混合 RBAC-DTE 策略的多角色管理

唐柳英[1),2),4)] 卿斯汉[2),3),4)]

[1)](中国科学院软件研究所基础软件国家工程研究中心 北京 100080)

[2)](中国科学院软件研究所信息安全技术工程研究中心 北京 100080)

[3)](北京中科安胜信息技术有限公司 北京 100086)

[4)](中国科学院研究生院 北京 100049)

摘 要 混合的基于角色访问控制-域型增强(RBAC-DTE)访问控制模型因其不同层次的保护机制近年来颇受关注,但是尚未见到公开的文献讨论混合 RBAC-DTE 策略中的多角色管理问题. 因此,从特权层面和访问许可权层面上,提出了一种角色划分粒度比域划分粒度粗的角色和域的划分方法,并引入域的静态继承关系. 这种混合 RBAC-DTE 策略的多角色管理方法解决了不同域的进程共享访问许可权集、控制策略代码尺寸的问题,特别是它可以充分支持极小特权原则.

关键词 安全策略;混合 RBAC-DTE 访问控制模型;极小特权原则;多角色管理;Capability 机制

中图法分类号 TP309

Administration of Multiple Roles in the Hybrid RBAC-DTE Policy

TANG Liu-Ying[1),2),4)] QING Si-Han[2),3),4)]

[1)](*National Engineering Research Center for Fundamental Software, Institute of Software, Chinese Academy of Sciences, Beijing* 100080)

[2)](*Engineering Research Center for Information Security Technology, Institute of Software, Chinese Academy of Sciences, Beijing* 100080)

[3)](*Beijing ZhongkeAnsheng Corporation of Information Technology, Beijing* 100086)

[4)](*Graduate University of Chinese Academy of Sciences, Beijing* 100049)

Abstract The hybrid Role Based Access Control-Domain and Type Enforcement (RBAC-DTE) access control model has recently been given much attention due to its different level of protect mechanisms. But no published literature has discussed administration of multiple roles in the hybrid RBAC-DTE policy. From the aspect of privilege and access right, this paper proposes an approach to dividing roles and domains that roles are more coarse-grained than domains, and introduces a static-inheritance relationship between domains. This method for multirole administration in the hybrid RBAC-DTE policy solves the problem of sharing access right set among processes in different domains and the problem of controlling policy code size, especially, supports the principle of least privilege sufficiently.

Keywords security policy; hybrid RBAC-DTE access control model; principle of least privilege; administration of multiple roles; capability mechanism

---

收稿日期:2006-03-31;**修改稿收到日期**:2006-06-02. **本课题得到北京市自然科学基金**(4052016)、**国家自然科学基金**(60573042)**和国家"九七三"重点基础研究发展规划项目基金**(G1999035802)**资助**.**唐柳英**,**女**,1973 **年生**,**博士研究生**,**工程师**,**目前主要从事操作系统安全与形式化开发方法的研究工作**. E-mail: tangly@ercist.iscas.ac.cn. **卿斯汉**,**男**,1939 **年生**,**研究员**,**博士生导师**,**主要研究领域为信息系统安全理论和技术**.

1 引 言

针对不同站点的安全目标所采取的安全策略有所不同.安全策略模型是安全策略的简单、抽象、无歧义的描述,通常为访问控制模型. RBAC(Role-based Access Control)策略模型[1]源于许多组织的安全需求,通过建立用户与角色、角色与操作的关系来控制信息的访问,是一种高层次抽象模型.而 DTE(Domain and Type Enforcement)策略模型[2~4]是将系统的主体(如进程)和域(domain)关联起来、系统的客体(如文件)和型(type)关联起来,基于域和型这两种安全属性来限制主体对客体、主体对主体的访问.跨不同域的操作必须经过成功的域转换,域转换包括自动转换(auto 域转换)和按请求转换(exec 域转换)两种方式,exec 域转换需要经过认证.域的隔离作用限制了进程的活动范围.与一个主体(用户进程)相关的访问集可能只是这个主体的角色职责范围内所必要的访问集的一个小子集. Hoffman 1997 年描述了在一个由 SCC(Secure Computing Corporation)开发的 TE(Type Enforced)操作系统 LOCK6 上实现的 RBAC 策略[5],首次提出将 RBAC 模型和 DTE 模型结合起来. TE 模型(DTE 模型是 TE 模型的扩展)被看成是介于角色和操作之间的一个额外的抽象层.传统的 RBAC 策略将角色和操作直接联系起来,而 Hoffman 提出了将角色和域(domain)、域和操作连接在一起的新观点,即在角色和操作之间引入了一个新的抽象层. Hoffman 最后得出了对于一个安全系统而言单一的 RBAC 并不是一种充分的保护机制的结论.根据 Hoffman 的思想我们可以将极小特权划分为高层次上的用户极小特权和低层次上的主体操作极小特权:通过划分角色满足用户极小特权原则;依据角色的职责确定必要的主体和域,再细分相应的域和型来满足主体操作极小特权原则.

Chandramouli 于 2001 年给出了一个支持多种认证类型的动态认证框架 DAFMAT[6]. DAFMAT 由混合 RBAC-DTE 访问控制模型和逻辑推导认证两部分组成,前者与 Hoffman 的方法一致. Chandramouli 同样认为 RBAC 作为高层模型要通过一个中间层在较低层访问控制机制(例如 Unix 中的许可权比特位)上实现它的抽象概念,而 DTE 模型可被选择作为这样的中间结构. DAFMAT 将讨论集中在应用系统(如健康医疗应用系统)而非操作系统,它充分描述了混合模型中实体间的映射关系,包括用户到角色的多对一映射、角色到域的多对一映射、主体到角色的多对多映射、主体到域的多对一映射等等.美国国家安全局(NSA)2001 年发布的 Security-Enhanced Linux(SELinux)系统[7],基于支持多安全策略的 Flask 架构采用 TE 模型与 RBAC 模型结合一起的安全模型[8]. SELinux 的发布已经表明了解决当前操作系统的保护机制不能充分支持机密性和完整性要求的可行性.季庆光等人 2004 年提出的基于权能、角色及 DTE 的特权控制模型 PCM_RBPC[9]将实现极小特权原理分为三层:顶层即管理层——RBAC 模型;中层即功能控制层——DTE 模型;底层即执行层——POSIX 权能机制.

混合 RBAC-DTE 访问控制模型能够提供给系统更充分的保护勿庸置疑.混合 RBAC-DTE 模型中,在角色和操作之间加入了中间层——域来实现底层的控制.一个角色的有效的普通访问许可权就是该角色当前所进入的域的普通访问许可权,一个角色的有效特权通常也与所进入的域的特权有关.这表明在混合 RBAC-DTE 模型中没有单纯的角色定义与划分,角色定义与划分应该结合域的定义与划分.由于角色对域的这种依赖关系,当一个高等级的安全系统对角色的要求比较高以保持与极小特权原则一致时,即要求复杂的角色来各司其职,如果采用混合 RBAC-DTE 策略,多角色的管理将对域的划分提出更高的要求.然而,并没有文献讨论混合 RBAC-DTE 策略所面临的多角色管理的问题.不恰当地划分角色或者不恰当划分角色所进入的域将造成:(1)角色所属的域会提供给角色其职责之外的特权或访问许可权,违背了极小特权原则;(2)忽略角色的不完全互斥性,即忽略相关角色共享部分特权或部分访问许可权,会产生域定义的冗余,策略配置变得复杂而不利于策略的分析.角色(role)的划分粒度、域(domain)的划分粒度以及两者之间的相互影响在有多角色需求的相关系统中非常重要,本文从角色特权及角色的访问许可权出发,提出一种角色粗粒度划分、域细粒度划分的方法,然后引入域的静态继承关系来描述一种角色获得访问许可权的方法,这不仅使不同域的进程能够共享访问许可权集、策略实施的代码尺寸控制等问题得以解决,而且更符合极小特权原则.

本文在第 2 节描述高安全等级系统对多角色的需求;第 3 节讨论混合 RBAC-DTE 策略的多角色管理;第 4 节用域的静态继承进一步解决第 3 节中多角色管理仍然存在的冲突;最后,在第 5 节进行总

结并指出将来的研究工作.

2　多角色需求

DG/UX(Data General UNIX)系统[10]是由 Data General 公司开发的基于 Unix 的达到可信计算机评估标准《TCSEC》[11] B2 级的安全商用操作系统. 它将事件(event)定义为能作出安全相关决策的一段代码. 该系统认为有四种类型的事件：可操作事件、访问事件、访问超越事件和审计事件. 前三者与控制系统特权的 Capability 权能机制密切相关. 可操作事件检查一个进程是否有执行受限操作的适当特权，如 mount 一个文件系统或者增加一个新的用户. 访问事件依据访问监控器中的访问控制策略决定一个主体是否能以所要求的访问模式(如 read)来访问客体. 访问超越事件检查一个进程是否有适当的特权超越被一个系统访问控制策略拒绝的访问(相对访问事件而言). 一般地，如果配置了 Capability 机制，那么进程的特权可分为两类：可操作事件的特权(如 MOUNT，CHROOT)和访问超越事件的特权，后者包括 DAC 超越特权(如 CHOWN，DAC_WRITE_OVERRIDE)和 MAC 超越特权(如 MAC_READ_OVERRIDE). 在高等级安全系统中，系统的管理任务从由原来的仅有一个超级用户(如 root)来完成细分到由不同的管理角色来执行. 管理角色其实是一些特殊的用户，每个角色都拥有恰当的特权来执行特定的受限操作，即可操作事件；或拥有恰当的特权来超越特定的访问控制，即访问超越事件. 角色的特权集由可操作事件的特权或访问超越事件的特权组成. 不同的角色拥有不同的特权集，从而管理系统的不同部分. 普通用户角色只能使用系统非常受限的功能，并不拥有任何特权，对于一个安全系统而言，多角色通常情况下指的是多管理角色. DG/UX 系统提供了 9 个缺省的系统管理角色：软件安装员 installer、系统管理员 sysadmin、操作员 operator、安全管理员 secadmin、网络管理员 netadmin、审计管理员 audadmin、审计操作员 auditor、备份管理员 backup 和行式打印机管理员 lpadmin. 尽管 SELinux 系统样本策略的用户角色只有执行系统维护的系统管理员 sysadm_r、作为普通用户的系统管理员 staff_r 和普通用户角色 user_r，但是这个样本策略实际上只是提供该系统策略配置的一个框架，可以根据多角色的需求对策略作相应的定制.

不失一般性，本文将着重讨论高等级安全系统的 8 个基本的管理角色，它们都拥有各自不同的特权集来执行系统不同部分的管理任务：

(1)与安全策略相关的安全管理员 secadm_r 和安全操作员 secopt_r.

安全管理员 secadm_r 负责系统安全策略的定义和维护等；安全操作员 secopt_r 负责系统安全策略的加载和启动等.

(2)与系统功能相关的系统管理员 sysadm_r 和系统操作员 sysopt_r.

系统管理员 sysadm_r 负责维护用户文件、添加或删除用户等；系统操作员 sysopt_r 负责系统的日常维护工作，如 mount 或 unmount 一个文件系统等.

(3)与网络应用相关的网络管理员 netadm_r 和网络操作员 netopt_r.

网络管理员 netadm_r 负责系统中网络的配置和维护；网络操作员 netopt_r 负责系统中网络的启动和停止.

(4)与审计模块相关的审计管理员 adtadm_r 和审计操作员 adtopt_r.

审计管理员 audadm_r 负责审计的配置如定义审计的事件集；审计操作员 audopt_r 负责审计分析工作.

3　角色和域的相对划分粒度

关于混合 RBAC-DTE 策略中的角色特权、域特权、角色特权与域特权的结合以及域的配置，前三者主要牵涉 Capability 机制的角色特权(role privilege)分配；后者指的是通常 DTE 中的域定义表(DDT)和域交互表(DIT)[2]，与访问事件有关而与访问超越事件、可操作事件无关，也就是说与普通的访问许可权(access right)相关而与特权(privilege)无关. 一个角色进入一个域，就代表该角色的主体被授予这个域的访问许可权.

SELinux 样本策略 1.22[6]将一个角色能进入的域的特权集直接赋给该角色. 通过分析这个策略我们得到两种角色特权集的分配情况. 第 1 种情况，将一个初始角色的特权集与代表该角色的主体所属的域的特权集取并集得到一个有效角色的特权集，是集合的∪关系. 这种特权集的并关系可间接实现多角色，比如安全角色 sec_r 进入管理域 admin_d，就成为安全管理员 secadm_r，其特权为 sec_r 的特权集∪admin_d 的特权集；进入操作域 opt_d，就成为安全操作员 secopt_r，其特权为 sec_r 的特权集∪opt_d 的特权集. 但是，当一个安全系统配置了较

多的初始角色和域时，很难找到唯一的一个域特权集，等于多个关联初始角色的特权集的交集. 也就是说，n 个关联角色的特权集两两相交往往产生 c_n^2 个集合，例如安全员 sec_r、系统员 sys_r 和网络员 net_r 的特权集两两相交后得到 3 个交集. 这使得初始角色的特权集定义变得复杂，甚至造成需要配置更多的初始角色而所进入的域的特权集为空的局面. 以上的特权集合的并关系另一缺憾是，造成角色特权集的扩张. 当发生域转换时，主体进入目标域后，当前主体所代表的角色的特权集包含了源域的特权集，而且随着域转换的增加，角色的特权集就会加入越来越多的域特权集，明显地违背了用户极小特权原则. 第 2 种情况，一个角色的特权集就是代表该角色的主体所属域的特权集. 这种情况很难有效地实现多角色，以上的 8 个角色就需要定义相应的 8 个管理域，而且显然当发生域转换时满足不了不同的角色进入相同的目标域拥有不同的特权集，为实现多角色的职责隔离，策略的配置会因过于细致复杂而难以理解，从而导致系统的安全性降低.

与季庆光等人的基于权能、角色及 DTE 的特权控制模型 PCM_RBPC 相应的安全策略则是将初始角色的特权集与代表该角色的主体所属的域的特权集取交集来得到一个有效角色的特权集，是集合的 $\cap$ 关系. 这种特权集的交关系同样间接实现了多角色，如角色 sec_r 进入域 admin_d，就可得到安全管理员 secadm_r 的特权：sec_r 的特权集 $\cap$ admin_d 的特权集；进入域 opt_d，就可得到安全操作员 secopt_r 的特权：sec_r 的特权集 $\cap$ opt_d 的特权集. 特权集 $\cap$ 关系相对于特权集 $\cup$ 关系的特点是限制了有效角色的特权集. 当网络管理员 netadm_r 从域 admin_d 经auto 域转换进入到 ftp 服务这个域 ftpd_d 后，就可得到角色 netadm_ftpd_r，其特权为 netadm_r 特权集 $\cap$ ftpd_d 特权集；当网络操作员 netopt_r 从域 opt_d 经 auto 域转换进入域 ftpd_d 后，就可得到角色 netopt_ftpd_r，其特权为 netopt_r 特权集 $\cap$ ftpd_t 特权集；角色 netadm_ftpd_r 负责 ftpd 服务的管理和配置，角色 netopt_ftpd_r 负责 ftpd 服务的启动等. 假设定义 4 个初始管理角色，分别为安全员 sec_r、系统员 sys_r、网络员 net_r、审计员 adt_r 以及这些角色可以以 exec 域转换方式进入的管理层的域，即管理域 admin_d 和操作域 opt_d. 由以上的特权集 $\cap$ 关系，可以得到 8 个有效的管理角色：安全管理员 secadm_r、安全操作员 secopt_r、系统管理员 sysadm_r、系统操作员 sysopt_r、网络管理员 netadm_r、网络操作员 netopt_r、审计管理员 adtadm_r、审计操作员 adtopt_r.

尽管以上所采用角色细粒度划分＋域粗粒度划分的方法在特权分配上是有效的，却在某种程度上违背了域配置中访问许可权分配的主体极小特权原则. 由图 1，系统管理员 sysadm_r 除了与安全管理员 secadm_r 拥有不同的特权外，两者的访问许可权是一样的，都是依据域 admin_d 的 DDT 表和 DIT 表来决策，其中 DDT 表规定域对型的可允许的访问模式，包括读 r、写 w、执行 x 等，DIT 表规定域与域之间可允许的交互模式，包括域转换方式、信号通信. 这样如果域 admin_d 可以修改型为 secpolicy_t 的安全策略配置文件，那么本应该只能由安全管理员(所属域为 admin_d)修改的安全策略配置文件就能被系统管理员(同属于域 admin_d)作不恰当的修改. 这是不符合极小特权原则的. 由此，所有的 4 个管理员(secadm_r，sysadm_r，netadm_r，adtadm_r)两两存在访问许可权冲突，4 个操作员(secopt_r，sysopt_r，netopt_r，adtopt_r)也两两存在访问许可权冲突，共有 12 个冲突.

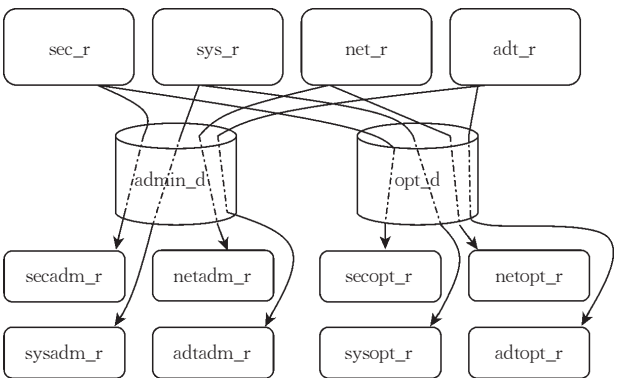


图 1 角色细粒度划分＋域粗粒度划分的方法

反之，相应于以上管理层配置，我们采用角色粗粒度划分＋域细粒度划分的方法定义两个初始管理角色，管理员 admin_r、操作员 opt_r 以及这两个角色可以以 exec 域转换方式进入的 4 个管理层的域，安全域 sec_d、系统域 sys_d、网络域 net_d 和审计域 adt_d. 同样由特权集 $\cap$ 关系也可得到以上的 8 个有效管理角色. 这两种方案的区别在于前者是 4 个角色进入 2 个域，而后者是 2 个角色进入到 4 个域. 前者造成域特权集很大，为 4 个角色的与域相关的特权集的并集；每个管理层域相关的 DDT 表项和 DIT 表项也很大(因为要考虑到 4 个角色的访问许可权)，造成该域中主体进程的活动范围很大. 而后者只是初始角色的特权集很大，为 4 个域的与角色相关的特权集的并集，但是实际发生作用的有效角色的特权集与前者是等价的，都只包含了有效角色

执行被授予任务的必要特权. 图 2 中操作员 opt_r 也可进入安全域 sec_d，那么安全员 secadm_r 和 secopt_r 之间因为相同的安全域 sec_d 存在访问许可权冲突. 类似地，系统员 sysadm_r 和 sysopt_r 存在冲突，网络员 netadm_r 和 netopt_r 之间存在冲突，审计员 adtadm_r 和 adtopt_r 之间存在冲突. 共有 4 个冲突. 角色粗粒度划分＋域细粒度划分的方法相比角色细粒度划分＋域粗粒度划分的方法，实现了同样的多角色，且执行效率是一样的，但是在安全性上，角色粗粒度划分＋域细粒度划分的方法由于域的细分要高于后者.

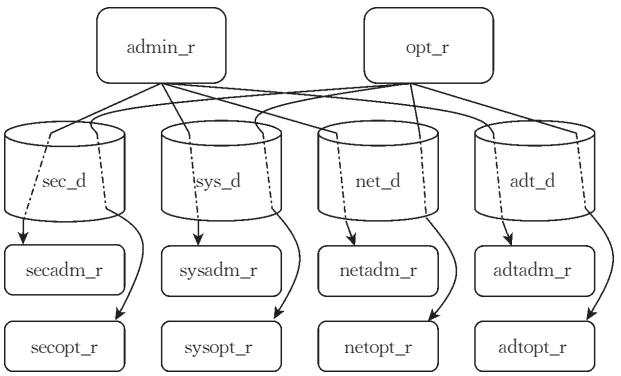


图 2　角色粗粒度划分＋域细粒度划分的方法

定理 1.　在采用角色特权集∩域特权集来获得有效特权集的混合 RBAC-DTE 安全策略的管理层配置中，角色粗粒度划分＋域细粒度划分的方法和角色细粒度划分＋域粗粒度划分的方法实现的有效角色相同，且执行复杂度相同.

证明.　由角色细粒度划分＋域粗粒度划分的方法，假设系统配置 m_1 个初始管理角色 $r_1, r_2, \cdots, r_{m_1}$ 与 m_2 个管理层域 $d_1, d_2, \cdots, d_{m_2}$ $(m_1 \geq m_2)$，每个初始角色都可以进入每个管理层域，因此可以得到 $m_1 \times m_2$ 个有效管理角色. $P_{r_i} = P_{r_i 1} \cup P_{r_i 2} \cup \cdots \cup P_{r_i m_2}$ $(1 \leq i \leq m_1)$，P_{r_i} 是初始角色 r_i 的特权集，$P_{r_i j}$ $(1 \leq j \leq m_2)$ 是角色 r_i 的与域 d_j 相关的部分特权，$P_{d_j} = P_{r_1 j} \cup P_{r_2 j} \cup \cdots \cup P_{r_{m_1} j}$ 是域 d_j 的特权集. 根据上述特权集的∩关系，初始角色 r_i 进入到域 d_j 得到的有效角色 vr_k 的特权集 $P_{vr_k} = P_{r_i j}$ $(1 \leq k \leq m_1 \times m_2)$.

由角色粗粒度划分＋域细粒度划分的方法，系统配置 m_2 个初始角色 $r_1, r_2, \cdots, r_{m_2}$ 与 m_1 个管理域 $d_1, d_2, \cdots, d_{m_1}$，与上同理，可得 $m_2 \times m_1$ 个有效角色. $P'_{r_j} = P'_{r_j 1} \cup P'_{r_j 2} \cup \cdots \cup P'_{r_j m_1}$ $(1 \leq j \leq m_2)$，P'_{r_j} 是初始角色 r_j 的特权集，$P'_{r_j i}$ $(1 \leq i \leq m_1)$ 是角色 r_j 的与域 d_i 相关的部分特权，$P'_{d_i} = P'_{r_1 i} \cup P'_{r_2 i} \cup \cdots \cup P'_{r_{m_2} i}$ 是域 d_i 的特权集. 那么同样由特权集的∩关系，初始角色 r_j 进入到域 d_i 得到的有效角色 vr_k 的特权集 $P'_{vr_k} = P'_{r_j i}$ $(1 \leq k \leq m_2 \times m_1)$.

$P'_{r_j i} = P_{r_i j}$，即 $P'_{vr_k} = P_{vr_k}$，从而可得：角色粗粒度划分＋域细粒度划分方法和角色细粒度划分＋域粗粒度划分方法都实现了相同的有效管理角色. 从上述分析过程不难看出，这两种方法都执行 $m_1 \times m_2$ 个交集操作来获得角色的有效特权，所以执行复杂度相同.　　证毕.

定理 2.　在采用角色特权集∩域特权集来获得有效特权集的混合 RBAC-DTE 安全策略的管理层配置中，角色粗粒度划分＋域细粒度划分方法的安全性高于角色细粒度划分＋域粗粒度划分方法.

证明.　假设在角色细粒度划分＋域粗粒度划分方法下，系统配置 m_1 个初始角色与 m_2 个管理域，$m_1 \geq m_2$，而在角色粗粒度划分＋域细粒度划分方法下，系统配置 m_2 个初始角色与 m_1 个管理域. 前者存在 $m_2 C_{m_1}^2$ 个域配置的访问许可权冲突，后者存在 $m_1 C_{m_2}^2$ 个域配置的访问许可权冲突，因为 $m_2 C_{m_1}^2 = m_2 \times \frac{m_1 \times (m_1 - 1)}{2} = \frac{m_1 m_2 (m_1 - 1)}{2}$，$m_1 C_{m_2}^2 = m_1 \times \frac{m_2 \times (m_2 - 1)}{2} = \frac{m_1 m_2 (m_2 - 1)}{2}$，$m_2 C_{m_1}^2 \geq m_1 C_{m_2}^2$，因此角色粗粒度划分＋域细粒度划分方法的安全性高于角色细粒度划分＋域粗粒度划分方法.　　证毕.

4　域的静态继承

在混合 RBAC-DTE 策略的多角色管理中，(1)如果从一个角色改变到另一个角色，除了通过运行 init 程序实现从系统管理员角色 sysadm_r 到系统操作员角色 sysopt_r 的转换，通常情况下每次角色的改变都应该经过一次角色认证，这只能或者在与 login 程序相关的域(如域 login_d)中实现，或者设计一个程序以及与其相关联的域来实现，如 SELinux 系统采用的 newrole 程序和域 newrole_t；(2)如果基于程序的执行要改变一个角色的权限而保持角色不变，可以利用 DTE 模型的域转换来实现；(3)角色要实现对信息的操作必须通过域来作许可权决策. 为了在低层的 DTE 模型中刻画高层 RBAC 模型的角色继承关系，这一节引入了新的域的特性.

从定理 2 的证明可以看到，角色粗粒度划分＋域细粒度划分方法确实降低了系统的危害性，但是域的细分有一定的限度，否则会造成 DTE 策略库过于膨胀产生冗余而难以维护. 另一方面，只要允许

多个管理角色进入同一个管理域,角色粗粒度划分+域细粒度划分方法仍然存在访问许可权的冲突.为了遵循主体操作极小特权原则,我们新引入了域的静态继承关系,这种关系描述了静态继承域继承父域所有的访问许可权,同时还拥有继承域本身的访问许可权(一般不同于父域的访问许可权).这种继承性是静态的,有别于域转换,从一个域转换到另一个域是动态的,并且无继承性. DTEX[12] 也提出了继承域的概念,它是基于域转换的,是动态的,也就是说,如果一个域是继承域的话,那么从一个域转换到这个继承域时继承域的访问许可权由源域的访问许可权和目标域(继承域)本身的访问许可权构成.然而,针对上述许可权冲突问题,这种动态继承域会要求连续几次进行 exec 方式域转换,并且重复认证转换请求.例如当角色 admin_r 以 exec 方式从一个域转换到安全域 sec_d 时,需要对 admin_r 的身份进行一次认证,为了获得不同于角色 opt_r 的特别的管理许可权,根据 DTEX 的动态继承域方法,角色 admin_r 必须再次以 exec 方式进入到安全域sec_d 的与管理相关的继承域 INHERITANCE DOMAIN-admin_d(INHERITANCE 和 DOMAIN 都是安全策略中的关键字),又重复一次角色认证.

继承域只需要在父域的基础上增加少量的访问许可权.而静态继承域可以不经过域转换就继承了父域的访问能力,我们用 static_inheritance 来声明一个静态继承域,其作用可由以下的部分 DTEL 策略语言[2] 样本得以体现.以下的部分样本策略中,域规范由域名称、入口程序的型、型访问、域转换和域间通信组成,访问模式 r,w,x 分别表示文件的读、写、执行,l,c,d 分别表示目录的读、写、查找.

```
…
//*_d 和 *_t 分别表示域和型,数字表明项数
//0->0 表示不发生域间通信
spec_domain login_d (1 login_t)(16 rld->base_t
rxld->bin_t … rwxlcd->tmp_t \
rld->user_t)(8 exec->secadm_d exec->secopt_d
exec->sysadm_d exec->sysopt_d \
exec->netadm_d exec->netopt_d exec->adtadm_d
exec->adtopt_d)(1 0->0)

spec_domain sec_d (1 shell_t)(28 rld->base_t
rxld ->bin_t … r->init_t \
rld->secpolicy_t rld->secpolicy_run_t)(10 auto->
mail_d auto->login_d …)(1 0->0)

//分配型 secpolicy_t 给安全策略文件或安全策略库
spec_domain secadm_d
static_inheritance sec_d
  (1 wc->secpolicy_t)(6 auto->mailadm_d …)
  (1 0->0)

//分配型 secpolicy_run_t 给可加载或可启动安全策略
//文件或安全策略库的客体
spec_domain secopt_d
static_inheritance sec_d
  (1 wxc->secpolicy_run_t)(6 auto->mailopt_d …)
  (1 0->0)

spec_domain sys_d (1 shell_t)(28 rld->base_t
rxld->bin_t … rxld->init_t \
rwxlcd->tmp_t rld->sysconf_t rld->sysconf_run_t)
(9 auto->mail_d auto->login_d …) \
(1 0->0)

//分配型 sysconf_t 给系统配置文件
spec_domain sysadm_d
static_inheritance sys_d
  (1 wc->sysconf_t)(6 auto->mailadm_d …)
  (1 0->0)

//分配型 sysconf_run_t 给可使系统配置生效的客体
spec_domain sysopt_d
static_inheritance sys_d
  (1 wxc->sysconf_run_t)(6 auto->mailopt_d …)
  (1 sigtstp->daemon_d)
…
//客体的型分配规则
```

域 secadm_d 和域 secopt_d 都是域 sec_d 的静态继承域,它们都继承了域 sec_d 的所有访问特性,包括 1 项入口程序的型 shell_t,rld->base_t、rxld->bin_t 等 28 项域对型的访问,auto ->mail_d 等 10 项域转换和 1 项域间信号通信 0->0.除此之外,域 secadm_d 和域 secopt_d 又分别拥有自己的访问特性.如由 wc->secpolicy_t,域 secadm_d 还可以写方式访问型为 secpolicy_t 的客体,但域 secopt_d 不能;由 wxc->secpolicy_run_t,域 secopt_d 还可以写方式或执行方式访问型为 secpolicy_run_t 的客体,但域 secadm_d 不能.对域 sys_d 的静态继承域,域 sysadm_d 和域 sysopt_d,我们也可得到类似的分析结果.从上述分析可以得到:域的静态继承关系可以减少策略代码的重复出现,从而有效控制了安全策略文件的代码尺寸;允许子域共享父域的所有访问权以支持多域共享访问许可权;最重要的是与 RBAC-DTE 安全策略配置的主体极小特权原则一致.值得注意的是,域的静态继承关系通过 DTE 机制刻画了在访问许可权上的角色继承关系,DTE 机制中不关心域特权,域特权和角色特权一起都在 Capability 权能机制中实现.也就是说,混合

RBAC-DTE 策略的多角色管理由实现特权的 Capability 机制和实现访问许可权的 DTE 机制来组成. 如果角色 admin_r 以 exec 方式从一个域转换到安全域 sec_d 时，成为安全管理员，其特权集为角色 admin_r 的特权集与域 sec_d 的特权集的交集，而访问许可权集为静态继承域 secadm_d 的访问许可权集.

5　结论及将来的工作

混合 RBAC-DTE 访问控制模型因其安全目标的灵活性受到了越来越多的重视. 既然混合 RBAC-DTE 模型的首要模型是 RBAC 模型，且高等级安全系统需要不同的角色执行复杂的管理任务，这个混合模型的实现就存在多角色管理的问题，遗憾的是目前的研究工作尚未讨论这个问题. 多角色管理通常情况下指多管理角色的管理. 本文从特权和访问许可权两个层面上针对混合 RBAC-DTE 策略中的多角色管理，讨论了角色和域的相对划分粒度，并认为在角色粗粒度划分＋域细粒度划分方法的基础上利用域的静态继承关系，既可以允许多域共享访问许可权又有效地控制了策略代码的尺寸，而且更充分地体现了主体极小特权原则.

通常情况下，DTE 策略的配置都在系统的安装和初始化配置期间进行，而系统运行期间不能改变 DTE 策略. DTE 策略静态配置的局限性也约束了混合 RBAC-DTE 策略配置的灵活性，在这种情况下不能动态地添加、删除和修改角色. 本文的工作是基于静态配置的混合 RBAC-DTE 策略，随着动态 DTE 模型的发展，可以将多角色的管理扩展到动态的 RBAC-DTE 策略上来讨论.

参　考　文　献

1　Sandhu R. S., Coyne E. J., Feinstein H. L., Youman C. E.. Role-based access control models. Computer, 1996, 29(2): 38～47

2　Badger L., Sterne D. F., Sherman D. L., Walker K. M.. A domain and type enforcement UNIX prototype. USENIX Computing Systems, 1996, 9(1): 47～83

3　Hallyn Serge E., Kearns Phil. Domain and type enforcement for Linux. In: Proceedings of the 4th Annual Linux Showcase and Conference, Atlanta, Georgia, USA, 2000, 247～260

4　Ji Qing-Guang, Qing Si-Han, He Ye-Ping. Based-DTE integrity protection formal model. Science in China, Series E, 2005, 35(6): 570～587(in Chinese)
(季庆光，卿斯汉，贺也平. 基于 DTE 技术的完整性保护形式模型. 中国科学，E 辑，2005，35(6)：570～587)

5　Hoffman J.. Implementing RBAC on a type enforced system. In: Proceedings of the 13th Annual Computer Security Applications Conference(ACSAC'97), Washington, DC, USA, 1997, 158～163

6　Chandramouli R.. A framework for multiple authorization types in a healthcare application system. In: Proceedings of the 17th Annual Computer Security Applications Conference(ACSAC'2001), Washington, DC, USA, 2001, 137～148

7　National Security Agency. Security-Enhanced Linux(SELinux). Available at http://www.nsa.gov/selinux

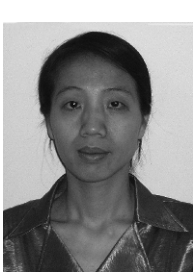
8　Smalley S.. Configuring the SELinux policy. NAI Labs, Network Associates, Inc., Glenwood, Maryland, USA: Technical Report # 02-007, 2002. Available at http://www.nsa.gov/selinux/info/docs.cfm

9　Ji Qing-Guang, Qing Si-Han, He Ye-Ping. A new formal model for privilege control with supporting POSIX capability mechanism. Science in China, Series E, 2004, 34(6): 683～700(in Chinese)
(季庆光，卿斯汉，贺也平. 支持 POSIX 权能机制的一个新的特权控制的形式模型. 中国科学，E 辑，2004，34(6)：683～700)

10　Data General. Managing security on DG/UX system. Data General Corporation, Westboro, Massachusetts, USA: Manual 093-701138-09, 2001

11　U. S. Department of Defense. Trusted computer system evaluation criteria. U. S. Department of Defense, Washington, DC, USA: DoD 5200.28-STD, 1985

12　Fox Charles Matthew. DTEX: Domain and type enforcement extensions in Linux[M. S. dissertation]. Information Networking Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, 2004



TANG Liu-Ying, born in 1973, Ph. D. candidate, engineer. Her mainly engaged in the research of operating system security and formal development methods.

QING Si-Han, born in 1939, professor and Ph. D. supervisor. His research interests include theories and technologies for information system security.

Background

The work of this paper is supported by the Beijing Natural Science Foundation under grant No. 4052016; the National Natural Science Foundation of China under grant No. 60573042; the National Basic Research Program (973 Program) of China under grant No. G1999035802.

As a fundamental and critical step during a high-level secure system development procedure, a security policy model is an abstract of the security policy describing a site's security goals. Security policy models, such as BLP secrecy model, Biba integrity model, Role-Based Access Control (RBAC) model and Domain and Type Enforcement (DTE) model, have gathered much attention in the past. Unlike BLP model and Biba model, RBAC model and DTE model each do not explicitly indicate the security goals of their own policy. RBAC model and DTE model can provide broader secure protect for the systems because they implicit the secure goals of the systems and both are neutral. RBAC model establishes the relationship between users and roles and restricts the operations available to a user according to the roles assigned to the user. Different roles perform their own specified system functions. DTE model being an extension of Type Enforcement (TE) model establishes the relationship between subjects and domains, and restricts the operations available to a subject according to the domain assigned to the subject. Thus, RBAC model is a high-level abstraction model, but DTE model is a low-level model. In 1997, Hoffman firstly described an implementation of RBAC mechanisms on LOCK6, a secure operating system developed at Secure Computing Corporation. According to Hoffman's opinion, DTE can be viewed as an intermediate layer of abstraction between roles and individual permission bits in the operating system. Since the work of Hoffman, much work has been done in hybrid RBAC-DTE policy model research and application, like Dynamic Authorization Framework for Multiple Authorization Types (DAFMAT) presented by Chandramouli and the famous Security-Enhanced Linux (SELinux) system released by the U. S. National Security Agency (NSA).

For a high level secure system applying the hybrid RBAC-DTE policy, multiple roles are usually required to perform complicate administrative tasks. Administration of multiple roles turns to be more complicate and more difficult due to the special role-domain relationship of the hybrid RBAC-DTE policy. However, there has been still no public discussion on the issue. Therefore, from the two layers of privilege and access right, the authors separately discuss existing problems raised by multirole configuration and further propose corresponding solution for multirole administration in the hybrid RBAC-DTE policy. The work aims to implement effective multirole administration under the principle of least privilege.