面向服务的角色访问控制技术研究

许 峰 赖海光 黄 皓 谢 立

(南京大学计算机软件新技术国家重点实验室 南京 210093)

(南京大学计算机科学与技术系 南京 210093)

摘 要 面向服务的体系结构具有开发效率高、响应快、费用低等优点.但是由于其结构的松散耦合性和计算的动态性,从而造成其安全管理更为复杂.文章首先回顾了访问控制技术的发展,然后提出了一个面向工作流和服务的基于角色访问控制模型.在这个模型中,通过引入服务和授权迁移的概念,加强了对动态服务架构的描述能力.模型对用户角色权限的控制,是通过实际任务和服务状态进行管理的,这样能够有效地加强访问控制的灵活性和系统的安全性.

关键词 面向服务;安全;访问控制;角色;工作流

中图法分类号 TP393

Service-Oriented Role-Based Access Control

XU Feng LAI Hai-Guang HUANG Hao XIE Li

(*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing* 210093)

(*Department of Computer Science and Technology, Nanjing University, Nanjing* 210093)

Abstract Service-oriented architecture (SOA) is an evolution of client/server architecture. A SOA-based system can transparently incorporate services running on different software platforms. It could drive the costs down by achieving automated code generation, reuse, and interoperability. But it will cause the complexity of security management due to its loose-couple and dynamic characteristics. The paper first reviews the development of access control technology, and then presents a workflow-based and services-oriented role-based access control (WSRBAC) model. In the model, the authors introduce two notions of services and authorization transfer to describe dynamic service-oriented architecture. In WSRBAC model, access control system can make its access control decisions by capturing practical relevant environmental context. It can realize access control with dynamic grant and adapt permissions based on the state of workflows and services. This model can enhance system security and provide flexibility in access control system.

Keywords service-oriented; security; access control; role; workflow

1 引 言

面向服务的体系结构(Service-Oriented Architecture,SOA)是一种重要的构建分布式系统的组件模型.由于这类应用系统具有开发效率高、响应快、费用低等优点,因而受到了广泛的关注.它使企业业务从重复流程向维护成本较低、可复用和共享服务

收稿日期:2004-12-05;修改稿收到日期:2005-03-04.本课题得到国家自然科学基金(60473091)和国家"八六三"高技术研究发展计划项目基金(2003AA142010)资助.**许 峰**,男,1970 年生,博士研究生,主要研究方向为信息安全和分布式计算.E-mail:njxuf@163.com.
赖海光,男,1975 年生,博士研究生,主要研究方向为网络安全.**黄 皓**,男,1957 年生,教授,博士生导师,主要研究领域为网络安全.
谢 立,男,1942 年生,教授,博士生导师,主要研究领域为信息安全和分布式计算.

的应用转变,能迅速适应和传送关键业务服务来满足市场需求,同时能降低业务流程的复杂性,达到节约时间和资金的目的.另外与传统的点到点的集成模式相比,SOA 具有基于标准的兼容特性,可以降低开发的复杂性.SOA 在体系结构、设计、实现与部署等方面比传统的分布式对象技术更加合理.与传统的分布式系统相比,基于 SOA 的系统更具标准化和开放性,以致于更易受到攻击,所以其安全性更加脆弱.

目前大多数访问控制模型主要解决后台应用数据的保护问题,不能适应日益复杂的分布式系统.在传统的分布式环境中,客体对象通常是一种静态的对象.例如对数据库的访问控制中,其授权对象是确定的数据表,授权的操作也是固定的几种操作,如查询或删除等操作.随着面向服务的应用系统的发展,对访问控制技术提出了新的要求.传统的面向数据对象的访问控制技术,很难满足应用系统的动态性和开放性的要求.

为了适应面向服务和工作流的应用系统需求,加强对 SOA 系统的访问控制,本文提出了基于工作流和服务的角色访问控制模型.本文第 2 节介绍了国内外相关研究工作和进展;第 3 节提出了基于工作流和服务的角色访问控制模型;第 4 节总结全文.

2　相关工作和进展

访问控制技术的研究一直是信息安全研究的热点问题.访问控制技术起源于 20 世纪 70 年代.最初的需求是保护大型计算机系统的数据集的安全[1].20 世纪 70 年代初,Anderson[2] 首先提出了引用监视器(reference monitor)的概念,为访问控制成为一项关键的安全技术奠定了理论基础.在随后的 30 多年中,先后出现了多种访问控制模型.目前最流行的访问控制模型有自主访问控制模型(Discretionary Access Control,DAC)、强制访问控制模型(Mandatory Access Control,MAC)和基于角色的访问控制模型(Role-Based Access Control,RBAC).

自主访问控制在操作系统和数据库系统中得到了广泛的应用.然而在 DAC 模型中,访问权限的授予是可以传递的.一旦访问权被传递出去将难以控制,会带来严重的安全问题.MAC 源于对信息机密性的要求以及防止特洛伊木马之类的攻击.虽然 MAC 增强了信息的机密性,但不能实施完整性控制.

随着计算机网络的发展,特别是 Internet /Intranet 的广泛应用使信息的完整性和可用性要求超过了机密性,对授权管理和策略配置的便捷性提出了要求.而传统的 DAC 和 MAC 策略难以提供这方面的支持.在 DAC 和 MAC 系统中访问权限直接授予用户,而系统中的用户数量众多且经常变动,这就增加了授权管理的复杂性.

1992 年 Ferraiolo 和 Kuhn 提出了 RBAC 模型[3].RBAC 模型的突出优点是简化了各种环境下的授权管理.通过引入角色这一中介,实现了用户与权限的逻辑分离.RBAC 的思想是将访问权限分配给角色.与用户相比角色是相对稳定的.角色实际上是与特定工作岗位相关的一个权限集,当用户改变时只需进行角色的撤消和重新分配即可.传统 RBAC 比较适合于访问控制对象和操作比较明确的系统,如典型的 RBAC96 模型[4] 和 NIST 的 RBAC 模型[5] 都是基于这种传统的分布式环境的,所以缺乏对主客体的动态描述.在传统的 RBAC 模型中,访问控制的对象是文件、数据库表等资源.由于系统的复杂和现代分布式系统对应用的动态要求,使访问控制的对象变动较大,这增加了授权管理的难度.通过对这些访问控制模型的研究可以看出,它们都是从系统的角度出发保护资源.这些模型都是被动的安全模型.其弱点在于都没有考虑主体执行操作时所处的环境,这样容易造成安全隐患.只要主体拥有对客体的访问权限,主体就可以无数次地使用该权限.而不能根据所处的环境,对主体所拥有的权限进行动态管理.进行授权时将权限提前授予,不符合“最小特权原则”.

随着面向服务架构的分布式计算的发展,对访问控制提出了新的要求.与传统的分布式系统相比,在面向服务架构的分布式系统中,提出请求的主体和提供服务资源的客体都具有较高的动态特性.主体动态特性是由于主体操作方式的多样性和用户的计算环境的异构性造成的.服务本身的动态特性表现为:服务提供的功能可能扩充或被修改,服务的组成也具有动态的变动性.这就要求访问控制系统应该能够动态地适应这种变化,能够根据安全相关的环境做出其访问控制决策.

为此,我们提出了面向工作流和服务的基于角色访问控制(Workflow-based and Service-oriented Role-Based Access Control,WSRBAC)模型.它与传统的 RBAC 模型相比:

(1)引入了服务(Service)的概念,简化了对访问控制对象的管理.该模型能够较好地适应面向服务的计算环境.这样访问控制系统就从对资源的静态保护转移到对服务动态的授权保护上.我们在文献[6]中提出了面向服务的角色访问控制模型.本文在此基础上进行了进一步阐述.

(2)采用角色的动态管理,加强了对用户实际行使角色权限的控制能力.从安全监控的角度来看,被激活的角色也是被管理的对象.传统的 RBAC 模型中的角色虽然能够表示角色在系统中的存在,但是不能表示角色被多个用户激活运行的特征.为此,我们提出了角色扮演者(Actor)的概念.

与传统的 RBAC 模型相比,WSRBAC 模型通过引入服务对象和角色扮演者的概念,增强了对动态特性的描述能力.

(3)采用工作流的控制方法,引入了授权迁移的概念,以实现"最小特权原则".类似的方法是基于任务的访问控制模型[7,8].该模型从任务和任务执行来建立安全模型并实现访问控制,在任务处理的过程中提供动态的权限管理.但是该方法没有考虑基于角色访问控制的问题.我们采用任务的流程作为一种上下文环境,对角色的权限进行约束.本文通过将任务和服务相联系,引入授权迁移的概念,实现面向服务的访问控制.

(4)安全管理的方便性.WSRBAC 模型与传统的 RBAC 模型的区别在于:WSRBAC 模型对用户的权限控制,是通过角色分配和根据任务和实际环境进行管理的.这样既方便了权限管理,又加强了授权的控制力度.

3 基于工作流和服务的角色访问控制模型(WSRBAC)

3.1 WSRBAC 的基本概念

在传统的 RBAC 中,用户一旦激活某个角色,就拥有了该角色的权限.在 WSRBAC 模型中,用户的权限既受限于其所扮演的角色,又取决于实际的工作流状态.这样既能充分满足任务流的完整性约束,又能够增强系统的安全性.在 WSRBAC 模型中,用户激活一个角色时,相应地激活一个动态临时的 Actor.Actor 作为一个用户拥有该角色的代理,执行后续的操作.这样传统 RBAC 中对用户及其角色的管理,就映射为对角色扮演者的管理.而 Actor 最终所获得的权限由工作流的状态(授权迁移)决定.在执行任务时只给用户分配所需的权限,未执行任务或任务终止后用户不再拥有所分配的权限;而且在执行任务过程中,当某一权限不再使用时,Actor将被释放,这样分配给用户的权限就可以被动态收回.

在定义 SRBAC 模型之前,我们给出以下几个定义.

定义 1(角色扮演者,Actor). 角色扮演者是由一个角色被用户激活后产生的一个动态对象,是用户执行该角色的代理.其集合 Actors 记为 A.可以用一个元组〈User,Role,Lifetime〉表示;其中:

(1)User 是 Actor 所代理的用户且 User$\in U$,U 是用户集;

(2)Role 是所激活的角色且Role$\in R$,R 是角色集;

(3)Lifetime 为 Actor 的生存时间,包括两个时间项:产生时间和截至时间.

用户激活一个角色时,相应地激活一个 Actor.对 Role 的约束相应地传递到 Actor 中.这样用户获得服务的过程就映射为角色扮演者获得服务的过程.这个过程是一个动态过程.在 WSRBAC 模型中,我们通过对角色扮演者(Actor)的控制,实现基于角色的访问控制.系统通过监控所有 Actor 的行为和状态,实现对所有用户的角色行为的安全监控和跟踪.

定义 2(原子功能,Atom Function). 原子功能是指对系统资源对象的基本操作,且该操作不能分解为其它操作.可以用一个二元组〈Operation,Object〉表示.记作 af,其集合记为 AF.

定义 3(服务,Service). 服务是由一组独立的原子功能组成的集合,记作 Service.其集合称为服务集(Services),记为 S.

从上述定义可以看出,服务是对操作和对象的封装,它为用户提供了调用接口.传统的 RBAC 模型中的授权对象,在服务中表现为一个嵌入的参数.用户通过服务接口获取服务内容.

定义 4(服务状态,Service State). 服务状态是服务实例在执行过程中可能经历的状态.令$s\in S$,则该服务的状态为 $s(state)$.在我们的模型中,有如下 5 种服务状态:

(1)睡眠状态.表示一个服务处于未被激活状态.

(2)就绪状态.表示一个服务的运行条件已经满足,完成了运行前的准备工作.

(3)挂起状态.表示一个服务由于某种原因(如

等待资源)被暂停运行.

(4)运行状态.表示一个服务被成功激活后正在运行.

(5)终止状态.表示一个服务被终止运行(可能是由于服务完成或出现异常而被终止).

图 1 显示了一个服务实例的状态迁移图.当一个服务被调用时,将生成一个服务实例.当一个用户调用一个服务时,访问控制系统将根据客户请求判断是否合法.如果请求合法将使服务进入就绪状态.如果此时该服务可用,那么服务将被激活进入运行状态;如果不可用则进入挂起状态.在运行时如果出现异常,则终止该服务.

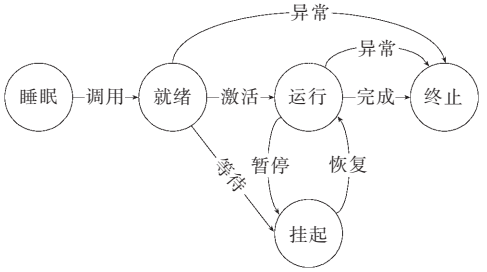

图 1　服务实例的状态迁移图

服务关系反映了业务系统中的工作流策略.服务间的依赖关系对我们制定策略是非常重要的.

定义 5(服务关系,Service Relation).　服务关系是服务间的依赖关系.

根据服务的状态,可以定义如下的服务关系:

(1)同步关系.是指服务间存在状态的同步.也就是对于任意两个服务 s_1 和 s_2($s_1 \in S, s_2 \in S$),存在 $s_1(state1) \leftrightarrow s_2(state2)$.这意味着当 s_1 进入状态 $state1$ 时,同时 s_2 必须进入 $state2$,反之亦然.这时我们称服务 s_1 和 s_2 间存在同步关系.

(2)顺序关系.是指服务间存在状态的顺序关系.也就是对于任意两个服务 s_1 和 s_2($s_1 \in S, s_2 \in S$),存在 $s_1(state1) < s_2(state2)$.这意味着只有当 s_1 进入状态 $state1$ 后,s_2 才能够进入 $state2$.这时我们称服务 s_1 和 s_2 间存在顺序关系.

(3)互斥关系.是指服务间存在状态的互斥关系.也就是对于任意两个服务 s_1 和 s_2($s_1 \in S, s_2 \in S$),存在 $s_1(state1) \rightarrow \neg s_2(state2)$.这意味着当 s_1 进入状态 $state1$ 后,s_2 不能够进入 $state2$,反之亦然.这时我们称服务 s_1 和 s_2 间存在互斥关系.

其中,符号$<$表示先后关系,符号$\leftrightarrow$表示同步关系.

定义 6(授权服务,Authorization Service).　授权服务是授权可被调用的服务.其集合称为授权服务集(Authorization Services),简称为AS,且$AS \subseteq S$.

定义 7(授权结构体,Authorization Unit).　授权结构体指在一个工作流程中对被处理对象的一次处理过程.授权结构体是一个基本授权执行单元,简称为 Au,其集合为 AU.一个授权单元由一个角色扮演者集和多个授权服务集组成.授权服务集则是角色扮演者集成员所拥有的访问许可.

当用户激活其分配的角色后,就相应地生成一个角色扮演者,该角色扮演者将获得该角色的授权服务集 AS.根据工作流的状态,其授权将要受到限制.在这个授权结构体执行过程中,该 Actor 能够获得的许可服务称为可执行服务集(Enable Services),记为 ES.执行授权服务集是其授权服务集的子集.图 2 是授权结构体的示意图.

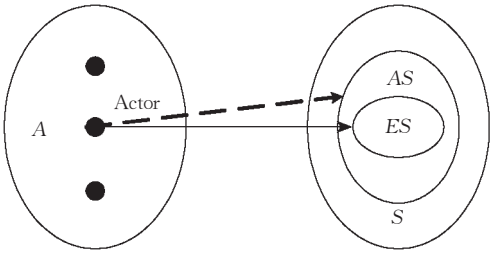

图 2　授权结构体

定义 8(任务,Task).　任务是工作流程中的逻辑单元.它是一个可区分的动作,可能与多个用户相关,也可能包括几个子任务.授权结构体是任务在计算机中进行控制的一个实例.

定义 9(授权依赖,Authorization Dependence).在工作流的执行过程中,授权结构体之间存在授权相互制约关系,我们称之为授权依赖关系,简称为 AD.可能存在以下依赖关系:

(1)同步依赖.对于任意的两个授权单元 AU_1 和 AU_2,如果 AU_1 和 AU_2 同时进入激活状态,即存在 $AU_1 \leftrightarrow AU_2$;

(2)顺序依赖.对于任意的两个授权单元 AU_1 和 AU_2,如果 AU_2 必须在 AU_1 已经完成后才能被激活,即存在 $AU_1 < AU_2$;

(3)失败依赖.对于任意的两个授权单元 AU_1 和 AU_2,如果 AU_1 失败后,则触发 AU_2.即存在 $\sim AU_1 \rightarrow AU_2$;

(4)互斥依赖.对于任意的两个授权单元 AU_1 和 AU_2,如果 AU_1 和 AU_2 不能同时被激活,即存在 $AU_1 \rightarrow \neg AU_2$ 或者 $AU_2 \rightarrow \neg AU_1$;

(5)分权依赖.对于任意的两个授权单元 AU_1 和 AU_2,必须由不同权限级别的用户执行,即存在 $AU_1 \overset{\times}{\longleftrightarrow} AU_2$.

造成授权依赖的主要原因是工作流的任务之间和服务本身具有的依赖关系以及根据安全需要而制定的依赖关系.授权依赖反映了服务的任务特性.

3.2 WSRBAC 模型

WSRBAC 模型是基于任务的 RBAC 模型.该模型由两个视图模型组成:基于任务层面的访问控制视图和基于服务层面的访问控制视图.图 3 是基于任务的 TRBAC 模型,也是 WSRBAC 的基于工作流的访问控制视图.

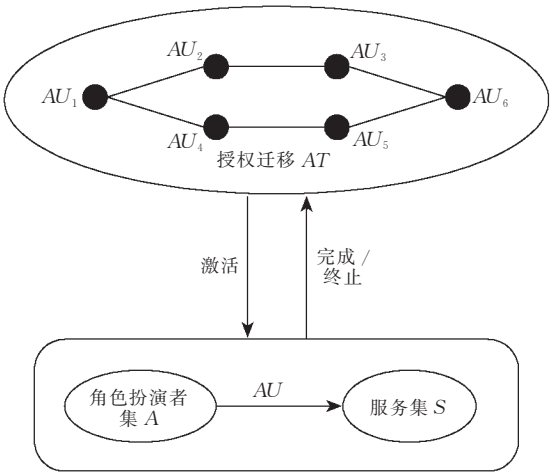

图 3 TRBAC 模型

定义 10(TRBAC 模型). $TRBAC=(T,A,P,AU,S)$,其中 T 是工作流的任务集合;A 是角色扮演者集合;P 是权限集合;AU 是授权结构体;S 是服务集合.定义下列关系:

(1)在访问控制中工作流的任务集 T 由一系列 Au 组成.Au 之间存在的关系为:$Au\times Au\subset 2^D$,$D=$ {同步依赖,顺序依赖,失败依赖,互斥依赖,分权依赖}.

(2)权限集 P 是 AU 和 A 的函数.AU 与 P 是 1:n 关系,$Inti(AU,Actor)\rightarrow P$,$Inti$ 是初始化执行者许可集函数;$G(a,p)\rightarrow Recycle(a)$,$a\in A$,$p\in P$,$G$ 为权限回收函数,$Recycle$ 为回收对象函数.在模型中动态权限的回收只要终止相应的 $Actor$ 即可.

(3)授权结构体 AU 与角色扮演者 $Actor$ 关系是 1:n 关系,$AU\rightarrow A$,是一个从集合 A 中选择一个角色扮演者的函数.

在传统的 RBAC 模型中,访问权限被定义为对被保护对象的授权操作[5~8].在我们提出的面向服务的 RBAC 模型[6]中,用授权服务表示访问权限.WSRBAC 模型中继承了这种表示,只是角色的授权是动态的,受控于工作流的状态.用授权单元序列可以表示授权变化.

在 WSRBAC 模型中,对任务的访问控制最终要分解为对服务的访问控制.图 4 是基于对授权服务控制的 WSRBAC 模型示意图,它通过 AU 和图 3 相联系.其中边 I,II 和 III 反映了角色分配和授权关系,反映了传统的 RBAC 访问控制的机理;而边 IV,V 和 VI 反映了用户激活角色后,它的角色扮演者获得服务的动态过程,反映了动态存取服务的机制.

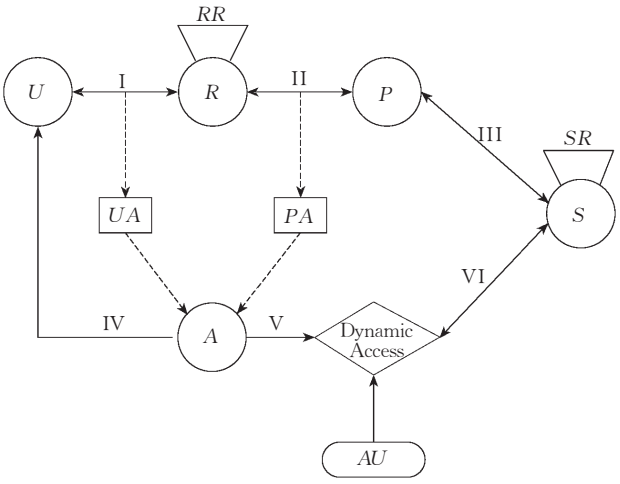

图 4 WSRBAC 模型

定义 11(WSRBAC 模型). $WSRBAC=(U,R,P,A,AU,S)$,其中,U 是用户集,R 是角色集,P 是权限集合,A 是角色扮演者集合,AU 是授权结构体,S 是服务集合.定义下列关系:

(1)用户分配关系(User Assignment)是用户到角色的多对多的映射关系,记为 UA,$UA\subseteq U\times R$;

(2)权限分配关系(Permission Assignment)是访问权限到角色的多对多的映射关系,记为 PA,$PA\subseteq R\times P$;

(3)角色关系(Role Relation)是角色间的相互关系,如角色的继承和派生等关系,角色关系的集合称为角色关系集(Role Relations),记为 RR,$RR\subseteq R\times R$;

(4)服务关系(Service Relation)是服务间的关系,在业务系统中主要表现为服务间的相互依赖关系.服务关系的集合称为服务关系集(Service Relations),记为 SR,$SR\subseteq S\times S$;

(5)授权迁移(Authorization Transition):用户在执行授权的过程中,其授权也在发生着变化.我们将这种变化过程称之为授权迁移,记为 AT.我们用授权结构体序列表示授权迁移:$AT=\{AU_1,AU_2,\cdots,AU_n\}$.其中,$AU_1,AU_2,\cdots,AU_n$ 是具体的授权项(授权单元).

(6)动态存取(Dynamic Access),是执行过程中基于授权步的决策函数.根据授权迁移的状况和角色的权限,系统动态决定角色扮演者的实际执行权限.

WSRBAC 模型的访问策略包含在 *AU-AU*, *AU-A*, *AU-AS* 关系中. *AU-AU* 的关系决定一个工作流的执行流程, *AU-A* 和 *AU-AS* 组合决定一个授权实例的运行.这些部件的关系一般由系统管理员直接配置.

3.3　访问控制策略和机理

WSRBAC 是一个随时间、进程和语境发生变化的动态访问控制模型.角色扮演者获得服务的过程,是一个动态交互的过程.在服务的进程中,系统根据授权结构体的状态和限制条件决定服务的许可.

在 WSRBAC 模型中,访问控制的实施过程为:权限分配、角色权限激活和动态权限调整 3 个阶段.

(1)权限分配

权限分配过程包括系统管理员对用户进行角色分配以及对角色的权限进行预定义.在 WSRBAC 模型中,用户权限分配可以用二元组$\langle u,r\rangle$表示,记为 $ua(u,r)$.我们将角色 r 授予的一组用户集,记为 $UA(r)$.

$$UA(r)=\{ua(u_i,r)\mid ua(u_i,r)\in UA, i=1,2,\cdots,n\}.$$

在系统中权限表现为调用服务的能力.这样 PA 关系就映射为 R 与 S 的关系.可用一个二元组$\langle r,s\rangle$表示,记为 $pa(r,s)$.同理,一个角色 r 被授予一组权限集,记为 $PA(r)$.

$$PA(r)=\{pa(r,s_i)\mid pa(r,s_i)\in PA, i=1,2,\cdots,n\}.$$

(2)角色权限激活

用户要获得实际的角色授权,需要在静态角色分配的基础上激活相应的角色.在 WSRBAC 模型中表现为用户激活角色,系统生成与之相应的Actor. Actor 代表用户执行所激活的角色.我们将 Actor 的生成函数记为 $Create\_Actor(u,r)$.

当用户提出服务请求后,系统将对用户进行身份认证,并查找用户是否被分配了具有访问该服务权限的角色,如果成功则系统生成对应的 Actor, Actor 将代表用户执行其激活的角色权限. $Create_Actor(u,r)$的生成条件可以如下表示:

$$request(u,s)\wedge u\in UA(r)\wedge s\in PA(r)\rightarrow Create\_Actor(u,r).$$

(3)动态权限调整

在 WSRBAC 模型中,其授权可以用$\langle U,R,P,AU,LF\rangle$表示.其中 U,R 的含义同前. AU 表示授权结构体. LF 表示生命周期(Lifecycle),即授权结构体 AU 的存活期限. P 是授权结构体所激活的权限. LF 和 AU 是 WSRBAC 模型与其它控制模型不同的显著特征.在 AU 被触发之前,它的权限是无效的,其中的授权服务是不可用的.当授权结构体被触发时,它的角色扮演者开始拥有调用授权服务的权限,同时其生命时钟开时计数.在 AU 的寿命期限中授权是有效的.当授权超过寿命期时授权无效.在生命终止时开始回收 Actor,从而收回授权.

一个用户 u 能够获得服务 s 的必要条件为,存在一个角色 r, u 是 r 的成员,且 r 对 s 有授权,即

$$Access(u,s)\rightarrow \exists r(u\in UA(r)\wedge s\in PA(r)\wedge s\in AS\wedge AS\subset AU),$$

其中, $Access(u,s)$表示用户 u 能够获取服务 s.

而用户 u 最终能否获得服务 s 取决于工作流是否能够执行 AU,即 AU 能否被触发.角色扮演者获得服务的过程是一个动态交互的过程.在服务的进程中,根据服务的状态和限制条件,系统动态决定其获取服务的许可.所以,授权是一个随时间、进程和语境发生变化的过程.

3.4　服务关系与角色授权迁移

WSRBAC 模型通过隐含的工作流条件和服务间的关系设定了用户的访问权限.这样用户的权限不仅取决于预定义的授权,而且取决于系统中服务的运行状态.在授权迁移过程中角色的权限是动态变化的.

我们将角色 r 动态变化着的权限,记为 $RA(r)$.它是角色的实际权限,是其预定义的角色授权 $PA(r)$的一个子集.即 $RA(r)\subseteq PA(r)$.

根据系统的运行状态和服务关系,可以确定 $RA(r)$:

$$PA(r)\wedge AU\wedge SR\rightarrow RA(r),$$

其中, SR 反映了服务间的逻辑约束关系.我们可以通过一个例子说明服务关系对角色权限的限制.

假设一个角色 r 的权限集 $PA(r)$可以表示一组授权服务,即 $PA(r)=\{s_1,s_2,\cdots,s_n\}$.其中任意一个服务 s_i与另一个服务 s_x($s_x\notin PA(r)$)为顺序关系,且 $s_x(running)<s_i(running)$, $s_x(running)$表示 s_x 处于运行状态.如果这时 s_x没有进入运行状态,则 s_i 也不能运行.也就是说此时角色 r 的实际权限不能包括 s_i,即 $s_i\notin RA(r)$.

3.5　WSRBAC 模型的特点

与原有的 RBAC 模型相比,WSRBAC 模型具有以下优点:

(1)支持动态更新

传统的 RBAC 模型以操作和对象作为访问控制的客体,操作和对象的任何变化都可能造成相应的授权变化.在 WSRBAC 模型中服务作为访问控制的客体,服务中的具体操作和对象的变化不会影响到具体的授权.这样正常的版本更新不会影响用户的正常使用.

(2)支持工作流模型

与我们在文献[6]中提出的 SRBAC 模型相比,WSRBAC 模型通过引入了 *AU* 的概念,加强了对工作流的支持.WSRBAC 实现了基于工作流的访问控制,能够较好地适应工作流环境.*AU* 间的相互关系反映了任务间的相互关系.

(3)更好的安全性

通过授权实例的动态权限管理,WSRBAC 支持两个著名的安全控制原则:

①最小特权原则.在执行任务时只给用户分配所需的权限,未执行任务或任务终止后用户不再拥有所分配的权限;而且在执行任务过程中,当某一权限不再使用时,授权实例自动将该权限回收.

②职责分离原则.有时一些敏感的任务需要不同的用户执行,这可通过授权结构体间的分权依赖实现.

(4)安全管理的灵活性、便捷性

WSRBAC 模型采用服务的概念使访问控制粒度更加适合人的思维.采用服务封装,可以实现更加便捷的安全管理.

3.6 WSRBAC 模型的适用场景

采用面向服务的架构开发和整合企业应用,已经成为一种必然的趋势.然而其安全性和管理的复杂性已经成为阻碍其发展的重要原因.WSRBAC 模型适用于面向服务架构的系统,尤其适合于基于工作流的企业计算、电子政务和电子商务等系统.另外如果我们对来自多个系统(或企业间)的 Web 服务进行动态重组和控制,还可以实现动态系统(或企业)联盟.

目前,WSRBAC 模型已经在我们开发的电子政务系统中得到应用.下面以其中的一个公文流转子系统简要说明.在该公文流转子系统中,公文流转过程比较复杂,而且对系统数据安全、用户权限控制等信息安全问题都有严格要求.由于政府办公的特殊性,对公文访问的权限是动态变化的.使得传统的访问控制在复杂的公文流转系统中进行安全管理变得非常困难.我们采用 Web 服务封装具体的业务应用,通过工作流定义具体的业务流程.根据各个职能部门在公文处理流程上的先后顺序和制约关系,用授权服务和角色授权迁移来控制公文在各相关部门的流动和处理.实践表明采用 WSRBAC 模型减轻了安全管理员的工作负担,提高了系统的工作效率,并且可有效保证公文流转的安全性.

4 总 结

以面向服务架构为基础的 Web 服务和网格计算,给企业计算带来了软件开发的高效率和低成本.但是由于其松散耦合的特性和计算的动态性造成其安全管理的复杂性.在传统的 RBAC 模型中,授权对象和操作的任何变化都会造成相应的权限分配变化,使之不能适应面向服务架构系统的动态特性.

近年来,面向服务系统的访问控制研究受到了广泛的关注.目前主要集中在 Web 服务环境中的访问控制问题.在文献[9]中提出对服务形式化描述方法,研究了复合服务的访问控制策略,但是缺乏对 SOA 系统的访问控制策略的整体描述.Bhatti 等人[10]提出了一个基于 XML 的 RBAC 策略描述框架,可以实现 XML 文档的基于角色的访问控制.但是未能解决对动态 Web 服务的访问控制问题.

为此,我们提出了面向服务的访问控制模型[6,11],主要针对 Web 服务环境下的访问控制问题.虽然能够描述面向服务的访问控制,但是还不能够描述面向服务的工作流系统的访问控制.为此本文在研究 SRBAC 模型[6]的基础上,提出了基于工作流和面向服务的 WSRBAC 模型.WSRBAC 不仅适用于 Web 服务环境,而且适用于其它基于 SOA 的计算环境(如网格环境).应该说它是一个更加全面的模型,其服务层视图更具有通用性,可以适用于所有的 SOA 系统;而其工作流视图可以描述工作流系统,通过对企业应用服务进行重组,可以构建动态的企业联盟.

WSRBAC 模型有效地加强了系统的安全性和访问控制的灵活性.WSRBAC 模型引入了服务和角色扮演者的概念,简化了对访问控制对象的管理,使之能够较好地适应面向服务的计算环境.服务开发商只要保持原有的接口定义,对服务的具体操作进行修改和变更,就不会影响到原有的权限分配.采用工作流的控制方法,引入了授权迁移的概念,实现了"最小特权原则".在任务处理的过程中提供动态的权限管理.WSRBAC 模型对用户的角色权限控制,

是通过实际任务和服务状态进行管理的.这样既方便了权限管理又加强了系统的安全.

致　谢　在此对审稿人提出的宝贵意见,致以诚挚的谢意!

参　考　文　献

1 Gladney H. M., Meyers J. J., Worley E. L.. Access control mechanism for computing resources. IBM Systems Journal, 1975, 14(3): 212～228

2 Anderson J. P.. Computer security technology planning study. Air Force Electronic Systems Division, Hanscom AFB, Bedford, MA: Technical Report ESDTR-73-51, 1972

3 Ferraiolo David, Kuhn Richard. Role-based access controls. In: Proceedings of the 15th NIST-NCSC National Computer Security Conference, Baltimore, MD, 1992, 554～563

4 Sandhu R., Conyne E. J., Lfeinstein H. L. *et al*.. Role based access control models. IEEE Computer, 1996, 29(2): 38～47

5 Ferraiolo D. F., Sandhu R., Guirila S., Kuhn D. R., Chandramouli R.. Proposed NIST standard for role-based access control. ACM Transactions on Information and System Security, 2001, 4(3): 224～274

6 Xu Feng, Lin Guo-Yuan, Huang Hao, Xie Li. Role-based access control system for Web services. In: Proceedings of the 4th International Conference on Computer and Information Technology (CIT'04), Wuhan, 2004, 357～362

7 Thomas R. K., Sandhu R.. Task-based authentication controls (TABC): A family of models for active and enterprise-oriented authentication management. In: Proceedings of the IFIP WG11.3 Workshop on Database Security, London, 1997, 166～181

8 Deng Ji-Bo, Hong Fan. Task-based access control model. Journal of Software, 2003, 14(1): 76～82(in Chinese)
(邓集波,洪　帆.基于任务的访问控制模型.软件学报, 2003, 14(1): 76～82)

9 Sudhir Agarwal, Barbara Sprick. Access control for semantic Web services. In: Proceedings of IEEE International Conference on Web Services (ICWS'04), San Diego, California, USA, 2004, 770～773

10 Bhatti R., Joshi J. B. D., Bertino E., Ghafoor A.. Access control in dynamic XML-based Web-services with XRBAC. In: Proceedings of the 1st International Conference on Web Services, Las Vegas, 2003, 243～249

11 Xu Feng, Xie Jun, Huang Hao, Xie Li. Context-aware role-based access control model for Web services. Lecture Notes in Computer Science 3252, 2004, 430～436

XU Feng, born in 1970, Ph. D. candidate. His research interests include information security, distributed systems, and image processing.

LAI Hai-Guang, born in 1975, Ph. D. candidate. His research interests include computer network and network security.

HUANG Hao, born in 1957, professor, Ph. D. supervisor. His research interests include computer network and network security.

XIE Li, born in 1942, professor, Ph. D. supervisor. His research interests include information security and distributed system.

Background

The work is supported by the National High Technology Research and Development Program (863 Program) of China under grant No. 2003AA142010, and the National Natural Science Foundation of China under grant No. 60473091.

The research directions of this group include information security in distributed system, new technology in distributed system and network security. The primary work of this research is to solve the problem of how to enhance system security and provide flexibility in access control system. The authors present a role-based access control model for service-oriented architecture, and a dynamic mechanism based on the state of workflows and services to enhance system security.