基于时间约束和上下文的访问控制模型研究

杨 晶,张永胜,孙翠翠,王 莹
(山东师范大学 信息科学与工程学院,山东 济南 250014)

摘要:随着网络应用的普及,分布式环境中的访问控制受到越来越多的关注。Web服务作为一种新型的分布式计算模式,其动态、开放、异构的特点决定其需要更加灵活、细粒度的动态授权机制。目前的访问控制模型越来越不能满足面向服务计算环境的安全需求。提出了一个基于时间约束和上下文的动态访问控制模型TCDAC。该模型对基于角色的访问控制进行扩展,引入时间约束和上下文的概念,使访问控制的授权与时间有关,并能根据上下文信息动态地做出访问控制决定,提高了访问控制的安全性,满足了面向服务系统访问控制对动态性的需求。

关键词:访问控制;时间约束;上下文;Web服务
中图分类号:TP309 **文献标识码**:A **文章编号**:1673-629X(2011)01-0143-04

Access Control Model Based on Time-constraint and Context

YANG Jing, ZHANG Yong-sheng, SUN Cui-cui, WANG Ying
(School of Information Science and Engineering, Shandong Normal University, Ji'nan 250014, China)

Abstract: With popularization of network application, access control in distributed environment has attracted increasing attention. As a new distributed computing model, inherent heterogeneity, opening and highly dynamic nature of Web services determine that it requires more flexible, fine-gained and dynamic authorization mechanism. It is more and more unsuitable for existing access control models and mechanisms to meet the requirement of securing the services in the SOC environment. A dynamic access control model based on time-constraint and context (TCDAC) was proposed in this paper. The model extends role-based access control. By introducing time-constraint and context concept, the authorization of access control is related with time, and access control decision can be made dynamically according to context information. It improves the security of resource access, and satisfies the demands of service-oriented system for dynamic access control.

Key words: access control; time-constraint; context; Web services

0 引 言

近年来,Web服务作为一种崭新的用于分布式环境中的计算模型,以其松散耦合、平台无关性、可复用性好、开放性等特点得到广泛应用和发展,使得Internet上信息资源共享更为有效、便利。然而,随着网络技术的快速发展和普及,Web服务安全越来越成为人们关注的焦点问题。访问控制是保护Web服务安全的不可或缺的重要安全措施。访问控制就是通过某种途径授权或限制对关键资源的访问,防止非法用户的侵入或合法用户的不慎操作所造成的破坏。

目前广泛使用的基于角色的访问控制模型主要依靠主体的标识、属性和角色信息进行授权,不能对用户所处的环境上下文进行判断而实现动态的管理,而且其主要工作均立足于与时间特性无关的其他方面,在现实生活中有很多与时间有关的访问控制不能够得到很好解决,特别是一些要求时间性很高或者周期性规律很强的访问控制(如面向服务计算环境中的访问控制),都需要进行时间约束的控制。因此,基于角色的访问控制模型并不能满足Web服务访问控制动态、灵活的需要。新型的访问控制如基于任务的访问控制(TBAC)、基于组的访问控制(TMAC)、基于时间的角色访问控制(TRBAC)等也不能充分反映动态访问控制的过程[1,2]。文献[3]描述了一个引入时间后的角色访问控制模型,并提出相应的算法解决时间约束的判断问题,但是并没有考虑上下文信息对访问授权的影响。文献[4]针对工作流系统的安全性需求提出了一种面向服务的工作流动态访问控制模型,该模型引入了上下文的概念,增强了访问控制的灵活性和系统的安全性。但是,该模型不能很好地满足面向服务系统对时间性的需求。

收稿日期:2010-05-25;**修回日期**:2010-08-01
基金项目:山东省自然科学基金(Y2008G22)
作者简介:杨 晶(1986-),女,山东德州人,硕士研究生,研究方向为Web服务安全、面向服务计算的访问控制;张永胜,教授,主要研究方向为软件工程环境、Internet/Intranet工程、网络信息安全。

针对以上不足，文中提出了一个基于时间约束和上下文的动态访问控制模型(TCDAC)，综合考虑了访问控制过程中的时间特性和上下文信息，对授权约束引入时间特性，并根据上下文信息动态地激活用户和角色、角色和权限的分配，使对资源的访问实现更加高效、动态、细粒度的控制。

1 相关概念

1.1 基于角色的访问控制 RBAC

RBAC 是目前国际上得到广泛关注和大量应用的访问控制模型。其核心思想是将访问权限与角色相联系，通过给用户分配合适的角色，让用户与访问权限相联系。角色是根据企业内为完成各种不同的任务需要而设置的，根据用户在企业中的职权和责任来设定它们的角色。用户可以在角色间进行转换，可以与一个或多个角色发生联系，角色也可以和一个或者多个用户发生联系。系统可以添加、删除角色，还可以对角色的权限进行添加、删除。

NIST(美国国家标准与技术研究院)制定的标准 RBAC 模型由 4 个部件模型组成，这 4 个部件模型包括 RBAC 的核心(Core RBAC)，RBAC 的继承(Hierarchal RBAC)，RBAC 的约束(Constraint RBAC)中的静态职权分离(SSD)和动态职权分离(DSD)两个责任分离部件模型。RBAC 模型如图 1。

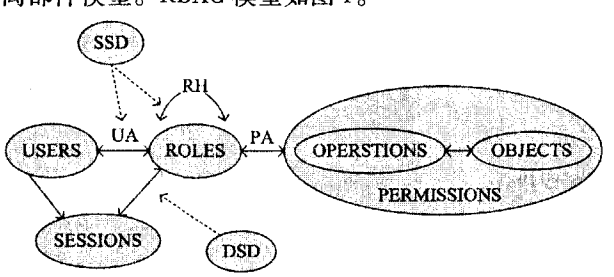

图 1 RBAC 模型

1.2 角色激活

角色激活是基于角色的访问控制中的重要环节。系统在用户首次得到认证后为其分配了静态角色，用户要获得实际的授权，需要激活相应角色。用户通过激活系统为他分配的角色子集来执行与角色相对应的权限。

1.3 时间约束

约束机制是 RBAC 的一个重要方面，RBAC 模型中，授权约束(Authorization Constraint)规定了访问权限被赋予角色时，或角色被赋予用户时，以及当用户在某一时刻激活一个角色时所应遵守的强制性规则。约束主要有关系约束、前提约束、数值约束、授权约束、势约束和时间约束等。时间约束是时间变化和角色权限的依赖关系的规则表示[5,9]。

1.4 上下文

上下文(Context)表示可能引发角色及权限变化的要素集合，包括动态的上下文环境(如时间、地点等)以及动态产生的敏感信息[10,12]。上下文决定当前分配的角色和权限即时有效性。活动角色(Active Role)表示一个分配给用户的角色在当前的上下文下是有效的；活动权限(Active Permission)是一个动态权限，表示分配的权限在当前的上下文下是有效的。上下文敏感的或者依赖上下文，是指角色或者权限分配要动态地根据上下文信息进行调整而不是过去的静态授权关系。

2 TCDAC 模型

2.1 模型结构

基于时间约束和上下文的动态访问控制模型(TCDAC)引入时间约束和上下文的概念，在角色-权限，用户-角色分配过程中充分考虑周期时间和用户所处的环境上下文对授权的影响，满足了 Web 服务对时间性日益增大的需求，并能够根据上下文信息对主体的访问权限进行动态的控制和调整。TCDAC 模型如图 2。

TCDAC 模型的形式化定义如下：

USERS：用户集，指系统中可执行操作的用户集合，用户是一个可以自主访问系统中资源的主体，可以是人、网络中的计算机或者具有智能的自治软件。

ROLES：角色集，包含一组用户和一个权限集，指定了用户以及用户拥有的权限。每个用户静态地从整个角色集中获得一个子集，即静态角色集。静态角色集被激活后才具有相应权限，并且其权限随上下文的变化而动态变化。

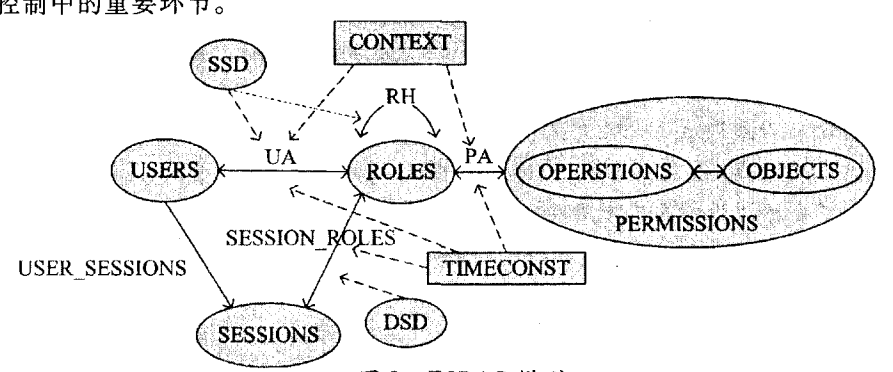

图 2 TCDAC 模型

PERMISSIONS：权限集，不可分割的一组操作的执行权限，是权限分配的最小单元。权限由操作和操作对象组成。

OBJECTS:对象集,系统中需要保护的信息。考虑到分布式环境中资源的动态特性,在该模型中,根据不同资源的动态变化程度将对象分为静态对象和动态对象两种。静态对象是相对比较稳定、不经常发生变化的对象,对于这类对象,将其权限定义到角色中;动态对象的生存期较短或其权限经常发生变化,如果将其权限定义到角色中,那角色权限会频繁变动,造成系统维护困难,因此直接将其权限定义到自身,这样,其权限就会随着对象的消亡而自动消失。

OPERATORS:操作集,定义在对象上的一组操作,操作即对系统中的资源进行访问的动作。

SESSIONS:会话集,用户每次通过建立会话来激活角色,得到相应的访问权限。

CONTEXTS:上下文信息,包括主体上下文信息(如访问时间、访问 IP 地址、访问连接加密等)和客体上下文信息(如系统负载、系统安全、宽带利用率、响应时间延迟等)。模型通过从应用环境中获取与安全相关的上下文信息来动态地改变用户的权限。

TIMECONST:时间约束,是安全管理员事先定义好的。用户在规定的时间内才能拥有某些角色,权限也只能在特定时间内可用。时间约束可用周期时间表达式 P 表示。P 表示周期时间时刻的一个无限集合,用多元组<[begin,end],P>来表示,[begin,end]是时间间隔,表示 P 中时刻的上下限。设 f_r,f_p 分别为角色激活和权限有效的时限布尔函数,则:

$$f_r(t)=\begin{cases}\text{true} & t\in \text{TIMECONST(r)}\\ \text{false} & t\notin \text{TIMECONST(r)}\end{cases}$$

其中,TIMECONST(r)表示角色 r 的时间约束。时间约束规则表示为:

$r\in R_U(u)\Rightarrow f_r(\text{U\_time(u)})=\text{true}$

其中,U_time(u)表示用户 u 所处的时刻。

$$f_p(t)=\begin{cases}\text{true} & t\in \text{TIMECONST(p)}\\ \text{false} & t\notin \text{TIMECONST(p)}\end{cases}$$

其中,TIMECONST(p)表示权限 p 的时间约束。时间约束规则表示为:

$P\in P_R(r)\Rightarrow f_p(\text{R\_time(r)})=\text{true}$

其中,R_time(r)表示角色 r 所处的时刻。

UA:用户分配,用户集到角色集的多对多关系,记录了用户所拥有的角色,UA ⊆USERS×ROLES×CONTEXTS×TIMECONST。用户 U 依据上下文信息在时间约束下被分配给角色 R,表示为 UA = UA ∪ (U,R,C,T)。

PA:权限分配,权限集到角色集的多对多关系,记录了角色所分配的权限。PA ⊆PERMISSIONS×ROLES×CONTEXTS×TIMECONST。把权限 P 分配给角色 R 时也要考虑上下文信息,表示为 PA = PA ∪ (P,R,C,T)。

user_sessions:用户登录,记录了用户所拥有的会话,一个用户可以同时拥有多个会话。user_sessions ⊆ USERS×SESSIONS。

SESSION_ROLES:激活和去活角色,用户在会话中在某一时刻可以激活某个角色,或者去活某个角色。激活的前提是在时间约束下用户拥有该角色并且未在当前会话中激活。一旦激活某角色,用户就拥有了该角色的所有特权。SESSION_ROLES ⊆ROLES×SESSIONS×TIMECONST。

RH:角色继承,若角色 R1 继承了角色 R2,则 R2 的权限也成为了 R1 的权限。RH ⊆ROLES×ROLES。

SSD:静态职责分离,若两个角色之间存在 SSD 关系,则用户不能同时分配这两个角色,并且不能在已有 SSD 关系的两个角色之间定义继承关系。

DSD:动态职责分离,若两个角色之间存在 DSD 关系,则用户可以同时分配这两个角色,但用户不能在同一个会话中同时激活这两个角色。

2.2 访问控制算法

主体获得授权的过程按照如下算法进行:

(1)根据当前用户所请求的对象和要执行的操作,找出对应的角色集。

(2)判断当前用户是否在上一步中的角色集中,如果是,则得到赋予该用户的角色集,转到(3);否则,拒绝该用户的访问请求。

(3)建立用户与角色的一次会话,并判断当前时间是否满足角色激活的时间约束,如果满足,则转到(4);否则拒绝请求。

(4)激活角色,判断角色在当前上下文下是否有效,如果是,则分配给用户的静态角色变为活动角色,转到(5);否则拒绝请求。

(5)判断分配的权限在当前上下文下是否有效,如果是,则该权限转为活动状态,转到(6);否则拒绝请求。

(6)判断当时是否满足该权限有效性的时间约束,如果满足,用户获得所请求的权限,并执行访问操作;否则拒绝请求。

3 模型分析

该模型是对 RBAC 的扩展,除具有 RBAC 的优点外,还有如下特点:

(1)支持动态授权,具有很高的灵活性。在面向服务架构的分布式系统中,由于计算环境的异构性和主体操作方式的多样性,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,该模型弥补了 RBAC 不能动态适应这种变化的不足,在主体获得授

权过程中，可以基于上下文信息的动态变化，对主体的权限做出相应的调整。

(2)实现更细粒度的访问控制。该模型引入了时间约束，使主体在时间约束下才能激活角色，角色在时间约束下才能拥有权限，使系统对资源信息的保护更加细粒度化。

(3)增强了系统的安全性。本模型是 RBAC 模型的扩展，角色的定义源于现实，用户的授权更接近于现实，避免了越权操作的发生。同时，时间约束的引入使角色授权的安全性在一定的时间周期内得以提高。另外，该模型还遵守最小权限、职责分离等安全原则。

(4)该模型可以广泛应用于分布式系统中，满足了分布式应用环境尤其是基于 Web 服务的应用环境对时间性的需求。

4 结束语

根据 Web Services 的动态性、开放性等特点及对访问控制的需求，提出了一个基于时间约束和上下文的动态访问控制模型。该模型综合考虑了角色-权限，用户-角色分配过程中的时间约束和上下文在主体获得授权过程中的影响，解决了一系列与时间有关的角色访问控制问题，使主体对资源的访问更加灵活、安全，提高了模型的描述能力，增强了访问控制的力度，能够更好地适应分布式环境尤其是 Web 服务的访问控制。

参考文献：

[1] 陈 怡，耿国华，李 喆. 动态访问控制的研究与应用[J]. 计算机技术与发展，2006，16(2)：223-225.

[2] 曾小超，张绍连，矛 兵，等. 访问控制技术的研究与进展[J]. 计算机科学，2001，28(7)：26-28.

[3] 杨 升，余文森. 带时间特性的基于角色的访问控制模型的研究[J]. 福建电脑，2007(9)：18-19.

[4] 张 宏，王 红，何绪堂. 面向服务的工作流动态访问控制模型[J]. 计算机工程与设计，2008，19(8)：1958-1960.

[5] 张新华，陈军冰. 时间约束的 RBAC 模型及应用[J]. 计算机技术与发展，2007，17(6)：246-249.

[6] 李席广，徐 蕾，石祥滨. 基于信任和时间约束的访问控制模型 TTRBAC[J]. 计算机应用与软件，2009，26(4)：266-268.

[7] 范艳芳，韩 臻，曹香港，等. 基于时间限制的多级安全模型[J]. 计算机研究与发展，2009，47(3)：508-514.

[8] 李秋敬，刘广亮，谢圣献，等. 基于时间约束的角色访问控制模型研究[J]. 计算机技术与发展，2009，19(8)：162-165.

[9] 杨 珍，刘连忠. 时间约束的角色访问控制系统的设计和实现[J]. 计算机应用研究，2008，25(1)：195-196.

[10] Wolf R，Keinz T，Schneider M. A Model for Context-dependent Access Control for Web-based Services with Role-based Approach [C]// In the Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA'03). Prague，Czech Republic：IEEE Computer Society，2003：209-214.

[11] Shen Hai-bo，Hong Fan. A Context-Aware Role-Based Access Control Model for Web Services[C]// In Grid and cooperative computing2004. Berlin：Springer，2004：430-436.

[12] 王小明，刘 丁，付争方. 工作流系统上下文相关访问控制模型[J]. 计算机科学，2006，33(12)：102-104.

(上接第 139 页)

工程的其他方面的监理工作。

当然，论文的研究尚存在一定的不足之处，概括起来有：(1)只对信息系统工程监理流程进行了一定的研究和探讨，而监理的工作远不止这些，还包含大量的方法和工作；(2)鉴于当前信息系统工程项目承建单位所处的能力成熟度等级，仅仅在 CMM 3 能力等级的水平下研究了监理过程的改进，而没有进一步讨论更高 CMM 等价水平下的过程改进的相关问题。因此，这些问题将称为下一步的研究工作重点。

参考文献：

[1] 于立军. 基于软件工程理论的信息工程监理的研究[D]. 济南：山东大学，2005.

[2] Wohlin C，Runeson P，Höst M. Experimentation in software engineering：an introduction [M]. Boston：Kluwer Academic Publishers，2000.

[3] 秦炳军，张圣坤. 动态过程风险评估方法[J]. 上海交通大学学报，1998，32(11)：7-20.

[4] 柳纯录. 信息系统监理师教程[M]. 北京：清华大学出版社，2005.

[5] 周申蓓. 信息系统工程监理过程改进研究[D]. 南京：东南大学，2004.

[6] Mathiassen L，Scrensen C. The capability maturity model and CASE[J]. Information Systems Journal，2008，6(3)：195-208.

[7] 郑人杰. 基于软件能力成熟度模型(CMM)的软件过程改进[M]. 北京：清华大学出版社，2003.

[8] 蒋依欣. 一种基于 CMM 的软件过程度量改进模型研究及应用[D]. 长沙：湖南大学，2009.

[9] 王艳慧. 基于 CMM 的软件过程改进实践[J]. 计算机技术与发展，2008，18(5)：141-143.

[10] 邓志宏，潘梅森. 信息系统监理能力成熟度模型研究[J]. 计算机工程与设计，2008，29(9)：2420-2425.

[11] 何新贵. 软件能力成熟度模型 CMM 的框架和内容[J]. 计算机应用，2001，21(3)：1-5.

[12] 龚 波. 能力成熟度模型集成及其应用[M]. 北京：中国水利水电出版社，2003.

[13] Florac W A，Park R E，Carleton A D. Practical software measurement：Measuring for process management and improvement[R]. Pittsburg：SEI，1997：180-181.

[14] 韩 杰，顾 庆. 基于 CMM 模型的软件质量保障支撑平台框架[J]. 计算机科学，2001，28(1)：12-15.