基于角色的域-类型增强访问控制模型研究及其实现

赵庆松[1], 孙玉芳[1], 张晓平[2]
(1. 中国科学院软件研究所, 北京 100080; 2. 山东建筑工程学院, 山东济南 250014)

摘 要: 安全系统只有能够支持多种安全政策才能满足实际需求. 基于角色的访问控制(Role-Based Access Control, RBAC)是一种政策中性(Policy Neutral)的新模型, 已经实现了多种安全政策. 域-类型增强(Domain and Type Enforcement, DTE)安全政策充分体现了最小特权(Least Privilege)和职责分离(Separation of Duty)的安全原则, 但是, RBAC96 不便于直接实现 DTE. 根据 RBAC 和 DTE 的思想, 本文提出了"基于角色的域-类型增强访问控制"(Role-Based Domain and Type Enforcement Access Control, RDTEAC)模型. 该模型继承了 RBAC96 的优点, 又体现了 DTE 的安全思想, 并易于实现 DTE 安全政策. 此外, 我们还在 Linux 上实现了 RDTEAC 模型的一个原型.

关键词: 信息安全; 多安全政策; 域-类型增强; 基于角色的访问控制; 基于角色的域-类型增强访问控制
中图分类号: TP309 文献标识码: A 文章编号: 0372-2112 (2003) 06-0842-05

Research and Implementation of Role-Based Domain and Type Enforcement Access Control Model

ZHAO Qing-song[1], SUN Yu-fang[1], ZHANG Xiao-ping[2]
(1. *Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China;*
2. *Shandong Institute of Architecture and Engineering, Jinan, Shandong 250014, China*)

Abstract: Only those systems which support multiple security policies can meet practical security requirements. The role-based access control (RBAC) is a new policy-neutral model, which has enforced multiple security policies. The domain and type enforcement security policy fully reflects the security principles of least privilege and separation of duties. According to RBAC and DTE, the role-based domain and type enforcement access control (RDTEAC) model is presented. RDTEAC not only inherits the advantage of RBAC, but also facilitates implementation of DTE security policy. A Linux prototype of RDTEAC was developed.

Key words: information security; multi-policy security; domain and type enforcement; role-based access control; role-based domain and type enforcement access control; Linux

1 引言

随着信息技术的不断发展和大规模应用, 信息安全成为越来越突出的问题. 实际应用对系统资源保护需求是多方面的, 如保护资源的"保密性"、"完整性"、"隐私性"、"可用性"、"可控性"等[1], 因此, 安全系统必须支持多种安全政策[2]. 当前, 支持多种安全政策的系统大都是用不同的安全模型实现不同的安全政策: 用 BLP 模型实现保密性强制访问控制政策[3], 用 Biba[4]或者 C-W[5]模型实现完整性强制访问控制政策, 用 B-N 模型实现 Chinese Wall 安全政策(CWSP)[6]等. 这种实现方式开发工作量大、周期长, 系统管理困难. 因此, 开发一种与安全政策无关的安全模型以实现多种安全政策成为迫切的需要.

作为政策中性的模型, RBAC 被看作是替代传统的自主访问控制(Discretionary Access Control, DAC)和强制访问控制(Mandatory Access Control, MAC)的一种全新的模型[7]. RBAC 受到普遍重视的原因正是在于它的"政策中性"的特点[7], 即 RBAC 不是为专门的安全政策而设计的, 而可以用来实现多种不同的安全政策, 如用来实现 DAC[8]、MAC[9]和 CWSP[10]等. RBAC 技术已经得到了广泛应用[11]. RBAC96[7]是被普遍接受的 RBAC 模型. RBAC96 仅是一个基于角色的安全框架, 因此, 在用它实现某一具体的安全政策时不如用专门的安全模型方便.

最小特权和职责分离是两个重要的安全原则, 前者是指系统分配给用户的权限是用户完成工作所必需的权限的最小集合[12]; 后者是指系统中不同的职责尽量由不同的用户承担, 避免权限过度集中[13]. UNIX/ Linux 中拥有"全能权限(all-powerful privilege)"的超级用户带来了严重的安全隐患, 实际上是"将所有的鸡蛋都放入一个篮子中"[14]. DTE 是一种强制访问控制安全政策, 很好地体现了最小特权和职责分离的安

收稿日期: 2002-01-01; 修回日期: 2002-08-29
基金项目: 国家 863 高科技项目(No. 863-306-ZD12-14-2); 国家自然科学基金项目(No. 60073022); 中国科学院知识创新工程项目(No. KGCX1-09)

全原则. 已经实现 DTE 的系统很多, 它们的大都是基于 DTEL 所描述的规则库实现的[14]. DTEL 是为 DTE 专门设计的语言, 这种基于 DTEL 规则库的方法难以同时灵活地实现其他安全政策.

本文将 DTE 的概念和 RBAC 模型相结合, 提出了 RDTEAC 模型. 该模型在保持了 RBAC 模型通用性的基础上引入了 DTE 的思想, 能够灵活地实现包括 DTE 在内的多种安全政策.

2　RBAC 和 DTE

2.1　基于角色的访问控制及 RBAC96 模型

RBAC96 是由 Sandhu 研究、定义及总结的一组模型[7], 它充分体现了角色的思想. 但是 RBAC96 仅是基于角色理论的访问控制框架模型, 它不是为某种安全政策而专门设计的, 因此也没有体现 DTE 的思想, 缺少对 type 的支持, 没有定义权限和权限之间、以及 type 和 type 之间的相互关系, 因此不能方便地实现 DTE 安全政策. RBAC96 是本文提出的 RDTEAC 模型的基础.

2.2　DTE 和 DTEL

DTE 把主体归到不同的域(Domain)中, 而把客体归到不同的类型(Type)中, 系统按照规则, 限制域中的用户对类型中的资源的访问, 同时限制域中的用户对其他域中的用户的访问.

作为一种访问控制技术, DTE 最初源自 Bobert 和 Kain 的早期工作[15], 后来又在 LOCK 系统[16]中重新定义. DTE 是对“类型增强(type enforcement)”访问控制的一种扩展, 在文献[15]中, Bobert 和 Kain 按照制定的安全策略来限制进程对客体的访问.

DTEL 是一种高级的符号语言, 它采用易于理解的形式来表达系统的 DTE 安全策略[15]. 实际上, DTEL 就是通过对主体所属的域的限制、对客体所属的类型的限制以及对主体访问客体的限制来实现安全策略的. DTEL 提供了 type、domain、initial_domain 和 assign 四种基本语句以表达 DTE 安全策略[14].

显然, DTEL 是为 DTE 专门设计的语言, 仅提供了有限的描述和控制功能. 一方面, 基于 DTEL 的规则库难以同时灵活地描述其他多种安全政策; 另一方面, 随着系统的运行, 系统的安全规则库会变得越来越庞大, 规则之间的关系也越来越复杂, 难以管理系统.

3　RDTEAC 模型

RDTEAC 模型定义如定义 1; 其结构可以用图 1 来表示.

定义 1: (1) U: 用户集合; DR 和 ADR: 域_角色和授权管理域_角色, $DR \cap ADR = \varnothing$; P 和 AP: 权限和授权管理权限, $P \cap AP = \varnothing$; T 和 AT: 类和管理类; O: 客体集合; S: 会话集合.

(2) $U\text{-}DR\_A \subseteq U \times DR$: 用户到域_角色的指派关系; $U\text{-}ADR\_A \subseteq U \times ADR$: 用户到授权管理域_角色的指派关系.

(3) $DR\text{-}P\_A \subseteq DR \times P$: 权限到域_角色的指派关系; $AP\text{-}ADR\_A \subseteq AP \times ADR$: 授权管理权限到授权管理域_角色的指派关系.

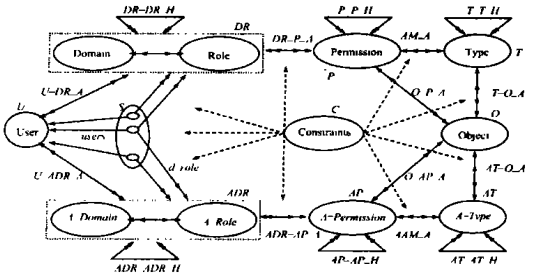

图 1　基于角色的域-类型增强访问控制模型

(4) $AM\_A \subseteq P \times T$: 权限到类型的指派关系; $AAM\_A \subseteq AP \times AT$: 管理权限到管理类型的指派关系.

(5) $T\text{-}O\_A \subseteq T \times O$: 客体到类型的指派关系; $AT\text{-}O\_A$: 客体到管理类型的指派关系.

(6) $O\text{-}P\_A \subseteq O \times P$: 客体到权限的指派关系; $O\text{-}AP\_A \subseteq O \times AP$: 客体到管理权限的指派关系.

(7) $DR\text{-}DR\_H \subseteq DR \times DR$: 域_角色与域_角色之间的继承关系; $ADR\text{-}ADR\_H \subseteq ADR \times ADR$: 授权管理域_角色与授权管理域_角色之间的继承关系.

(8) $user: S \rightarrow U$: 会话和用户之间的映射关系; $d\_roles: S \rightarrow 2^{DR \cup ADR}$ 会话到域_角色和授权管理域_角色之间的映射关系; $d\_roles(s_i) \subseteq \{r \mid (\exists r' \geqslant r)[user(s_i), r' \in U\text{-}DR\_A \cup U\text{-}ADR\_A]\}$

(9) Constrains: 在 RBAC 的各种关系和各个组件中起作用的一组约束.

模型中的元素及其含义如表 1 所示.

表 1　RDTEAC 元素及其含义

符号	含义
O	系统中所有客体的集合
o	系统中的某一客体, 是系统中可被访问的最小单元
T/AT	系统中所有类型/授权管理类型的集合
t/at	系统中的某一类型/授权管理类型
P/AP	系统中所有客体的集合
p/ap	系统中的某一项权限/授权管理权限
A	所有的元访问模式的集合
a	某一元访问模式, 是系统中不能再细划的访问模式
M	所有的访问模式的集合
m	某一访问模式
DR/ADR	系统中所有域-角色/授权管理域-角色的集合
dr/adr	系统中的某一域-角色/授权管理域-角色
S	用户启动的所有的会话的集合
s	用户启动的某一会话
U	系统中的所有用户的集合
u	系统中的某一用户
C	系统对 $U\text{-}DR\_A$、$DR\text{-}P\_A$、$U\text{-}ADR\_A$、$ADR\text{-}AP\_A$、d_roles 的制约的集合
c	具体的某一制约

3.1　模型中基本元素间的关系

与 RBAC96 不同, RDTEAC 中的 type 和 type 之间、permission 和 permission 之间以及 domain role 和 domain role 之间都存在相互关系. 下面对模型中的 type、permission、domain role 及其

它们之间的关系作系统的描述.

(1) 访问模式 m

定义 2: m 表示某访问模式, $m(m)$ 表示取得组成 m 的元访问模式的集合.

$m_i \subset m_j \Leftrightarrow m(m_i) \subset m(m_j)$, 表示访问模式 m_i 包含于 m_j 中;

$m_i = m_j \Leftrightarrow m(m_i) = m(m_j)$, 表示访问模式 m_i 等于 m_j;

$m_i \leftrightarrow m_j \Leftrightarrow m_i \not\subset m_j \cap m_j \not\subset m_i \cap m_i \neq m_j$, 表示访问模式 m_i 和 m_j 不可比较;

如"读写"操作模式就包含"读"操作模式, 其中"读"和"写"操作是不可再分的操作, 叫作元访问模式.

(2) 类型 type

定义 3: 类型 t 所包含的客体的集合记作: $t(t)$, 其中 $t(o) = o$.

对于任意 t_i 和 t_j, 有:

$t_i \subset t_j \Leftrightarrow t(t_i) \subset t(t_j)$, 表示类型 t_i 包含于类型 t_j 中;

$t_i = t_j \Leftrightarrow t(t_i) = t(t_j)$, 表示类型 t_i 等于类型 t_j;

$t_i \leftrightarrow t_j \Leftrightarrow t_i \not\subset t_j \cap t_j \not\subset t_i \cap t_i \neq t_j$, 表示类型 t_i 和类型 t_j 不可比较;

(3) 权限

定义 4: 在 RDTEAC 中, 权限的形式有 $p(m, t)$ 和 $p(m, o)$, 其中, m 是访问 t 或 o 的访问模式, 记作为: $pm(p)$; o 或 t 是 p 的访问对象记作: $pt(p)$.

$p_i \subset p_j \Leftrightarrow pm(p_i) \subset pm(p_j) \cap pt(p_i) \subset pt(p_j)$

$p_i \leftrightarrow p_j \Leftrightarrow pm(p_i) \leftrightarrow pm(p_j) \cup pt(p_i) \leftrightarrow pt(p_j)$

任何权限都可以分解成用元访问模式和系统客体的二元组表示.

对于权限 $p = (m, t) = (\{a_1, a_2, \cdots, a_{k-1}, a_k\}, \{o_1, o_2, \cdots o_{p-1}, o_p\})$, 有: $ao(p) = ao(m, t) = \sum_{i=1}^{k} \sum_{j=1}^{l} (a_i, o_j)$

其中 $a_i \in m(m) = \{a_1, a_2, \cdots, a_{k-1}, a_k\}$; $o_j \in t(t) = \{o_1, o_2, \cdots o_{p-1}, o_p\}$.

权限是可比较的, 权限组也是可以比较的.

对于权限组 $P_m = \{p_{m1}, p_{m2}, \cdots, p_{mk-1}, p_{mk}\}$ 和权限组 $P_n = \{p_{n1}, p_{n2}, \cdots, p_{np-1}, p_{np}\}$, 有:

$$P_m \subset P_n \Leftrightarrow \sum_{i=m1}^{mk} ao(p_i) \subset \sum_{j=n1}^{np} ao(p_j)$$

$$P_m = P_n \Leftrightarrow \sum_{i=m1}^{mk} ao(p_i) = \sum_{j=n1}^{np} ao(p_j)$$

如果同时具有权限 $p_1, p_2, p_3, \cdots, p_m, p_{m+1}, \cdots, p_n$ 的主体不会违反安全政策 SP, 则称 $p_1, p_2, p_3, \cdots, p_m, p_{m+1}, \cdots, p_n$ 基于 SP 是相容的, 记作: $comp\_p(SP: p_1, p_2, p_3, \cdots, p_m, p_{m+1}, \cdots, p_n)$.

推论 **1** 对于基于安全政策 SP 相容的一组权限 P_m 有:

◇ P_m 中的任一项权限 p_i 与其本身基于 SP 是相容的;

◇ 若 $p_i \in P_m$, $p_j \subset p_i$, 则 p_j 与 P_m 是相容的;

◇ P_m 中的任意项权限项组合成的一组权限是相容的.

基于任一安全政策 SP, 与权限 p_i 相容的所有权限的集合称作 p 基于 SP 的相容集: $comp\_ps(SP: p_i) = \{p \mid comp\_p(SP: p_i, p)\}$.

如果同时具有权限 $p_1, p_2, p_3, \cdots, p_m, p_{m+1}, \cdots, p_n$ 的主体会违反安全政策 SP, 则称 $p_1, p_2, p_3, \cdots, p_m, p_{m+1}, \cdots, p_n$ 基于 SP 是相冲突的, 记为: $conf\_p(SP: p_1, p_2, p_3, \cdots, p_m, p_{m+1}, \cdots, p_n)$.

推论 **2** 对于基于安全政策 SP 相冲突的一组权限 P_m 有:

◇ 任何权限 p_i 与 P_m 是相互冲突的;

◇ P_m 其中至少有两项权限是相互冲突的;

◇ 与 P_m 中的任意权限项 p_i 有关系 $p_i \subset p_j$ 的 p_j 与 P_m 是相冲突的;

基于任一安全政策 SP, 与权限 p_i 相冲突的所有权限的集合称作 p 基于 SP 的冲突集: $conf\_ps(SP: p_i) = \{p \mid conf\_p(SP: p_i, p)\}$.

(4) 域-角色 DR

定义 5: dr 是一组权限的集合, $p(dr)$ 表示 dr 所具有的权限组.

$dr_i \subset dr_j \Leftrightarrow p(dr_i) \subset p(dr_j)$

$dr_i = dr_j \Leftrightarrow p(dr_i) = p(dr_j)$

$dr_i \leftrightarrow dr_j \Leftrightarrow dr_i \not\subset dr_j \cap dr_j \not\subset dr_i \cap dr_i \neq dr_j$

如果同时具有域-角色 $dr_1, dr_2, dr_3, \cdots, dr_m, dr_{m+1}, \cdots, dr_n$ 的主体不会违反安全政策 SP, 则称 $dr_1, dr_2, dr_3, \cdots, dr_m, dr_{m+1}, \cdots, dr_n$ 基于 SP 是相容的, 记为: $comp\_dr(SP: dr_1, dr_2, dr_3, \cdots, dr_m, dr_{m+1}, \cdots, dr_n)$.

推论 **3** 基于某安全政策 SP 相容的一组 DR_m 有:

◇ 其中的任一 DR 与其本身是相容的;

◇ 若 $dr_i \in DR_m$, 且 $dr_j \subset dr_i$, 则 dr_j 与 DR_m 是相容的;

◇ 其中的任意项域-角色组合成的 DR 组仍是相容的;

基于任一安全政策 SP, 与域-角色 dr_i 相容的所有域-角色的集合称作 dr_i 基于 SP 的相容集, 记作: $comp\_drs(SP: dr_i) = \{dr \mid comp\_dr(SP: dr_i, dr)\}$.

如果同时具有 $dr_1, dr_2, dr_3, \cdots, dr_m, dr_{m+1}, \cdots, dr_n$ 的主体违反安全政策 SP, 则称 $dr_1, dr_2, dr_3, \cdots, dr_m, dr_{m+1}, \cdots, dr_n$ 基于 SP 是相冲突的, 记为: $conf\_dr(SP: dr_1, dr_2, dr_3, \cdots, dr_m, dr_{m+1}, \cdots, dr_n)$.

推论 **4** 基于某安全政策 SP 相冲突的一组 DR_m 有:

◇ 任何 $dr_i \cup DR_m$, 基于该安全政策仍是相互冲突的;

◇ 其中至少有两个 DR 是相互冲突的;

◇ 相互冲突的两 DR 与 *senior* 于其中之一的任何 DR 是相互冲突的;

基于任一安全政策 SP, 与 dr_i 相冲突的所有 DR 的集合称作 dr 基于 SP 的冲突集: $conf\_drs(SP: dr_i) = \{dr \mid conf\_dr(SP: dr_i, dr)\}$.

(5) 主体用户 U

定义 6: 对于系统中的用户 u, 其所具有的域-角色记作: $dr(u)$.

对于模型中的 A_type、A_permission 和 A_domain_role 等

权限管理元素, 它们之间的关系与以上描述的 type、permission 和 domain-role 等普通元素之间的关系是一样的, 只是它们所涉及到的客体是系统中被管理的权限项和用户. 为突出系统中 DTE 的实现, 我们简化了对该部分的深入讨论, 认为所有的管理功能由系统安全管理员(具有域-角色 secoff 的用户)统一管理.

3.2 授权和授权撤销

RDTEAC 中, 对系统中的用户权限的管理主要是通过 $U\text{-}DR\_A$/ $U\text{-}ADR\_A$、$DR\text{-}P\_A$/$ADR\text{-}AP\_A$ 等授权管理部分和会话管理实现的. 系统中的 constrains 对这些部分加以控制. 以下对 $U\text{-}DR\_A$、$DR\text{-}P\_A$ 和 roles 逐一讨论.

(1)将权限 P 授予 DR 的管理过程记作 $DR\text{-}P\_A$, 只有满足 can_assign_drp 时才可以对 DR 的权限进行配置. $can\_assign\_drp \subseteq ADR \times CR \times P^2$, 具有授权管理域-角色 ADR 的主体对满足 CR 的域-角色授予权限 P. 如 $can\_assign\_drp(x, y, \{a, b, c\})$ 表示具有授权管理域-角色 x(或者任意其他授权管理域-角色 x', 其中 $x(x')$ 的主体, 可以给域-角色(该域-角色已经具有的权限满足条件 y)授予权限 a、b 或 c.

如上所述, 我们认为系统中的授权管理工作由系统安全管理员同一管理, 系统安全管理员可以管理系统中的一切 DR、P、T 的设置和配置. 因此, 基于系统的安全政策 SP, 主体用户 u_i 对域-角色 dr_j 授予权限 p 的关系如下: $can\_assign\_drp(ADR, CR, P) \Leftrightarrow secoff \in dr(u_i) \cap comp\_p(\mathrm{SP}: p, p(dr_j)) \cap p \in P$.

(2)将权限 P 授予 DR 的管理过程记作 $U\text{-}DR\_A$, $can\_assign\_udr \subseteq ADR \times CR \times DR^2$, 具有授权管理域-角色 ADR 的主体对满足 CR 的用户授予域-角色 DR. 如 $can\_assign\_udr(x, y, \{a, b, c\})$ 表示具有域-角色 x(或者任意其他域-角色 x', 其中 $x \subset x'$)的主体用户, 可以给用户(该用户已经具有的域-角色满足条件 y)授予域-角色 a、b 或 c.

相应的, 基于系统的安全政策 SP, 主体用户 u_i 对用户 u_j 授予域-角色 dr_k 的关系如下: $can\_assign\_udr(ADR, CR, DR) \Leftrightarrow secoff \in dr(u_i) \cap comp\_dr(\mathrm{SP}: dr_k, dr(u_j)) \cap dr_k \in DR$.

4 RDTEAC 在 Linux 中的实现

我们基于 Linux 初步实现了一个 RDTEAC 的试验原型, 如图 2 所示.

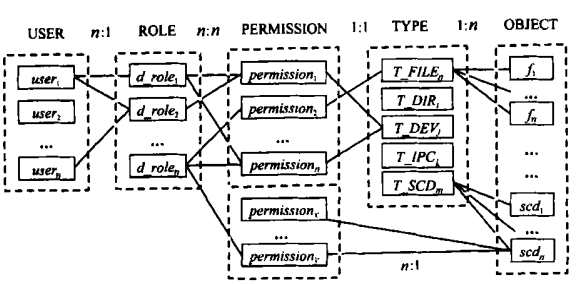

图 2　基于 Linux 的 RDTEAC 原型

在该原型中, 我们把系统中的客体分为五类: 文件(FILE)、目录(DIR)、设备(DEV)、进程间通信(IPC)和系统控制数据(SCD). 为便于说明 RDTEAC 的本质, 系统允许管理员定义的每一类型只含某一类的某些客体, 例如定义类型 $file_x$ 包含文件/tmp/err 和/tmp/secc, 则 $file_x$ 不能再包含其他如何四种类型的任何客体; 同样, permission 也只是指定对某一类型的客体的操作, 或者, 不经过类型 type 直接指派权限到客体的访问. 一个 permission 可以授予多个域-角色, 同时, 一个域-角色可以包含多个 permission. 如图 2 所示, 用户和域-角色之间的关系也是一种多对多的关系.

在该原型中, 用户被授予不同的域-角色, 即相当于属于在 DTE 中属于不同的 domain; 系统中的客体被分配到不同的类型中, 相当于 DTE 中的 type; 每类客体有与之对应的操作集, 对客体的操作定义为权限, 多个权限抽象成域-角色, DTE 中的 DDT 正是由 RDTEAC 中的 $DR\text{-}P\_A$、AM_A、$T\text{-}O\_A$ 以及 $ADR\text{-}AP\_A$、AAM_A、$AT\text{-}O\_A$ 等指派关系体现出.

DTE 中一个域中的用户可以迁移到其他的域中, 即所谓的"域迁移". RDTEAC 给每一个可执行程序分配一个域-角色, 意味着只有属于该域-角色的用户才能执行该程序. 例如给/usr/bin/ftp 分配域-角色"数据传输", 则只有具有"数据传输"域-角色的用户才能在会话过程中激活该域-角色并执行 ftp. 用户在会话过程中激活他所直接具有或者隐含具有的域-角色的过程就是 RDTEAC 的"域-角色迁移". 控制用户激活域-角色就能控制用户的域-角色迁移.

比较 RDTEAC 和 RBAC96 可知, RDTEAC 并没有对 RBAC96 明显作任何删减, 而是对该模型的扩充. 因此, 直接用 RDTEAC 模型完全可以实现 DAC、MAC 和 CWSP 等各种安全政策.

5 总结

本文将 DTE 的思想与 RBAC96 想结合, 提出了一种新的安全模型: RDTEAC. RDTEAC 继承了 RBAC96 原有的政策中立、便于管理等优点; 扩充了 RBAC96 中权限的概念, 定义了权限之间的相互关系; 引入了类型的概念并定义了类型之间的关系. RDTEAC 不但便于实现 DTE, 而且还可以实现其他多种安全政策.

RDTEAC 只是一种可以实现多种安全政策的模型, 如何在一个系统中协调多种安全政策, 如何检测并解决安全政策之间的冲突等问题还有待进一步研究.

参考文献:

[1] Butler W Lampson. Requirements and Technology for Computer Security [R]. Computers at Risk. Washington: National Academy Press, 1991.74-101.

[2] C Bidan, V Issarny. Dealing with multi-policy security in large open distributed systems [A]. Proc of 5th European Symposium on Research in Computer Security [C]. Louvain-la-Neuve, Belgium, Sep. 1998. 51-66.

[3] D E Bell, L J LaPadula. Secure Computer System: Unified Exposition and MULTICS Interpretation [R]. Technical Report MTR-2997, rev 1, MITRE Corp., Bedford, Mass, March 1976.

[4] K J Biba. Integrity Considerations for Secure Computer Systems [R]. Technical Report ESD-TR-76-372, ESD/AFSC, Hanscom AFB, Bedford, Mass, 1977. Also MITRE MTR-3153.

[5] D Clark, D Wilson. A comparison of commercial and military computer security policies [A]. Proc of the IEEE Symposium on Security and Privacy [C]. Los Alamitos: IEEE Computer Society Press, 1987. 554-563.

[6] D Brewer, M Nash. The Chinese Wall security policy [A]. Proc of the 1989 IEEE Symposium on Security and Privacy [C]. Oakland, California, USA: IEEE Computer Society Press, May 1989. 206-214.

[7] Ravi Sandhu. Role-based access control [J]. Advances in Computers, 1998, 46: 237-286.

[8] R Sandhu, Q Munawer. How to do discretionary access using roles [A]. Proc of the Third ACM Workshop on Role Based Access Control [C]. Barkley, Cincotta, 1998. 47-54.

[9] S Osborn. Mandatory access control and role-based access control revisited [A]. Proc of the Second ACM Workshop on Role Based Access Control [C]. Fairfax, Virginia USA: ACM Press, November 1997. 31-40.

[10] Qingsong Zhao, Yufang Sun. Configuring role-based access control to enforce Chinese Wall policy [A]. Proc of the Sixth International Conference for Younger Computer Scientists [C]. Hang Zhou, China, 2001. 563-568.

[11] Joon S Park, Ravi Sandhu, SreeLatha Ghanta. RBAC on the web by secure cookies [A]. Proc of the IFIP WG 11.3 Workshop on Database Security [C]. Chapman & Hall, July, 1999. 49-62.

[12] D S Wallach, D Balfanz, D Dean, E W Felten. Extensible Security Architectures for Java [R]. Technical Report 546-97, Department of Computer Science, Princeton University, Apr. 1997.

[13] Sandhu, R. Separation of duties in computerized information systems [A]. Proc of IFIP WG11.3 Workshop on Database Security [C]. North Holland, 1990. 179-189.

[14] Kenneth M Walker, Daniel F Sterne. Confining root programs with domain and type enforcement [A]. Proc of the Sixth USENIX UNIX Security Symposium [C]. San Jose, California, 1996. 21-36.

[15] W E Boebert, R Y Kain. A practical alternative to hierarchical integrity policies [A]. Proc 8th DoD/NBS Computer Security Initiative Conference [C]. Gaithersburg, MD, September, 1985. 18-27.

[16] R O' Brien, C Rogers. Developing applications on LOCK [A]. Proc of 14th National Computer Security Conference [C]. Washington, DC, Oct, 1991. 147-156.

作者简介:

赵庆松 男, 1973 年 9 月生于山东日照, 1997 年毕业于山东工业大学计算机系, 获学士学位, 1999 年于该校获硕士学位, 现为中科院软件所博士生, 主要研究方向为系统软件安全性.

孙玉芳 男, 1947 年 2 月生于江苏无锡, 研究员, 博士生导师, 主要研究方向为系统软件、信息安全和中文信息处理.

张晓平 女, 1971 年 1 月生于山东枣庄, 现为山东建筑工程学院讲师, 主要研究方向为模式识别与系统辨识.