计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
44(4):615~622, 2007

RBAC 实施中国墙策略及其变种的研究

何永忠[1,2] 李晓峰[2] 冯登国[2]
[1](北京交通大学计算机与信息技术学院 北京 100044)
[2](中国科学院软件研究所信息安全国家重点实验室 北京 100080)
(heyongzhong@bigfoot.com)

Implementing Chinese Wall Policies on RBAC

He Yongzhong[1,2], Li Xiaofeng[2], and Feng Dengguo[2]
[1](*School of Computer and Information Technology, Beijing Jiaotong University, Beijing* 100044)
[2](*State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing* 100080)

Abstract The purpose of Chinese Wall policy is to prevent conflict of interest between competing companies. The BN model proposed by Brewer and Nash, and the lattice-based interpretation of Chinese Wall policy proposed by Sandhu are two examples of the Chinese Wall policy models. However, these models are severely restricted and awkward to implement. RBAC is a prevalent model with policy-neutrality. This paper makes a thoroughly research on how to configure RBAC to enforce Chinese Wall policy and its variations. Based on role hierarchies and RBAC constraints, the detailed configurations are presented and the constraints in the configurations are formalized. Compared with traditional models, schemes are more flexible and can be directly enforced in systems supporting the RBAC model.

Key words Chinese Wall policy; access control; RBAC

摘 要 中国墙策略的目的是防止在有竞争关系的企业间访问信息导致的利益冲突. 著名的中国策略模型包括 Brewer 等人提出的 BN 模型和 Sandhu 提出的基于格访问控制模型实施的中国墙模型. 然而这些模型都存在严重的缺陷:灵活性不够或者实施不方便. RBAC 是目前主流的访问控制模型,该模型的特点是策略中立. 基于 RBAC 模型全面研究了实施中国墙策略及其各种变种策略的方法,利用角色体系和 RBAC 的约束机制,给出了具体的构造方法,讨论了这些构造中约束的形式化描述方法. 与传统的中国墙模型相比,更加灵活,并可在支持 RBAC 的系统中直接实施.

关键词 中国墙策略;访问控制;RBAC

中图法分类号 TP309.2

中国墙策略(Chinese Wall policy, CHW)是在商业环境中的一种典型的访问控制策略,这种策略最初来源于证券咨询业务的安全需求. 根据有关法律法规要求,证券公司的咨询人员不能在知道一个企业的竞争对手的内部信息后为该企业咨询,以防止"利益冲突". 根据这些规定,Brewer 等人[1]提出了一种类似于多级强制策略 BLP(Bell Lapudula)[2]模型的实施中国墙策略的安全模型(简称 BN 模型). 该模型提出后受到高度的重视,被认为是商业领域的 BLP 模型. BN 模型中每个公司的数据属于一个公司数据集,具有竞争关系的公司数据集构成一个利益冲突类(conflict of interest, COI). 该模型的简单

收稿日期:2006-05-10;修回日期:2006-11-28
基金项目:国家"八六三"高技术研究发展计划基金项目(2004AA147070);国家自然科学基金项目(90304007,60373048);北京交通大学科技基金项目(2007RC004)

安全性要求一个用户不能同时访问同一个利益冲突类中不同公司数据集. 简单安全性直接反映了相关规章的要求,即如果一个咨询师为某个企业提供咨询后,不能给该企业的竞争对手提供咨询. 然而,仅仅简单安全性不能保证竞争对手的公司数据集之间没有信息流动,因此,"写"访问还必须满足模型的另一个性质:BN-*-特性. 文献[1]进一步证明了如果 BN 模型满足简单安全性和 *-特性就不存在非授权的信息流. 中国墙策略虽然具有广泛的应用,但是实施该策略的 BN 模型却有很大的限制. BN 模型的安全性定义蕴涵的一个限制是,如果一个用户读了两个不同的公司数据集,就不能写任何的公司数据集了. 显然,这个限制将大大影响 BN 模型的应用. Sandhu[3]提出用基于格(lattice-based)的访问控制模型 LBAC 来实施中国墙策略,然而,该方案需要构造的格关系与 BLP 模型实现中典型的格关系相比复杂得多,不便于在实际系统中实现和使用.

由于 RBAC 是一种策略中立的访问控制模型[4],得到广泛的应用,因此,通过配置 RBAC 来实施中国墙策略具有很强的实用价值. Osborn 等人[5]提出了配置 RBAC 实施自主访问控制策略和强制访问控制策略 LBAC 的完整方案. Zhao 等人[6]提出了一个配置 RBAC 实施中国墙策略的初步方案. 本文全面研究了基于角色的访问控制 RBAC 模型实施中国墙策略的机制. 首先给出了配置 RBAC 模型模拟典型的中国墙策略 BN 模型的方案,并证明了两者的一致性. 由于 BN 模型的局限性,本文进一步研究了如何有效的增强实施中国墙策略灵活性的机制,提出了利用用户和主体的分离以及会话(session)约束的方式配置 RBAC 的方案,该方案允许一个用户在不同的会话期"写"访问不同的公司数据集. BN 模型的另一个局限是假设利益冲突关系是具有传递性的,而实际情况并非如此,对于这类中国墙策略,本文提出了相应的 RBAC 配置方案. 在这些配置方案中,RBAC 约束都采用了非形式化描述,而在实际实施策略时,必须采用机器可读的形式化的描述形式. 实际采用的约束描述语言的表达能力都是受限的语言,非形式化的约束不一定能用这些语言描述. 因此,本文进一步研究并给出了采用一种 RBAC 约束描述语言对相关配置方案中的约束进行形式化描述的方法.

1 基本模型概述

Brewer 和 Nash 提出了一种实施中国墙策略的安全模型(BN 模型),它可以防止由于信息在竞争对手间的流动带来的利益冲突 COI. BN 模型还支持清洁信息(sanitized information),所有的用户都可以读清洁信息(所谓清洁信息是指所有可以公开的信息,包括公司的非敏感信息或者隐藏了公司身份的信息). 模型定义一个独立的公司数据集 g_1 和利益冲突类 t_1 来刻画清洁信息. 形式化的定义如下:

定义 1. BN 模型. BN 模型由下面的基本元素和两个安全性质组成.

O,S,G 和 T 分别为对象集合、主体集合、公司数据集的集合、利益冲突类集合.

$group(o)$:$O\rightarrow G$ 把对象映射到公司数据集的函数.

$type$:$G\rightarrow T$ 把公司数据集映射到利益冲突类的函数.

g_1:特殊公司数据集,只有清洁数据属于该数据集.

t_1:特殊利益冲突类,只有 g_1 属于该冲突类.

1) BN-简单安全性质. 一个主体 s 可以访问(读、写)数据对象 o 的条件是满足下列两者之一:

① 主体 s 已经访问过与对象 o 属于同一个公司数据集的对象 o',即

$$group(o) = group(o');$$

② 对于主体 s 访问过的每一个对象 o',它属于的公司数据集都和对象 o 的公司数据集不在同一个利益冲突类中,即 $type(group(o))\neq type(group(o'))$.

2) BN *-特性. 一个主体 s 可以写数据对象 o 的条件是下面两个条件都满足:

① 据 BN-简单安全性质,主体 s 可以访问该对象 o;

② 对于所有的对象 o',如果主体 s 访问过该对象 o',则该对象必须与主体 s 将要写的对象 o 在同一个公司数据集中,或者是清洁数据,即

$group(o)=group(o')$ 或者 $group(o')=g_1$.

RBAC 模型是目前主流的访问控制模型,不仅有大量的研究论文,而且很多商业系统都实现了对 RBAC 的支持. 我们下面描述 RBAC 基本模型的定义.

定义 2. RBAC 模型. RBAC 的组成部分包括:

① U,R,P,和 S:分别表示用户集、角色集、权限集和会话集;

② $PA\in P\times R$:权限—角色的多对多关系;

③ $UA\in U\times R$:用户—角色的多对多关系;

④ $RH \in R \times R$:角色体系,是角色集合上的偏序关系,如果$(r_1, r_2) \in RH$,则称角色 r_1 支配角色 r_2,记为 $r_1 \geqslant r_2$;

⑤ $user: S \rightarrow U$:一个会话对应一个用户;

⑥ $roles: S \rightarrow 2^R$:一个会话对应多个角色,表示该会话当前激活角色集;

⑦ 一系列关于 PA,UA,会话-角色关系($roles$)的约束.

对RBAC模型中角色体系的管理,PA,UA 关系的管理等属于RBAC管理特权.在配置RBAC实施中国墙策略时,需要对 UA 关系管理特权的配置.由于目前还没有公认的RBAC管理模型,本文不对 UA 管理进行形式化描述.

定义3. LBAC模型. LBAC模型的主要元素包括主体、客体、操作模式和安全标记.安全标记是一个标记符合集合,并且在该集合上定义"支配"关系,该支配关系构成格关系.操作模式至少包括读操作和写操作.每个主体和客体都有安全标记.主体对客体的访问请求必须满足下面的两个性质.

① 简单安全性:对所有的访问请求、主体的安全标记必须支配客体的安全标记;

③ *-特性:如果访问请求是写操作,主体的安全标记必须被客体的安全标记支配.

2 配置RBAC模拟BN模型

配置RBAC来实施中国墙策略需要模拟中国墙策略的两个重要的特征是:①基于历史的访问控制特征,即是否准许一个主体的访问请求取决于该主体以前访问了哪些公司数据集;②自主和强制结合的特征,即一个主体访问一个数据对象受到了强制策略的限制,但是主体可以在满足强制策略限制情况下主动选择需要访问的对象.根据文献[7]中的定义,中国墙策略属于基于扁平访问历史(shallow access history)的访问控制策略.在这类策略中,访问控制决策只与一个访问事件是否发生过有关,与访问历史中访问事件的顺序以及次数没有关系.我们在配置RBAC模拟中国墙策略时正是利用了该策略的基于扁平访问历史的特点.配置RBAC的思想是允许每个用户都拥有可以授予自己任何角色的管理特权,但不具有取消角色的特权.一个用户在访问一个数据对象时,如果它不具备相应的访问权限,就需要先授予自己相应的角色从而获得权限.为了模拟中国墙策略,要求一个用户只有在必须的时候才授予自己相关的角色(访问一个数据对象但没有相应的授权时),如果一个用户不需要访问某个公司数据集中的数据,就不授予自己相应的角色.该要求是合理的,因为安全系统一般都基于合法用户不是恶意的假设,并且,即使一个用户不遵守上面的"必须时授权"原则要求,那么只会导致限制该用户的自主访问权限,但也不会导致在竞争公司间形成非法信息流.这样,任何状态下,一个用户是否访问过某个公司数据集就与拥有对应权限的角色是否授予了该用户等价,所以用户-角色关系 UA 的状态变化就等价于用户对公司数据集的扁平访问历史.通过对RBAC的配置,一方面利用用户-角色关系 UA 的状态来模拟中国墙策略的基于扁平访问历史的特性;另一方面,通过允许用户拥有授予自己角色的管理权限(具有自主性),并通过对用户-角色关系进行限制的策略(具有强制性)模拟中国墙策略的自主和强制的两种特征.

BN模型是最早提出的中国墙策略,我们基于上述配置RBAC的思想,首先配置RBAC来模拟BN模型.

构造1. RBAC模拟BN模型的配置方案.

假设BN模型中有 n 个公司数据集 $\{g_1, g_2, \cdots, g_n\}$,$m$ 个利益冲突类 $\{t_1, t2, \cdots, t_m\}$,其中包含公共数据对应的公司数据集 g_1 和利益冲突类 t_1.那么相应的RBAC配置为

1) 角色集合 $R = \{R_1, R_2, \cdots, R_n, W_1, W_2, \cdots, W_n, T_1, T_2, \cdots, T_m\}$.

2) 角色体系 RH,由安全管理员配置,定义如下.角色集合 $Rd = \{R_1, R_2, \cdots, R_n\}$ 中的角色之间没有任何关系,同样,在 $Wt = \{W_1, W_2, \cdots, W_n\}$ 和 $T = \{T_1, T_2, \cdots, T_m\}$ 集合内部的角色之间没有任何关系.如果在BN模型中 $type(g_j) = t_i$,那么,RBAC中有 $T_i \geqslant W_j$,$W_j \geqslant R_j$ 成立.

3) $P = \{(o, r), (o, w) \mid o \in O\}$.

4) 关于 UA 的约束:每个用户最多可以拥有的角色包括 $\{W_1, W_2, \cdots, W_n\}$ 中的一个角色和 $\{R_1, R_2, \cdots, R_n\}$ 中的多个角色.并且,任何用户拥有的所有角色中,不存在属于 $\{R_1, R_2, \cdots, R_n\}$ 的两个角色都受到同一个 $T_i (i \in [1, m])$ 角色支配.如果一个用户拥有 $\{W_1, W_2, \cdots, W_n\}$ 中的一个角色 W_i,那么他必须同时直接(不含通过继承)拥有角色 R_i,并且也可以拥有 R_1(显然,当 $i = 1$ 时该用户只能有 Rd 中的一个角色 R_1),除此之外,不能拥有其他 $\{R_1, R_2, \cdots, R_n\}$ 中的角色.

5) 关于会话的约束:无.

6) 关于 PA 的约束:(o,r)被授予 R_i 当且仅当(o,w)被授予 W_i;(o,r)仅被授予一个角色. 如果 $group(o)=g_i$,则(o,r)被授予 R_i.

7) 角色管理权限的配置:每个用户都拥有授予自己角色的管理权限(并且要求主体在访问数据对象时如果没有相应的权限才给自己授予相应的角色),所有用户可以取消自己拥有的 Wt 中的角色.

注意,在角色管理权限配置中,用户可以取消属于 Wt 的角色(写角色),不可以取消 Rd 中的角色,这是为了和 BN 模型保持一致的需要. 因为根据 BN 模型,除了清洁数据外,一个主体只有在没有访问其他的公司数据集前,才能对一个公司数据集执行写操作;之后,该主体还可以读别的公司数据集,但是一旦读了两个不同的公司数据集,它就再不能写任何公司数据集了. 因此,我们在构造 1 中通过取消用户拥有的写角色来精确地模拟 BN 模型的特性. 这样,构造 1 中,Rd 中的读角色授予用户就表示用户访问过对应公司数据集,因此,一个用户拥有的所有读角色对应于 BN 模型的访问历史矩阵.

下面看一个具体的例子. 假设有 7 个公司数据集 $\{g_1,g_2,\cdots,g_7\}$和 4 个利益冲突类 $\{t_1,t_2,t_3,t_4\}$(包含公共数据集 g_1 和公共冲突类 t_1). 并且有 $group(g_1)=t_1$,$group(g_2)=group(g_3)=group(g_4)=t_2$,$group(g_5)=t_3$,$group(g_6)=group(g_7)=t_4$. 那么,相应的 RBAC 配置的角色体系如图 1 所示.

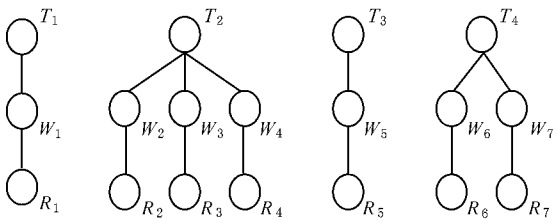

Fig. 1 Role hierarchies configuration example.

图 1 角色体系配置示意图

构造 1 是否对 BN 模型准确模拟可以通过它们之间的一致性来说明. 首先定义模型的一致性.

定义 4. 如果从访问控制模型的初始状态开始,对于任何一组访问请求序列,两个访问控制模型的访问控制决策序列都是相同的,则称这两个模型是一致的. 其中一个访问请求是三元组(主体,客体,访问模式),一个访问决策是准许或者拒绝.

定理 1. 构造 1 和 BN 模型是一致的.

构造 1 和 BN 模型的初始状态定义如下. 构造 1 的初始状态为所有用户都没有被授予任何角色. BN 模型中访问历史用矩阵 $\boldsymbol{A}(S,N)$表示,其中 S 是主体集,N 是自然数集,对应公司数据集的编号. 主体 s 访问公司数据集 g_i 的数据则有 $\boldsymbol{A}(s,i)=$ true. g_1 代表清洁数据. 初始状态时 $\boldsymbol{A}(S,N)$元素全为 false. 从直观上看,构造 1 和 BN 模型的一致性比较明显,然而,由于 BN 模型中允许的合法操作序列是较为微妙的,因此两者的一致性必须经过严谨的证明. 从证明中可以看到,构造 1 在各种不同的情况下是如何模拟 BN 模型的. 证明的思路是采用对访问请求序列的长度进行结构归纳证明方法. 我们证明对任何的访问请求序列中的一个访问请求,如果构造 1 中 RBAC 允许访问,则对应的 BN 模型也允许访问,反之亦然. 为了证明的方便,我们还需要定义 BN 模型和构造 1 的状态一致性,即如果 $\boldsymbol{A}(s,i)=\text{true}\Leftrightarrow(s,R_i)\subseteq UA$,则 BN 模型和构造 1 状态一致,证明见本文附录.

3 基于用户和主体分离的 CHW 实施

BN 模型是最早实施中国墙策略的强制模型,然而,进一步的研究发现,该模型存在严重的问题(不考虑清洁数据):①一个主体如果从两个不同的公司数据集读访问过数据,那么这个用户不能写访问任何的数据;②一个主体只有在仅仅访问过一个公司数据集时才能写访问这个公司数据集.

因此,BN 模型在使用中"写"操作存在重大的限制. 如何增强实施中国墙策略的灵活性成为一个重要的研究课题. Sandhu[3] 提出通过区分主体和用户概念来改进 BN 模型的思想,并在此基础上,采用了格访问控制模型(lattice-based access control, LBAC)来实施中国墙策略. Sandhu 方案基于这样的假设,即系统用户不是恶意的,不会有意泄漏信息,系统安全策略的目标主要是要防范恶意进程,因此在系统中引入用户和主体的概念,它们具有不同的权限. 在 Sandhu 方案中一个用户可以对应多个本体(principal),一个本体对应多个主体(subject). 本体和其对应的主体具有相同的权限,而不同的本体可以有不同的权限. 一个用户可以通过启动不同的本体来实现对不同公司数据集的读写访问,从而增加中国墙策略的灵活性.

但是,Sandhu 方案采用的安全标记非常特殊,不利于实现. 该方案安全标记为

$$LABEL=\{[i_1,i_2,\cdots,i_m]\mid i_1\in COI_1\cup\{\perp\},\cdots,i_m\in COI_m\cup\{\perp\}\},$$

其中,m 是系统中利益冲突类的个数,COI_i 是第 i 个冲突类中的公司数据集(假设数据为 M_i 个)的编号集合,即 $COI_i = \{1,2,\cdots,M_i\}$.

该方案定义的安全标记与典型的安全标记方法差异很大. 典型的安全标记由敏感级别和范畴组成:$LABEL = \{[l,c] \mid l \in Level, c \in Category\}$,其中,敏感级别 $Level$ 是服从严格线性关系的集合,从低级到高级. 范畴 $Category$ 是满足偏序关系的集合.由于典型安全标记和 Sandhu 方案差异如此大,因此在支持典型的安全标记的 LBAC 系统中,实现 Sandhu 方案非常不方便.

实际上,Sandhu 方案中的 $Principal$ 等价于 RBAC 中的会话的概念,因此可以利用 RBAC 模型中会话的概念来改进构造 1,实施与 Sandhu 方案类似的中国墙策略. 这里的一个前提是,同一个用户的不同会话之间不能通过用户的本地存储(local memory)通信. 所谓本地存储就是一个用户的私有存储空间,每次会话都可以读写这个私有存储空间. 因为,如果用户存在本地存储,那么每次会话之间就存在通过用户的本地存储进行通信并泄漏信息的可能. 具体做法是,取得某个用户信任的恶意程序先启动一个主体读利益冲突类 c 中某个公司数据 i,把结果暂时保存,然后再启动另一个主体把暂时保存的结果写到另一个利益冲突类 d 中的公司数据集 j 中;之后,取得了另一个用户信任的恶意程序把数据从 j 读出后写入冲突类 c 中不同于 i 的一个公司数据集 k 中,从而导致数据从 i 泄漏到 k. 这样必然会导致违反中国墙策略不能同时读利益冲突公司数据的原则.

当用户不存在本地存储时,可以利用会话约束,改进构造 1,限制每个会话只能读写一个公司数据集(都可以读清洁数据集),而一个用户通过建立不同的会话就可以读写多个不同的公司数据集,从而增强中国墙策略实施的灵活性.

构造 2. 无用户本地存储的 RBAC 实施中国墙策略. 构造 2 对构造 1 进行改进,除了关于 UA 的约束和关于会话的约束不同,其他的与构造 1 相同.

1) 关于 UA 的约束:每个用户可以拥有 $\{R_1, R_2,\cdots,R_n\}$ 和 $\{W_1, W_2,\cdots,W_n\}$ 中的多个角色. 任何用户拥有的角色中,不存在属于 $\{R_1, R_2,\cdots,R_n\}$ 的两个角色,它们都受到同一个 T_i($i \in [1,m]$)角色支配. 如果一个用户拥有 $\{W_1, W_2,\cdots,W_n\}$ 中的一个角色 W_i,那么他必须同时直接拥有角色 R_i.

2) 关于会话的约束:任何一次会话最多有 3 个角色 $\{W_i, R_i, R_1\}$.

在构造 2 中,每个用户的最多可以拥有的权限包括"读、写"不同的利益冲突类中一个公司数据集,而该用户的一次会话只能读写一个公司数据集以及读清洁数据. 由于不同的会话间不能通过用户本地存储通信,可以证明构造 2 不会导致利益冲突,即任何用户都不可能读到有竞争关系的两个公司的信息.详细证明从略.

4 非传递性利益冲突的 CHW 实施

除了前面提到的 BN 模型的缺点外,该模型还有一些不合理的限制. BN 模型要求一个公司数据集只能属于一个利益冲突类,而实际应用中,一个公司数据集可能属于多个利益冲突类. 用 RBAC 实施中国墙策略时可以很容易解决这个问题. 比如,假设公司数据集 g_2 同时属于冲突类 t_2 和冲突类 t_3,那么在构造 1 和构造 2 的角色体系中,设定 W_2 的上级角色是 T_2, T_3.

BN 模型根据利益冲突关系划分为利益冲突类,其中蕴涵的假设是利益冲突关系是有传递性的. 但实际情况是,利益冲突关系并不一定具备这一特征.比如一个跨国公司和两个在不同国家的国内公司具有利益冲突关系,但是这两个国内公司的市场不同,并不具有利益冲突. 所以,这时,利益冲突关系不具有传递性. 因此,RBAC 的角色体系的配置应该相应改变. 一种方法是把每一对有竞争关系的公司组成一个利益冲突类,一个公司可以出现在多个利益冲突类中. 该方法的缺点是会引入很多的利益冲突类角色. 下面给出另一种更简洁的构造方法. 先定义中国强策略的利益冲突函数 $Goi: G \to 2^G$,该函数计算与公司数据集 g 有利益冲突的公司数据集的集合 $Coi(g)$. 下面的构造 3 基于该利益冲突函数定义角色体系,其中增加了刻画利益冲突函数的角色集合 C.

构造 3. 非传递性利益冲突的 RBAC 配置. 构造 3 对构造 2 进行改进,除了角色集合、关于 RH 的构造和 UA 的约束不同,其他的与构造 2 相同.

1) 角色集合 $R = \{R_1, R_2,\cdots,R_n, W_1, W_2,\cdots,W_n, T_1, T_2,\cdots,T_n, C_1, C_2,\cdots,C_n\}$.

2) 角色体系 RH 由安全管理员配置,定义如下. 角色集合 $Rd = \{R_1, R_2,\cdots,R_n\}$ 内部的角色之间没有任何关系,同样,在 $Wt = \{W_1, W_2,\cdots,W_n\}$, $T = \{T_1, T_2,\cdots,T_n\}$ 和 $C = \{C_1, C_2,\cdots,C_n\}$ 几个集合内

部的角色之间没有任何关系. 对所有的 i,有 $T_i \geqslant W_i$,$T_i \geqslant C_i$,$W_i \geqslant R_i$. 如果 $g_j \in Coi(g_i)$,那么,$C_i \geqslant W_j$ 成立.

3) 关于 UA 的约束:每个用户可以拥有 $\{R_1, R_2, \cdots, R_n\}$ 和 $\{W_1, W_2, \cdots, W_n\}$ 中的多个角色. 任何用户拥有的所有角色中,不存在属于 $\{R_1, R_2, \cdots, R_n\}$ 的两个角色,它们都受到同一个 $T_i(i \in [1,n])$ 角色支配并且其中一个角色不受角色 C_i 所支配. 如果一个用户拥有 $\{W_1, W_2, \cdots, W_n\}$ 中的一个角色 W_i,那么他必须同时直接拥有角色 R_i.

在角色体系构造中,一个公司数据集对应的角色(W_i,R_i)受到角色 T_i 的支配,而与该公司数据集有利益冲突的公司数据集对应的角色都受到角色 C_i 的支配,C_i 又受到 T_i 的支配. 这种角色体系是对利益冲突函数的模拟,有利于对 UA 约束进行形式化描述.

5 RBAC 约束的形式化

在前面配置 RBAC 实施中国墙策略时,我们对 RBAC 约束的配置采用的都是自然语言. 采用自然语言存在的问题是:①自然语言容易导致歧义;②在实际系统中,必须采用机器可读的形式化语言描述,但因为效率的因素,约束语言的描述能力都是受限制的,这样就产生了约束描述的可行性问题,即配置的约束能否以及如何用实际约束语言进行描述. 因此有必要研究采用适当的形式化语言对本文提出的几个构造中的约束进行描述. RCL2000[8] 是特别针对 RBAC 设计的约束条件描述语言,该语言等价于一种受限的一阶逻辑. 基于 RCL2000,我们对构造 1 和构造 3 中的约束进行形式化描述. 下面先简单介绍 RCL2000 的语法和语义.

定义 5. RCL2000.

$op ::= \in \mid \cup \mid \cap$,

$size ::= \varnothing \mid 1 \mid \cdots \mid N$,

$set ::= U \mid R \mid OP \mid O \mid P \mid S \mid CR \mid CP \mid CU$,

$function ::= user \mid roles \mid roles^* \mid sessions \mid permissions \mid permissions^* \mid operations \mid object \mid OE \mid AO$,

其中,该定义中与定义 2(RBAC)中相同的符号的意义也相同,或者进行了扩展. op 是通常的集合运算,$size$ 是自然数,OP 是访问模式的集合,O 是对象的集合,CR,CP,CU 分别表示冲突的角色集的集合、冲突权限集的集合和冲突用户集的集合,比如 $CR = \{cr_1, cr_2, \cdots, cr_s\}$,$cr_i \subseteq R$.

函数 $roles$,$roles^* : U \cup P \cup S \rightarrow 2^R$ 和 $roles(u) = \{r \in R \mid (u,r) \in UA\}$. 与 $roles$ 不同,$roles^*$ 考虑了角色继承关系,定义为 $roles^*(u) = \{r \in R \mid \exists r' \geqslant r \cdot (u,r') \in UA\}$. 其他的函数定义略. OE(one element)是在一个集合中取一个元素的函数,比如 $OE(U) = u, u \in U$. AO(all other)是在一个集合中去除了 OE 的其他元素,比如 $AO(U) = U \setminus \{OE(U)\}$.

我们用一个例子来说明 RCL2000 如何描述约束条件. 假设一个约束为任何用户都不能被赋予有冲突关系的角色,则形式化表示为 $|roles(OE(U)) \cap OE(CR)| \leqslant 1$,其中 $roles(OE(U))$ 返回赋予某个用户 $OE(U)$ 的所有角色,$OE(CR)$ 表示某一个冲突角色集. 把上述表达式转换成谓词逻辑,形式为 $\forall u \in U \wedge cr \in CR \times |roles(u) \cap cr| \leqslant 1$. 另外,在下面的约束形式化描述中,还需要用到关于角色关系的函数 $juniors^* : R \rightarrow 2^R$,其中 $juniors^*(r) = \{r' \mid \exists r' \neq r \times r \geqslant r'\}$.

定义 6. 构造 1 中 UA 约束的形式化描述如下:

$$
\begin{aligned}
&|roles(OE(U)) \cap T| = \varnothing \wedge \\
&|roles^*(OE(U)) \cap Wt| \leqslant 1 \wedge \\
&|juniors^*(OE(T)) \cap \\
&roles^*(OE(U)) \cap Rd| \leqslant 1 \wedge \\
&|roles^*(OE(U)) \cap OE(Wt)| = 1 \Rightarrow \\
&\quad |juniors^*(OE(Wt)) \cap \\
&\quad\ roles(OE(U)) \cap Rd| = 1 \wedge \\
&\quad |roles(OE(U)) \cap Rd \setminus \{R_1\}| \leqslant 1 \wedge \\
&W_1 \in roles^*(OE(U)) \Rightarrow \\
&|roles(OE(U)) \cap Rd| = 1.
\end{aligned}
$$

构造 2 中 UA 约束的形式化表示是构造 1 中 UA 约束的简化形式,具体描述略.

定义 7. 构造 3 中 UA 约束的形式化描述如下

$$
\begin{aligned}
&|roles(OE(U)) \cap T| = \varnothing \wedge \\
&|roles(OE(U)) \cap C| = \varnothing \wedge \\
&|juniors^*(OE(T)) \cap juniors^*(OE(C)) \cap \\
&\quad roles^*(OE(U)) \cap Rd| \geqslant 1 \Rightarrow \\
&\quad |juniors^*(OE(T)) \cap juniors^*(OE(C)) \cap \\
&\quad\ roles^*(OE(U)) \cap Rd| = \\
&|juniors^*(OE(T)) \cap \\
&\quad roles^*(OE(U)) \cap Rd| \wedge
\end{aligned}
$$

$|\ roles^{*}(OE(U)) \cap OE(Wt)\ | = 1 \Rightarrow$

$|\ juniors^{*}(OE(Wt)) \cap$

$roles(OE(U)) \cap Rd\ | = 1.$

上面给出的 UA 形式化描述的特点是不依赖于具体的公司数据集的竞争关系函数,因此对任何情况的公司竞争关系,UA 约束都是一样的,从而大大方便了管理员的配置.

6 结束语

中国墙策略是在商用环境下的一种重要强制访问策略,但由于缺少有效的实施模型和机制,所以没有在实际系统中广泛采用. 我们利用目前主流的RBAC模型来模拟中国墙策略及其各种变种策略,给出了详细的配置方案,证明了配置方案与中国墙策略的一致性,并且对关键的 UA 约束给出了可行的形式化约束. 与格访问控制模型模拟中国墙策略相比,用RBAC模拟中国墙策略更加简洁和灵活,实施更加有效. 由于RBAC已经获得了广泛的认可,越来越多的实用系统都支持RBAC模型,在这些系统上,本文提出的配置方案可以直接实现,因此大大方便了中国墙策略的实施.

参 考 文 献

[1] D Brewer, M Nash. The Chinese Wall security policy [C]. In: Proc of the IEEE Symposium on Research in Security and Privacy. Los Alamitos, CA: IEEE Computer Society Press, 1989. 206-214

[2] K Bell, L J LaPadula. Secure computer systems: Unified exposition and multics interpretation [R]. MITRE Corporation, Tech Rep: MTR-2997, 1976

[3] R Sandhu. A lattice interpretation of the Chinese wall policy [C]. In: Proc of the 15th NIST-NCSC National Computer Security Conference. Washington: United States Government Printing Office, 1992. 329-339

[4] R S Sandhu, E J Coyne, H L Feinstein, *et al*. Role-based access control [J]. IEEE Computer, 1996, 29(2): 38-47

[5] S Osborn, R Sandhu. Configuring role-based access control to enforce mandatory and discretionary access control policies [J]. ACM Trans on Information and System Security, 2000, 3(2): 85-106

[6] Q Zhao, Y Sun. Configuring role-based access control to enforce Chinese Wall policy [C]. The 6th Int'l Conf for Yonger Computer Scientists, Hangzhou, 2001

[7] Philip W L Fong. Access control by tracking shallow execution history [C]. In: Proc of the 2004 IEEE Symp on Security and Privacy. Los Alamitos, CA: IEEE Computer Society Press, 2004. 43-55

[8] G-J Ahn, R Sandhu. Role-based authorization constraints specification [J]. ACM Trans on Information and System Security, 2000, 3(4): 207-226

He Yongzhong, born in 1969. Received his Ph. D. degree in computer science from the Institute of Software, the Chinese Academy of Sciences. His current research interests are computer security and cryptographic protocols.

何永忠,1969年生,博士,主要研究方向为计算机安全和密码协议.

Li Xiaofeng, born in 1973. He is a Ph. D. candidate in the Institute of Software, the Chinese Academy of Sciences. His current research interests include database security, security evaluation and security management.

李晓峰,1973年生,博士研究生,主要研究方向为数据库安全、安全评估和安全管理.

Feng Dengguo, born in 1965. Research professor and Ph. D. supervisor in the Institute of Software, the Chinese Academy of Sciences. His current research interests are computer security and cryptography.

冯登国,1965年生,研究员,博士生导师,主要研究方向为计算机安全和密码学.

Research Background

There are many different access control models, typically including mandatory access control models (MAC), discreet access control models (DAC), role based access control model (RBAC). It is of both theoretical and practical importance to investigate the relationship among these models. Chinese Wall policy model is one of MAC models with some features of DAC models. The purpose of Chinese Wall policy is to prevent conflict of interest between competing company. There are some research about how to configure RBAC models to simulate Bell Lapudula model and DAC models. However, how to configure RBAC to simulate Chinese Wall policy is unknown. This paper makes a thoroughly research on this problem. Not only the traditional Chinese Wall policy is simulated by RBAC in this paper, but also some key variations of the Chinese Wall policy are studied. Based on role hierarchies and RBAC constraints, the detailed configurations are presented and the constraints in the configurations are formalized. Our results means that RBAC is a very general model and can simulate various access control models even as the Chinese Wall policy models which are

sensitive to access history. Practically, our schemes are more flexible than previous models and can be directly enforced in systems supporting RBAC model. This work is supported by the Natural Science Foundation of China under grant No.90304007, 60373048, and High Technology Development Project under Grant No. 2004AA147070.

附录 A

定理 A1. 构造 1 和 BN 模型是一致的.

证明. 对访问请求序列的长度进行结构归纳证明.

1) 访问序列长度为 1,即只有一个访问请求$(s,o,r/w)$时,显然,不论是对哪个数据的读、写请求,BN 模型的访问决策都是准许的. 而对于构造 1,主体对应的用户可以根据访问请求,如果是读请求,就授予自己 Rd 中一个相应角色R_i,如果是写请求,就授予自己相应的 Wt 中和 Rd 中的一个角色 $\{R_i,W_i\}$,显然,这样授予的角色是符合 UA 约束的. 主体激活角色就可以获得相应的访问权限. 所以 BN 模型和构造 1 的访问决策此时是一致的. 另外,访问导致状态转移,有 $\boldsymbol{A}(s,i)=\text{true},(s,R_i)\subseteq UA$,因此 BN 模型和构造 1 两者的状态也是一致的.

2) 假设访问序列长度不大于 k 时,BN 模型和构造 1 的决策是一致的,并且状态也是一致的. 那么我们需要证明,访问序列长度不大于 $k+1$ 时,两者也是一致的. 显然,由于访问序列中后面的访问请求不能影响对前面的访问情况的访问决策,所以,根据假设对前 k 个访问请求的访问决策是一致的,下面只需要考虑对访问序列中第 $k+1$ 个请求的访问决策.

我们根据不同的情况,分别讨论两者的一致性.

① 如果第 $k+1$ 个请求是读清洁数据,根据 BN 模型访问被准许的,并且 $\boldsymbol{A}(s,1)=\text{true}$. 而对于构造 1,如果 R_1 已经授予主体 s 对应的用户,那么该主体激活此角色就可以获得对应的访问权限;否则,就把 R_1 授予自己(容易看到,这是满足 UA 约束的),激活后就可以访问清洁数据. 并且该访问导致$(s,R_1)\subseteq UA$.

② 如果第 $k+1$ 个请求是写清洁数据,根据该主体是否访问过其他公司数据集的情况,BN 模型作出拒绝或者准许的访问决策. 如果主体 s 访问过其他数据集$j(j\neq 1)$,必然有 $\boldsymbol{A}(s,j)=\text{true}$ 所以$(s,R_j)\subseteq UA$. 根据构造 1 的 UA 约束,拥有 W_1 角色的用户不能有除了 R_1 外的其他 Rd 中的角色,所以当$(s,R_j)\subseteq UA$ 时 W_1 不能授予用户 s,写访问被拒绝. 反之,如果主体 s 没有访问过其他数据集,则可以把 W_1 授予用户 s,同时根据构造 1 的 UA 约束的要求要把 R_1 授予用户 s,这样,写访问被准许,并且有 $\boldsymbol{A}(s,1)=\text{true},(s,R_1)\subseteq UA$.

③ 如果第 $k+1$ 个请求是读某个公司数据集 i 的数据$(i\neq 1)$. 根据该主体是否访问过同一个利益冲突类中的其他公司数据集,BN 模型作出拒绝或者准许的访问决策. 相应的,根据构造 1 中 UA 约束,用户不能或者可以把角色 R_i 授予自己.

因为,如果主体 s 访问过同一个利益冲突类中的其他公司数据集 j,则 $\boldsymbol{A}(s,j)=\text{true},(s,R_j)\subseteq UA$. 根据 RH 的定义,因为 g_i 和 g_j 属于同一个利益冲突类,所以 R_i 和 R_j 都受到 T 中同一角色的支配;而根据 UA 约束,一个用户不能同时拥有这样的两个角色. 所以用户 s 拥有角色R_j 时就不能同时把角色 R_i 授予自己.

如果主体 s 没有访问过同一个利益冲突类中的其他公司数据集 j,那么它可以把角色 R_i 授予自己.特别的,如果该主体还拥有一个角色 $W_l(l\neq i)$ 时,根据 UA 约束,它最多可同时再拥有角色 R_1 和 R_l,所以该主体必须先取消角色 W_l 把再把角色R_i 授予自己才能满足 UA 约束.

④ 如果第 $k+1$ 个请求是写某个公司数据集 i 的数据$(i\neq 1)$. 根据主体是否访问过任何其他的公司数据集(不包括清洁数据),BN 模型作出拒绝或者准许的访问决策. 如果主体只是访问过清洁数据,则 BN 模型准许.

如果主体 s 没有访问过任何其他公司数据集 $j(j\neq i)$,则 $\boldsymbol{A}(s,j)=\text{false},(s,R_j)\not\subset UA$. 并且注意到根据 UA 约束有$(s,W_j)\subseteq UA\Rightarrow(s,R_j)\subseteq UA$,所以这时必然有$(s,W_j)\not\subset UA$. 那么主体 s 可以把角色 $\{W_i,R_i\}$(如果在这以前主体 s 没有这两个角色)或者 W_i(如果在这以前主体 s 只有角色 R_i)授予自己且不违反 UA 约束.

如果主体 s 访问过某个公司数据集 j(不是清洁数据),则 $\boldsymbol{A}(s,j)=\text{true},(s,R_j)\subseteq UA$,那么它把角色 $\{W_i,R_i\}$或者 W_i 授予自己就会违反 UA 约束规定(当一个主体拥有 W_i,那么它不能拥有 $\{R_1,R_i\}$ 之外的角色 R_j).

如果主体 s 访问过的其他数据集只有清洁数据集 $j=1$,则 $\boldsymbol{A}(s,1)=\text{true},(s,R_1)\subseteq UA$,那么它可以把角色 $\{W_i,R_i\}$或者 W_i 授予自己(特别的,如果$(s,W_1)\subseteq UA$,就需要取消角色 W_1)并且不违反 UA 约束.

综述上面①~④共 4 种情况,构造 1 和 BN 模型的访问决策和状态都是一致的. 得证.