基于角色的访问控制

Role-based Access Control

尹 刚 王怀民 滕 猛

(国防科技大学计算机学院 长沙410073)

Abstract As the complexity of enterpriese applications increasing continuously, how to enforce flexible and multiple security policies, reliable and efficient policy management in these application systems is the problem that must be resolved. This paper firstly introduces the basic conception and features, and then describes the typical RBAC policies using precise language.

Keywords RBAC, Policy, Constraints, Operation, Duty, Task

1. 引言

基于角色的访问控制(Role-based Access Control,下简称RBAC),其核心概念是用户不能对应用对象随意访问。RBAC系统中,用户作为某种角色的成员,其访问权限在管理上与其角色相关。这种思想大大简化了用户授权的管理,为定义和实施系统安全策略提供极大的弹性。用户可根据权力和资格被赋予不同的角色,并且用户角色易于重新分配,无须改变基本访问结构。如果有其它应用程序或操作加入,可以往角色中添加或删除权限。目前RBAC机制已广泛应用于各种系统中,包括:Oracle, NetWare, Java, DG/UX, object-oriented systems, databases, MS Windows NT, enterprise security management systems 等等。

2. RBAC的基本概念

RBAC中的安全策略以用户、主体、角色、角色层次、操作和对象来描述。要执行RBAC安全框架中某对象的操作,用户必须具有相应角色。在用户以某角色发出动作之前,安全管理员必须首先使其成为该角色的一个成员。

RBAC允许管理员在角色分配、角色激活和操作执行等方面实施各种限制(限制是RBAC中安全策略重要表现形式,本文以规则或假设形式给出)。这些策略有多种形式,包括角色成员集的基数限制,角色互斥性限制,以及在为角色添加权限或者执行对象操作时实施的时间和位置限制。

下面几节给出RBAC特性的定义并对几种典型的安全限制给出准确描述。

2.1 用户、角色、操作

在RBAC框架里,用户指人,角色指工作职责的集合,操作指访问一个或多个RBAC对象的方式。主体表示一个活跃的用户进程,用户与主体之间的关系为一对多。下述函数用来描述用户、主体和角色间的映射关系:

主体用户集(s:主体)={与主体s关联的用户}

主体角色集(s:主体)={与主体s关联的角色}

角色成员集(r:角色)={被授予角色r的用户}

用户角色集(u:用户)={与用户u关联的角色}

注:主体用户集的基始终为1。

与主体关联的用户由唯一的用户ID决定。每个主体仅与一个用户关联,但可以同许多角色关联。RBAC还要求如果主体角色集(s)=R并且主体用户集(s)=u,那么u必须同角色集R关联。下面是更准确的描述:

假设1(主体一致性假设) ∀s:主体,u:用户,R,r:角色:

(主体用户集(s)=u)∧(主体角色集(s)=R)∧(u∈角色成员集(r))⇒r∈R

例如,某用户u通过主体s访问对象的操作。u仅同"业务员"角色关联,而s却以"经理"的角色执行某些操作(这时R={"业务员"},而r="经理",r∉R),这就违背了主体一致性假设,显然这种情况是很危险的。

对象及其操作的类型依赖于其实现所处的系统类型。例如,在操作系统中,操作可能包括读、写和执行;在数据库管理系统中,操作可能包括插入、删除、追加和更新;在事务管理系统中,操作可能表现出所有事务的属性。在面向对象领域,操作可以是对象的方法。RBAC系统中的对象包括所有RBAC操作可以访问到的对象。但并非所有文件系统和系统对象都需要包含进RBAC方案中。例如,同步对象(如信号量、管道、消息)和临时对象(如临时文件和目录)就无须在RBAC系统里作为受保护对象。

操作表示RBAC框架里由角色使用、体现各种限制的控制条目。正确认识简单的访问模式(如读、写、修改等)和操作之间的不同很重要。操作能够体现出各种RBAC限制和安全相关细节(这些细节将体现出访问粒度),而简单的访问模式却做不到。

为了说明RBAC中操作的重要性,我们考虑一下银行出纳员和核算经理在访问需求方面的差别。企业定义出纳员角色可以执行存款业务。这要求在一个文件特定字段中执行读和写操作。企业也可以定义一个核算经理角色,可以执行校正操作。这些操作要求在该文件相同字段(即出纳员操作的字段)中执行读和写操作。但核算经理可能不被允许新建或注销帐号,只是在这些操作执行后进行校正。同样,出纳员在业务完成后不能执行任何校正操作。两角色的区别是执行了不同操作以及在业务日志中的不同记录(但他们的访问模式相同)。

为了说明控制粒度的重要性,考虑一下药剂师的例子。药

尹 刚 硕士生,主要从事分布计算及网络安全方面的研究。王怀民 教授,博士生导师,研究方向为分布计算环境及网络安全。滕 猛 博士生,主要从事分布计算及安全机制的研究。

剂师需要访问病例以检查药物疗效，并在病例中添加新的药物名称。尽管该操作是必需的，但药剂师却不能读写病例的其它字段。

如图1所示，操作与对象相关联。

图1 操作与对象

把用户加入角色中时，意味着该用户有权执行该角色管理的所有操作。每个操作有唯一标识与其对应。操作同角色、关系和对象间的关系由下述函数描述：

角色操作集(r:角色)={与角色r关联的操作}

操作对象集(op:操作)={与操作op关联的对象}

2.2 角色和角色层次

角色可能有重复的职责和权限，即属于不同角色的用户有时需要执行公共的操作。把这些操作添加到每个角色中是繁琐易错的工作。为了给企业提供自然的层次结构，RBAC 提出角色层次的概念。角色层次定义了包含特有属性和其它角色的角色。角色层次是一种能够反映职权、责任和能力层次关系的管理角色的方法。图2是一个角色层次的例子。其中“专门医师”角色包含“医生”角色和“实习医师”角色。这意味着“专门医师”角色成员集将包含“医生”和“实习医师”的操作、限制和对象，而无须管理员逐一列出“医生”和“实习医师”的属性。权限最大的角色位于图的最上层。图2还表明并非所有角色间都有关联。角色“心脏病专家”和“风湿病专家”在层次上无关联但却可能包含相同的角色。

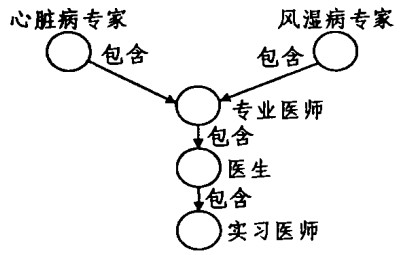

图2 角色层次实例

角色层次可以用继承关系描述。直接的父子关系可用有序对$((R_{i+1},R_i),>)$表示，其中R_{i+1}是父亲，R_i是孩子，连接词“>”表示“包含”关系。因此$R_{i+1}>R_i$表示R_{i+1}包含R_i。如图2，角色“专业医师”是“医生”的父亲，角色“实习医师”是“专业医师”的祖先(即“专业医师”包含“实习医师”)。所以把一个角色授予某用户意味着该用户具有这个角色“包含”的所有角色。下面是准确的描述：

规则1(角色继承规则) $\forall$ s:主体;r_i,r_j:角色：

$(r_j\in$主体角色集(s))$\wedge(r_j>r_i)\Rightarrow(r_i\in$主体角色集(s))

3. RBAC 的典型策略

3.1 角色授权

用户与角色的关联具有下述特性：1. 用户不能授予超过其工作范围的权限；2. 授予用户的角色同其用户角色集不冲突；3. 任何角色，其角色成员集的基有上限。

第一条特性确保最小权限原理。最小权限原理要求用户的权限不能超过其职权范围。确保最小权限要求对用户的职权有明确的认识，定义完成工作所需的最小权限族，把用户限制在该权限族对应的域里。

上述第二条特性是为了体现职权分离和防止内部冲突等安全策略。这意味着不能同时具有两个或两个以上互斥的角色。例如，银行系统中，被授予“出纳员”角色的用户不能具有“审计员”角色。因为“出纳员”和“审计员”这两个角色是互斥的。

职权分离策略可以集中描述，然后在特殊角色上实施。角色的互斥角色和职权分离特性可由下式描述：

角色互斥集(r:角色)={与角色r互斥的所有角色}

只有当某角色同用户的用户角色集没有互斥时，才能将其授予该用户：

规则2(静态职权分离规则) $\forall$ u;用户;r_i,r_j:$i\neq j$:

$(u\in$角色成员集$(r_i)\wedge u\in$角色成员集$(r_j)\Rightarrow r_i\notin$角色互斥集$(r_j))$

上述第三个特性在为用户授予角色时起作用，它是一种基数特性。某特定时间段里，一些角色只能授予一定数目的职员，例如“经理”角色。尽管其它职员可以使用该角色，但任意时刻只能有一个职员行使经理的权力。只有当角色的角色成员集数目没有超过上限时，用户才能添加到该角色的成员集中。角色成员集上限和角色成员集的基由下述函数描述：

角色成员集上限(r:角色)=角色成员集的基的上限($\geq$0)

角色成员集的基(r:角色)=N($\geq$0,角色r成员集的基)

对角色成员集的基的限制可描述为：

规则3(基的限制) $\forall$ r:角色：

角色成员集上限(r)$\geq$角色成员集的基(r)

3.2 角色激活

主体是用户一个或多个角色的映射。用户在会话期间以其相关联的角色代表其身份。为用户授予角色(使用户持有一个角色的序列)是用户执行操作的必要但非充分条件。为此，还需考虑其它组织内部策略或安全限制，决定用户是否有权执行操作。

角色激活为策略提供环境上下文。RBAC 要求用户在用户执行某动作前首先被授权(激活)。

根据组织内部策略检查待激活的角色、将要执行的操作，以及将被访问的对象。如果下列条件满足，角色将被激活：1. 待激活角色在用户角色集中；2. 待激活角色不与用户其它角色互斥；3. 将被执行的操作在待激活角色的角色操作集中。

下述函数关系对主体能否执行 RBAC 操作，以及主体的活跃角色集给出定义：

exec(s:主体,op:操作)=TRUE

iff 主体s能执行操作op;

exec(s:主体,op:操作)=FALSE

iff 主体s不能执行操作op。

活跃角色集(s:主体)={主体s当前活跃角色集合}

主体的待激活角色必须在该主体的主体角色集中，下述规则将作出更准确的描述：

规则4(角色授权规则) $\forall$ s:主体

主体活跃角色集(s)$\subseteq$主体角色集(s)

如果某角色属于主体的主体角色集，且主体可以执行操作，则表明该角色是活跃的。尽管待激活角色在角色集中，仍然会有组织内部策略(如下面将要描述的动态职权分离策略)禁止其被激活。下述规则将作出描述：

规则5(角色执行规则) $\forall$ s:主体,op:操作：

exec(s,op)⇒主体活跃角色集(s)≠∅

RBAC 允许管理员实施动态职权分离策略。前述静态职权分离策略可为企业消除角色分配时存在的潜在冲突。但有些组织允许用户角色集里存在互斥角色(这些角色分别单独使用),这就要求引入新策略实施动态管理。例如,静态策略可能要求具有"采购员"角色的个人不能同时具有"会计"角色。尽管该策略足以描述某些组织的权限要求,但对其它组织可能过于严格。动态职权分离策略的优点是允许操作具有更多的灵活性。动态职权分离策略在角色并发激活时施加限制。如某用户可被授予"采购员"和"会计"两种角色,但在某一时刻只能使用其中一个。待激活角色的互斥角色集由下述函数描述:

互斥活跃角色集(r:角色)={同待激活角色 r 互斥的活跃角色集合}

如果主体的待激活角色不与该主体当前任何活跃角色互斥,则该待激活角色可以成为主体的活跃角色。RBAC 动态职权分离规则定义为:

规则6(动态职权分离规则) ∀ s:主体,r_i,r_j:角色:i≠j:

r_i∈活跃角色集(s)∧r_j∈活跃角色集(s)⇒r_i∉互斥活跃角色集(r_j)

仅当操作已被授予主体的待激活角色时,主体才能执行该操作。有如下规则:

规则7(操作授权规则) ∀ s:主体,op:操作,r:角色:

exec(s,op)⇒r∈活跃角色集(s)∧op∈角色操作集(r)

3.3 任务的操作分割

RBAC 支持系统管理员实施任务操作分割策略。操作分割策略能有效防止欺诈行为。因为在关键业务活动(任务)中往往存在相关职务的协作问题,可能出现欺诈行为。例如,购买商品可能包括下述操作:批准定购单;记录支票是否到达;记录商品是否到达;最后批准付款。如果每个操作由不同角色执行,欺诈的可能性就很小。如果允许一个用户执行所有操作就可能发生欺诈行为。

任务操作分割策略要求单个用户不能执行与特定任务相关的所有操作,否则会被组织发现。RBAC 中任务操作分割策略在角色被授予单个用户或操作被赋予角色时起作用。任务操作分割策略可由下述函数定义:

任务操作集(t:任务)={任务 t 包含的所有操作}

角色与操作关联,仅当该角色属于主体的角色集,且先前没有被赋予任务的任何其它操作:

规则8(任务的操作分割规则) ∀ u:用户,r:角色,t:任务:

¬(任务操作集(t)⊆∪角色操作集(r))

其中:r∈用户角色集(u)

3.4 对象访问

为了在 RBAC 对象中实施企业内部策略,主体对 RBAC 对象的访问必须受到控制。下述函数用来决定当主体是否能访问 RBAC 对象:

access (s:主体,o:对象)=TRUE

iff 主体能访问对象 o

access (s:主体,o:对象)=FALSE

iff 主体不能访问对象 o

根据上述角色授权和角色执行特性(规则4和5),下面定义的对象访问授权特性规定,主体必须根据其活跃角色的操作访问 RBAC 对象,即主体能访问某对象的操作,仅当该主体的角色属于其当前活跃角色集,且其角色能执行该对象的操作,且该对象属于操作的对象集:

规则9(对象访问规则) ∀ s:主体,o:对象:

access(s,o)⇒∃ r:角色,op:操作:

(r∈活跃角色集(s)∧op∈操作集(r)∧o∈对象集(op))

结束语 RBAC 的主要目的是描述并实施企业内部的安全策略,减小典型安全策略管理的复杂性。如上所述,RBAC 是一个具有丰富策略机制的框架,其配置与具体组织的策略相关。这就允许 RBAC 能够适应任何组织结构和业务手段,并且能够随着企业和组织的结构、安全需求的改变而更新。同时,RBAC 极大提高了安全管理员的工作效率,减少了管理疏漏。

参 考 文 献

1 Oralce Corporation. ORACLE7 Server SQL Language Reference Manual. Dec. 1992
2 Sandhu R S,et al. Role-based Access Control Models,unpublished journal article
3 Department of Defense. Trusted Computer Security Evaluation Criteria. DoD 5200. 28-STD,1985
4 National Computer Security Center. A Guide to Understanding Discretionary Access Control in Trusted Systems. NCSC, Sep. 1987
5 Ferraiolo D F,et al. Role-based Access Control(RBAC):Features and Motivations

---

(上接第87页)

在 Mogent 平台上开发了一个原型系统,证实了使用 XML 描述市场空间 Ontology 库带来的好处:实现了市场空间中信息的理解、应用的独立开发和业务处理的集成;并保证了市场空间的开放性、分布性等。

目前 XML 在电子商务的应用还处于发展阶段。当各类领域知识的 XML 描述统一和完善后,将这些标准纳入市场空间 Ontology 库,将全面实现 MABEMS 的开放性。同时,使用 XML 描述 Ontology 库仍然有大量的工作要做,如信息的存储冗余、对 XML 文档进行基于语义的查询、XML 文档集的知识推导和自动产生等,都处于启步阶段。随着 MABEMS 市场空间 Ontology 库中知识的不断积累,Ontology 库的管理将成为十分突出的问题,这些将是我们进一步研究和探索的方向。

参 考 文 献

1 Fingar P, Kumar H, Sharma T. 21st Century Markets: From Places to Spaces,First Monday,1999,4(12)
2 CommerceOne Corp, MarketSite Portal Solution 3. 0, The Most Comprehensive B2B Portal Framework For,Driving Successful E-Commrece
3 Microsoft, Microsoft ® Biztalk™ Server 2000 White Papers. http://www. biztalk. org,2000/3/30
4 苏敏,郭瑞景,陶先平,吕建. 软件 agent 在电子商务中的应用初探. 计算机科学,2001,28(11)
5 陶先平,冯新宇,李新,张冠群,吕建. Mogent 系统的通讯机制. 软件学报,2000,11(8)
6 Amann B, Fundulaki I. Integrating Ontologies and Thesauri to build RDF Schemas. In Research and Advanced Technologies for Digital Libraries, Lecture Notes in Computer Science, Paris, France, Third European Conference ECDL'99, Springer-Verlag, 1999,234～253
7 http://www. ontology. org
8 Dublin Core Metadata Initiative. http://dublincore. org
9 Wide Web Consortium. http://www. w3c. org/XML