基于时间角色访问控制授权模型的研究

夏启寿[1,3],殷晓玲[1,4],范训礼[1,2]
(1.西北大学 信息科学与技术学院,陕西 西安 710069;
2.西北工业大学 电子信息学院,陕西 西安 710072;
3.池州学院 计算机中心,安徽 池州 247000;
4.池州学院 数学计算机系,安徽 池州 247000)

摘 要:基于角色的访问控制作为一种新型的访问控制技术成为近年来访问控制领域研究的热点,但目前主要工作均立足于与时间特性无关的其他方面。在基于角色访问控制的几种主要模型的基础上,提出了一种基于时间的角色访问控制模型。引入时间的目的是使系统更加安全有效。充分体现了TRBAC模型更能满足信息安全管理的需求,使扩展的TRBAC模型更能适应系统的需求。

关键词:访问控制;角色;时间;TRBAC
中图分类号:TP393.08 **文献标识码**:A **文章编号**:1673-629X(2008)11-0138-04

Research on Authorization Model Based on Time RBAC

XIA Qi-shou[1,3], YIN Xiao-ling[1,4], FAN Xun-li[1,2]
(1. School of Information Science and Technology, Northwest University, Xi'an 710069, China;
4. School of Electronic Information, Northwest Polytechnical University, Xi'an 710072, China;
3. Computer Centre, Chizhou College, Chizhou 247000, China;
4. Department of Maths and Computer, Chizhou College, Chizhou 247000, China)

Abstract: As a new access control technology, role-based access control has been very popular in that research field in recent years, but the main work focus on some other characters which has nothing to do with the time character. Synthesizes main models of role-based access control and designs a time role-based access control model which is called TRBAC model. Introducing time is to make the system more safe and effective. TRBAC can meet the needs of information security and satisfy the systems better.

Key words: access control; role; time; TRBAC

1 RBAC概述

R.S.Sandhu等人在1996年提出了一个基于角色的访问控制参考模型,即RBAC96模型[1],该模型系统地、全面地描述了RBAC多层次的意义而得到了广泛的认可。之后他们很快就提出了著名的分布式角色管理模型ARBAC97(Administrative RBAC)[2],从理论上给出了RBAC模型中角色管理的办法。ARBAC02模型[3]是对ARBAC97模型的一次改进,它保留了ARBAC97模型的主要特征,修改了ARBAC97模型的URA97和PRA97模型,增加了新的概念"组织结构",RRA97模型保持不变。NIST(National Institute of Standards & Technology)提出的NIST模型实质上是RBAC96中RBAC3模型的一个扩展。NRBAC是国内学者提出的一个RBAC模型,NRBAC模型[4]主要是细化了RBAC的实现机制,从其内容来看,它属于RBAC3的扩展。

OASIS(Open Architecture for Secure Interworking Services提出的OASIS模型[5,6]是针对分布式系统的访问控制而设计的。它建立在开放的、非集中式方法的基础上。

根据实际需要在访问控制方面的诸多特点,基于改进RBAC96的实现模型,对RBAC96模型的基本要素重新进行描述和进行形式化定义将它提升为一种扩展的基于角色的访问控制模型TRBAC(Time Role-Based Control Model)。

收稿日期:2008-03-07
基金项目:博士后基金资助项目(20070410381);安徽省自然科学研究项目(KJ2008B116);池州学院自然科学研究项目(XK0714)
作者简介:夏启寿(1975-),男,硕士,讲师,研究领域为信息安全、数据挖掘。

2　时间的概念

目前在 RBAC 系统中引入时间约束的主要有两类：

(1)周期时间约束[7]；

(2)连续时间约束[8]。

文中结合两种时间约束方案提出一种有周期性的连续时间约束模型。

3　对 RBAC 的时间扩展

3.1　具有周期性的连续时间约束授权模型

在基于角色的转授权模型中，发起转授权动作的用户称为授权用户记做 u_{ing}，转授出去的角色称为被转授角色记做 r_{ed}，接受被转授角色的用户称为被授权用户记作 u_{ed}。用 $[t_b, t_e]$ 来表示时限 duration，$[t_b, t_e]$ 是一个时间段[9]，t_b 是时间段 duration 的开始，t_e 是时间段 duration 的结束，其中，$t_b \in N, t_e \in N, t_b < t_e$。

定义 1：周期的表示。

给定日历 $C_d, C_1, C_2, \cdots, C_n$，周期 P 定义为

$$P = \sum_{i=1}^{n} O_i \cdot C_i \triangleright r \cdot C_d$$

这里 $O_1 = \text{all}, O_i \in 2^{IN} \cup \{\text{all}\}, C_i \subseteq C_{i-1} \text{for } i = 2, \cdots, n, C_d \subseteq C_n$，且 $r \in \text{IN}$。其中 all 表法 C_i 的所有时间区间，$O_i \in 2^{IN} \cup \{\text{all}\}$ 是时间区间子集，$C_i \subseteq C_{i-1} (i = 2, \cdots, n), C_d \subseteq C_n$ 是时间区间的长度单位，IN 是自然数集合，r 为时间区间长度。符号 $\triangleright$ 表示将第一部分的周期表示分离出来，即从它所代表的时间区间的开始点持续的日历长度 C_d 的数量。周期 P 对应的时间区间无限集合用 $\prod(P)$ 表示。

定义 2：Function $\prod(\,)$。

已知周期表示：$p = \sum_{i=1}^{n} O_i \cdot C_i \triangleright r \cdot C_d$，$\prod(P)$ 是一个周期时间区间集合，它的一般持续时间是 $r \cdot C_d$。

定义 3：Function Sol()。

设 t 表示时间，P 是一个周期表达式，begin 和 end 是以天为单位的日期表示。$t \in \text{Sol}(< [\text{being}, \text{end}], P)$，当且仅当存在 $\tau \in \prod(P)$，那么 $t \in \tau$ 并且 $t_b \leqslant t \leqslant t_e$，这里 t_b, t_e 分别代表开始时间和结束时间。

定义 4：时间长度的表示。

给定时间 t_i, t_j，其中 $i \leqslant j$，时间长度 L 定义为：$L = t_j - t_i$。

具有周期性的连续时间约束的转授权的具体含义是：u_{ing} 在转授权操作中，仅仅赋予 u_{ed} 在时间周期 P 时间段 duration 中具有 r_{ed}，即 u_{ed} 仅仅在时间段 duration 中可以行使 r_{ing} 所具有的权限，一旦当前时间 $t_b > t_e$，则系统自动撤销 u_{ed} 的 r_{ed}，而且行使 r_{ed} 所具有的权限最大时间长度为 L。文中将 RBAC96 中有关授权管理的部分简化为 secoff 统一行使所有的授权管理工作。

具有周期性连续时间约束的授权 t_ auth 的形式是：{ing, ed, time}，ing = (u_{ing}, r_{ing})，ed = (u_{ed}, r_{ed})，time = $(P, \text{duration}, L, t_d)$，duration = $[t_b, t_e]$，即 tauth = $\{(u_{ing}, r_{ing}), (u_{ed}, r_{ed}), (P, [t_b, t_e], L, t_d)\}$ 其中，$u_{ing} \in U, u_{ed} \in U, r_{ing} \in R, r_{ed} \in R, t_b \in N, t_e \in N, t_d \in N$。

定义以下几类函数：

定义 5：

(1)授权用户和转授角色函数：

ing($t\_$ auth) = ing = (u_{ing}, r_{ing})，

ing$u$($t\_$ auth) = ing$u$(ing) = u_{ing}，

ing$r$($t\_$ auth) = ing$r$(ing) = r_{ing}；

(2) 被授权用户和被转授角色函数：

ed($t\_$ auth) = ed = (u_{ed}, r_{ed})，

ed$u$($t\_$ auth) = ed$u$(ed) = u_{ed}，

ed$r$($t\_$ auth) = ed$r$(ed) = r_{ed}；

(3) 时限和授权时间函数：

time($t\_$ auth) = time = $(P, \text{duration}, L, t_d)$，

time$d$($t\_$ auth) = time$d$(time) = $(P, \text{duration}, L)$，

time$t$($t\_$ auth) = time$t$(time) = t_d；

基于以上的定义，有：

uoa $\in$ UOA$\Leftrightarrow$uoa = $\{(\text{secoff}, A), (u_{ed}, r_{ed}), (P, [t_b, t_e], L, t_d)\}$，其中 $t_e \geqslant t_b \geqslant t_d, L \geqslant 0$，$A$ 表示 secoff 具有系统授予用户任何角色的权限，尽管 secoff 可能并不具有这些角色。

uda $\in$ UDA$\Leftrightarrow$uda$\{(u_{ing}, r_{ing}), (u_{ed}, r_{ed}), (P, [t_b, t_e], L, t_d)\}$，其中 $t_e \geqslant t_b \geqslant t_d, L \geqslant 0, u_{ing} \neq$ secoff。

3.2　具有周期性的连续时间约束模型

具有周期性的连续时间约束模型的基本元素和系统功能函数在定义 5 中已给出。

定义 6：

(1) U：用户的集合；O：客体的集合；A：访问的集合；P：权限的集合；R：角色的集合；S：会话的集合；UOA：用户到初始角色之间的多对多的关系；UDA：用户到被转授角色之间的多对多的关系。

(2)用户角色分配集(URA)：URA = UOA $\cup$ UDA = $\{(u, r) \mid u \in U \wedge r \in R\}$。

(3)角色权限分配集(RPA)：RPA = $\{(r, p) \mid r \in$

$R \wedge p \in P\}$。

(4)RH$\subseteq R \times R$:角色与角色之间的继承关系,该关系是一偏序关系。

(5)用户委派集(UAP):用户委派(*uap*)是指某用户指定另一用户暂时代替自己执行全部或部分自身的权限。

(6)会话集(S):可以简单地将其定义为三元组(ID,u,R)$\in N \times u \times 2^{\text{role}(u)}$,其中ID用于标识某一特定的会话,role($u$)是计算机用户角色集的函数。

(7)users_ $o(r,t)$:$R \to 2^U$返回时刻t具有初始角色r的所有用户

users_ $o(r,t) = \{u \mid (\exists r' \geqslant r)(\text{uoa} = \{(\text{secoff}, A), (u,r'), (P,[t_b,t_e],L,t_d)\} \in \text{UOA}) \cap t \in \text{timed}(\text{uoa})\}$;

(8)users_ $d(r,t)$:$R \to 2^U$返回时刻t具有转授角色r的所有用户

users_ $d(r,t) = \{u \mid (\exists r' \geqslant r)(\text{uda} = \{(u_{\text{ing}}, r_{\text{ing}}), (u,r'), (P,[t_b,t_e],L,t_d)\}) \in \text{UDA}\} \cap t \in \text{timed}(\text{uda})$;

(9)users(r,t):$R \to 2^U$返回时刻t具有角色r的所有用户

users(r,t) = user_ $o(r,t) \cup$ user_ $d(r,t)$;

(10)user(s_i,t):$S \to U$返回时刻t会话s_i所属的用户

user(s_i,t) = $\{u \mid \text{ura} = \{(u_{\text{ing}}, r_{\text{ing}}), (u,r'), (P,[t_b,t_e],L,t_d)\} \in s_i \cap t \in \text{timed}(\text{ura})\}$;

(11)role(s_i,t):$S \to 2^R$返回时刻t会话s_i所具有的所有角色

role(s_i,t) $\subseteq \{r \mid (\exists r' \geqslant r)[\text{ura} = \{(u_{\text{ing}}, r_{\text{ing}}), (\text{user}(s_i,t), r'), (P,[t_b,t_e],L,t_d)\} \in \text{URA} \cap t \in \text{timed}(\text{ura})]\}$;

(12)permissions(s_i,t):$S \to 2^P$返回时刻t会话s_i所具有的所有权限

permissions(s_i,t) = $\{p \mid (\exists r' \geqslant r)[(p,r') \in \text{PA} \cap r' \in \text{role}(s_i,t)]\}$;

(13)role_ $u(u,t)$:$U \to 2^R$返回时刻t用户u所具有的所有角色

roles_ $u(u,t) = \{r' \mid u \in \text{users}(r',t), r' \geqslant r\}$;

(14) N:是一个以自然数为元素构成的集合;

(15)DLGT$\subseteq$URA$\times$URA = $U \times R \times U \times R$;

(16)ODLGT$\subseteq$UOA$\times$UDA;

(17)DDLGT$\subseteq$UDA$\times$UDA;

(18)DLGT$\subseteq$ODLG$\cup$DDLGT;

(19)DTA$\subseteq$URA$\times$URA:转授权树;

(20)path: $U \times R \times ((u_0,r_0),\cdots,(u_i,r_i))$返回授权($u,r$)的授权路径

path(u_0,r_0) = $\{(u_0,r_0),(u_1,r_1),\cdots,(u_i,r_i),\cdots,(u_n,r_n) \mid \{\text{ing}_1, (u_i,r_i), \text{time}_1\} = \text{prior}\{\text{ing}_2, (u_{i-1},r_{i-1}), \text{time}_2\} \in \text{UDA}, i > 0\}$;path$\{(\text{secoff},A), (u,r), \text{time}_1\} = \{\Phi \mid \{(\text{secoff},A), (u,r), \text{time}_1\} \in \text{UOA}\}$;

(21)depth: $U \times R \to N$返回授权(u,r)的授权路径的长度

depth(u,r) = $\{n \mid n = \mid \text{path}(u,r) \mid (u,r) \in \text{UDA}\}$;depth($u,r$) = $\{0 \mid \{(\text{secoff},A),(u,r),\text{time}_1\} \in \text{UOA}\}$;

(22)width: $U \times R \to N$返回同是由($u_{\text{ing}}, r_{\text{ing}}$)转授的授权($u,r$)的个数

width($\{(u_{\text{ing}}, r_{\text{ing}}), (u,r), \text{time}_1\}$) = $\{n \mid n = \mid \text{broths}(\{(u_{\text{ing}}, r_{\text{ing}}), (u,r), \text{time}_1\}) \mid\}$;

(23)valid_ d: $U \times R \times T \to [t_b,t_e]$返回时刻$t$用户$u$具有$r$的有效时限

$$\text{valid\_}\, d(u,r,t) = \begin{cases} [t_b,t_e], & \text{当 } t \leqslant t_b \cap t \in \text{Sol}(< P,[t_b,t_e] >) \cap 1 > 0 \\ [t,t_e], & \text{当 } t_b < t \leqslant t_e \cap t \in \text{Sol}(< P,[t_b,t_e] >) \cap 1 > 0 \\ \varnothing & \text{当 } t_e \leqslant t \cup t \notin \text{Sol}(< P,[t_b,t_e] >) \cap 1 < 0 \end{cases}$$

其中$\{(u_{\text{ing}}, r_{\text{ing}}), (u,r), (P,[t_b,t_e],L,t_d)\} \in$ DLGT。

4 授权与授权撤销

4.1 约 束

约束[10]即某种限制,也可以理解为一种规则,TRBAC模型中主要有三种约束:

(1)先决角色约束集(CR)。

定义7:先决条件CR是用操作符"&"(and,与)和"|"(or,或)将元素cr结合起来的布尔表达式。其中,cr可以是x或者$-x$的形式,前者表示具有角色x,后者表示不具有角色x。这种约束即为先决角色约束,可以定义如下:

$cr = (r_1,r_2,r_3,\cdots,r_n \vdash r_j)$其中$r_i \in R \wedge r_j \in R (1 \leqslant i \leqslant n)$,它表示只有在获得了全部的$r_i (1 \leqslant i \leqslant n)$之后,才可以得到角色$r_j$。系统中所有的先决角色约束的集合即构成先决角色约束集:

$CR = \{cr \mid cr = (r_1,r_2,r_3,\cdots,r_n \vdash r_j), r_i \in R \wedge r_j \in R, (1 \leqslant i \leqslant n)\}$

先决角色约束作用于所有与角色有关的分配,用户组分配、用户委派。

(2)静态权责分离(SSD)。

静态权责分离(SSD) 它定义了相互排斥的角色,在同一个静态权责分离集中的角色不可以分配给同一个用户。静态权责分离可以定义为角色的子集,SSD $\subseteq 2^R \times K$,这里 K 是一个 $\geqslant 2$ 的整数集。即该子集中的角色同时分配给同一用户时,必须小于指定个数 k。

定义 8:静态权责分离集(SSD)。

$SSD = \{(rs,k) \mid rs \subseteq 2^R \wedge k \geqslant 2\}$

$\forall (rs,k) \in SSD(u \in U \Rightarrow |Role(u) \cap rs| < k)$

其中'$|\cdots|$'表示集合的元素个数。该定义表示在给用户分配角色时,用户获得的角色集与 SSD 中任何一个集合的交集,其元素个数必须小于指定的 k。

(3) 动态权责分离(DSD)。

动态权责分离是指一个角色集,该集合中的角色在分配给用户时不受限制,但用户不能同时激活这些角色,由于用户可以同时创建多个会话。因此这一约束要求跨越同一用户同一时间创建的所有会话。

定义 9:动态权责分离集(DSD)。

$DSD = \{(rs,k) \mid rs \subseteq 2^R \wedge k \geqslant 2\}$

$\forall (rs,k) \in DSD(u \in U \Rightarrow |permission(s_i) \cap rs| < k)$

4.2 转授权

转授权判定公式是基于以上几方面进行判断外还要判断系统允许的最大转授权步数 ,某一角色被转授的次数(width) 加以限制。

定义 10:转授权的判定如下:

Can_ delegate $\subseteq R \times CR \times SSD \times DSD \times N \times Y \times TIME$;

$dlgt = \{r, cr, ssd, dsd, n, y, (P, [t_b, t_e], L, t_d)\} \in$ can_ delegate $\Leftrightarrow u_{ing} \in \{u \mid users(r'), r' \geqslant r\} \cap$ roles_ $u(u_{ed}, t) \in cr \cap$ roles_ $u(u_{ed}, t) \in ssd \cap$ roles_ $u(u_{ed}, t) \in dsd \cap n(dlgt) \leqslant n \cap y(dlgt) \leqslant y \cap [t_b, t_e] \subseteq$ valid_ $d(u_{ing}, r, t_d)$

其中,R,CR,SSD,DSD,N,Y 和 TIME 分别是角色、先决条件、静态权责分离,动态权责分离,最大转授权深度、最大转授权宽度、时间的集合;n(dlgt) 表示转授权操作 dlgt 中被转授角色 r 的再次转授次数,即转授权深度;y(dlgt) 表示当前操作中被转授角色被同一用户转授次数, 即转授权宽度。

5 结束语

文中在分析访问控制模型的基础上提出了一个基于时间的角色访问控制模型,并对该模型的授权规则进行系统的论述,这是主要工作。正如文中所述,角色访问控制本身具有很多特性,如何正确地刻画这些性质,以及扩充现有模型或提出新模型以支持多种特性,是今后的一个重要工作。

参考文献:

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

[2] Sandhu R S, Bhamidipati V, Munawer Q. Role - Based administration of User - Role Assignment: The URA97 Model and Its Oracle Implementation[J]. The Journal of Computer Security, 1999, 2: 105 - 130.

[3] Sejong O, Sandhu R. A Model for Role Administration using Organization Structure[C]//Proc. 7th ACM Symposium on Access Control Models and Technologies. Monterey, Calif.: [s.n.], 2002: 155 - 162.

[4] 乔　颖,须　德,戴国忠. 一种基于角色访问控制(RBAC)的新模型及其实现机制[J]. 计算机研究与发展, 2000, 37(1): 37 - 44.

[5] Hayton R J, Bacon J M, Moody K. OASIS: Access Control in an Open, Distributed Environment [C]//In Proceedings of IEEE Symposium on Security and Privacy. Oakland, CA: [s.n.], 1998: 3 - 14.

[6] Hine J, Yao W, Bacon J, et al. Architecture for distributed OASIS services[C]//Proc. Middleware 2000, Lecture Notes in Computer Science. New York: Springer - Verlag, 2000: 107 - 123.

[7] Elisa B, Andrea B P, Elena F. TRBAC: A Temporal Role - Based Access Control Model[J]. ACM Transactions on Information and Systems Security, 2000, 4(3): 21 - 30.

[8] 黄　建,卿斯汉,温红子. 带时间特性的角色访问控制[J]. 软件学报, 2003, 14(11): 1944 - 1954.

[9] Elisa B, Claudio B, Elena F, et al. Atemporal access control mechanism for database systems[J]. IEEE Trans on Knowledge and Data Engineering, 1996, 8(1): 67 - 80.

[10] Ferraiolo D F, Kuhn D R, Chandramouli R. Role - Based Access Control[M]. London: Artech House, 2003.

(上接第 133 页)

[7] Orso A, Harrold M J, Rosenblum D. Component metadata for software engineering tasks[C]//In: Proc of Int'l Workshop on Engineering Distributed Objects (EDO). Lecture Notes in Computer Science 1999. Berlin: Springer - Verlag, 2000: 129 - 144.

[8] Orso A, Harrold M J, Rosenblum D, et al. Using component metacontent to support the regression testing of component - based software[C]//In: Proc of IEEE Int'l Conf on Software Maintenance (ICSM 2001). Los Alamitos, CA: IEEE Computer Society Press, 2001: 716 - 725.