多自治域协同的数据库访问控制

葛丽娜,贺忠华
GE Li-na,HE Zhong-hua

广西民族大学 计算机科学与信息学院,南宁 530006
School of Computer Science & Information,Guangxi University for Nationality,Nanning 530006,China
E-mail:gelina100@gmail.com

GE Li-na,HE Zhong-hua.Access control scheme for database in multi-domain autonomous collaborative environment. Computer Engineering and Applications,2007,43(16):180-182.

Abstract: In multi-domain autonomous collaborative environment,the resource providers of database have the ultimate authority over their resources to decide how to access their database resource and what granularity is,and the security administrators of domain user located manage user roles.Apparently,directly mapping user role to permission of other autonomous domains is not reasonable in multi-domain collaborative environment.In this paper,four lay RBAC(4-RBAC) model and its implement for database access is proposed in multi-domain autonomous collaborative environment.Mapping resource role to permission and defining resource role are done by the resource provider in the domain resource located.Then domain user located has to negotiate with domain resource located to map user role to resource role.The scheme in this paper is simple and reasonable.It is separation of duty and guarantee for database's security.

Key words: multi-domain autonomous collaboration;database;RBAC;resource role

摘 要:多自治域的协同工作领域中,资源拥有者应该对数据库资源的操作方式及粒度有最终决定权;用户域对用户的职能作明确规定。显然角色直接映射权限的RBAC模型在多自治域协作环境中是不合理的。针对多自治域协同的数据库访问,提出基于角色的四层访问控制模型及其设计,资源域定义资源角色与权限的映射,用户域与资源域协商来映射用户角色与资源角色。该方案简单合理,分清职责,符合数据访问安全需求。

关键词:多自治域协同;数据库;基于角色访问控制;资源角色
文章编号:1002-8331(2007)16-0180-03 **文献标识码**:A **中图分类号**:TP311

1 引言

在许多协同工作领域中,如智能交通、合作科研、远程医疗、军方任务合作等,组成一个基于网络的虚拟协作环境,必须分布式地、安全地存取虚拟协作的数据库。这就必须解决异域的数据库访问控制策略问题。基于角色的访问控制集中自主访问控制和强制访问控制的优点,是目前数据访问控制策略的首选。本文认为当用户与数据库资源处于不同的自治域中,由资源提供者决定他的数据库的访问方式和粒度,并在用户每次的访问中实现动态访问控制。提出多自治域协同环境下,具有用户、用户角色、资源角色和权限4个层次的基于角色的数据库数据访问控制,定义了用户静态的预权限,并在用户每次的访问请求中实现动态细粒度访问控制。

2 访问控制的相关方法

访问控制就是针对越权使用资源的防御措施。从策略上说,常用的访问控制可以分为强制访问控制、自主访问控制、基于角色的访问控制和基于任务的访问控制等。

(1)传统的访问控制,一是自主访问控制(DAC)[1],UNIX安全模型类似于DAC模型。优点是具有相当的灵活性,但安全级别较低、难管理;二是强制访问控制(MAC)[1],它管理集中,适宜于对安全性要求较高的应用环境,不适用于主体或客体经常更新的应用环境。

(2)基于角色的访问控制(RBAC)模型,角色是实现访问控制的基本语义实体,它综合了MAC与DAC的特点,用户不能进行自主授权和权限转移,引入了一种抽象的中介元素"角色"来传递信息,不再直接将用户与权限捆绑。RBAC模型使用严密的数学语言描述了复杂的用户访问控制的规则;提供对安全管理策略的支持,比如说任务完整性说明、管理的最小权限、冲突检测规则等等。1992年David Ferraiolo等人率先提出基于角色的访问控制模型框架[2],Ravi Sandhu等人于1996年提出著名的RBAC96模型[3],Ferraiolo和Sandhu等于2001年发表美国国家标准化和技术委员会(NIST)的RBAC推荐标准[4]。随着

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.60461001);广西民族大学重点科研基金项目(No.03ZSY00013)。

作者简介:葛丽娜(1970-),女,副教授,主要研究方向:网络与信息安全;贺忠华(1970-),女,讲师,主要研究方向:数据库与Web服务。

XML 技术、多域交互、Web services 以及网格服务的发展,出现处理基于 XML 的 X-RBAC[5];文献[6]提出基于多自治域安全访问的 RBAC 策略集成框架,通过域间的角色映射与角色集成实现跨域的访问控制,但由于产生易角色回路而使管理变得复杂。

(3)基于任务的访问控制(TBAC),TBAC[7,8]安全模型是从应用和企业层角度来解决安全问题。它采用"面向任务"的观点,从任务角度来建立安全模型和实现安全机制。

3 多自治域协同数据库访问控制

本文设计的 4 层 RBAC 模型(4-RBAC)兼顾多自治域协同的环境下数据库访问拥有的自治性和安全性的特点。

3.1 4 层结构模型概要

4-RBAC 有 4 个主要:用户、用户角色、资源角色和对数据库资源的访问权限,关系如图 1。

图 1 多自治域的数据库访问控制 4 层模型

4-RBAC 与传统的 RBAC 相比的优势:

(1)权限分配更合理,用户域管理用户角色,不能直接分配异域数据库资源的权限;资源域管理资源角色,资源角色直接与权限关联,方便管理,加强授权的控制力度;

(2)权限的局部性,资源提供者对自己的资源有决定权,实现自治域间的松散耦合;

(3)适应于分布式多自治域的环境;

(4)引入资源角色,强化了职责分离。

3.2 模型的形式化定义

定义 1 4-RBAC 模型的组成:

(1)由用户、用户角色、资源角色和权限(或称许可,含操作与操作对象)构成,即记为 4-RBAC=$\{U,UR,SR,P\}$,U 表示用户集合,UR 表示用户角色集合,SR 表示资源角色集合,P 表示权限集合;

(2)用户分配关系,用户到用户角色的 $N:M$ 映射 UUR,$UUR\subseteq U\times UR$,用户角色 ur 所授予的用户 $user(ur)=\{u|u\in U\wedge(u,ur)\in UUR\}$,用户 u 的用户角色 $u\_userole(u)=\{ur|(u,ur)\in UUR,ur\in UR\}$;

(3)用户角色分配关系,用户角色到资源角色的 $N:M$ 映射 $URSR$,$URSR\subseteq UR\times SR$,资源角色 sr 授予的用户角色集 $userrole(sr)=\{ur|ur\in UR\wedge(ur,sr)\in URSR\}$;用户角色 ur 授予的资源角色集 $ur\_resourverole(ur)=\{sr|sr\in SR,(ur,sr)\in URSR\}$;

(4)资源角色分配关系,资源角色到权限的 $N:M$ 映射 SRP,$SRP\subseteq SR\times P$,权限 p 所授予的资源角色集 $resourcerole(p)=\{sr\in SR,(sr,p)\in SRP\}$;资源角色 sr 的权限集 $sr\_permission(sr)=\{p|p\in P,(sr,p)\in SRP\}$;

(5)权限分配关系,权限表示对资源的操作,$P\subseteq Op\times O$,Op 表示操作集合,O 表示对象(资源)集合;权限 p 的操作集 $p\_operationSet(p)=\{op|op\in Op,(op,o)\in P\}$。

定义 2 4-RBAC 模型的授权由三元组(S,O,Op)表示,S 表示主体,O 表示客体,Op 表示操作。该授权的产生要经过如下步骤:管理员给用户(u,这里也是 S)分配用户角色,用户角色向资源域请求,由资源域的策略决策点批准获得资源角色,再由资源角色获得权限(O 及 Op)。

定义 3 角色关系:

(1)用户角色层次 URH,$URH\in UR\times UR$ 是一种 UR 上的偏序关系,称为继承关系,记为:$\geqslant$,设 $ur1,ur2\in UR$,则 $ur1\geqslant ur2$ 当且仅当 $ur1$ 拥有 $ur2$ 的所有资源角色;

(2)资源角色层次 SRH,$SRH\in SR\times SR$ 是一种 SR 上的偏序关系,称为继承关系,记为:$\geqslant$,设 $sr1,sr2\in SR$,则 $sr1\geqslant sr2$ 当且仅当 $sr1$ 拥有 $sr2$ 所有权限。

定义 4 约束关系:

(1)冲突角色。冲突的用户角色不能同时授予同一个用户;冲突的资源角色不能同时授予同一个用户角色。使用集合 CUR 来定义冲突用户角色,$CUR\subseteq UR\times UR$ 对于一个用户角色 ur,定义与其冲突的用户角色集合 $CRWith(ur)=\{uri|(ur,uri)\in CUR\}$;使用集合 CSR 来定义冲突资源角色,$CSR\subseteq SR\times SR$ 对于一个资源角色 sr,定义与其冲突的资源角色集合 $CRWith(sr)=\{sri|(sr,sri)\in CSR\}$。

(2)冲突权限。在一个操作中不能够赋予同一个用户的权限,使用集合 CP 来定义冲突权限,$CP\subseteq P\times P$,对于一个权限 p,定义与其冲突的权限集合为 $CPWith(p)=\{pi|(p,pi)\in CP\}$。

算法 1 用户访问数据库的权限计算算法

输入:用户名,服务请求

输出:授予用户权限

Step 1 用户的预定义权限获取

Step1.1 获得用户 u 的用户角色集,即 $urSet=u\_userrole(u)$;

Step1.2 获得用户 u 的资源角色集,用户角色 ur 的资源角色集为 $ur\_resourcerole(ur)$,则用户 u 的资源角色集为 $resourceSet=u\_resourcerole\text{-}Set(u)=\bigcup_{ur\in urSet}ur\_resourcerole(ur)$;

Step1.3 获得用户 u 的权限集,资源角色 sr 的权限集为 $sr\_permission(sr)$,用户 u 的权限集为 $permissionSet=u\_permissionSet(u)=\bigcup_{sr\in resourceSet}sr\_permission(sr)$;

Step 2 本次用户的动态权限获取:利用用户的请求 $request(u,o)$ 与用户的预定义权限集 $permissionSet$ 获得用户的动态有效权限 $NowpermissionSet(u,o)$,即 $NowpermissionSet(u,o)=request(u,o)\cap permissionSet$。

用户的本次权限与他的本次请求有关,并且仅授予他与本次请求相关的权限,故用户所得的权限是他预先定义的权限与本次请求所需权限的交集,既 $NowpermissionSet\subseteq permissionSet$,体现了授权的动态性与最小特权原则。

3.3 安全分析

(1)最小特权原则。执行时只给用户分配所需权限,通过结合权限的静态授予(预分配)与动态授予来实现。

(2)职责分离原则。有时,一些敏感操作需不同的用户执行或用户不能同行拥有某些操作,这可通过冲突角色约束与冲突权限约束来实现。

(3)自治原则。用户角色由用户管理员管理;资源角色由资源提供者维护。体现多自治域协同的权利自治原则。

4 多自治域协同数据库访问控制体系结构

4.1 体系结构构件

含有 8 个主要构件:(1)用户角色管理器,创建、修改和删

除用户角色,定义用户角色与资源角色的映射;(2)资源角色生成器,管理资源角色,定义资源角色与权限的映射;(3)策略管理器,生成访问控制信息和访问控制规则,如策略"允许域 D_1 的用户角色 ur 拥有域 D_2 的资源角色 sr", 策略信息保存在策略知识库中;(4)资源角色授予模块(PEP),接受并转发用户角色的请求给 PDP, 根据 PDP 的回答授予/拒绝用户角色请求;(5)策略决策引擎(PDP),依据策略知识库的知识执行 PEP 转发的请求,返回结果给 PEP;(6)用户角色授权请求模块,定义用户角色与资源角色的映射;(7)资源角色检索模块,在 LDAP 目录上检索某资源所提供的资源角色,用户角色在捆绑资源角色之前要首先检索到有用的资源角色;(8)知识库,含用户角色信息库、资源角色信息及其 LDAP 目录和策略知识库。

当给域 D_1 的用户 u_1 预分配一个访问域 D_2 某数据库信息的权限时,管理员运行用户角色管理器,调用用户角色授权请求模块,若存在用户角色满足权限要求,则建立用户与用户角色的映射,权限分配结束;否则 D_1 向 D_2 的资源角色检索模块请求检索,获取相应的资源角色,D_1 与 D_2 的资源角色授予模块协商,D_2 的策略决策引擎决定是否把该资源角色授予 D_1 中的用户角色,最后把用户角色分配给用户。

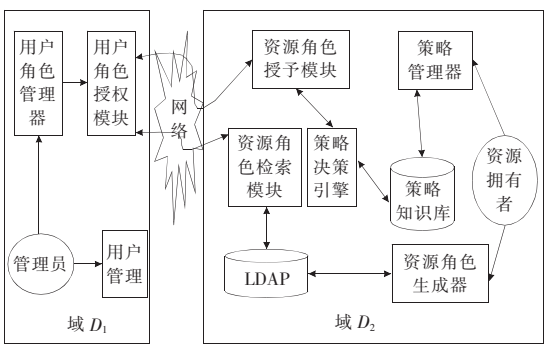

图 2 实现多自治域数据库访问的 4-RBAC 体系结构图

4.2 应用实例

场景:多个医院合作诊疗。一个医院(D_1 域)的专家可为另一个医院(D_2)的病人异地诊疗、开医嘱。病历表(病人 ID,医生,症状,医嘱,异常,进展,病历状态),病历记录有医生创建并登记内容,在治疗过程中医生填写进展及异常,主管护士记录执行医嘱中的病人异常,病历状态含"未提交、提交、完成",只有"未提交"的病历可删除,而已经"完成"的病历不能删除及修改。医院中与医疗直接相关的角色有"主治医生 ur_1,主任医生 ur_2,科室主任 ur_3,涉外专家 ur_4,护长 ur_5,主管护士 ur_6…",显然科室主任>主任医生>主治医生,护长>主管护士,对病历表资源 b 的管理也有如下的资源角色:读表 sr_1,受限写表(写异常的权限)sr_2,创建病历记录 sr_3,受限修改与删除表中自己创建的记录 sr_4。

假设 D_1 域的医生 a 有主任医生和涉外专家两个角色,涉外专家角色可映射域 D_2 的资源角色为$\{sr_1,sr_2,sr_3,sr_4\}$。现在,医生 a 要修改表 b(b 是域 D_2 的资源)的他所诊疗的病人的病历,则通过如下步骤来获得动态权限:

(1)预权限:$permissionSet(a)=sr\_permission(sr_1)\cup sr\_permission(sr_2)\cup sr\_permission(sr_3)\cup sr\_permission(sr_4)$。

(2)本次操作权限:$NowpermissionSet(a,b)=request(a,modify,b)\cap permissionSet(a)=\{(a,restrict\_modify,b)\}$。

5 4-RBAC 模型的原型实现

资源角色信息存放在 LDAP 目录以备各个域的管理员检索并协商分配给用户角色,其它的信息如角色、操作、用户和策略等采用关系数据库(如 Orocle)表示。

5.1 数据库设计

(1)用户端所在域使用的表:用户信息表、用户角色映射表、角色信息表、角色继承关系定义表、用户角色资源角色映射等。

(2)资源端使用的表:操作与操作码映射表、资源角色与操作映射表、资源角色与资源角色号映射表、资源拥有者与资源拥有者号映射表、资源信息表、策略信息表等。

5.2 软硬件环境与开发

(1)操作系统:域 D_1 的操作系统为 Windows XP;D_2 的操作系统为 Fedora Linux 3。

(2)数据库:域 D_1 采用了 Oracle 9.i;D_2 的采用了 Oracle 9.i 和 LDAP。

(3)开发平台:Tomcat5.5,JBOSS,JBuild 2005。

系统的开发采用了 B/S/S 的分布式软件体系结构模式、J2EE 平台 EJB 架构,用户界面采用基于浏览器的 JSP 技术,如权限管理的用户界面用 JSP 编写, 以部署在 Tomcat 容器中的网络服务应用的方式实现; 核心功能主要以 Java 实现并封装在 Java 包中,数据访问采用 EJB 来实现;后台数据库支持为 Oracle9.i 以及 LDAP;域之间的数据通信采用安全通道 SSL。

5.3 关键算法

这里仅简单描述用户角色映射算法, 即域 D_1 管理员给用户角色定义域 D_2 资源角色,算法如下:

算法输入:用户名。

算法输出:建立用户名与资源角色的映射。

Step1.资源查询角色:

Step1.1 打开资源域的检索 LDAP 的界面;

Step1.2 选择搜索条件,并提交;

Step1.3 获取资源角色信息。

Step2.映射用户角色到资源角色

Step2.1 域 D_1 管理员向域 D_2 请求资源角色;

Step2.2 请求被转发到 D_2 的资源角色授予模块,调用决策策略引擎;

Step2.3 调用决策策略引擎根据策略知识库,作出允许/拒绝应答;

Step2.4 由资源角色授予模块把应答返回给 D_1。

Step2.5 D_1 管理员把资源角色映射到用户角色。

6 小结

多自治协同环境是今后应用发展的一个重要方向。RBAC 的设计仅仅针对于同一个安全域,显然已经不能满足多自治域协同安全访问。本文就多自治协同环境下安全访问数据库问题,提出了具有 4 层结构的 RBAC 方案,给出了相应的模型、形式化定义、体系结构和原型实现。下一步工作将方案应用到 Web Service、网格服务等多自治协同环境中的访问控制。

(收稿日期:2007 年 1 月)

参考文献:

[1] Snyder L.Formal models of capability-based protection systems[J]. IEEE Transactions on Computers,1981,30(3):172-181.

(下转 188 页)