ISSN 1000-9825 Journal of Software 软件学报 2000,11(10):1320~1325

强制访问控制在基于角色的安全系统中的实现*

李立新 陈伟民 黄尚廉

(重庆大学智能结构中心 重庆 400044)

E mail: wmchen@cqu.edu.cn

摘要 讨论了在基于角色的安全系统中实现强制访问控制(mandatory access control,简称MAC)的问题.首先介绍了角色的基本概念及其在安全系统中的应用和MAC的基本概念,然后给出了一个利用角色机制实现强制访问控制的方法,通过将每个角色上下文处理为独立的安全级并施行非循环信息流要求,实现了强制访问控制.

关键词 安全级,信息流,强制访问控制,角色,基于角色的安全机制.

中图法分类号 TP309

基于角色的安全机制为管理大量的访问权限提供了一种灵活的、动态的方法,可以对角色灵活地配置和再配置,尤其适合于大型数据库系统[1,2].传统的基于角色的安全机制在对信息完整性的要求超过了保密性的环境中得到了应用,但这并不排除利用角色的安全机制的优点来实现保密性.本文给出了一种利用基于角色的安全机制实现保密性的方法.通过对读、写操作增加新的规则,能够实现MAC(mandatory access control)强制访问控制机制.

1 角色与基于角色的安全

1.1 角色的定义与基本概念

定义1(特权). 一个特权是指二元组(x,m),其中x是指一个客体或客体标识符,m是指一个客体x的非空的访问模式集.它是不可遗忘的,对它的修改必须经过授权.

一个特定的权限定义可被看成是从属于客体x与其访问模式m的计算.依赖于集合m,一个特权能导致某些变化(如客体状态的变化),泄露和增加系统信息.它能引起客体的创建或删除,创建新的特权和删除已有的特权、创建新的角色或删除已有的角色,等等.一个特权是以一个客体的访问模式来定义的,每一个这样的模式对相关客体都有特定的效果.必须确保在执行时角色的定义获得预期的效果.另一点需要指出的是,特权从属于简单或复杂的客体.在访问某些客体时,一个访问方式不需要访问与此客体相关的所有信息.因此,在考虑一个特定方式的访问时,我们感兴趣的是以这种访问方式所访问到的客体部分.执行一种特权的净效果是所有访问模式执行的积累.特权的特性是依赖于应用的.定义1在特定的应用中可以具体化.

定义2(角色). 一个角色是一个命名的特权集合,形式为二元组(*rname*,*rpset*),其中*rname*是角色名,*rpset*是特权集合,*r*.*rname*与*r*.*rpset*分别代表*r*的名字与*r*的特权集合.从计算的观点来看,一个角色指定了通过授权为此角色而可能的计算,引用所有此角色的计算结果等同于角色集合中各个特权所导致的计算结果的总和.将角色的总集合称为R.

---

* 本文研究得到重庆市资助科技项目基金(No.99-5517)资助.作者李立新,1967年生,博士生,助理研究员,主要研究领域为信息系统安全监测系统,建筑结构安全监测系统.陈伟民,1955年生,博士,教授,主要研究领域为光纤传感与网络,光纤通信.黄尚廉,1936年生,教授,博士生导师,中国工程院院士,主要研究领域为智能结构系统,光纤通信,传感技术.

本文通讯联系人:李立新,重庆 400044,重庆大学智能结构中心

本文2000-05-31收到原稿,2000-06-30收到修改稿

1.2 基于角色的安全

角色方便了对系统资源的访问,用户-角色授权是3种基于角色授权保护方法中的一种.在这种方式下,一个用户/组被授权通过一个角色访问已有特权,此授权必须在一个角色的访问控制列表中指定.除了在角色访问控制列表中所指定的访问之外,其他访问没有被允许的.故每个角色必须拥有一个相关联的访问控制列表.

定义3(角色访问控制列表). 一个角色访问控制列表 $racl$ 的形式如下:$\langle id_1,\ldots,id_n\rangle$,其中 $id_i\in ID$ 是一个用户($uid\in UID$)标识符或一个组标识符($gid\in GID$),其中 $ID=UID\cup GID$.

定义4(安全角色). 一个安全角色是一个形式为三元组($rname$,$rpset$,$racl$)的命名的特权集合,$rname$ 是角色名,$rpset$ 是它的特权集,而 $racl$ 是它的访问控制列表.它只有一个访问控制列表.

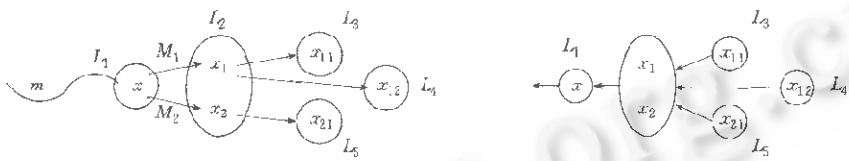
判断一个特定用户对某些客体是否能以某些访问方式进行授权访问是一个两阶段的过程.首先,必须保证一个用户有一角色授权,即用户或组的标识符在角色的访问控制列表上;其次,必须保证对客体所希望的访问方式在特权集中存在.后者可以称为用户-特权授权.即一个两阶段的过程来证实一个授权.主体被授权为一个角色,角色包含了访问客体所需要的相关的特权.后者称为通过一个角色的客体可访问性.在相关联的特权中指定的访问模式称为通过一个角色的合法访问模式.角色的执行权限是与角色的管理权限相分离的.对一个单一的主体不应同时拥有两类权限,这会导致冲突.

1.3 角色与角色信息上下文(context)

特权的定义保证特权的执行是从一个命名的来源获得输入(客体 x 与任何相关的访问模式参数),特权的执行导致了一种潜在的变换,导致某些信息可获得和信息在过程中流动.并且由于此执行从属于系统信息,它提供了一种访问系统信息的方式,故每一个特权都可以被看成是一个信息访问的子上下文.

定义5(特权信息子上下文). 特权信息子上下文是指通过一个特定的特权可以被访问的一段系统信息.

以O-O技术中的"方法"来构成特权定义的基础.设 x 是客体标识符,m 是 x 的有效"方法",而方法可以有不同于 x 的有效对象的参数,其执行可能因引用其他方法而创建新的客体(对象).一个方法引用系列可以用一棵树来代表.在图1中,m_i 代表方法引用,L_i 是相关的系统定义的安全级别,x_i 代表了与一个方法引用相关的客体(对象),箭头代表跨安全级别/客体的引用.



(a) Method invocation sequence
(a) 方法引用序列

(b) Corresponding information flow
(b) 相应的信息流

Fig. 1
图1

这种引用使得访问不同安全级别的信息更加方便.通过一个引用从一个安全级上可以访问的信息不会超过此安全级上包含的信息.一个引用可以产生对超过一个以上安全级的访问和在每个安全级别上获得的一段信息.一个引用子上下文是与此引用有关的各个安全级上涉及的信息的聚集.在本例中,在安全级别 $c_1,c_2,\ldots,c_5$ 上,信息片断的聚集构成了 m 的子上下文.这些信息片断依次与此引用所涉及的客体相关联,所以,m 的子上下文将会包含从属于 x_i 的信息片断.这种方法可以看成是从属于此引用相关客体的系统信息的一个小窗口,本文中把通过这个小窗口可获得的信息叫做一个特定特权的子上下文,相关的信息流如图1(b)所示.

角色充当了系统信息的一个窗口,通过一个角色获得的信息由角色的特权集决定.它至少是通过特权集上的单独权限可获得信息的总和.

给定某些角色 $r\in R$,其特权集为 $r.rpset$,设 $INF(t)$ 代表通过某些角色或特权集而获得的信息量,其中 pv 在 r 的角色特权集中,则

$$INF(pv)\leqslant INF(r).$$

根据"聚集"原则[5],一个全体的信息大于等于各独立部分信息的总和.在一个特权中的信息的"量"不可能

超过它所关联的角色.而所有在一个角色特权中信息的总和总是小于或等于通过角色可获得的信息.即

$$INF(r)\geqslant\bigcup INF(pv),$$

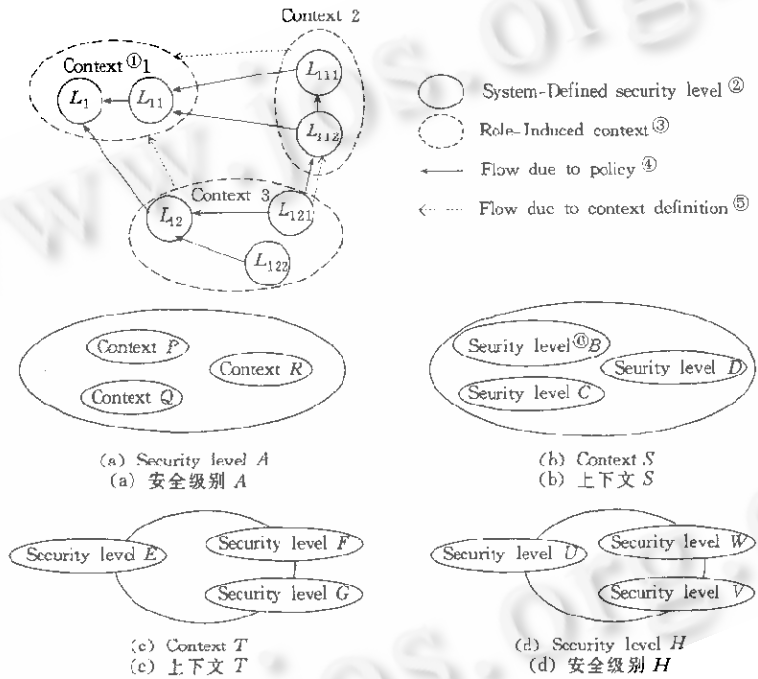
其中 $pv\in r.rpset$.

一个角色 r 的信息窗口定义为 $INF(r)$.

故一个基于角色的系统可看成是把系统信息分区,通过以角色定义的窗口获得各区的信息.通过此窗口可得的信息构成一个角色相关的上下文,通过将某个角色授权给用户,每个这样的上下文对用户是可获的.

定义 6(角色信息上下文). 它是指通过角色可获得的系统信息.角色信息上下文与系统定义的安全级的区别是。前者与一个特定的角色有关,后者是一个系统定义的范围,信息可在其中自由流动.

这些系统定义的安全级不需要与角色上下文相一致.这些子上下文可能分布在多个系统定义的安全级,也可能多个安全级属于同一个上下文,更普通的情况是数个上下文属于同一个系统安全级别,如图 2 所示.



(a) Security level A
(a) 安全级别 A

(b) Context S
(b) 上下文 S

(c) Context T
(c) 上下文 T

(d) Security level H
(d) 安全级别 H

①上下文,②系统定义的安全级别,③角色引起的上下文,④策略导致的流,⑤上下文定义导致的流,⑥安全级别.

Fig. 2
图 2

由一个角色方案所导致的上下文之间可能有不同的关系.一个子上下文(context)可能是另一个子上下文的子集,或与另一个相等(当它们与具有相同特权集的角色相联系时).

1.4 角色与信息流

本节给出一个基于角色系统的信息流分析[4].首先讨论信息流问题:给定一个程序和它的输入、输出变量集合,对于每个输出变量,判断/确定输入变量的子集,此子集可能包含程序执行之后的信息.

可以认为,信息是从输入子集流向包含了输入子集信息的输出子集.

公理 1(基本信息流公理). 一个流 $x\rightarrow y$ 仅当 y 的值被修改时才发生.

公理 2(信息流传递性公理). 信息流具有传递性,即 $x\rightarrow y\wedge y\rightarrow z\Rightarrow x\rightarrow z$.

在一个系统中有两种类型的操作:读操作,不改变相关客体/对象的状态;修改操作,改变/修改/写相关客体/对象的状态.一个包含读和修改的操作导致信息从所读的输入流向所修改的输出.

而在一个角色上下文内的修改操作则可被看成是定义了此上下文的修改范围,与这些上下文相关的信息通过这些操作被修改.而操作的修改效果可以在其他安全级别上感觉到,信息能从一个上下文流向另一个上下文.角色读的范围也可类似地规定,对一个角色 r 的读写操作有下式成立(以 r_scope 和 u_scope 分别表示其读、写范围):

(1) 对所有的 $r\in R$,$r\_scope(r)\rightarrow u\_scope(r)$,即在一个角色内有信息流,即 $r\rightarrow r$.

(2) 对所有的 $r\in R$,$u\_scope(r)\rightarrow u\_scope(r)$.

(3) 对于 $r_i,r_j\in R$,$r\_scope(r)\not\rightarrow r\_scope(r_j)$,即在两个不同角色的读范围之间无信息流(除(4)之外).

(4) 在不同角色的读写范围有交叉时,在两个相关上下文间有信息流发生.

设有两个角色 r_i 和 r_j,具有如下的范围:

$$r\_scope(r_i)=\{x,y,z\},u\_scope(r_i)=\{p,q\};r\_scope(r_j)=\{a,b,c\},u\_scope(r_j)=\{d,e,f\}.$$

由(1)和(2)有 $\{x,y,z\}\rightarrow\{p,q\}$ 和 $\{a,b,c\}\rightarrow\{d,e,f\}$.

从 r_i 到 r_j 的信息流发生,即 $r_i\rightarrow r_j$ 发生,当且仅当通过 r_j 可以访问与 r_i 相关的修改,即或者 p 或者 q 或者 p 和 q 都属于 r_j 的范围.

换句话说,如果有 $r_i\rightarrow r_j$,那么,或者 $u\_scope(r_i)\cap r\_scope(r_j)\neq\varnothing$,或者 $u\_scope(r_i)\cap u\_scope(r_j)\neq\varnothing$.

多向信息流当 $r_i\rightarrow r_j$ 与 $r_j\rightarrow r_i$ 同时成立时发生,记作 $r_i\leftrightarrow r_j$.

单向的信息流可以是 $r_i\rightarrow r_j$ 或者 $r_j\rightarrow r_i$,如果要求 $r_i\rightarrow r_j$,则有 $u\_scope(r_i)\cap r\_scope(r_j)\neq\varnothing$ 或 $u\_scope(r_i)\cap u\_scope(r_j)\neq\varnothing$,$u\_scope(r_j)\cap r\_scope(r_i)\neq\varnothing$ 或 $u\_scope(r_j)\cap u\_scope(r_i)\neq\varnothing$.

定义 7(包含信息流). 所有来自于一个上下文的信息流也流进一个上下文.假设 $r_i\rightarrow r_j$ 是这样一个流,则其条件是 $u\_scope(r_i)\subseteq u\_scope(r_j)$ 或 $r\_scope(r_i)\subseteq u\_scope(r_j)$.

2 角色与强制访问控制

本节对 MAC 进行回顾,并提出使用角色实现 MAC 的方法.

2.1 MAC 基础

强制存取控制(MAC)[5]是一种安全策略,是指通过无法回避的存取限制来阻止直接和间接非法入侵.系统为主体和客体分配了不同的安全标识,代表用户的应用程序不能改变自身或任何客体的安全标识,在多级安全系统中,主体与客体的安全标识是安全证书和敏感级.它们是主体和客体的属性,访问基于它们进行.主体对客体的访问基于两个规则:简单安全特性和 * 特性.通过引用监督器进行访问管理,所有的访问必须通过它.

2.2 在基于角色的安全中实现 MAC

MAC 机制能够阻止特洛伊木马的攻击,而阻止特洛伊木马的策略是基于非循环信息流的,故在一个级别上读信息的主体一定不能在另一个违反非循环规则的安全级别上写.同样,在一个安全级别上写信息的主体也一定不能在另一个违反非循环规则的安全级别上读.由于在 MAC 中在不同安全级别上的信息流必须是非循环的,主体都给予一个安全证书而客体被指定敏感级别,访问由两个关键的规则控制:不向上读和不向下写.结合这些和引用监督器的要求,在此提出一种基于角色系统中实现 MAC 的方法,将每个角色上下文作为一个安全级别,然后在构造角色的方式上增加非循环信息流要求,并增加了 MAC 中规则等价的规则.

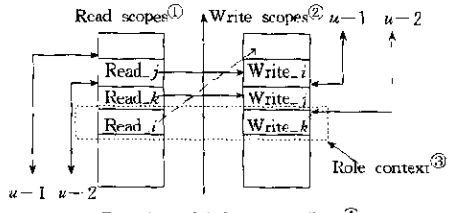
约束 1(信息一致性约束). 系统对于某些安全策略是安全的,只要它的系统实现的信息流与安全策略定义规定的信息流是一致的.

为了保证保密性,基于角色的方法中的信息流图必须是非循环的.而在有循环之处,循环中所有角色/范围必须处于同一个上下文中.为了限制特洛伊木马攻击,必须构造一个 * 特性的等价规则以控制授权:

约束 2(强制授权约束). 一个主体只能以一个已获授权的角色并以此角色中相关联的特权规定的方式访问一个客体.

授权为某个角色是在角色的 ACL 中规定的.并且在一个安全系统中,所有的角色都是安全角色(见定义 4).用户授权为某个角色意味着用户能够以角色访问客体,其访问方式是合法的.为了简化和避免冲突,任何

用户在一个时刻只能充当一个角色,这样就只需保证用户授权为此角色,而不必关心是否与非循环信息流要求相矛盾.在传统的MAC中,用户不能同时以不止一个安全证书登录到系统中.为了执行被指定的不同角色,一个用户必须在需要改变角色时退出登录.



①读范围,②写范围,③角色上下文,④信息流方向.

Fig. 3
图3

这也意味着一个用户的授权的角色集能够被划分为不矛盾的角色组,即在任何一个组内,一个授权用户能够自由执行任何角色而不违反非循环信息流.在登录进入系统时,用户将被强制获得这些角色组之一的执行权.必须保证用户在任何时刻只执行一组权限.如果希望执行另一个组中的一个角色,用户必须重新登录以获得所希望的执行权.

用户授权本身并不足以保证信息的保密性和完整性.必须确保没有此类授权会引起非法信息流.下面的约束,对于保证在角色中施行类似于MAC的保证是必须的,与BLP模型的不向上读和不向下写的规则有同样的效果.图3给出了一个例子.

为了确保保密性不能因为域的交叉而违反,有以下约束:

约束3(读访问约束).设有两个用户u_1,u_2以及两个角色r_i和r_j,设u_1可访问两个角色的读范围,u_2只能访问r_i的读范围.则有$r_scope(r_j)$必须是$r_scope(r_i)$的子集,即$r_scope(r_j)\subseteq r_scope(r_i)$.

回想一下从一个角色的读范围到它的写/修改范围的信息流,即

$$r\_scope(r_i)\rightarrow u\_scope(r_i)\text{和}r\_scope(r_j)\rightarrow u\_scope(r_j).$$

假设$r_scope(r_i)\not\subseteq r_scope(r_i)$,这意味着在$r_scope(r_j)$中存在不属于$r_scope(r_i)$的信息.但是,如果$u_1$授权访问两个范围:$r_scope(r_i)$和$r_scope(r_j)$,则有$r_scope(r_j)\rightarrow u_scope(r_i)$.

所以,如果$r_scope(r_j)\not\subseteq r_scope(r_i)$,则在$r_scope(r_i)$中的信息就不能保证全部流向$u_scope(r_i)$.

在指定合法信息流时和进行用户授权时,必须保证通过不同角色进行的读/写操作不违反规定的流策略,即使特洛伊木马把信息泄露给提供非授权的上下文中成为不可能.下面两个约束的目的是防止特洛伊木马的攻击.

约束4(修改访问约束).一个主体不能访问一个角色的读范围和修改/写另一个角色的写范围,如果没有从前者到后者的合法信息流.

约束4的目的是确保一条信息流按修改的方向定义,这是根据信息流公理1定义的.

约束5(读/修改约束).一个主体能够访问一个角色的读范围和修改另一个角色的写/修改范围,当且仅当第2个角色的读范围包含第1个角色的读范围.即给定两个角色r_i和r_j,主体能够以角色r_i修改其他在r_j中主体能够读的信息,当且仅当有一个定义的合法信息流(直接或非直接),从由r_i的上下文规定的信息到r_j的上下文规定的信息.

至此可得出结论:如果角色定义和用户角色授权遵守约束1~5,就可以实现一个MAC一类的保护.这些约束保证了一个特定具体策略的正确实现,通过被授权的角色,这个策略控制了用户-角色授权以及对信息流访问的控制.

3 总　结

本文中给出了用基于角色的保护来实现一个模拟强制访问控制的方法.由于在许多应用场所,对MAC中信息的完整性和保密性需要同样关心,在MAC中信息流必须是非循环的.而在MAC中以主体与客体的属性作为基本批准授权的基础,并且这种授权必须遵守引用监督器的规则.相应地,为了实现基于角色保护的MAC,把每个角色上下文看成一个安全标识,并确保信息流(由角色执行或用户-角色授权)是非循环的.同时给出了5个约束以实现BLP模型的不向上读和不向下写规则的等价物.

参考文献

1 Sanhu R S, Coyne E J, Feinstein H L *et al*. Role-Based access models. IEEE Computer, 1996,29(2):38~47

2 Nyanchama M, Osborn S L. Access rights administration in role-based security system. In: Biskup J, Morgenstern M eds. Database Security VIII: Status & Prospects, Proceedings of the 8th Annual IFIP TC11 Working Conference on Database Security. North-Holland: IFIP, 1994. 35~76

3 Lunt T F. Research Directions in Database Security. New York: Springer-Verlag, 1992. 98~99

4 Nyanchama M, Osborn S L. Information flow analysis in role-based security system. Journal of Computing and Information, 1994,1(1):1368~1384

5 Trusted Computer System Evaluation Criteria. DoD 5200. 28-STD, Washington DC: Department of Defense, 1985

Realizing Mandatory Access Control in Role-Based Security System

LI Li-xin CHEN Wei-min HUANG Shang-lian

(*Smart Structure Center Chongqing University Chongqing* 400044)

Abstract The realization of MAC (mandatory access control) in role-based protection system is discussed. First, the definition of role and the application in security are discussed. Then the concept of MAC is introduced and a scheme of role-based protection which realizes MAC is developed, by viewing each of the role contexts as an independent security-level and imposing non-cyclic information flow requirement.

Key words Security level, information flow, MAC (mandatory access control), role, role-based security mechanism.