访问控制技术综述

1 引言

随着网络技术的发展和网上应用的日益增加, 信息安全问题日益凸现。 目前, 世界各国都深刻认识到信息、计算机、网络对于国防的重要性,并且投入大量资金进行信息安全的研究 和实践。以美国为例, 从总统令PDD63, 到《信息系统保护国家计划》的颁布和实行, 美国在原有基础上大力加强其信息系统的安全性保障。 日本则强调, 信息安全保障是日本综合安全保障体系的核心。《俄罗斯国家安全构想》中明确提出, 保障国家安全应把保障经济安全放在第一位, 而信息安全又是经济安全的重中之重等。可见, 这些国家对于信息安全的重视程度日益增加。

网络信息安全是对网络信息及其服务的管理, 而服务最终是面向用户的。 因此, 必须从服务信息和使用者两个角度出发来分析网络信息的安全概念。与网络信息有关的安全概念有保密性(confidentiality) 、完整性(integrity) 和可用性(availability) 等, 与使用者有关的安全概念有认证(authentication) 、 授权(authorization) 和抗抵赖(non-repudiation) 等。

当前信息安全技术主要包括密码技术、身份认证、访问控制、入侵检测、风险分析与评估等诸多方面, 在实际应用中, 这些安全技术相互支持和协作, 各自解决安全问题的某一方面。 但目前人们关注的重点是密码技术、身份认证、入侵检测等, 访问控制技术却没有得到应有的重视。事实上, 访问控制技术是一个安全信息系统不可或缺的安全措施, 对保护主机系统和应用系统的安全都有着

重要的意义。

2 传统访问控制策略

访问控制技术起源于 70 年代，当时是为了满足管理大型主机系统上共享数据授权访问的需要。但随着计算机技术和应用的发展，特别是网络应用的发展，这一技术的思想和方法迅速应用于信息系统的各个领域。 在 30 年的发展过程中，先后出现了多种重要的访问控制技术，它们的基本目标都是为了防止非法用户进入系统和合法用户对系统资源的非法使用。 为了达到这个目标，访问控制常以用户身分证为前提，在此基础上实施各种访问控制策略来控制和规范合法用户在系统中的行为。

2.1 自主访问控制

自主访问控制（DAC: Discretionary AccessControl）的基本思想是系统中的主体（用户或用户进程）可以自主地将其拥有的对客体的访问权限全部或部分地授予给其它主体。 实现方法一般是建立系统访问控制矩阵，矩阵的行对应系统的主体，列对应系统的客体，元素表示主体对客体的访问权限。为了提高系统性能，在实际应用中常常是建立基于行（主体）或列（客体）的访问控制方法。

基于行的方法是在每个主体上都附加一个该主体可以访问的客体的明细表，根据表中信息的不同可分为三种形式：权能表（Capabilities）、前缀表（Profiles）和口令（Password）。

权能表决定用户是否可以对客体进行访问以及进行何种形式的访问（读、写、改、执行等）。一个拥有某种权力的主体可以按一定方式访问客体,并且在进程运行期间其访问权限可以添加或删除。

前缀表包括受保护的客体名以及主体对它的访问权。 当主体欲访问某客体

时，自主访问控制系统将检查主体的前缀是否具有它所请求的访问权。

至于口令机制，每个客体（甚至客体的每种访问模式）都需要一个口令，主体访问客体时首先向操作系统提供该客体的口令。

基于列的自主访问控制是对每个客体都附加一个可访问它的主体的明细表。它有两种形式：保护位（Protection bits）和访问控制表（ACL：AccessControl List）。保护位是对所有的主体指明一个访问模式集合，但由于它不能完备地表达访问控制矩阵，因而很少使用。

访问控制表可以决定任一主体是否能够访问该客体，它是在客体上附加一主体明细表的方法来表示访问控制矩阵。表中的每一项包括主体的身份和对客体的访问权。访问控制表是实现自主访问控制的最好的方法。尽管DAC已在许多系统中得以实现（如UNIX），然而DAC的一个致命弱点是：访问权的授予是可以传递的。一旦访问权被传递出去将难以控制，访问权的管理是相当困难的，会带来严重的安全问题；另一方面，DAC不保护受保护的客体产生的副本，即一个用户不能访问某一客体，但能够访问它的拷贝，这更增加了管理的难度。而且在大型系统中，主、客体的数量巨大，无论使用哪一种形式的DAC，所带来的系统开销都是难以支付的，效率相当低下，难以满足大型应用特别是网络应用的需要。

2.2 强制访问控制

MAC（Mandatory Access Control）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。MAC通过无法回避的存取限制来阻止直接或间接的非法入侵。系统中的主/客体都被分配一个固定的安全属性，利用安全属性决定一个

主体是否可以访问某个客体。 安全属性是强制性的，由安全管理员(Security Officer) 分配，用户或用户进程 不能改变自身或其它主 / 客体的安全属性。

MAC 的本质是基于格的非循环单向信息流政策。 系统中每个主体都被授予一个安全证书，而每个客体被指定为一定的敏感级别。 访问控制的两个关键规则是: 不向上读和不向下写，即信息流只能从低安全级向高安全级流动。任何违反非循环信息流的行为都是被禁止的。

MAC 起初主要用于军方的应用中，并且常与 DAC 结合使用，主体只有通过了 DAC 与 MAC 的检查后，才能访问某个客体。 由于 MAC 对客体施加了更严格的访问控制，因而可以防止特洛伊木马之类的程序偷窃，同时 MAC 对用户意外泄漏机密信息也有预防能力。 但如果用户恶意泄漏信息，则可能无能为力。由于 MAC 增加了不能回避的访问限制，因而影响了系统的灵活性; 另一方面，虽然 MAC 增强了信息的机密性，但不能实施完整性控制; 再者网上信息更需要完整性，否则会影响 MAC 的网上应用。 在 MAC 系统中实现单向信息流的前提是系统中不存在逆向潜信道。逆向潜信道的存在会导致信息违反规则的流动。但现代计算机系统中这种潜信道是难以去除的，如大量的共享存储器以及为提升硬件性能而采用的各种 Cache 等，这给系统增加了安全隐患。

3 基于角色的访问控制

随着网络的 发展，特别是 Internet 的 广 泛 应用，使网上信息的完整性要求超过了机密性，而传统的 DAC- MAC 策略难以提供这方面的支持。 90 年代以来, NIST(National Institute of Standards andTechnology) 提 出 了 RBAC (Role - Based AccessControl) 的概念并被人们广泛接受。 RBAC 的突出优点是简化了各种环境下的授权管理。 在 DAC /MAC 系统中，访问权限直

接授予用户，但系统中的用户数量众多且经常变动，这增加了授权管理的复杂性。RBAC 的思想是将访问权限分配给角色,系统的用户担任一定的角色，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。 虽然 RBAC 仍处于发展阶段,但已经在某些系统中得到了应用，例如通过 x.509 证书来实现对用户身份的认证，把用户和密钥结合起来，在验证用户身份的同时，实现基于角色的访问控制。

3.1 RBAC 的基本概念

(1) 主体(Subject) : 可以对其它实体实施操作的主动实体。通常是系统用户或代理用户行为的进程。

(2) 客体(Object) : 接受其 它实体 动 作 的 被 动实体。 通常是可以识别的系统资源，如文件。 一个实体在某一时刻是主体而在另一时刻又可能成为客体，这取决于该实体是动作的执行者还是承受者。

(3) 用户(User) : 企图使用系统的人员。每个用户都有一个唯一的用户标识(UID) ，当注册进入系统时，用户要提供其 UID，系统进行用户身份认证以确证用户身份。

(4) 角色(Role) : 是系统中一组职责和权限的集合。角色的划分涉及组织内部的岗位职责和安全策略的综合考虑。

(5) 访问权限(Permission) : 在受系统保护的客体上执行某一操作许可。 在客体上能够执行的操作常与系统的类型有关，这是 RBAC 系统复杂性的一个重要方面。

(6) 用户角色分配(User- to- Role Assignment) :为用户分配一定的角色，即建立用户与角色的多对多关系。

(7) 角色权限分配(Permission - to - RoleAssignment): 为角色分配一组访问权限, 即建立角色与访问权限的多对多关系。 这样通过角色把用户与访问权限联系起来。 用户具有其所属诸角色的访问权限的总和。

(8) 会话(Session): 在特定环境下一个用户与一组角色的映射, 即用户为完成某项任务而激活其所属角色的一个子集, 激活角色权限的并集即为该用户当前有效的访问权限。

3.2 RBAC96 模型

RBAC96 模型是 Sandhu 等人提出的一个 RBAC 模型簇, 共包括四个模型。 Sandhu 等人认为 RBAC 是一个内涵广泛的概念, 难以用一个模型全面地描述。 RBAC0 是基本模型, 描述任何支持 RBAC 系统的最小要求。 RBAC0 包含四个基本要素: 用户、角色、会话和访问权限。 用户在一次会话中激活所属角色的一个子集获得一组访问权限即可对相关的客体执行规定的操作, 任何非显式授予的权限都是被禁止的。

RBAC1 是对 RBAC0 的扩充, 增加了角色等级的概念。 实际组织中职权重叠现象的客观存在为角色等级实施提供了条件。 通过角色等级, 上级角色继承下级角色的访问权限, 再被授予自身特有的权限而构成该角色的全部权限, 这极大地方便了权限管理。譬如: 销售部经理应具有销售部职员的访问权限, 同时还应具有普通职员不具备的权限, 如制定和修改销售计划, 考核每个销售员的业绩等。

RBAC2 也是 RBAC0 的扩充, 但与 RBAC1 不同, RBAC2 加进了约束的概念。约束机制久已有之, 如在一个组织中会计和出纳不能由同一个人担当(称为职责分离)。 RBAC2 中的约束规则主要有:

(1) 最小权限: 用户被分配的权限应是完成其职责所需的最少权限, 否则会导致权力的滥用。

(2) 互斥角色: 组织中的有些角色是互斥的, 一个用户最多只能属于一组互斥角色中的某一个, 否则会破坏职责分离, 如上面提到的会计和出纳。 权限分配也有互斥约束, 同一权限只能授予互斥角色中的某一个。

(3) 基数约束与角色容量: 分配给一个用户的角色数目以及一个角色拥有的权限数目都可以作为安全策略加以限制, 称作基数约束。 一个角色对应的用户数也有限制, 如总经理角色只能由一人担当, 这是角色容量。

(4) 先决条件: 一个用户要获得某一角色时必须具备某些条件, 如总会计必须是会计。 同理, 一个角色必须先拥有某一权限后才能获得另一权限,如在文件系统中先有读目录的权限后才能拥有写文件的权限。

RBAC3 是 RBAC1 和 RBAC2 的结合。 将角色等级与基数约束结合起来就产生了等级结构上的约束:

(1) 等级间的基数约束: 给定角色的父角色(直接上级) 或子角色(直接下级) 的数量限制。

(2) 等级间的互斥角色: 两个给定角色是否可以有共同的上级角色或下级角色, 特别是两件互斥角色是否可以有共同的上级角色, 如在一个项目小组中程序员和测试员是互斥角色, 那么项目主管角色如何解释(它是程序员和测试员的上级) 。

3.3 ARBAC97 模型

RBAC96 模型假定系统中只有一个安全管理员(SO) 进行系统安全策略设计和管理。 大型系统中用户和角色数量众多, 单靠一个 SO 是不现实的, 通常的

做法是指定一组 SO, 如有首席安全员(CSO) 、系统级安全员(SSO) 、部门级安全员(DSO)等。 因此又提出了 RBAC96 的管理模型 ARBAC97(Administration RBAC Model) 。

在 ARBAC97 中角色分为常规角色和管理角色, 二者是互斥的。 管理角色也具有等级结构和权限继承。 访问权限分为常规权限和管理权限, 两者也是互斥的。 ARBAC97 包括三个组成部分:

(1) 用户—角色分配管理(User - RoleAssignment: URA97)

描述管理角色如何实施常规角色即用户成员的分配与撤消问题。 成员分配常常会涉及先决条件问题, 如工程部 SO 只能在本部门内分配用户角色, 而被分配的用户必须是工程部的职员(哪些职员属于工程部由更高级的 SO 分配) , 这是一个先决条件。 用户成员的撤消简单得多, 如部门 SO 可以依据部门安全策略在本部门内任意撤消角色的用户, 但这种撤消是一种弱撤消。如用户 M 是角色A 和 B 的成员, 同时 B 是 A 的上级角色, 假如 SO撤消了 M 在 A 上的用户成员关系, 那么通过继承M 仍然具有 A 的权限。 要实现强撤消可以采用级联撤消, 即从指定角色及其所有上级角色中撤消指定用户。 但若某个上级角色超出了此 SO 的管理范围, 又会出现新的问题。

(2) 权限—角色分配管理 (Permission- RoleAssignment: PRA97)

讨论常规角色访问权限 的分配与撤 消问题。从角色的角度看访问权限与用户具有 对称性, 通过角色关联, 可以看出, 权限—角色分配与用户—角色分配具有相似的特点, 可以通过类似的办法来处理, 但权限的级联撤消是沿角色等级结构向下级联的。

(3) 角色—角色分配管理 (Role - RoleAssignment: RRA97)

讨论常规角色的角色成员分配规则以构成角色等级的问题。 为了便于讨论将角色分成以下三种类型:

能力角色: 只有访问权限成员或其它能力角色成员的角色, 即没有用户成员。

组角色: 只有用户成员或其它组角色成员的角色, 即没有权限成员。

用户—权限角色: 成员类型不受限制的角色。

这样区分是由建立角色之间关系的管理模型决定的。 我们分析一下能力角色, 它实际上是一组必须同时授予某一角色的访问权限的集合。 因为有的操作需要用户同时具备多项权限, 缺一不可,为了管理方便, 将这组权限提取为能力角色, 且禁止为其分配任何用户。 同样, 对于组角色, 它实际是应同时分配给某一角色的一组用户, 它们形成一个团队共同完成某一任务, 将它们抽象称为一个组角色, 禁止为其分配权限。基于这种思想, PRA97 可用于能力—角色分配管理(Abilities - RoleAssignment: ARA97) , URA97 可用于组—角色分配(Group- Role Assignment: GRA97) 。

三者的关系可以这样理解, 能力角色只能用能力角色作为其子角色, 可以用能力角色或用户—权限角色作为父角色, 用组角色或用户—权限角色作为子角色, 管理角色可以在自己的管辖范围内(相对于常规角色等级而言) 进行系统要素的创建、修改、删除等管理活动。

4 NIST RBAC 建议标准

2001 年 8 月 , NIST 发表了 RBAC 建议标准 。此建议标准综合了该领域众多研究者 的共识, 包括两个部分: RBAC 参考模型(the RBAC ReferenceModel) 和功能规范 (the RBAC FunctionalSpecification) 。 参考模型定义了 RBAC 的通用术语和模型构件, 并且界定了标准所讨论的 RBAC 领域范

围，功能规范定义了 RBAC 的管理操作。 它们均包括如下四个部分:

4.1 基本 RBA（C Core RBAC)

包括任何 RBAC 系统都应具有的要素，如用户、角色、权限、会话等。 基本思想是通过角色建立用户和访问权限的多对多关系，用户由此获得访问权限。

4.2 等级 RBA（C Hierarchical RBAC)

在基本 RBAC 上增加对角色等级的支持。角色等级是一个严格意义上的半序关系，上级角色继承下级角色的权限，下级角色获得上级角色的用户。根据半序关系中有无限制又可分为通用等级的 RBAC 和有限等级的 RBAC 两种。 通用等级的 RBAC 支持任意的半序关系。 对于有限等级的 RBAC，可在半序关系中加入某 种限制，一般 是使等级结构趋于简单，如使其成为树结构。

4.3 静态职责分离(SSD: Static Separation of Duties)

用于解决角色系统中潜在的利益冲突(Conflict of Interest) 。利益冲突源于用户被授予相互冲突的角色。 一种解决办法是在分配用户时实施限制，如禁止为一个用户同时分配一组互斥的角色。 考虑到等级结构的影响可分为两种情形: 基本静态职责分离和等级结构中的静态职责分离。

4.4 动态职责分离(DSD: Dynamic Separation ofDuties)

与 SSD 类似，DSD 也是限制可提供给用户的访问权限，但实施的机制不同。DSD 在用户会话中对可激活的当前角色进行限制，用户可被授予多个角色，包括有冲突的角色，但它们不能在同一个会话中被激活。DSD 约束可视作一个二元组(roleset, n) ，表示任何用户在某个角色子集中不能同时激活 n 个以上的角色。 DSD 是最小权限原则的扩展，每个用户根据其执行的任务可以在不同的环境下拥有不同级别的访问权限, DSD 确保访问权限不会在时间上

超越它们对履行职责的必要性，这种机制称作信任的适时变更。

在具体实现一个 RBAC 系统时，除了必须有基本的 RBAC 构件外，其它构件可根据应用的需要取舍，因此参考模型具有较大的弹性。

在上述每一个构件中都定义了相应的功能规范，将抽象的模型概念映射为可以提供的管理操作、会话管理以及管理审查。 RBAC 功能规范定义了用于创建和维护 RBAC 模型构件和提供系统支持的各种功能原型。 可以分为如下三类:

管理功能: 用于创建和维护构成 RBAC 模型构件的各种系统要素及相互关系。

系统支持功能: 用于在用户与系统交互时支持 RBAC 模型构建的各种功能，如建立会话、添加和去除活跃角色、确定访问、决定逻辑等。

审查功能: 用于审查由管理功能和系统支持功能所产生的各种活动的结果。

RBAC 的目的是简化安全策略，管理并提供弹性的、个性化的安全政策。 从思想上讲, RBAC 是目前最为深入的访问控制方法，但由于提出的时间较晚，在理论上尚未达成共识，也没有制定统一的标准。但这似乎并不影响它的应用，已有许多厂商开始提供基于 RBAC 的解决方案，呈现出理论与应用同步发展的态势。 这一状况必然要求尽快制定通用的标准, NIST 综合众多学者的观点并参考了许多厂商的产品提出了这个建议标准，旨在提供一个权威的、广泛接受的、可用的 RBAC 参考规 范,为进一步研究指明方向。 当然，这个建议标准只描述了 RBAC 系统最基本的特征，在实际应用中可以在此基础上扩展其它更强的访问控制功能。

5 结论

本文介绍并分析了三种访问控制模型：DAC基于访问者或访问组的身份进行访问控制，MAC基于信息敏感度实现访问控制，RBAC模型根据用户的角色获得相应访问权限。RBAC方法比DAC和MAC更易于实现安全访问控制的管理。事实上，RBAC模型并不是完全独立于传统的访问控制模型，甚至在某种程度上可以说，RBAC是DAC和MAC在应用范围、有效性和灵活性方面的扩展。利用RBAC96模型就可以实现多种DAC和MAC。由于使用了角色继承、约束、角色管理、授权管理等机制，使得访问控制的实现和管理更加灵活。

访问控制是个古老而又新颖、简单而又复杂的课题，本文主要介绍了三种主流的访问控制方法,除此之外，还有一些访问控制方法，如：基于组、基于任务、基于所有者（DAC的一种简化情形）等访问控制方法，这些方法由于应用很少，这里就不再介绍。网络技术的发展使访问控制技术的研究成为热点；应用系统的多样性又决定了访问控制的复杂性，很难统一到一个简单的标准之内。由于RBAC能够很好地适应实际组织的安全策略,具有很好的灵活性，能够减轻系统安全管理的负担，因此必将得到更加广泛的应用。