维普资讯 http://www.cqvip.com

基于角色访问控制模型及其在操作系统中的实现

刘 伟 孙玉芳
(中国科学院软件研究所 北京100080)

Role-Based Access Control Model and its Implementation in Operating System

LIU Wei SUN Yu-Fang
(Institute of Software, Chinese Academy of Sciencs, Beijing 100080)

Abstract Since Role-based access control shows great advantage in meeting the security need in large-scale, enterprise-wide system, RBAC becomes the hot topic in access control research area. Researchers have proposed several RBAC models, which include the famous RBAC96 model. However, these frameworks are sometimes hard for system developers to understand because the models defined are too abstract or focus on application-oriented solutions. In this paper, a new model (OSRBAC)is discussed, which is the improved model to RBAC3 model in RBAC96 model family. Compared with RBAC3 model, OSRBAC model is more concrete and easilier to understand. At the end, this paper describes the implementation of OSRBAC model in RedFlag Secure Operating System(RFSOS).

Keywords RBAC, Role, Access control, Secure operating system

1 引言

随着电子商务和电子政务的发展,计算机系统安全的重要性进一步增强。访问控制策略(Access Control Policy)[1]作为系统安全策略的重要部分受到广泛的关注。基于角色的访问控制(RBAC)在70年代提出以后,一直是访问控制领域的研究热点。由于在满足政府、军队、商业等部门的安全需求方面显示了极大的优势,RBAC[2,3]作为自主访问控制和强制访问控制等传统访问控制策略的替代和补充引起了广泛的关注,并应用于操作系统、数据库等[4,5]实际系统中。

RBAC 的中心思想是将权限与角色相结合,同时用户被赋予若干角色从而获得相应的权限。这样就能极大地简化权限管理,减少管理访问控制策略的开销,并且易于描述和理解。RBAC 能够按照用户内部的组织结构指定和加强安全策略。任何一个简单的 RBAC 模型都能够满足管理员根据组织内部个人作用描述访问控制策略的要求。

近年来研究人员已经提出若干 RBAC 模型,然而,这些模型定义往往过于抽象或者是面向应用的解决方案。在目前所提出的 RBAC 模型中,Sandhu 提出的 RBAC96模型族[6]由于系统全面地描述了 RBAC 的层次含义及关系而成为角色访问控制研究领域的经典模型,但由于 RBAC96模型族并非针对具体系统实现提出,因此过于抽象,难以实现,其框架很难被系统开发人员理解,因此很多系统[7,8]对 RBAC96模型族进行了修改以适应不同实现的需要。本文在 RBAC96模型族的基础上做出适合操作系统实现的改进[9],提出一个新的模型——OSRBAC 模型,同时给出该模型在 RFSOS 系统上的具体实现。RFSOS 是中科院软件所基于 CC 标准和 Linux 操作系统历经两年的开发并经中科红旗公司商品化后的安全产品,通过专家鉴定、评估取得安全部的认证,并获得销售许可证,达到国家 GB17895中的3级安全标准。

2 现有的模型——RBAC3模型

2.1 RBAC3模型描述

RBAC96模型族由四个概念模型组成:RBAC0,RBAC1,RBAC2,RBAC3。

RBAC0作为基本模型,是任何支持 RBAC 的系统最小要求。RBAC1和 RBAC2都包含 RBAC0,但是增加了不同的特性。RBAC1增加了角色层次的概念,RBAC2增加了角色限制条件。RBAC1和 RBAC2本身并不相互包含。RBAC3包含 RBAC1和 RBAC2,通过传递同时包含 RBAC0。在实际应用中,只有 RBAC3模型才能反映组织内部的权限和能力的对应关系,因此,我们选择在 RBAC3模型的基础上进行改进。

RBAC3模型定义包含若干集合,以及集合上的映射、函数关系,如图1所示。

·U,R,P,S 分别代表用户(User),角色(Role),权限(Permission),会话(Session)

·$PA \subseteq P \times R$,权限到角色的映射关系

·$UA \subseteq U \times R$,用户到角色的映射关系

·$RH \subseteq R \times R$,偏序集,表示角色间继承关系,可以用"≥"或"≤"表示

如果(x,y)∈RH,可以表示为 x≥y 或 y≤x,描述角色 x 继承角色 y 的所有权限

·$RC \subseteq R \times R$,描述角色之间的限制关系

·$user: S \rightarrow U$,将每个会话 s_i 映射到单个用户 $user(s_i)$的函数

·$roles: S \rightarrow 2^R$,将每个会话 s_i 映射到角色集合映射函数 $roles(s_i) \subseteq \{r \mid (\exists\ r1 \geqslant r)[(user(s_i), r1) \in UA]\}$,

同时会话 s_i 获得权限 $\cup_{r \in roles(s_i)} \{p \mid (\exists\ r2 \leqslant r)[(p, r2) \in PA]\}$

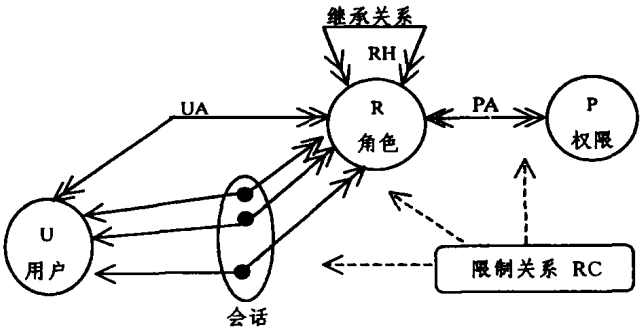

图1 RBAC3模型

维普资讯 http://www.cqvip.com

2.2 RBAC3模型存在的问题及解决方案

由于RBAC3模型在设计的时候并没有考虑具体实现，因此存在下列问题：

(1)RBAC3模型中，会话(Session)的概念描述的是单个用户与角色集合的映射关系，形式过于宽泛，灵活性较差，存在冗余数据较多。在大型系统应用重负载情况下，系统中存在的会话过多将导致系统角色控制数据急剧增加，系统性能严重下降，同时还会造成逻辑关系混乱。针对这种情况，使用活动角色(Active Role)表示用户和角色集合之间动态的对应关系，限制了使用者对系统及数据进行存取所需要的最小权限，既保证了用户能够完成所操作的任务，又确保非法用户或异常操作所造成的损失最小，从而使模型满足“最小特权”(Least Privilege)原则的要求。

(2)对于操作系统的具体实现，用户并不能代表所有操作的执行者，大部分情况下是进程对文件、目录等数据进行处理，RBAC3模型无法准确表示系统的所有操作的执行者。因此，需要对RBAC3模型中的用户(User)的概念进行扩展。我们使用主体(Subject)和客体(Object)描述权限的概念。权限可以理解成为主体对客体进行操作的许可。

3 改进后的模型—— OSRBAC 模型

3.1 基本集合

S：系统中具有主动行为能力的对象。包括用户(User)和进程(Process)

O：系统中能够保存信息的实体，只能被动地接受主体的访问。包括用户(User)、进程(Process)、文件目录(FD)、设备(DEV)、进程间通信(IPC)、系统控制数据(SCD)。

OPERATIONS = {addUser, rmUser, addRole, rmRole, addMaxRole, rmMaxRole, addActiveRole, rmActiveRole, addInheritance, rmInheritance}，所有操作包含系统管理操作(如新增、删除用户)和安全管理操作(如新增、删除活动角色集合)

3.2 模型定义

OSRBAC 模型定义如下：

·S，R，P 分别代表主体(Subject)，角色(Role)，权限(Permission)

·$PA \subseteq P \times R$，权限到角色的映射关系

·$SA \subseteq S \times R$，主体到角色的映射关系

·$PH \subseteq R \times R$，偏序集，表示角色间继承关系，可以用“$\geqslant$”或“$\leqslant$”表示

如果$(x,y) \in RH$，可以表示为 $x \geqslant y$ 或 $y \leqslant x$，描述角色 x 继承角色 y 的所有权限

·$RC = \{RSC, RDC\}$，描述角色之间的冲突关系的集合

·$RSC \subseteq R \times R$，描述角色之间的静态冲突关系的集合

·$RDC \subseteq R \times R$，描述角色之间的动态冲突关系的集合

·$RSC \cap RDC = \varnothing$，角色之间的静态和动态冲突关系集合没有交集

·$max\_roles: S \rightarrow 2^R$，主体到角色的最大角色集合映射关系

$max\_roles(s) \subseteq \{r \mid (\exists\ r1 \geqslant r)[(s,r1) \in SA]\}$

·$active\_roles: S \rightarrow 2^R$，主体到角色的活动角色集合映射关系

$active\_roles(s) \subseteq \{r \mid (\exists\ r1 \geqslant r)[(s,r1) \in SA] \cap (\forall\ r)[(s,r) \in max\_roles(s)]\}$

·$authorized\_roles: S \rightarrow 2^R$，主体到角色的有效角色集合映射关系。例如，对于一个主体 abc 被赋予角色“研发工程师”，角色“研发工程师”继承角色“职员”(即研发工程师也是一种职员)，则“研发工程师”和“职员”都是 abc 的角色

$authorized\_roles(s) \subseteq \{r \mid p \in max\_roles(s) \cap (p,r) \in RH\}$

·$authorized\_users: R \rightarrow 2^S$，角色到主体的有效用户集合映射关系

$authorized\_roles(r) \subseteq \{s \mid r \in authorized\_roles(s)\}$

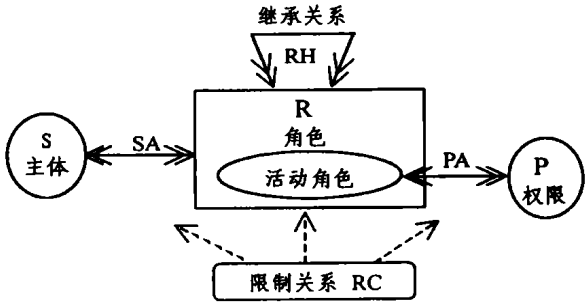

图2 OSRBAC 模型

3.3 状态及转换

系统状态是一个多元组：STATES(S，R，max_roles，active_roles，RH，RSC，RDC)

用 STATES 表示系统状态集合，状态之间的转换通过操作触发，包括安全管理员实施的安全管理操作和主体实施的激活、撤销活动角色集合的操作等。描述如下：

$\delta: STATE \times OPERATIONS \rightarrow STATES$

3.4 模型描述

主体是系统中具有主动行为能力的对象，包括用户和进程。主体拥有两个角色集合(Role Set)，最大角色集合和活动角色集合。最大角色集合是主体拥有的所有角色集合，活动角色集合是主体当前生效的角色集合。最大角色集合由安全管理员指定，活动角色集合是最大角色集合的子集，活动角色集合可以由安全管理员和主体进行调整，但是不能超过最大角色集合的范围。主体可以通过活动角色集合拥有刚够完成工作的最小权限，从而实现“最小特权”原则。用户所赋予的角色由安全管理员指定，进程由于是动态产生的，因此进程拥有的权限包含进程属主的权限集合和其所需要的特殊权限的集合，体现在进程角色集合上就是由进程属主的角色集合和可执行文件被执行后生成进程将拥有的角色集合共同决定。

角色继承描述角色间的层次关系。如果角色 x，y 之间存在继承关系($x \geqslant y$ 或 $y \leqslant x$)，则表示角色 x 继承角色 y 的所有权限，在现实生活中可以对应于上下级之间的关系。

角色冲突描述角色间的互斥关系，包括静态冲突和动态冲突。包含静态冲突关系的多个角色不能赋予同一个主体，静态冲突可以通过继承关系传递。比如企业中的会计和出纳就是具有静态冲突的两个角色，因此不能由同一个人担任。包含动态冲突关系的多个角色可以赋予同一个主体，但是不能同时生效，动态冲突可以通过继承关系传递。

权限可以理解为主体对客体操作的许可，主体拥有某个权限表示允许主体对客体进行某类操作。用一个三元组表示：(主体，客体，访问类型)。

将权限分化为三个系统特权：系统管理特权、安全管理特权和审计管理特权。安全管理员可以将部分系统权限赋予角

维普资讯 http://www.cqvip.com

色。

4 在RFSOS上的实现

4.1 系统逻辑结构

RFSOS的逻辑结构如图3所示。系统内核可以分为安全决策(SD)和决策实施(DE)两个部分，安全决策依赖于安全政策，负责判断一个安全相关行为是否可以执行；决策实施与安全政策无关，负责执行一个已经得到许可的行为所要执行的任务。安全决策内部设立相互独立的政策支持机制，每个安全政策对应一个政策支持机制，这样做能获得灵活的安全政策支持。RBAC作为一个独立的安全政策，参与安全决策判断。

由于传统操作系统中存在的特权用户(如root)拥有的权限过大，因此成为系统安全的重大隐患。在改进的模型中依据"最小特权"原则对特权用户的特权进行分化，根据系统管理任务设立系统角色：

系统管理员：拥有维护系统正常运行所必需的权限，但不具有与安全有关的权限；

安全管理员：拥有管理主客体安全属性的权限，但不具有维护系统的权限和其他权限；

审计管理员：拥有进行安全审计工作所必需的权限；

任意两个系统角色之间都是静态冲突的关系。

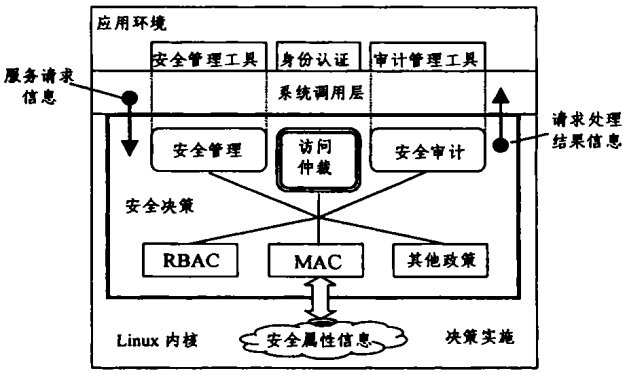

图3 RFSOS系统逻辑结构

4.2 安全决策

在安全决策的过程中，安全决策的相应机制首先将安全请求提交访问仲裁，访问仲裁把决策任务转交给所有的政策支持机制，政策支持机制借助由核心的其他部分提供给安全决策部分的安全属性信息决定是否允许操作，并将结果返回给访问仲裁。

RBAC安全决策分为系统管理特权决策部分和访问控制决策部分。

4.2.1 *系统管理特权决策* 依据主体请求的特权，通过函数调用取得主体的活动角色集合，在活动角色集合的有效特权向量中查找对应的特权，如存在，则允许操作，否则，拒绝操作。通过改造的capability机制中的特权决策函数实现。

4.2.2 *访问控制决策* 从核心的角度看，安全相关行为是由系统调用触发的。以open系统调用打开文件为例，首先主体把打开文件的请求提交给安全决策子系统，访问仲裁机制受理这个请求，并由RBAC政策支持机制作出判断，RBAC政策支持机制首先取得当前主体的活动角色集合，然后检验活动角色集合对文件客体类型的操作权限中是否包含打开文件的权限，如包含，则允许主体打开文件，否则，拒绝访问，最后的结果由安全决策子系统返回给open系统调用。RBAC政策支持机制由裁决函数实现。裁决函数根据主体发出的请求、客体的种类、客体的标识判断是否允许主体操作。

4.3 决策实施

在安全决策子系统允许安全请求以后，决策实施负责执行一个已经得到许可的安全相关行为。仍然以open系统调用为例，在得到安全决策子系统的访问许可以后，决策实施子系统将进行打开文件的操作，最后设置文件客体的相关安全属性。RBAC的决策实施部分由设置函数实现。设置函数主要是系统允许主体对客体进行操作以后，客体的相关安全属性可能会发生变化，需要重新进行设置。

结论 我们在RBAC3模型的基础上作出改进并定义了OSRBAC模型，细化了角色间的限制关系，提出活动角色的概念以代替原有模型中会话的概念，同时扩展了原有模型中的用户范围，使用主体描述系统中具有主动行为能力的对象。最后，我们描述了OSRBAC模型在RFSOS操作系统上的实现机制。在RFSOS操作系统中的具体实现表明OSRBAC模型与RBAC3模型相比具有容易被系统开发人员理解、易于实现等优点，同时验证了模型的正确性。

参考文献

1 McLean J. Security Models and Information Flow. In: IEEE Symposium on Security and Privacy, pp. 180～189

2 Ferraiolo D, Cugini J, Kuhn D R. Role Based Access Control: Features and Motivations. In Annual Computer Security Applications Conf. IEEE Computer Society Press, 1995

3 Ferraiolo D, Kuhn D R. Role Based Access Control. In: the Proc. of the 15th National Computer Security Conf. Vol Ⅱ, 1992. 554～563

4 RBAC in the Solaris [tm] Operating Environment. wwws. sun.com/software/whitepapers/wp-rbac/

5 Chandramouli R, Sandu R. Role Based Access Control Features in Commercial Database Management Systems. In: 21st National Information Systems Security Conf. Oct. 1998

6 Sandhu R S, Coyne E J, Feinstein H L, Youman C E. Role-based access control models. IEEE Computer, 1996, 29(2): 38～47

7 Sandhu R S, Coyne E J, Feinsten H L, Youman C E. Role-Based Access Control: A Multi-Dimensional View. In: Proc. of 10th Annual Computer Security Applications Conf. 1994. 54～62

8 Jansen W A. A Revised Model for Role Based Access Control. NIST-IR 6192, July 1998

9 Gavrila, Barkley. Formal Specification for Role Based Access Control User/Role and Role/Role Relationship Management. In: Third ACM Workshop on Role-Based Access Control