2006-09-07 **E-mail：**lenelee@hnu.cn

扩展 RBAC 系统的基本元素是用户、角色、职能、权限和会话，与 RBAC96 相比，RBP-ERBAC 模型增加了抽象的角色权限控制实体—职能，职能分离角色和权限可实现角色分解和权限控制的目的。

扩展 RBAC 系统的主要功能是：

1)在用户建立会话期间，根据用户访问请求，授权相应的“静态”角色；

2)根据用户的角色，分配相应的职能，即不同的客体权限组合和权限执行策略；

3)根据角色的不同职能，按要求访问客体权限，把结果返回用户。

用户-角色关系将在“用户-角色分配系统”中自动获取。在企业信息系统的日常安全管理中，无需始终启用“用户-角色分配系统”,只需要根据分配系统的运行结果就可实现有效地访问控制管理。静态角色管理可以简化管理过程，降低系统的运行开销。

(2)基于规则的用户-角色分配系统

用户-角色分配系统主要包括企业人力资源数据库、用户属性和规则引擎。该系统的主要功能是在需要更新用户-角色关系时，根据企业人力资源数据库里最新的用户信息更新用户属性，再使用规则引擎进行计算后，得到新的用户-角色分配关系。规则的引入实现了角色职责的自动分离。

用户-角色分配系统采用 If-Then 的规则推理形式。规则由 LHS(条件)和 RHS（行动）组成：当条件表达式 LHS 为真时，执行 RHS 定义的行动。在这里，规则的 LHS 是用户的属性表达式，RHS 是一个或多个角色。如果用户 u 满足属性表达式，u 就被赋予规则的 RHS 所指定的角色，如规则 rule_i：

$$ae_i \Rightarrow r_g \qquad (1)$$

其中，ae_i是用户属性表达式，r_g是规则产生的角色。为了维护用户-角色的授权关系，定义了用户-角色关系集：

$$\text{URAuth} = \{(u, r) \mid (\exists \text{rule}_i)[\text{u满足ae}_i \wedge r \subseteq \text{RHS(ae}_i)]\} \qquad (2)$$

如果 $(u, r) \in \text{URAuth}$，则意味着 u 被赋予了角色 r。该集合描述了基于规则的用户-角色分配的语义，是基于规则的用户-角色分配模型的关键成分。

在每一次会话建立过程中不直接动态分配用户角色，而是采用静态用户-角色分配模式，把规则引擎系统当作系统安全管理员为用户手动分配角色工作的自动化替代品，每隔一段时间就自动运行，更新用户属性和用户-角色关系，用新的角色分配表取代旧表。属性-角色推理规则的变更也只有在系统安全管理员进行充分检查后才能被激活使用，避免了规则变化的高动态性带来的不可预知的影响。

2.2 访问控制策略和机理

2.2.1 权限分配的控制机制

RBAC 模型里定义的角色是一个被命名的工作职责，抽象描述角色成员具有的权限、信任、责任和能力。每个角色在企业的业务流程中包含了用户被授权执行的一系列任务。每个任务有一个或多个权限，因此可以定义为一个权限的集合。如果需要访问一个客体，则可以把必要的权限赋予该任务来完成一个期望的工作。

RBAC 模型最主要的特点就是权限直接与角色相关，这种访问控制结构使得将 RBAC 模型产生了一些缺陷：

(1)只要主体拥有对客体的访问权限，主体就可以无限次地使用该权限。由于这个局限性，访问控制系统不能有效限制由许可操作组成的攻击。

(2)权限是直接与角色相关的，决定了权限的粒度最多只能细化到角色一级。无论采用静态职责分离还是动态职责分离约束，最小权限约束只能是角色的最小权限约束，而现实世界中一个角色往往可以执行多项任务，同时不管用户是否执行任务，只要用户激活某一角色就拥有该角色的全部权限。

(3)实际应用中为了实现对系统权限的精确控制，需要定义和继承大量的角色，导致角色层次过于复杂而难以管理。

为了解决上述问题，必须改变 RBAC 模型的三层访问控制结构，在此将职能控制集的概念明确引入 RBAC 模型中，形成新的基于角色的访问控制模型。本文将角色职能和权限控制结合起来作为一个单独的概念，将职能控制集置于角色集和权限集之间，将传统 RBAC 模型的三级结构修改为四级结构。这样就能增强对客体权限的控制管理，同时减少需要的角色数量，降低角色层次的复杂度。

图 2 是企业流程里的一个角色分解的例子，其中客体代表可识别的信息实体，如文件、目录、服务等，p 是对客体的操作权限，可以是 read、write、execute、create、delete、edit 等。用户主管 A 拥有两个角色，分别为营销经理和品牌经理。每一个角色分有各自不同的职能要求，这些职能之间相对独立，不需要在角色激活期间拥有所有职能权限。当一个职能被激活后，职能控制机制将自动关闭其他职能直到该任务结束，从而实现最小权限约束。

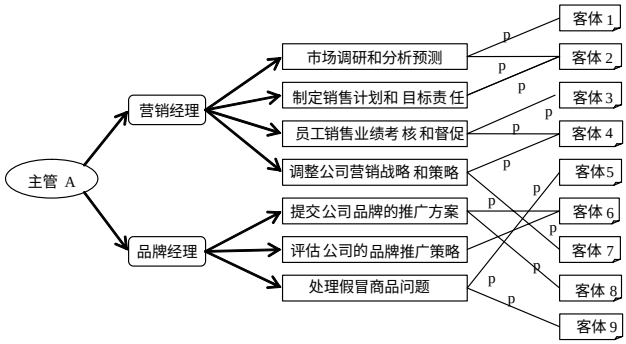

图 2 用户-角色-职能-权限

这里提出的职能控制集合与一般的任务集合不同，不但设定完成此任务需要的权限集合，还包含了各个权限的执行顺序和次数约束。访问顺序和次序的控制有助于更加准确地实现最小特权原则，增强系统的访问控制能力。

在每一次会话建立的时候，系统从用户获得角色信息，从客体获得访问权限信息。这些访问控制信息被立刻用来检查用户权限并作出相应访问决策。但在传统 RBAC 机制下，给角色赋予的权限通常都超出所需，且没有考虑权限之间存在的组合关系，因此限制了系统的访问控制功能。假定有一个文件的权限集{A, B, C, D}。如果完成一个角色职能需要这个权限集，我们按传统 RBAC 把所有权限赋予该角色。但是，如果实现此职能仅需要权限组合（A, C, B, D），我们无需将权限的所有组合赋予该角色。如果检查访问顺序，就能够防止权限以不期望的次序被滥用。同样的道理，增加每一次会话期间权限的执行次数约束，就能有效地限制由许可权限组合构成的重复攻击探测。

2.2.2 否定授权

文献[6]中提出了两种著名的访问控制策略：

(1)封闭策略：如果存在一个相应的肯定授权就允许访

问，否则就拒绝访问。

(2)开放策略：如果存在一个相应的否定授权就拒绝访问，否则就允许访问。

在RBAC的文献中很少提到否定授权，主要是由于RBAC模型如RBAC96 和建议的NIST标准模型是基于肯定授权的，只允许权限拥有者执行某些权限规定的行动[2~4]。文献[7]中指出：对一个缺乏某个授权的用户使用封闭策略，并不会阻止其以后获得这个权限。因此他们建议用明确的否定授权来阻塞授权。当用户获得否定授权时，他的肯定授权就被阻塞了。

传统的 RBAC 在意识上是为一个封闭的企业环境设计的，安全高级职员们为用户手动分配角色，执行肯定授权策略。然而，商业和信息技术的前景在近年来急剧变化。服务提供企业的数量剧增，他们通过 Internet 为用户提供各种服务，纯粹的肯定授权难以适应开放的网络环境的安全要求。

由于用户的属性可能随时会发生变化，采用基于规则的自动角色分配管理难以预见到用户基于其属性所能获得的所有角色组合，动态职责分离难以实现。为了避免用户通过激活某些角色组合来绕过访问控制策略的情况，引入了否定授权，表现为否定角色，语法为

$$ae_k \Rightarrow \neg r_k \tag{3}$$

该规则表明一旦用户满足属性ae_k，系统就阻止用户获得角色r_k。明确否定授权的引入，无论其用户是谁，只要其属性满足特定条件，都能够阻止该用户获得某些特定角色的授权，弥补了肯定授权可能存在的一些安全漏洞。

2.2.3 角色层次的一致性

在RBP-ERBAC模型里，根据用户属性，基于由企业安全策略定义的有限的授权规则集合，为用户自动分配角色。可引入优先级概念来捕捉不同授权规则之间可能存在的关系。符号“ ≥ ”表示规则之间的优先级，如规则 rule_i 优先于rule_j表示为

$$\text{rule}_i \geq \text{rule}_j \Leftrightarrow (\text{ae}_i \rightarrow \text{ae}_j) \tag{4}$$

其中ae_i和ae_j分别为规则rule_i和rule_j的LHS条件。该表达式说明只要用户的属性ae_i蕴含属性ae_j，规则 rule_i 导出的角色r_i将继承规则rule_j导出的角色r_j，即 $\text{r}_i \geq \text{r}_j$，从而拥有角色$r_i$的用户集合一定是拥有角色$r_j$的用户集合的子集。因此，导出的角色层次（IRH）能够捕捉用户-角色分配中的角色继承关系。

一般来说，企业的系统安全管理者是从企业的实际业务流程出发，根据企业安全策略确定执行不同任务所需的权限集合和操作顺序，然后再根据这些职能关系确定不同的角色关系，构成一个符合企业安全管理要求的反映企业组织结构的角色层次。这个角色层次被称为给定角色层次（GRH）。在RBAC96 中，这个角色层次是权限驱动的。在 RBP-ERBAC 模型中，通过引入职能控制集，GRH 变成了职能驱动

$$(\text{r}_i \geq_{\text{GRH}} \text{r}_j) \Rightarrow \text{authorized\_functions}(\text{r}_j) \subseteq \text{authorized\_functions}(\text{r}_i) \tag{5}$$

其中，$\geq_{\text{GRH}}$同样表示角色的继承关系。

理想情况下，IRH 和 GRH 应该是一致的。但是在实际的 Web 服务环境下，随着企业业务流程的变化，访问控制系统的职能控制集与授权规则必然需要进行调整，很难保证 IRH 和 GRH 的一致性。在 RBP-ERBAC 模型中，通过增加职能控制集并采用静态的用户-角色分配机制，系统安全管理者能够方便地进行角色层次的一致性检查，及时调整授权规则和角色-职能分配关系以消除角色冗余、缺失和冲突现象。

3 访问控制模型的评估

3.1 模型的应用

图 3 所示为采用 sever-pull 结构的 RBP-ERBAC 模型应用系统协作图。在 server-pull 结构中，用户不需要访问自己的角色，只需要提供自己的身份验证信息。角色获取机制对用户透明。

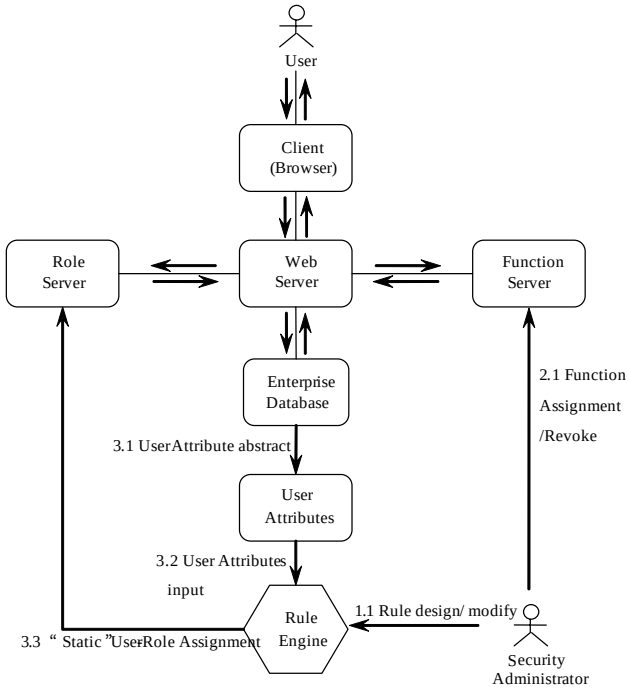

图 3 采用sever-pull结构的RBP-ERBAC模型应用系统协作

访问控制系统正式对用户开放之前，系统的安全管理者需要设计和测试角色分配规则，并根据企业职责划分策略确定角色-职能分配和职能-权限分配关系。然后系统从企业数据库里提取出用户的属性信息，经过规则引擎计算后得出-用户-角色分配表，存入角色服务器。系统每隔一段时间可以再次启动规则引擎，及时更新用户的角色信息。

系统配置完成后，用户通过浏览器输入用户名、密码等身份验证信息，登录 Web 服务器。在进入系统相关模块之前，Web 服务器向角色服务器提出查询用户-角色信息的请求，获得用户角色后向职能服务器提出查询角色-职能信息的请求，激活相关职能权限，阻塞其他无关职能，根据激活的职能控制信息进入相关模块，按规定访问企业数据库里的相关数据。结果信息由 Web 服务器返回到客户端浏览器。

3.2 访问控制模型的比较

将 RBP-ERBAC 模型与 RBAC、RBRBAC 和 RBP-RBAC 模型作一个比较，如表 1。

通过比较可以看出，RBP-ERBAC 模型通过增加职能控制实体，可以降低角色层次的复杂性，增强客体权限控制能力和安全性。由于采用规则“ 静态 ”管理角色授权，可以实现角色分配的自动化，同时克服了规则变化带来的不利影响，保证了访问控制系统的审计能力。这些特性很适合于企业环境，尤其是用户众多、信息客体数量大、业务流程和角色层次复杂的大企业环境下的访问控制。 （下转第 125 页）