文章编号:1001 - 9081(2009)10 - 2809 - 03

面向用户角色的细粒度自主访问控制机制

魏立峰,孟凯凯,何连跃
(国防科学技术大学 计算机学院,长沙 410073)
(wei_lifeng@yahoo.com.cn)

摘 要:基于访问控制表(ACL)的细粒度自主访问控制机制可以实现针对单个用户或用户组的访问授权,但是在实际使用中可能造成不适当授权或权限撤销不及时的缺陷。基于可信 Kylin 操作系统的角色定权(RBA)机制,在自主授权中引入了用户角色约束,提出了一种面向用户角色的细粒度自主访问控制机制,实现了针对单个用户在承担特定角色时的访问授权,一旦用户不再承担该角色,访问授权可以及时撤销,有效解决了 ACL 不适当授权的问题。

关键词:自主访问控制;访问控制表;角色;授权
中图分类号: TP309 **文献标志码**:A

Fine-granularity discretionary access control based on user's role

WEI Li-feng, MENG Kai-kai, HE Lian-yue
(*College of Computer Science, National University of Defense Technology, Changsha Hunan* 410073, *China*)

Abstract: Fine-granularity discretionary access control based on Access Control List (ACL) may grant authority to one user or group, but it may grant unapt authority or remove authority not timely. Based on Role-Based Authorization (RBA) of trusted Kylin OS, introducing user's role restriction in discretionary access control, a fine-granularity discretionary access control mechanism to user's role was given. It can which may grant authority to the user with special role, and remove authority from the user without special role, therefore solves the problem of granting unapt authority.

Key words: Discretionary Access Control (DAC); Access Control List (ACL); role; authorization

0 引言

自主访问控制是安全操作系统的一项基本访问控制功能,其特点就是授权的自主性,并基于一种访问控制规则实现主体对客体的访问。用户对信息的控制是基于用户身份的鉴别和存取访问规则的确定。客体的拥有者可以自主地给其他用户或用户组进行访问授权,能直接或间接地将访问权或访问权的某些子集授予其他主体。最典型的自主访问控制是 Linux/Unix 的 UGO 权限管理方式,它是一种粗粒度的按照"拥有者"、"拥有者用户组"和"其他"进行授权的 9 位权限控制方式。为了实现一些比较复杂的权限管理,IEEE POSIX 1003.1e 定义了访问控制表(Access Control List,ACL)标准。ACL 是一个文件/目录的访问控制列表,可以针对任意指定的用户或组进行访问授权,从而实现细粒度的自主访问控制。

基于角色的访问控制(Role-Based Access Control,RBAC)模型在用户和权限之间引入角色,角色与权限关联,用户与角色关联,从而大大降低了系统的复杂度,同时 RBAC 体现了系统的组织结构,简洁并具有灵活性,大大降低了系统管理员误操作的可能性。文献[1]作者论述了基于角色的自主访问控制和强制访问控制,讨论了用 RBAC 实现强制访问控制(Mandatory Access Control, MAC)和自主访问控制(Discretionary Access Control,DAC)的方法,讨论了角色和自主访问控制结合方法,针对三种 DAC 类型,设计了文件管理角色和正规角色。实际上与一般的 ACL 没有太大差别,仅仅是将授予相同权限的用户集定义为角色,同一角色的用户具有相同的权限。

通过自主访问控制,用户一旦被授予对某客体的访问权限,就能按照许可随意访问该客体,没有清晰地表达出访问权限可适用的范围。客体的拥有者在赋予其他用户访问权限时,通常是带有某种限制条件的,在限制条件不满足时,权限应及时撤销。在授权时引入时间限制[2-5],可以保证只在给定的时间段内保持有效的授权机制。但是仅仅引入时间机制,并不能完全保证权限的可适用范围。客体的拥有者在为其他用户进行自主授权时,通常是因为被授权的用户具有某种职位才被授权,一旦其不再承担该职务时,权限应及时撤销。由于基于 ACL 的自主访问控制仅仅针对用户 ID 或组 ID 进行访问授权,因此其可能存在权限没有及时撤销的问题。同时,如果原来不少用户针对某特定 uid 进行了访问控制授权,因为某些原因需要统一变更对该 uid 用户的授权,传统的 DAC、ACL 等操作繁琐,因此可能会出现不适当授权现象。

为了解决不适当授权或授权撤销不及时的问题,本文在可信 Kylin 角色定权的基础上,提出了面向角色的访问控制列表(Role Access Control List,RACL)机制,通过在 ACL 中引入针对用户角色的授权,实现了针对用户授权的角色约束,使得客体的拥有者可以保证只有某用户在承担某角色时,该用户才能获得相应的授权。

1 可信 Kylin 角色定权机制

自主访问控制本身的安全性不高,用户可能因为疏忽或恶意导致敏感信息的泄漏,甚至系统一旦遭受缓冲区溢出攻

收稿日期:2009 - 04 - 17。

作者简介:魏立峰(1973 -),男,山东聊城人,副研究员,博士,主要研究方向:信息安全、系统软件; 孟凯凯(1984 -),男,浙江义乌人,硕士研究生,主要研究方向:信息安全; 何连跃(1971 -),男,浙江金华人,副研究员,博士,主要研究方向:计算机安全、系统软件。

击,系统将被完全控制,因此单纯依靠自主访问控制不能完全满足敏感信息保护的要求。强制访问控制是一种在系统范围内实施的对主体访问客体进行访问控制检查的访问控制机制。访问控制检查的依据是主体和客体的安全属性标记,最终是否允许访问是由系统实现的各安全策略决定,任何用户都无法改变控制规则。

可信 Kylin 操作系统提出了角色定权技术,并设计实现了基于角色定权的强制访问控制框架 KACF。KACF 将访问控制实施与访问控制决策分离,访问控制实施部分由框架本身完成,并在涉及访问控制标记和决策的关键点提供一系列的钩子函数,通过这些钩子函数,由访问控制决策部分完成访问控制逻辑。通过访问控制框架,极大简化了访问控制策略的开发,支持多访问控制策略。在基于角色定权的访问控制框架下,可信 Kylin 操作系统研究和实现了不同安全目的强制访问控制策略,包括:体现最小权限管理的进程权能控制策略;实现安全域隔离的基于类型的访问控制策略,实现信息机密性保护的多级安全策略,实现信息完整性保护的完整性控制策略等。

可信 Kylin 操作系统的角色定权机制将传统主流操作系统的 root 的管理功能分解为不同管理权限,分配给不同的角色,实现管理员分权。系统缺省包含系统管理员、安全管理员、安全审计管理员和一般用户 4 种角色。其中系统管理员角色负责系统的用户管理、网络管理等日常管理工作等;安全管理员负责与安全配置、管理相关的工作,包括:角色创建与删除、角色权限设置与修改、用户角色设置与修改、文件安全属性的修改等;安全审计管理员负责安全审计工作,缺省角色是系统中具有最小权限的角色。当没有为用户明确关联角色时,则系统为用户关联一般用户角色。

角色定权技术通过角色与权限关联、用户与角色关联,实现对用户权限的设置,最终实现对主体(进程)的权限设置。安全管理员可以在系统使用过程中根据需要创建新的系统角色,比如创建特权不同的角色,满足角色特权需求,创建不同安全级的角色,满足角色安全级需求。通过角色定权技术实现了用户权限设置的灵活性。

2 ACL 授权分析

在实际生活中,角色常常对应职位。假设一个部门拥有多个职位,每个职位对应一个角色,比如普通员工具有一个“员工角色”,部门经理具有一个“经理角色”,部门经理助理具有一个“经理助理角色”,部门小组 A 具有一个“部门组 A 角色”,部门小组 B 具有一个“部门组 B 角色”,部门 A 组组长具有“部门 A 组组长角色”,部门 B 组组长具有“部门 B 组组长角色”。同一部门内的文件可能需要共享,部门之间也可能需要共享。

A 组、B 组之间员工可能互相调动,也可能职位升迁。随着职位的变动,其角色也发生变化,访问控制权限也应发生变化。比如:某用户 u 由部门 A 组变为部门 B 组,其角色发生变化,其访问权限也应发生变化,其在 A 组时的原有权限可能减少,在 B 组的权限可能增加;再比如,部门经理助理换人,那么新任经理助理和离任经理助理权限都发生了变化。

针对以上情况,传统的自主访问控制可能需要由系统管理员更改用户 u 的用户组,如果原来有不少用户针对用户 u 进行了访问控制授权,那么可能需要统一变更对该用户 u 的授权。这样的操作相当繁琐且可能会出现授权没有及时变更的现象。

正是由于职位变迁、角色变化导致权限变化,因此在进行自主授权时,引入用户角色约束,那么用户角色发生变化时,授权自然改变,方便及时。

3 面向用户角色的 RACL 访问控制机制

3.1 RACL 机制

结合可信 Kylin 操作系统的角色定权机制,在自主授权时引入用户角色,使得客体的拥有者可以针对用户角色进行授权,做到不仅可以针对单个用户或用户组进行访问授权,还可以实现针对具有特定角色的单个用户进行授权。通过 RACL 机制,实现用户自主授权的角色约束,使得被授权的用户在不满足角色要求时,可以及时撤销被授予的访问权限。

3.2 RACL 数据结构

面向用户角色的访问控制列表(RACL)是客体的一个 DAC 实体,包含一个项目列表,其中每一项是标识符(如:用户、用户组或用户角色)和相应访问许可权集合。RACL 通常包含两种 RACL 类型,分别是访问 RACL(access RACL)和缺省 RACL(default access RACL)。访问 RACL 可以和文件、目录、连接等文件系统客体关联,控制对它们的访问许可权。缺省 RACL 仅仅和目录关联,在具有缺省 RACL 的目录下创建的客体将把父目录的缺省 RACL 继承为自己的访问 RACL。

RACL 入口项数据结构定义为:

```
struct posix_acl_entry {
  short e_tag;
  unsigned short e_perm;
  unsigned int e_id;
  unsigned int e_rid;
};
```

其中:e_tag 表示具有许可权的对象,也就是 ACL 的标志类型;e_perm 表示许可权;e_id 表示用户或组的 id,e_rid 表示用户角色的 id。e_tag 的取值范围如表 1 所示。

表 1 RACL 标志类型取值

标志类型	含义
ACL_USER_OBJ	指定文件属主的访问许可权
ACL_GROUP_OBJ	指定文件属组的访问许可权
ACL_USER	指定由 e_id 所标识的用户的访问许可权
ACL_USER_ROLE	指定由 e_id 所标识的用户在承担 e_rid 所标识角色时的访问许可权
ACL_ROLE	指定由 e_rid 所标识角色的访问许可权
ACL_GROUP	指定由 e_id 所标识的组的访问许可权
ACL_OTHER	指定那些在该 ACL 中没有任何其他 ACL 项与之匹配的进程访问许可权
ACL_MASK	指定 ACL_USER、ACL_GROUP_OBJ 或 ACL_GROUP 项所能赋予的最大访问许可权

3.3 RACL 访问控制检查算法

访问控制检查算法决定是否赋予主体访问客体的权力,检查算法采用首次匹配决定进程访问客体的许可权。如图 1 所示,RACL 按照如下顺序进行访问控制检查。

1) 如果进程的有效 UID 和文件属主的 UID 匹配,那么如

果 ACL_USER_OBJ 项包含请求的许可权,则允许访问;否则拒绝访问。

2）如果进程的有效 UID 和其中的一个 ACL_USER_ROLE 项的 e_id 标识的用户 UID 相匹配,则说明针对该用户在承担特定角色的情况下进行了授权。检查规则为:如果进程的活跃角色与该 ACL_USER_ROLE 项的 e_rid 标识的角色 ID 相匹配,那么如果匹配的 ACL_USER_ROLE 项和 ACL_MASK 项都包含请求的许可权,则允许访问;否则拒绝访问。

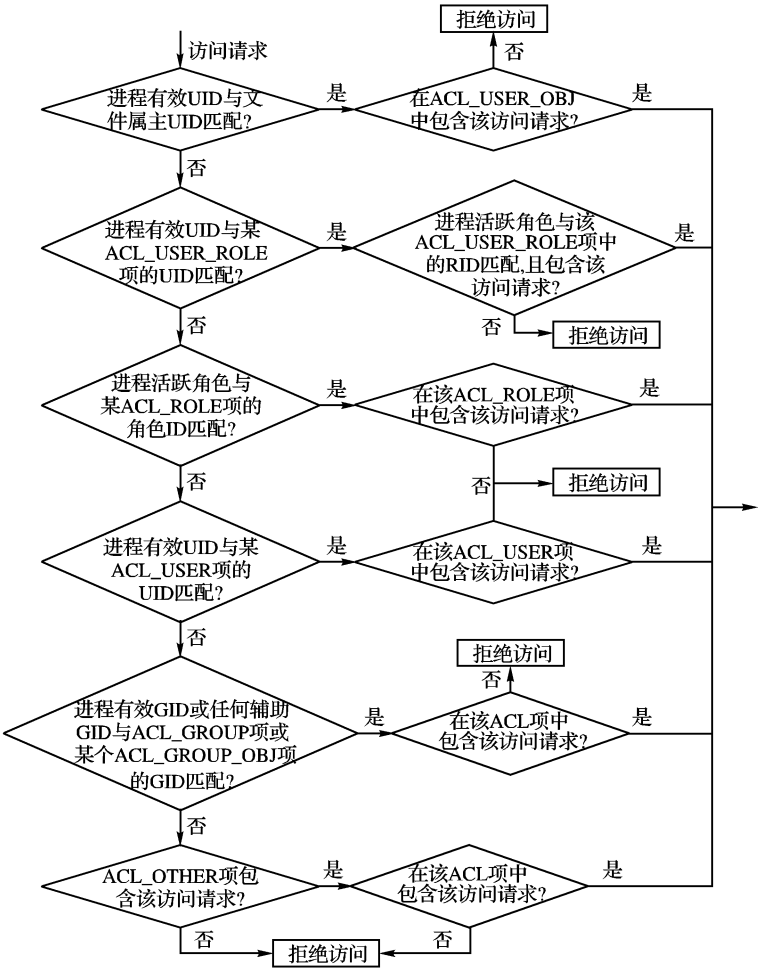

图 1　RACL 访问控制检查算法

3）如果进程的活跃角色和其中的一个 ACL_ROLE 项的 e_rid 标识的角色 ID 相匹配,那么如果匹配的 ACL_ ROLE 项和 ACL_MASK 项都包含请求的许可权,则允许访问;否则拒绝访问。

4）如果进程的有效 UID 和其中的一个 ACL_USER 项的 e_id 标识的用户 UID 相匹配,那么如果匹配的 ACL_USER 项和 ACL_MASK 项都包含请求的许可权,则允许访问;否则拒绝访问。

5）如果进程的有效 GID 或任何辅助 GID 和其中的 ACL_GROUP 项或 ACL_GROUP_OBJ 项的 e_id 标识的 GID 项匹配,那么如果匹配的 ACL_GROUP 项或 ACL_GROUP_OBJ 项和 ACL_MASK 项都包含请求的许可权,则允许访问;否则拒绝访问。

6）如果 ALL_OTHER_OBJ 和 ACL_MASK 项包含请求的许可权,则允许访问。

7）否则拒绝访问。

4　结语

基于 ACL 的细粒度自主访问控制可以针对单个用户或用户组进行访问自主授权。通常情况下,客体的拥有者在为其他用户进行授权时,是因为被授权的用户承担了某种职务或者具有某种特殊身份才为其进行访问授权,一旦该用户不再承担该职务或不再具备该特殊身份时,先前的访问授权应及时撤销。本文分析了 ACL 在不适当授权或授权没有及时撤销方面的不足,结合可信 Kylin 的角色定权机制,提出了一种面向用户角色进行自主授权的细粒度自主访问控制 RACL 机制,该机制有效解决了 ACL 不适当授权或授权没有及时撤销的问题。

参考文献:

[1] BERTINO E, BETTINI C, FERRARI E, *et al*. A temporal access control mechanism for database systems[J]. IEEE Transactions on Knowledge and Data Engineering, 1996, 8(1):67 - 80.

[2] BERTINO E, BETTINI C, FERRARI E, *et al*. An access control model supporting periodicity constraints and temporal reasoning[J]. ACM Transactions on Database Systems, 1998, 23(3):213 - 285.

[3] 张宏,贺也平,石志国.基于周期时间限制的自主访问控制委托模型[J].计算机学报,2006,8(29):1427 - 1436.

[4] 谭良,周明天.带时间特性的自主访问控制政策及其在 Linux 上的设计与实现[J].计算机应用,2006,12(26):2906 - 2909.