

基于 RBAC 扩展的网格访问控制的研究

祁正华¹, 王汝传^{1,2}, 任勋益¹

(1. 南京邮电大学, 江苏 南京 210003; 2. 南京大学, 江苏 南京 210093)

摘 要:访问控制是众多计算机安全解决方案中的一种,是最直观最自然的一种方案。而基于角色的访问控制(RBAC)是最具影响的高级访问控制模型。然而,由于网络的跨组织、动态、异构的特点,建立访问控制还需要对 RBAC 扩展。文中对 RBAC 做了简单介绍,分析了其在网格环境下的不足,重点给出了扩展 RBAC 定义,并以此建立了基于 RBAC 扩展的网格动态访问控制模型,给出了访问控制流程。

关键词:访问控制;基于角色的访问控制;网格

中图分类号:TP311.52

文献标识码:A

文章编号:1673-629X(2006)06-0017-03

Study on RBAC Extention - Based Grid Access Control

QI Zheng-hua¹, WANG Ru-chuan^[1,2], REN Xun-yi¹

(1. Nanjing Univ. of Posts and Telecommunications, Nanjing 210003, China;

2. Nanjing Univ., Nanjing 210093, China)

Abstract: Access control is one solution of computer security, which have straight and nature features. Role - Based Access Control has most important influence among all access control models. But, in grid environment, grid access happens on many heterogeneous groups, and has dynamic characters, so must extend RBAC to adapt to these. This paper introduces RBAC, analyses its deficiency in grid environments, focuses on extending RBAC and one grid access model. Lastly give access control process.

Key words: access control; RBAC; grid

0 引言

David Ferraiolo 和 Rick Kuhn 在 1992 年提出了 RBAC (Role - Based Access Control) 模型, RBAC 在管理大型网络应用安全时所表现出的灵活性和经济性迅速成为最具影响的高级访问控制模型。Sandhu 等于 1996 年提出的 RBAC96 模型由于系统和全面地描述了 RBAC 多层次、多方面的意义而得到了广泛认可。RBAC96 的主要贡献是意识到 RBAC 可以很简单也可以很复杂,不存在一个模型可以应用于所有的场景,而只能根据实际的需要选择合适的模型。然而,因为网格环境的分布式、异构性、不可控制等特点, RBAC 模型并不能很好地适应网格访问控制,鉴于此,文中对 RBAC 模型进行了扩展,重点解决上下文敏感问题,并以此建立了一个网格动态访问控制模型。

收稿日期:2005-09-23

基金项目:国家自然科学基金(60173037, 70271050);江苏省自然科学基金和江苏省自然科学基金预研项目(BK2004218);江苏省高技术研究计划(BG2004004);江苏省计算机信息处理技术重点实验室基金(kjs04)

作者简介:祁正华(1975-),女,陕西宝鸡人,硕士研究生,助教,研究方向为计算机软件、计算机网络和网格、信息安全、移动代理和虚拟现实技术等;王汝传,教授,博导,研究方向为计算机网络和网格计算、移动代理和信息安全技术等。

1 RBAC 及其在网格环境下的不足

RBAC 是策略中立的、一种特殊的强制访问控制^[1],通过角色的继承和职责分离(separation of duty)等约束条件可以表达和支持多种安全策略。在 RBAC 中,角色是用户和权限之间的数据中间层,一方面权限与角色关联,限制每个主体不必要的访问权利^[2]。另一方面,用户作为相关角色的成员。由于一个组织的行为特征和功能是比较稳定的,从而其角色是比较稳定的,而相比之下,角色所关联的权限和用户是动态的。通过用户-角色、角色-权限的关联, RBAC 与直接的用户-权限关联的访问控制模型相比,简化了授权的管理,减少了授权管理工作量的复杂度。

然而,由于网格是一个异构的、跨域的、分布式环境,也包含主体和客体两种实体^[3],标准 RBAC 在网格环境实现访问控制存在着多个组织 RBAC 问题、上下文感知 RBAC 问题等。多个组织 RBAC 问题是指参与网格计算的各个企业、组织或者个人形成了网格计算访问控制的多个管理域,多个管理域之间的信息访问需要交互和协同^[4]。这些管理域之间的交叉验证和授权问题成为网格计算必须要解决的问题。文中集中研究上下文感知 RBAC 问题。

对网格计算的提供者而言要根据当时的情况来决定网格计算的访问授权,把这种授权机制称为上下文感知的

(Context-Aware)授权。这些上下文信息包括:时序、实体间的关系、访问资源的利用情况等。总之,上下文是一系列可能影响安全决策的状态信息集合。

传统的 RBAC 对于一些情况难以描述和定义。比如:情形 1:只允许用户在晚上 12:00 到早上 6:00 对某网格资源进行访问;情形 2:只允许用户在 CPU 负载处于 < 80% 时对 CPU 资源进行访问;情形 3:只允许用户访问 MP3 文件,并且 MP3 文件的大小在 1MB~3MB 之间;情形 4:对资源 B 的访问必须在对资源 A 的访问之后;情形 5:如果一个事务中的所有资源没有全部被授权访问则授权中止。

上述这些情形在基本 RBAC 的几个扩展模型中都有考虑,如 Bertino 等提出了时序访问控制模型来阶段性(按时间)激活/钝化某角色,这种方法可解决情形 1 的问题,通过动作行为间的依赖关系可以实现如情形 4 和 5 的事务控制。但因为约束和依赖关系定义在角色级别,这样对资源的时间访问控制以及角色和资源之间的关系控制就没有办法处理了。RBAC 模型通过引入主题角色、对象角色以及环境角色来策略性地管理访问控制,这种方法可以用于解决情形 1 到情形 5 的问题,但因为引入了独立于用户和角色的多种角色将必然导致潜在的冲突和管理上的复杂性,给系统的扩展也带来了问题。其他的解决方法还有基于扩展状态的方法,Wullems 等则给出了内联网环境下上下文感知的安全授权架构。下文将对 RBAC 进行扩展,解决上下文敏感问题,提出的扩展定义具有简单、易扩展、易实现等特点,能够克服以上方法的不足。

2 基于 RBAC 的扩展

尽管基于角色的访问控制是一个复合的策略^[5],但是 RBAC 方法不能满足网格计算访问控制的要求,在这里笔者提出 RBAC 扩展来重点解决上下文敏感的网格访问控制问题。其定义基础基于核心 RBAC,其主要目标是向前兼容以及简单实用。下面是 RBAC 扩展定义,标准核心 RBAC 的修改扩展部分用黑体字表示。

定义: RBAC 扩展模型中,角色表示用户、对象和操作的对应关系,其中:USERS 为用户;ROLES 为角色;OPS 表示操作;OBS 表示对象;ACT-ROLES 表示活动角色;ACT-PERMS 为活动权限;CONTEXTS 表示上下文特征信息。

基于上下文的用户和角色关系映射表示为:

$$UA \subseteq \text{USERS} \times \text{ROLES} \times \text{CONTEXTS}$$

该式表示一个多对多的用户和角色的上下文敏感的分配映射关系。

已经分配到角色的用户表示:

$$(r: \text{ROLES}, c: \text{CONTEXTS}) \rightarrow 2^{\text{USERS}}$$

将角色 r 依据上下文映射到一个用户集合形式化描述为:

$$\text{assigned_users}(r, c) = \{u \in \text{USERS} | (u, r, c) \in UA\}$$

多对多的权限和角色的上下文敏感的分配映射关系:

$PA \subseteq \text{PRMS} \times \text{ROLES} \times \text{CONTEXTS}$ 。PRMS 表示权限集合: $\text{PRMS} = 2^{\text{OPS} \times \text{OBS}}$

将角色 r 依据上下文映射到一个权限集合: $(r: \text{ROLES}, c: \text{CONTEXTS}) \rightarrow 2^{\text{PRMS}}$ 。形式化描述为: $\text{assigned_permissions}(r, c) = \{p \in \text{PRMS} | (p, r, c) \in PA\}$ 。

权限对操作的映射: $\text{Ob}(p: \text{PRMS}) \rightarrow \{op \subseteq \text{OPS}\}$ 。它赋予权限 p 一个操作的集合。

权限和对象的映射,它赋予权限一个对象的集合: $\text{Ob}(p: \text{PRMS}) \rightarrow \{ob \subseteq \text{OBS}\}$ 。

用户 u 映射到一个会话集: $\text{user_sessions}(u: \text{USERS}) \rightarrow 2^{\text{SESSIONS}}$, SESSIONS 为会话集。

会话 s 映射到一个角色集: $\text{session_roles}(s: \text{SESSIONS}) \rightarrow 2^{\text{ROLES}}$,形式化描述为: $\text{session_roles}(s_i) \subseteq \{r \in \text{ROLES} | (\text{session_users}(s_i), r) \in UA\}$ 。

可用的会话权限: $(s: \text{SESSIONS}) \rightarrow 2^{\text{PRMS}}$, 对一个会话中用户而言可用的权限:

$$\bigcup_{r \in \text{session_roles}(s)} \text{assigned_permissions}(r)$$

上下文(Context)表示和时间、地点、统计相关的一些动态信息的集合,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。动态调整的方法依赖具体的实现,可以采取很多现有的成熟技术加以实现,例如有限状态机、状态转移矩阵、图标法等等。

3 基于 RBAC 扩展的动态网格访问控制模型

由于网格访问控制具有动态性和灵活性,所以一个典型的网格访问控制模型应当具有下列几个典型的特征:策略驱动,访问控制的策略驱动,包含多个方面,例如用户策略、角色继承策略、角色分配策略、权限分配策略、事件策略、角色转移策略以及权限转移策略等等。网格访问控制比较复杂,网格访问控制模型应当提供好的配置方法来对策略、参数等做出配置从而实现安全管理的目的。鉴于此,提出的基于 RBAC 扩展的动态网格访问控制模型如图 1 所示。

验证及授权服务:负载网格计算的验证和授权服务功能,验证及授权是可管理和配置的并且依据策略来动态控制。验证和授权服务使用一个认证机构(CA)来实现。

主体:主体本质上是用户,它是网格资源的访问请求者。每个主体建立后被分配相应的角色和一个角色控制的状态机,状态机用于根据主体环境上下文控制角色的状态转换。在任意时刻,主体被赋予一个活动的角色子集。

客体:客体是资源或者服务,是网格服务的提供者。每个客体有一个角色集合用来定义对客体的访问。角色

是权限的集合,一个权限状态机用来动态地控制角色的权限集合。

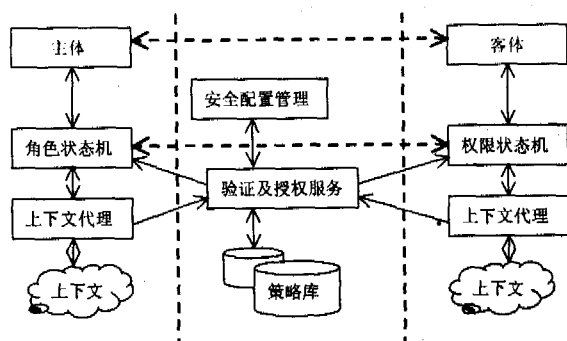


图1 动态网络访问控制模型

上下文代理:由 AAS 分配到指定的主体和客体处,它们使用中间件服务来监测上下文并产生事件触发状态机的状态迁移。

角色状态机:用于维护主体的活动角色子集,它受上下文代理驱动。

权限状态机:用于维护客体角色的活动权限子集,它受上下文代理驱动。

基于该模型的网络访问控制基本流程图如图2所示。

用户首先登录,当通过登录验证后,验证及授权服务授予用户角色集合并启动相关的角色状态机,委派上下文代理到用户宿主环境,上下文代理控制角色状态机动态调整活动角色。资源和服务一旦被允许进入网络提供服务,则对应的角色的权限状态机启动,一个角色可能和多个资源相关。权限状态机根据资源和服务以及其他情形对角色的权限集进行动态调整。对某一时刻的资源访问意味着在此刻的上下文下,用户属于一定的角色,并且该角色在一定的上下文下对资源进行某些操作。

4 结束语

介绍了基于角色的访问控制的标准 RBAC,着重于解

决网络计算访问控制问题,提出了基于标准角色访问控制的扩展方法来解决网络计算访问控制的动态控制、自主上下文感知等问题,并且给出了网络访问控制模型描述。不难看出扩展的 RBAC 方法的主要有内容简单、多粒度控制、易于实现、易于扩展等优点。基于扩展的 RBAC 给出了一个动态模型,该模型强调了上下文对用户的角色进行调整,比较适合网络环境下的资源访问控制。未来工作是结合对该模型的实际应用做进一步研究。

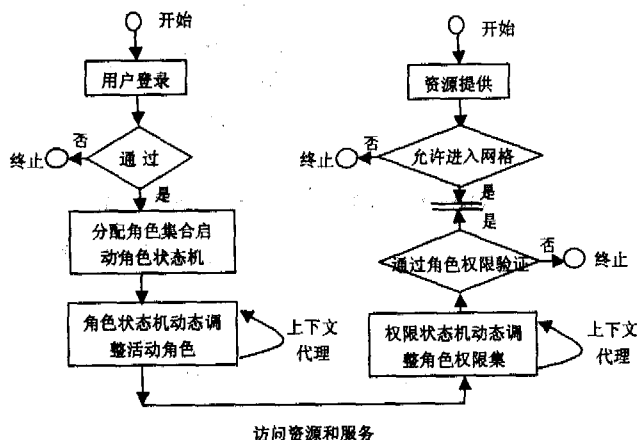


图2 网络访问控制基本流程

参考文献:

- [1] 宁 葵,严 毅,李陶深.一种基于角色访问控制的数据库安全模型[J]. 微机发展,2005,15(10):8-10.
- [2] 赵 亮.访问控制研究综述[J]. 计算机工程,2004,30(2):1-2.
- [3] 徐志伟,冯百明,李 伟.网络计算技术[M]. 北京:电子工业出版社,2004.
- [4] 张 纲,李晓林,游赣梅,等.基于角色的信息网格访问控制的研究[J]. 计算机研究与发展,2002,39(8):952-956.
- [5] 陈 华.网络的访问控制模型[J]. 微机发展,2004,14(8):27-29.

(上接第16页)

在实际编程时,只要将要连接的 DLL 文件、相应的头文件和用上述方法生成的 lib 文件拷贝当前工程项目的目录中即可。

5 结束语

由于 Borland 公司和 Microsoft 公司在 obj 文件、lib 文件格式方面的不同,因此二者的 obj 文件、lib 文件不能混用。C++ Builder 在开发硬件接口方面的应用程序时是提供了相应的工具来解决 lib 文件格式不兼容的问题,即 C++ Builder 的命令行工具 implib。Implib 能由动态连接文件(DLL 文件)生成相应的 lib 文件。

参考文献:

- [1] Cant C. Windows WDM 设备驱动程序开发指南[M]. 北京:

机械工业出版社,2000.

- [2] Richter J. Windows 核心编程[M]. 北京:机械工业出版社,2000.
- [3] Pietrek M. Under the hood: Link-time Code Generation[J/OL]. <http://msdn.microsoft.com/msdnmag/issues/02/05/Hood/>. 1996.
- [4] Pietrek M. Inside Windows: An In-Depth Look into the Win32 Portable Executable File Format[J/OL]. <http://msdn.microsoft.com/msdnmag/issues/02/02/PE/default.aspx>. 2002.
- [5] Pietrek M. Inside Windows: An In-Depth Look into the Win32 Portable Executable File Format (Part2) [J/OL]. <http://msdn.microsoft.com/msdnmag/issues/02/03/PE2/default.aspx>. 2002.