信息系统安全策略研究

李守鹏[1],孙红波[2]
(1. 四川大学数学学院,四川成都 610063; 2. 北京电子科技学院计算机科学技术系,北京 100070)

摘 要: 安全策略是信息系统安全的关键. 信息系统安全的前提是确保安全策略的完备、正确和一致. 安全策略的复杂性与系统本身的复杂程度密切相关. 安全策略必须得到有效的实施. 本文对安全策略的实施、要求和一致性进行了研究,给出了访问控制策略的一致性定理和一致性检查方法.

关键词: 安全策略; 策略实施机制; 一致性; 粒度
中图分类号: TN918 **文献标识码**: A **文章编号**: 0372-2112 (2003) 07-0977-04

Security Policies Study for Information Systems

LI Shou-peng[1], SUN Hong-bo[2]
(1. *College of Mathematics, Sichuan University, Chengdu, Sichuan* 610063, *China*;
2. *Dept of Computer Science & Technology, Beijing Electronic Science and Technology Institute, Beijing* 100070, *China*)

Abstract: Security policies are the key to the security of information systems. It is the precondition for the security of information systems to assure the completeness, correctness and consistency of security policies. The complexity of security policies is closely related to the degree of complexity of a system itself. Security policies must be enforced effectively. The following is studied in this paper: the enforcement of policies, the requirement for policies, and the consistency of policies. Especially a theorem is given for the consistency of access control policies and a method used to check the consistency is presented.

Key words: security policy; policy enforcing mechanism; consistency; granularity

1 引言

系统安全策略是否"完备、正确和一致"是能否确保系统安全的关键. 安全策略的复杂性与系统本身的复杂程度密切相关. 一般来说,系统的规模越大,安全策略的说明以及在系统中相应的实现机制就越复杂. 例如对于由单个计算机构成的系统来说,系统的安全策略将仅仅围绕该计算机系统,而在一个由计算机网络构成的 IT 系统内,系统的安全策略可能涉及到多个不同的设备,整个系统的安全策略的实施最终都将由系统中具体设备的策略实施机制来实现.

2 系统安全策略的实施

可信系统概念在不断发展和完善[1~3],对信息系统来说,安全策略是系统安全的关键,反映系统安全需求的系统安全策略,因其作用的层次和对象的不同,多种多样,各不相同. 策略的多样性,也为实现策略的机制提出了挑战. 最典型的安全策略是自主访问控制策略和强制访问控制策略[3],这两类策略描绘了大部分系统的实际需求,并且这两类策略在多数安全系统中都提供了相应的实施机制. 图 1 描述了这种情况下的策略实施机制.

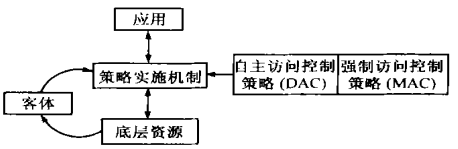

图 1 MAC 和 DAC 策略实施机制

图中没有表示出来的是策略的层次,目前有许多应用还执行特定于应用的安全策略,因此应用程序中也需要有相应的策略实施机制.

在说明不同的安全策略前,下面先给出适合于安全策略多样性的策略实施机制模型,见图 2 所示. 为简单起见,这里不考虑策略的层次性.

图 2 说明,系统安全策略可以分为两大类来考虑,即强制安全策略和自主安全策略. 系统的策略实施机制也划分为两部分,强制安全策略实施机制和自主安全策略实施机制,强制安全策略具有更好的普遍适用性,由系统强制提供,可涉及保密性、完整性、可用性、责任可查性等. 自主安全策略将反映用户自主的安全需求,由于用户自主安全需求的多样性,为尽可能实施灵活的自主安全策略,系统应为用户提供方便的自主安全策略表达机制,如安全规则的说明工具. 用户说明的与自

收稿日期:2002-03-29;修回日期:2002-12-10

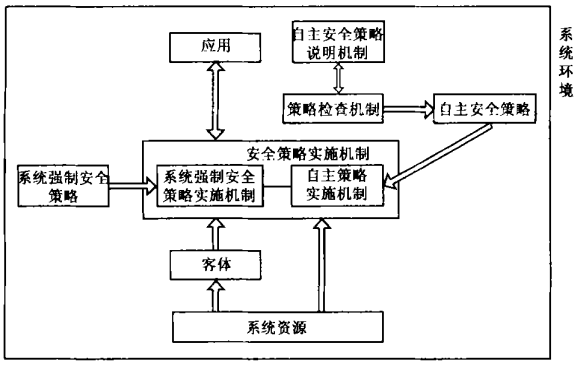

图 2 安全策略实施机制

主安全策略对应的规则集有时非常复杂,需要有专门的策略检查机制来确保规则的完备性、正确性和一致性.这些自主说明的安全规则只有通过检查处理后,才能形成适合于自主安全策略实施机制使用的系统内部自主安全策略(或规则集).规则的内涵及内部表示方式的不同,又对自主安全策略实施机制提出了不同的要求,自主策略实施机制应能够为不同类型的规则提供执行能力.因此图 2 给出的模型,其自主安全策略实施机制能否适当地得以实现,是系统设计时必须研究的重点.

3 对安全策略的要求

为说明对系统安全策略的要求,首先给出系统状态及相关的定义.

系统状态:系统状态(V)是对某一时刻系统状况的反映,可表示为一个二元组:

$$V = (C, P)$$

其中,C 是四元组 (s, o, R_j, α) 组成的集合,表示系统的当前操作集.α 表示一个操作,若以 $A = \{\alpha_1, \alpha_2, \cdots, \alpha_k\}$ 表示系统的可能操作集,则 $\alpha \in A$.系统的可能操作有:读、写、创建、删除、执行、分配、释放等.s 是单个主体,o 是单个客体,$R_j \subset R$ (其中 R 表示系统的全体资源集)是当前涉及的资源集.$P \subseteq P$ 是系统在当前操作集下的策略.P 表示系统所有可能的策略集,表示为 $P = \{p_1, p_2, \cdots, p_q\}$,其中 $p_i (1 \leq i \leq q)$ 表示个别策略.

安全状态:安全状态是指这样的系统状态,状态中的策略说明完备、正确、一致,不存在矛盾、缺陷和漏洞,状态中的当前功能行为集或当前操作集满足状态中的安全策略.

从安全状态的定义可知,系统的安全策略应保持完备、正确和一致,其含义如下:

完备:设系统当前策略 $P = \{p_{k_1}, p_{k_2}, \cdots, p_{k_\beta}\}$,对 $\forall p_\lambda \notin P$ 且 p_λ 与 P 一致,设 $P' = P \cup \{p_\lambda\}$,若 $C = C'$,则说策略 P 是完备的.这里 C 和 C' 分别是符合策略 P 和 P 要求的当前操作集.

设系统的当前策略为 P,P 将作用于系统中的主体、客体、资源、功能或操作及其组合.根据策略的完备性定义可知,对系统中的任何主体、客体、资源、功能或操作在 P 中都应有对应的制约策略,但是对于复杂的系统环境出现策略的不完备还是非常可能的,解决这一问题的办法之一是采用隐含策略说明的方式,即说明一隐含策略集 Q,使得 Q 作用于一切在 P 中没有相对应策略的全体主体、客体、资源、功能或操作等.这样新的策略集 $P' = P \cup Q$ 将是一个完备的策略集.

正确:设 P 为状态中的策略集,若 $\exists (s, o, R_j, \alpha)$ 或 (s', o', R_j', α'),使得 $(s, o, R_j, \alpha) \in C$ 或 $(s', o', R_j', \alpha') \notin C$,但按照系统的预期安全行为应有 $(s, o, R_j, \alpha) \notin C$ 或 $(s', o', R_j', \alpha') \in C$,则称策略是不正确的,相反策略就是正确的.

正确的安全策略来源于对系统任务的正确理解,当安全策略确定后,如何将它们映射成合理的安全规则是策略能否得到正确实施的前提.系统本身提供的策略执行机制是能否正确有效地在系统中实施策略的必要条件,例如,如果系统不支持强制访问控制,那么进行强制访问控制就是不现实的.

一致:设 $P = \{p_{k_1}, p_{k_2}, \cdots, p_{k_\beta}\}$,若 $\exists p_{k_i}$ 和 $\exists p_{k_j} (1 \leq i, j \leq \beta, i \neq j)$,可找到一 (s, o, R_j, α),使得按策略 p_{k_i} 有 $(s, o, R_j, \alpha) \in C$,而按策略 p_{k_j} 有 $(s, o, R_j, \alpha) \notin C$,则称策略 P 是不一致的,反之则称策略 P 是一致的.

在复杂的系统环境中,实现安全策略的规则集是否存在矛盾,是否存在不一致,是系统能否正确地履行其安全保障能力的关键.复杂系统环境中安全策略的层次性和分布性,使得策略的一致性需考虑多个方面的问题.

对于不同层次上的安全策略,为使策略保持一致,可通过管理上的协调及定义策略层次间的逻辑关系来解决.例如:设 P_1, P_2, P_3 是三个不同层次上的策略集,系统总的策略 P 可定义为"$P_1 \wedge P_2 \wedge P_3$","$\wedge$"表示"同时满足".即系统策略需同时满足三个层次上的策略 P_1、P_2、P_3.

对于复杂系统中分布在不同安全域(安全域可以看成是处于同一管理职权边界内的一个系统组成部分,既可以是逻辑的也可以是物理的,但物理上的区分更便于实际处理.如一个应用、一台主机或一个子网都可以构成一个安全域,一个系统可以包含多个安全域)之间的安全策略的一致性,要靠管理协调和技术措施来解决.在同一安全域内,集中在同一层次上的安全策略,为保持策略的一致性,需要有效的一致性检查机制.

对于以网络为基础的分布式信息系统来说,系统安全策略的说明、处理[4~6]与实施比以往都要复杂,无论采用什么方法说明系统安全策略都必须遵守这三个基本要求.

4 系统安全策略的一致性检查

在系统安全管理过程中,为说明域中的安全策略有时需配置数目较多的规则,如:一个用于复杂网络环境中的防火墙,有可能配置数百条,甚至是上千条访问控制规则.这些访问控制规则在粒度上一般也是各不相同的.如果规则的数目较多,同时系统的规则配置又允许"肯定"与"否定"类的规则并存(防火墙的访问控制规则一般采用"缺省"的办法)的话,那么规则之间的不一致就是非常现实的问题.另外,规则未必一定是单纯的"肯定"或"否定",还有可能存在这两种处理之外的多种决策的情况.既使是采用单纯的缺省拒绝("否定")

方式说明规则,对一个庞大的规则集,也有可能出现交叉、包含或重复的规则,使得规则集显得"痈肿".管理员阅读或检查起数百条规则,将是一件非常头痛的事情.另一种情况是,当管理员从规则集中删除一条规则,以便调整访问控制行为时,因存在规则的交叉和包含,删除一条规则有可能达不到预期的目的,也就是有可能还存在另一条规则在某种意义上仍起到与删除的那一条规则同样的作用.因此为了方便系统的安全管理,提高安全保障,对规则的一致性进行自动检查是一个非常值得重视的课题.

下面详细研究以访问控制表(ACL)方式给出的访问控制规则的一致性.

ACL 通常用于控制主体对客体的访问.不同的系统或系统实现,ACL 的结构、表示和特征也不相同.但是 ACL 总要精确地反映组织对其存储在系统设备中的信息的访问控制安全策略.ACL 可以以列表,访问矩阵或其它形式集中或分布地存储在系统中.

ACL 的一致性意味着:在 ACL 中不应存在冲突的表项,定义相互矛盾的访问权.以防火墙为例,防火墙通常安装在两个相邻网络(安全域)的边界上,是信息唯一可以流经的通路.两个邻接的网络通常称为内部网和外部网.对防火墙来讲,网络中的主体可以粗略地定义为:主机、主机组、子网、客户程序、应用程序、进程、用户等,严格地讲它们都代表运行在这些独立的网络中的进程.同样,网络中的客体可定义为主机、主机组、子网、服务、文件等,它们都可以抽象为包含在文件或服务中的信息.由于在安装有防火墙的网络环境中,主体或客体的定义,粒度变化范围较大,容易出现规则的交叉或包含,因此通常会导致拥有许多表项的 ACL 不一致.但是防火墙的正确运行,需要访问控制表 ACL 能够恰当地反映安全策略,具有一致性.下面以实例说明 ACL 不一致的情况.

设 ACL 中有一条规则说明外部网络中的一台主机 A(IP 地址为 202.92.10.100)可全面访问内部网络中的主机 B(IP 地址为 202.93.2.10),ACL 中还有另一条规则说明子网 C(IP 地址为 202.92.110.*)只能使用子网 D(202.93.*.*)中的 FTP 服务,则上述两条规则将出现冲突.在实际的复杂网络环境中,ACL 中可能有数百条乃至上千条规则,每一条规则都从不同粒度上说明访问权,规则的不一致极易出现.

4.1　数学表示

设 $S=\{s_1, s_2, \cdots, s_m\}$ 表示全体主体集,S_i 是 S 的子集,$\rho(S)$ 为 S 的幂集,即,$\forall S_i \in \rho(S)$,$(i=1,2,\cdots,2^m)$,有 $S_i \subseteq S$.

设 $O=\{o_1, o_2, \cdots, o_n\}$ 表示全体客体集,O_j 是 O 的子集,$\rho(O)$ 为 O 的幂集,即,$\forall O_j \in \rho(O)$,$(j=1,2,\cdots,2^n)$,有 $O_i \subseteq O$.

定义　$A=\{a_1, a_2, \cdots, a_l\}$ 代表主体对客体所有可能的访问或操作类型集,A_k 是 A 的子集,$\rho(A)$ 是 A 的幂集,即

$\forall A_k \in \rho(A)$,$(k=1,2,\cdots,2^l)$,有 $A_k \subseteq A$

设 $D=\{d_1, d_2, \cdots, d_h\}$ 代表系统的访问控制机制依据安全策略可能作出的各种决策的集合.

由此 ACL 是一个 V 到 D 的函数,即

$f: V \rightarrow D$,其中 $V \subset \rho(S) \times \rho(O) \times \rho(A)$

ACL 中的每一条规则都可表示成一个三元组 $T=(S, O, A)$ 和相应的决策 d.

三元组 T 表示一组主体 S 希望对一组客体 O 执行一组操作 A(访问属性),d 表示系统根据三元组 T 作出的访问控制决策(如允许、拒绝、丢弃、重定向、错误等),因而访问控制功能是系统实现的一个函数:

$$d=f(T)=f(S, O, A)$$

访问控制表可以认为是定义 f 的一个表.为简单起见,上述表示没有考虑规则中还可能包含的约束条件,如访问时间、位置等,但这并不影响最终的分析.

4.2　ACL 的一致性

定义　ACL 是一致的是指 ACL 中不应存在两个以上的表项,指定相互冲突(矛盾)的访问控制决策.

也就是说,对 $\forall T=(S, O, A) \in V$ 和 $T'=(S', O', A') \in V$,若 $\exists s, o, a$,使得 $s \in S$ 且 $s \in S'$,$o \in O$ 且 $o \in O'$,$a \in A$ 且 $a \in A'$,即两个三元组 T 和 T' 存在交叉,则相应的两个决策 $d=f(T)=f(S, O, A)$ 和 $d'=f(T')=f(S', O', A')$ 应该是一致的(不能矛盾).

例如,最简地,假设一个典型的访问控制决策集 $D=\{$允许,拒绝$\}$,如果在 ACL 中存在两个表项,它们的三元组 T 和 T' 构成的主体、客体和操作相交,则这两个表项的访问控制决策就应该同时为"允许(或拒绝).否则就会产生不一致.

由上述定义,关于 ACL 的一致性,很容易得出下面的定理.

定理 1　如果在 ACL 中,$\forall T=(S, O, A) \in V$ 和 $T'=(S', O', A') \in V$ 满足如下条件:

$S \cap S' \neq \varnothing$ 或 $O \cap O' \neq \varnothing$ 或 $A \cap A' \neq \varnothing$

则 ACL 是一致的.

证明　设 $d=f(T)=f(S, O, A)$ 和 $d'=f(T')=f(S', O', A')$,则 d 和 d' 是 ACL 中两个表项对应的访问控制决策.

假设 $S \cap S' \neq \varnothing$,则 $\forall s \in S$(或 $s \in S'$),有 $s \notin S'$(或 $s \notin S$),由此对 s, o, a 来说 d 和 d' 只有一个起作用,不管 d 和 d' 取值如何,都不会发生矛盾,因此 ACL 是一致的.

同样地,可证明 $O \cap O' \neq \varnothing$ 或 $A \cap A' \neq \varnothing$ 的情况.

定理 2　在 ACL 中,$\forall T=(S, O, A) \in V$ 和 $T'=(S', O', A') \in V$,$f(T)=d$,$f(T')=d'$,d 和 $d' \in D$,如果

$S \cap S' \neq \varnothing$ 且 $O \cap O' \neq \varnothing$ 且 $A \cap A' \neq \varnothing$

则 ACL 是一致的,当且仅当 $d R d'$,这里 R 是 D 上的一个关系,表示两个决策彼此不冲突.

证明　先证必要性.

根据定理给出的条件,$S \cap S' \neq \varnothing$,$O \cap O' \neq \varnothing$ 和 $A \cap A' \neq \varnothing$,表明与 ACL 中的两个表项相对应的三元组 T 和 T 是相交的,也就是说 $\exists s, o, a$,使得 $s \in S$ 且 $s \in S'$,$o \in O$ 且 $o \in O'$,$a \in A$ 且 $a \in A'$,如果 d 和 d 是相互矛盾的(不一致),即 $(d, d') \notin R$,则对 s, o, a 来说将存在两个相矛盾的访问控制决策.因此 ACL 一致必须有 $(d, d') \in R$.

再证充分性.

如果 $(d, d') \in R$,则对 $\forall s, o, a$,若 $s \in S$ 且 $s \in S'$,$o \in O$

且 $o \in O'$, $a \in A$ 且 $a \in A'$,作用于 s, o, a 的访问控制决策 d 和 d 将是一致的,因此 ACL 是一致的.

因此定理成立.

4.3 一致性检查

设有一个具有 n 个表项的 ACL,ACL 中的每个表项可表示为 (S_i, O_i, A_i, d_i) 下面我们建立四个矩阵 M_S, M_O, M_A, M_d:

$$M_S = \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1n} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \phi_{n1} & \phi_{n2} & \cdots & \phi_{nn} \end{bmatrix}, \quad M_O = \begin{bmatrix} \psi_{11} & \psi_{12} & \cdots & \psi_{1n} \\ \psi_{21} & \psi_{22} & \cdots & \psi_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \psi_{n1} & \psi_{n2} & \cdots & \psi_{nn} \end{bmatrix}$$

$$M_A = \begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1n} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \varphi_{n1} & \varphi_{n2} & \cdots & \varphi_{nn} \end{bmatrix}, \quad M_d = \begin{bmatrix} \theta_{11} & \theta_{12} & \cdots & \theta_{1n} \\ \theta_{21} & \theta_{22} & \cdots & \theta_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \theta_{n1} & \theta_{n2} & \cdots & \theta_{nn} \end{bmatrix}$$

其中:

$$\phi_{ij} = \begin{cases} 1, & S_i \cap S_j \neq \varnothing \\ 0, & S_i \cap S_j = \varnothing \end{cases}; \quad \psi_{ij} = \begin{cases} 1, & O_i \cap O_j \neq \varnothing \\ 0, & O_i \cap O_j = \varnothing \end{cases}$$

$$\varphi_{ij} = \begin{cases} 1, & A_i \cap A_j \neq \varnothing \\ 0, & A_i \cap A_j = \varnothing \end{cases}; \quad \theta_{ij} = \begin{cases} 1, & (d_i, d_j) \in R \\ 0, & (d_i, d_j) \notin R \end{cases}$$

(R 是上述定义的 D 上的关系)

现在我们对这些矩阵定义一个运算.

设 $M_V = M_S \overline{\odot} M_O \overline{\odot} M_A = (\phi_{ij} \odot \psi_{ij} \odot \varphi_{ij}) = (\alpha_{ij})$,其中"$\odot$"运算是计算两个一位二进制数的布尔积,"$\overline{\odot}$"运算表示对两个矩阵中相同行和列上的两个元素执行布尔积运算,并且得到一个新的矩阵.

容易证明:如果 $M_V \overline{\odot} M_d = (\alpha_{ij} \odot \theta_{ij}) = M_V$,则 ACL 就是一致的,这是因为:对任意的 $\alpha_{ij} = 1$,由定理 2 可知,为达到两条规则 $(i$ 和 $j)$ 的一致,需保证 $(d_i, d_j) \in R$,即 $\theta_{ij} = 1$,这时布尔积 $\alpha_{ij} \odot \theta_{ij} = 1 = \alpha_{ij}$.

其次,根据定理 1 可知,对 $\alpha_{ij} = 0$,ACL 中的两条规则 $(i$ 和 $j)$ 永远都不会矛盾,不管 $(d_i, d_j) \in R$ 还是 $(d_i, d_j) \notin R$,也就是对 θ_{ij} 的值没有要求,θ_{ij} 既可以是"0"也可以是"1",但这时同样可得到 $\alpha_{ij} \odot \theta_{ij} = 0 = \alpha_{ij}$.

因此上面的陈述是正确的.

4.4 进一步说明

采用上面的一致性检查方法,当 ACL 含有上千条规则时,计算量将是相当大的. 尽管考虑到 $\phi_{ij} = \phi_{ji}$, $\psi_{ij} = \psi_{ji}$, $\varphi_{ij} = \varphi_{ji}$ 以及 $\theta_{ij} = \theta_{ji}$,计算每一个矩阵 M_S, M_O, M_A, M_d 都需要执行 $n(n-1)/2 \approx n^2/2$ 次确定集合是否相交的运算.

幸运的是,在实际系统中 ACL 的一致性检查,不需每一次都执行所有的这些运算. 管理员配置安全规则的过程,是一个从无到有,从小到大的过程. 因此一致性检查也是从一开始就在进行,ACL 中每增加一条规则,系统只需将该条规则与原有 ACL 中的规则(已经是一致的)进行相应的运算即可. 这时每个矩阵涉及的运算次数只是 $(n-1)$, n 为当前规则数. 另外,由于"$\odot$"运算的特点,对矩阵 M_S, M_O, M_A, M_d 和 M_V,并不是每个矩阵中的元素都需计算,假设 $\theta_{ij} = 0$,则 α_{ij}, ψ_{ij}, φ_{ij} 和 ϕ_{ij} 都不必计算;若 $\theta_{ij} \neq 0$,但 $\phi_{ij} = 0$,则 $\alpha_{ij} = 0$, ψ_{ij} 和 φ_{ij} 都不必计算.

如果是在 ACL 中删除一条规则,为保持安全策略的一致性,也应进行规则集的一致性检查. 设删除的规则对应的三元组为 $T = (S, O, A)$,其决策为 $f(T) = d$,那么如果删除后的 ACL 中 $\exists T' = (S', O', A')$,其决策为 $f(T') = d'$,满足 $S \cap S' \neq \varnothing, O \cap O' \neq \varnothing$ 和 $A \cap A' \neq \varnothing$,并且 $(d, d') \in R$ 则规则的删除是不完整的,这时需要对所有满足这种关系的表项 $T' = (S', O', A')$ 进行处理,处理方法可以是将每个 $T' = (S', O', A')$ 对应的规则变成一条新规则,在这条新规则中的三元组 $T'' = (S'', O'', A'')$ 满足 $S'' = S' - S$, $O'' = O' - O$, $A'' = A' - A$. 也可以采用其他处理方式,但最终的目的都是一个,即保持 ACL 的一致性. 删除规则时,如何寻找每个 $T' = (S', O', A')$,同样也是采用上述矩阵元素运算的方法,这里就不再赘述. 由此可知,无论是在 ACL 中增加规则,还是删除规则,都可在系统中实现一致性检查机制,确保 ACL 的一致性.

参考文献:

[1] Marshall D. Abrams, and Michael V. Joyce. New thinking about information technology security [J]. Computers & Security, 1995, 14(1): 69 - 81.

[2] Marshall D. Abrams, and Michael V. Joyce. Trusted computing update [J]. Computers & Security, 1995, 14(1): 57 - 68.

[3] Marshall D. Abrams, and Michael V. Joyce. Trusted system concepts [J]. Computers & Security, 1995, 14(1): 45 - 56.

[4] Jonathan Moffett, Morris Sloman and Kevin Twidle. Specifying discretionary access control policy for distributed systems [J]. Computer Communications, 1990, 13(9): 571 - 580.

[5] Tatyana Ryutov and Clifford Neuman. Representation and Evaluation of Security Policies for Distributed System Services[A]. DARPA Information Survivability Conference and Exposition[C]. Hilton Head Island, SC, USA: DISCE, 2000.

[6] C Bidan and V Issarny. Dealing with Multi-Policy Security in Large Open Distributed Systems[A]. Proceedings of 5th European Symposium on Research in Computer Security[C]. Louvain-la-Neuve, Belgium: ESRCS, 1998. 51 - 66.

作者简介:

李守鹏 男,1964 年生于辽宁省凌源,1987 年毕业于西安电子科技大学,现为四川大学数学学院应用数学专业博士生,中国国家信息安全测评认证中心信息安全实验室主任,研究兴趣是计算机与通信、信息安全.

孙红波 男,1965 年生于黑龙江省,1989 年毕业于西安电子科技大学,获硕士学位,现为北京电子科技学院副教授,研究方向为密码学、信息安全.