信息系统安全模型研究

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: 信息系统安全模型的建立是获得信息系统安全的基础. 针对信息系统来说现有的安全模型都显现出了明显的不足. 为适应当今以网络为基础的高度分布与开放的信息系统的特点,在划分安全域的基础上,分别对单域系统、简单系统和复杂系统加以研究,给出了适应于信息系统的安全模型,从而为信息系统安全奠定理论基础.

关键词: 安全模型; 安全策略; 信息系统安全; 安全域
中图分类号: TN918 **文献标识码**: A **文章编号**: 0372-2112 (2003) 10-1491-05

Research on Information System Security Models

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

Abstract: It is the basis for security of information systems to establish an information system security model. For information systems the security models in existence have showed insufficiency obviously. To adapt the properties of highly opened and distributed network based information systems today, through the definition of security domains, the new model gives study on unique domain system, simple system and complex system respectively. A model adaptable for the security of information systems is presented. Thus a basis for the security of information systems is found theoretically.

Key words: security model; security policy; information system security; security domain

1 引言

目前以网络为基础的信息系统呈现出高度分布与开放的特点,尽管安全模型的研究由来已久,并且提出了许多信息安全模型,但还没有哪一个安全模型适用于这样的系统,这些模型[1~9]都仅仅是面向问题的某一个侧面. 例如:Bell La Padula (简称 BLP) 模型[5]就是最早提出的信息安全模型之一,也是最为重要的安全模型,它针对机密信息的保护(即信息的保密性)需要,主要适用于多用户分时系统. 为适应当前信息系统的特点,本文将给出具有普遍意义的信息系统安全模型.

2 系统描述

对于复杂的分布式信息系统,由于安全策略上的复杂多样,需对系统的组成和结构进行抽象,给出必要的定义,以便提供研究上的方便.

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,安全域为系统的描述提供了一种现实灵活的手段.

任意一个信息系统都可以划分为安全域,图 1 展示了组成系统的安全域及安全域之间的关系.

图中的椭圆表示安全域,从图中可以看出一个安全域可以包含多个其他安全域(子域)如: D_1, D_2, D_3, D_4 都是整个安全域 D 的子域,并且它们还都包含更小的子域. 安全域间还可能存在交叉,如安全域 D_2, D_3 之间存在交叉,它们的交叉可用另一个安全域 D_{23} 表示.

安全域间通常还存在通信或交互,为简单起见,图中未加以表示.

安全域间的层次(包含)和交叉关系使系统安全问题变得复杂.

每个安全域 D 都可以表示为一个五元组,即 $D = (P, S, O, F, R)$,其中 P 表示域中的安全策略集, S 表示域中的主体集, O 表示域中的客体集, F 表示域中的功能集, R 表示域中的资源集. 它们的定义是:

主体 起主动作用的实体,一般与用户和其进程相对应. 全体主体的集合为 $S = \{s_1, s_2, \cdots s_n\}$,其中 $s_i (1 \leq i \leq n)$ 表示单个主体.

客体 起被动作用的实体,如文件,接口等. 全体客体的集合为 $O = \{o_1, o_2, \cdots o_m\}$,其中 $o_j (1 \leq j \leq m)$ 表示单个客体.

收稿日期:2002-06-06;修回日期:2002-12-10

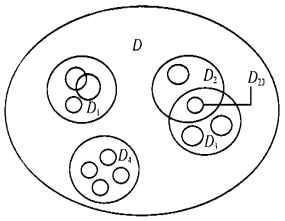

图 1 系统中的安全域

功能 主体的执行赋予系统的每一个可能的完整外在行为. 如鉴别、审计查阅等. 系统具有的全体可能的功能集合为 $F = (f_1, f_2, \cdots, f_p)$,其中 $f_i (i \leq i \leq p)$ 表示单个功能.

策略 系统应符合的安全行为要求. 如自主访问控制策略,强制访问控制策略,完整性策略,可用性策略,组织自身的特定策略等,系统需遵循的全体策略的集合为 $P = (p_1, p_2, \cdots, p_q)$,其中 $p_i (1 \leq i \leq q)$ 表示个别策略.

资源 资源是系统创建实体、执行功能的源泉. 可能的系统资源有 CPU 时间,内外存空间,通信基础等. 全体资源集为 $R(r_1, r_2, \cdots, r_h)$,其中 $r_i (1 \leq i \leq h)$ 表示单个资源.

根据组成系统的域的情况,可以将系统区分为单域系统、简单系统和复杂系统(这里只进行理论抽象,不考虑域的实际组成和规模).

单域系统 仅含单个安全域的系统称为单域系统.

简单系统 系统中含有有限个安全域,且域间无交叉和层次关系.

复杂系统 系统中含有的域的数量较多,且域间存在交叉和层次(包含)关系.

系统安全 是指系统从初始状态开始,随着时间的推移,系统所历经的每一个状态都是安全的(即:符合安全策略的要求).

3 单域系统的安全

为考察单域系统的安全,首先需定义系统的状态,然后研究系统在输入(请求)的作用下产生的输出(决策)及满足的状态转移关系. BLP 模型[5]中的系统状态描述、一般机制和相关的安全定理等都值得借鉴.

系统状态是对某一时刻系统状况的全面反映,可以从宏观和微观两个方面来描述.

系统的宏观状态 V_M 可表示为一个二元组:

$$V_M = (C_M, P_M)$$

其中,C_M 是四元组 (S_i, O_i, R_i, f_i) 组成的集合,表示系统的当前功能行为,f_i 代表系统的一个功能,$S_i \subset S$ 是 f_i 涉及的主体集合,$O_i \subset S$ 是 f_i 涉及的客体集合. $R_i \subset S$ 是 f_i 涉及的资源集.

$P_M \subseteq P$ 是系统的当前策略集.

因此 $C_M \in \mathrm{C}_M \subset \rho(S) \times \rho(O) \times \rho(R) \times F$,其中 C_M 是各种可能的当前系统行为集.

系统的宏观状态描述重在功能特性,然而系统的每一个功能都是一系列基本动作(操作)的组合,每一功能的完成都将引起更细致的系统状态的变迁,因此为了更深入地刻划系统的行为还需定义系统的微观状态.

系统的微观状态反映系统的基本操作对系统状况的影响,微观状态 V_m 是一个二元组:

$$V_m = (C_m, P_m)$$

其中,C_m 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,$R_j \subset R$ 是当前操作涉及的资源集. $P_m \subseteq P$ 是系统在当前操作集下的策略.

$C_m \in \mathrm{C}_m \subset S \times O \times \rho(R) \times A$,其中,$\mathrm{C}_m$ 是各种可能的当前操作集. 从微观状态定义可知,系统的每一个功能 $f_i (i = 1, 2, \cdots, p)$ 可表示为 $f_i \in F \subset A^T$,其中 $T = \{1, 2, \cdots, t, \cdots\}$ 为系统操作粒度上的时间序列. 系统运行过程中整个生存期可能的全部功能组成的集合可表示为:

$F_{all} = F^{\mathrm{T}}$ 的元素,其中 $\mathrm{T} = \{1, 2, \cdots, T, \cdots\}$ 为系统功能粒度上的时间序列.

为达到系统安全,必须保证系统宏观状态的变迁经过的每一个宏观状态都是安全状态,并且任何一个宏观状态内经历的每一个微观状态上的变迁也都是安全的. 由上述讨论不难看出,为保证系统安全,只需保证系统历经的一切微观状态都是安全状态. 区分宏观与微观状态的意义在于:可以从不同层次上考察系统,便于实际安全策略的说明与实现.

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. 下面仅以微观状态来研究系统的安全问题,所得出的结论对宏观状态也同样适用.

为说明系统安全,首先给出系统的形式化表示.

系统输入 系统输入也称为请求,是导致系统输出(也称决策)和状态转移的原因. 设可能的系统输入的集合为 $I = \{I_1, I_2, \cdots, I_h\}$,其中 $I_j (1 \leq j \leq h)$ 为单个请求,从粒度上讲,可分为功能上的请求或操作上的请求,这里为简单起见,不予细分.

系统输出 系统输出也称为决策,是系统在当前输入和当前状态下产生的决策,设可能的输出的集合为 $U = \{U_1, U_2, \cdots, U_l\}$,其中 $U_i (1 \leq i \leq l)$ 表示单个决策,系统决策一般为"是"、"否"、"错误"或"?"等.

由此系统的输入序列集可表示为 $X = I^T$,系统的输出序列集可表示为 $Y = U^T$,其中 $x \in X, y \in Y$,分别表示某一输入或输出序列,x_t, y_t 表示 x 和 y 中的第 t 个输入或输出.

系统的状态序列集表示为 $Z = \mathrm{V}_m^T$,其中 $\mathrm{V}_m = \mathrm{C}_m \times \mathrm{P}_m$,$\mathrm{P}_m = \rho(P)$ 为各种可能的当前策略,$z \in Z$ 为某一状态序列,z_t 表示 z 中的第 t 个状态.

系统 这里采用类似 BLP 模型的方法给出系统的形式定义.

设 $W \subset I \times U \times \mathrm{V}_m \times \mathrm{V}_m$ 表示系统的状态转移关系,系统 $\sum(I, U, W, z_0) \subset X \times Y \times Z$ 定义为:$(x, y, z) \in \sum(I, U,$

$W, z_0)$,当且仅当对每一个 $t \in T$,有$(x_t, y_t, z_t, z_{t-1}) \in W$. 这里 z_0 是系统的初态,形式一般为(Φ, P_m).

从定义可以看出,系统的状态转移关系 W 至关重要,它应反映系统的安全策略需求. 在现实系统中 W 一般要反映一组规则的要求,关于规则,说明如下:

规则　一条规则是一个函数 $\gamma: I \times V_m \rightarrow U \times V_m$,制约着系统的请求与状态向决策与新状态过渡的关系.

规则 γ 保持安全状态是指每当 $\gamma(I_i, V_m) = (U_j \times V'_m)$,且 V_m 为安全状态的话,V'_m 也是安全状态.

设 $\omega = \{\gamma_1, \gamma_2, \cdots, \gamma_e\}$ 是规则集,则可定义 ω 上的状态转移关系 $W(\omega)$ 为$(I_i, U_j, V'_m, V_m) \in W(\omega)$,当且仅当$(U_j \times V'_m) = \gamma_a(I_i, V_m)$,其中 $1 \leq a \leq e$. 该定义的前提条件是 ω 中的规则必须满足一致性.

从规则的定义看,系统的规则集是系统安全策略的集中体现,因此必须正确反映安全策略的内涵. 为保障系统的安全,由安全策略导出的安全规则集也必须满足完备、正确和一致的要求.

设任一安全策略集 $P = \{p_1, p_2, \cdots\}$,各种可能的规则的集合为 $\theta = \{\gamma_1, \gamma_2, \cdots\}$,令 $\psi = \rho(\theta)^P$,则一个完全符合策略 P 的规则集 ω 可表示为 ψ 的一个元素,即 $\omega = \psi$,并且 ψ 中的其他元素都是不符合策略 P 的规则集,因此求得该元素 ω,也就是确定准确的策略与规则的对应关系成为系统安全的关键.

安全系统　系统的一个状态 V_m 是安全状态,当且仅当 V_m 满足与状态 V_m 相关的策略 P_m(表示为 $V_m| \Rightarrow P_m$). 一个状态序列 z 是安全状态序列,当且仅当对每个 $t \in T$, z_t 是安全状态. 系统的一个表现$(x, y, z) \in \sum(I, U, W, z_0)$ 是安全表现,当且仅当 z 是一个安全状态序列. $\sum(I, U, W, z_0)$ 是安全系统,当且仅当系统的每一个表现都是安全表现.

这里的系统表现可粗略地理解为宏观状态里讲到的系统功能行为.

定理 1(一般安全定理)　任何初始状态 z_0 是安全的系统 $\sum(I, U, W, z_0)$,系统 $\sum(I, U, W, z_0)$ 是安全的系统的充要条件是,对系统的每一个动作,$(I_i, U_j, (C'_m, P'_m), (C_m, P_m))$, W 满足如下条件.

①每一个$(s, o, R_k, \alpha) \in C'_m - C_m$ 都满足安全要求,即$(s, o, R_k, \alpha)| \Rightarrow P'_m$,和

②任意的$(s, o, R_k, \alpha) \in C_m$ 若不满足安全要求,即$(s, o, R_k, \alpha)| \Rightarrow P'_m$ 不成立,有$(s, o, R_k, \alpha) \notin C'_m$.

定理 1 使 BLP 模型得到推广和一般化,证明略.

定理 2　设 ω 是保持安全的规则集,z_0 是安全的初态,则系统 $\sum(I, U, W, z_0)$ 是安全系统.

定理证明略.

定理 1 是一个广泛适用的一般安全定理,它使系统安全策略的说明更加灵活,并使我们认识到,一切系统安全问题都应将焦点集中在正确合理的安全策略制定和安全机制的实施上. 另外定理成立的前提是系统的全部安全策略都只与系统的历史有关,安全策略中不能涉及系统状态的未来信息,同样实际系统的安全机制也以不能使用系统的未来信息为条件. 由此只要制定的安全策略符合组织的预期安全目标,并且与策略相应的安全机制得到正确有效的实施,系统就是安全的.

定理 2 告诉我们为使系统保持安全,反映安全策略的规则集必须保持安全. 也就是说安全规则的定义很重要,系统中实际定义规则集时不要出现违反安全的规则.

4　简单系统的安全

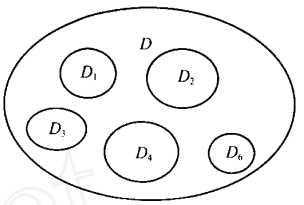

图 2　简单系统的组成

图 2 展示了简单系统的组成特点. 简单系统由 $D_1, D_2, \cdots$等多个互不交叉,互不包含的安全域构成,这些安全域中的策略相互独立. 当然整个系统也可看成一个安全域.

在简单系统中,每个域除具有域内行为外,还有域外行为,即各个域之间存在交互作用或通信. 因此最直观的想法是:简单系统的安全需要每个组成域都是安全的,并且各域间的交互作用也能保持所涉及域的安全.

简单系统中的每个安全域 D_i 都可表示为:

$$D_i = (P_i, S_i, O_i, F_i, R_i)$$

其中 P_i 表示 D_i 中的安全策略集, S_i 表示 D_i 中的主体集, O_i 表示 D_i 中的客体集, F_i 表示 D_i 中的功能集, R_i 表示 D_i 中的资源集. 除 P_i 外, P_i 中的其余四部分与单域系统中的定义没有区别.

设简单系统由 N 个域组成,即$\{D_1, D_2, \cdots, D_N\}$,由于存在域间交互,因此每个安全域 D_i 中的安全策略 P_i 都包含三个方面的内容:

①域内策略: P_i^S

②对它域作用策略: $P_i^O = \bigcup_{j=1,(j\neq i)}^{N} P_{ij}^O$,其中 P_{ij}^O表示域 D_i 对域 D_j 的作用策略.

③受它域作用策略: $P_i^I = \bigcup_{j=1,(j\neq i)}^{N} P_{ij}^I$,其中 P_{ij}^I是域 D_j 对域 D_i 的作用策略.

因此, D_i 中的策略集 $P_i = P_i^S \cup P_i^O \cup P_i^I$.

域策略的这种划分,使得安全域中策略的说明变得清晰简单,也便于在系统中分别实现相应的策略实施机制.

为研究简单系统的安全问题,可以将系统中的每一个域 D_i 都作为一个子系统来考虑. 域 D_i 对应的子系统为: $\sum_i (I_i, U_i, W_i, z_{i0})$

其中, I_i 和 U_i 为子系统 $\sum_i$ 中的请求与决策, W_i 为子系统的状态转移关系, z_{i0}为子系统的初态.

与安全策略类似,子系统 $\sum_i$ 中的请求 I_i 也可以划分为三个部分:

①子系统内部的请求: I_i^S

②对其他子系统的请求: $I_i^O = \bigcup_{j=1,(j\neq i)}^{N} I_{ij}^O$,其中 I_{ij}^O为子系统 $\sum_i$ 对子系统 $\sum_j$ 的请求.

③来自其他子系统的请求：$I_i^I = \bigcup_{j=1,(j\neq i)}^{N} I_{ij}^I$，其中 I_{ij}^I 为来自子系统 $\sum_j$ 的请求.

从而，$I_i = I_i^S \cup I_i^O \cup I_i^I$

对子系统的决策 U_i 同样可以采用上述划分方法. 这里不再细述.

子系统 $\sum_i$ 的状态可以采用与单域系统中相同的方式来定义：

子系统的宏观状态 $V_{i_M} = (C_{i_M}, P_{i_M})$，其中 C_{i_M} 是四元组 $(S_{ij}, O_{ij}, R_{ij}, f_{ij})$ 组成的集合，表示子系统的当前行为，f_{ij} 是子系统 $\sum_i$ 的一个功能，$S_{ij} \subset S_i$ 是涉及的子系统 $\sum_i$ 中的主体集合，$O_{ij} \subset O_i$ 是涉及的子系统 $\sum_i$ 中的客体的集合，$R_{ij} \subset R_i$ 是子系统 $\sum_i$ 中的当前资源集，P_{i_M} 是子系统 $\sum_i$ 的当前策略集.

子系统的微观状态 $V_{i_m} = (C_{i_m}, P_{i_m})$，其中 C_{i_m} 是四元组 $(s_{ij}, o_{ij}, r_{ij}, \alpha_{ij})$ 组成的集合，表示系统的当前操作集，α_{ij} 代表子系统 $\sum_i$ 中的一个操作，$\alpha_{ij} \in A_i = \{\alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{ik_i}\}$（$A_i$ 为子系统 $\sum_i$ 中的可能操作集），s_{ij}，o_{ij} 分别表示子系统 $\sum_i$ 中的单个主体和单个客体，$R_{ij} \subset R_i$ 是子系统 $\sum_i$ 中当前操作所涉及的资源集. P_{i_m} 是子系统 $\sum_i$ 的当前操作下的策略集.

子系统 $\sum_i$ 的状态转移关系 W_i 可表示为：

$$W_i \subset I_i \times U_i \times V_{im} \times V_{im}$$

简单系统的表示 一个简单系统 $\sum$ 可表示为：

$$\sum = \bigcup_{i=1}^{N} \sum_i (I_i, U_i, W_i, z_{i0})$$

也就是说经过上述描述的子系统，合并起便构成了完整的简单系统.

定理 3(简单系统安全定理) 一个简单系统 $\sum$ 是安全的，当且仅当简单系统在整个活动期内，组成它的每个子系统都是安全的.

该定理是显然的，因此研究简单系统的安全问题可归结为研究组成它的子系统的安全问题.

为进一步研究子系统的安全性，下面给出子系统片段的定义.

子系统片段 设 (x_i, y_i, z_i) 是子系统 $\sum_i$ 的任意一个表现，其中 $x_i \in X_i \in I_i^T$，$y_i \in Y_i \in U_i^T$，$z_i \in Z_i \in V_{im}^T$ 分别表示子系统 $\sum_i$ 的请求，决策和状态序列. 子系统片段是根据请求序列 x_i 对系统表现 (x_i, y_i, z_i) 的划分，该划分具有如下性质：

①设序列 $x_i = \{x_{i1}, x_{i2}, \cdots x_{it}, \cdots\}$，子系统请求片段序列 $\varphi_i = \{\varphi_{i1}, \varphi_{i2}, \cdots x_{ib}\}$，其中 φ_{ik}，$(1 \leq k \leq b)$ 表示子系统 $\sum_i$ 相对于表现 (x_i, y_i, z_i) 的一个片段，则任意的 $x_{ij} \in x_i$，都应有请求片段 φ_{ik} 使得 $x_{ij} \in \varphi_{ik}$，且对任意的其他请求片段序列 φ_{il}，$(l \neq k)$ 有 $x_{il} \notin \varphi_{il}$.

②任意的请求片段序列 $\varphi_{ik} = \{x_{ip}, x_{ip+1}, \cdots, x_{iq-1}, x_{iq}\}$，其中 $p \leq q$，满足：

$$\begin{cases} x_{im} \in \varphi_{ik}, \text{若 } p \leq m \leq q \\ x_{im} \notin \varphi_{ik}, \text{若 } m < p \text{ 或 } m > q \end{cases}$$

③φ_{ik} 中的任一请求 x_{im} 都具有同样的请求性质.

根据上述定义，可将 (x_i, y_i, z_i) 按请求是内部、对域外或是来自域外划分为连续的片段序列. 例如下面就是一种划分.

$$x_i = \{\underbrace{x_{i1}, x_{i2}, \cdots, x_{ip1}}_{\varphi_{i1} = I_i^O}, \underbrace{x_{ip1+1}, \cdots, x_{ip2}}_{\varphi_{i2} = I_i^S}, \underbrace{x_{ip2+1}, \cdots, x_{ip3}}_{\varphi_{i3} = I_i^I}, \underbrace{x_{ip3+1}, \cdots, x_{ip4}}_{\varphi_{i4} = I_i^S}, \cdots\}$$

其中，φ_{i1} 为对其他域的请求，φ_{i2}，φ_{i4} 为内部请求，φ_{i3} 为外部域产生的请求.

关于子系统的安全有下面的定理.

定理 4(子系统安全定理) 对于初态为 z_{i0}，且 z_{i0} 是安全的子系统 $\sum_i (I_i, U_i, W_i, z_{i0})$ 来说，子系统安全的充要条件是：子系统 $\sum_i (I_i, U_i, W_i, z_{i0})$ 的每一个片段都是安全的.

根据前面有关的系统安全定义，定理 4 也是显然的.

为保障子系统的安全，确定 W_i 应满足的条件尤为重要. 可喜的是，前面单域系统研究部分给出的系统安全定理(定理 1)，对子系统的关系 W_i 依然有效. 这是因为子系统 $\sum_i (I_i, U_i, W_i, z_{i0})$ 中，请求集 I_i、决策集 U_i 和策略集 P_i 的进一步定义和区分，没有影响到定理中的条件，如果不作区分依然可以用同样的方法得到证明.

同样的道理定理 2 的结论对简单系统中的子系统也是成立的.

另外子系统的规则集 ω_i 也可以根据子系统的请求 I_i 和策略 P_i 的三种不同类型划分为三种不同的规则集，即内部规则 ω_i^S，外来交互规则 ω_i^I 和外出交互规则 ω_i^O.

定理 4 以及上述规则集等划分的意义在于：子系统在实现不同策略的执行机制时有相对的独立性，可以采用不同的机制和技术手段实现子系统的安全功能，这为子系统的结构和设计实现带来了方便.

5 复杂系统的安全

5.1 复杂系统中的域关系

复杂系统中的安全域存在包含与交叉关系，图 3 展示了这种关系. 事实上，复杂系统的定义只是一种相对关系. 例如图 3(a) 和(b) 中的安全域包含与交叉关系，通过“屏蔽”、“合并”与“分离”的办法都可以转换成新的互不交叉的域，进而又与简单系统的定义相吻合.

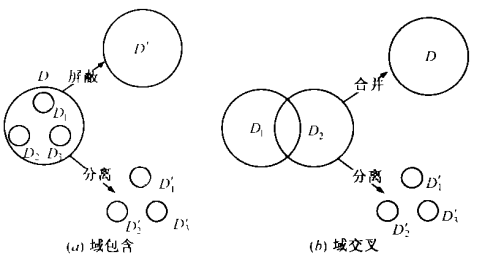

图 3 域间关系

从图 3(a) 可知：$D = \{D_1, D_2, D_3\}$，若不考虑 D 中的细节，即只在更高的层次上考虑问题，则我们可以屏蔽掉 D_1，D_2，D_3 的划分，即通过屏蔽只剩下最后一个域 D；另一方面

也可以从更低层次上考虑问题,即只考虑 D 的构成域,这样经过域的分离就只剩 D_1', D_2' 和 D_3'. 再看图 3(b),两个域 D_1 和 D_2 存在交叉,通过域合并,可以将它们变成一个域 D,且 $D = D_1 \cup D_2$,其中,"∪"表示域合并;通过域分离,又可将互相交叉的两个域 D_1 和 D_2 划分为三个无交叉的域 D_1', D_2' 和 D_3',且 $D_2' = D_1 \cap D_2$, $D_1' = D_1 - D_2'$, $D_3' = D_3 - D_2'$,其中"∩"表示取 D_1 和 D_2 的交叉部分.

通过上述处理办法,无论域间的交叉或包含关系多么复杂都可将它们变成无交叉、无包含的新的域. 从而将复杂系统简单化.

但是,必须说明的是上述"屏蔽"、"分离"和"合并"的办法不是单纯的集合运算,里面涉及到安全策略的处理. 并且这种处理办法只是为了便于对复杂系统的安全性的研究. 简单化后的系统中,应使各个新的安全域的安全策略准确反映系统原有的安全策略,并且各个新的安全域的策略都相互独立,互不影响.

实际的系统由于管理、构成等多种因素的现实状况,使这种从理论上将系统域间的关系简单化的处理难于实现. 不过幸运的是,在实际系统中也不必将系统中的域一定进行这些去"交叉"和去"包含"处理. 从理论上研究域交叉与域包含关系的目的,是为了更好地说明系统的安全策略,实现系统的策略执行机制.

复杂系统域间的"交叉"和"包含"尽管从理论上能够被简单化,但并不等于说系统就变简单了、不复杂了. 因为,复杂系统中域的规模或数量都较实际的简单系统大或多,并且复杂系统中的全体安全策略也较简单系统复杂得多.

5.2　复杂系统的安全性

复杂系统的域间关系在去除了"交叉"和"包含"后,从理论上可以用处理简单系统安全性的方式来处理,其前提是经过处理后的每个域的安全策略都能准确、完整地反映处理前的安全策略. 保持整个系统安全策略的不变性是安全策略处理方面的问题,也是实际情况中通过管理协调能够做到的.

定理 5(复杂系统安全定理)　任何组成确定的复杂系统,设系统包含的域为 D_1, D_2, …, D_N,其中,某些域间存在"交叉"或"包含"关系,在经去"交叉"和去"包含"后,系统包含的域为 D_1', D_2', …, D_M',(域间无交叉和包含关系). 系统是安全的,当且仅当:

①由域 D_1', D_2', …, D_M' 构成的简单系统是安全的,以及

②处理前后的系统策略一致,即, $\bigcup_{i=1}^{M} P_i' \Leftrightarrow \bigcup_{j=1}^{M} P_j$, ("⇔"表示策略的一致性)

定理证明略.

定理 5 为复杂系统安全性的分析提供了理论依据和思路. 若将复杂系统区分为独立的子系统(安全域),对这些子系统的安全策略表达采用更易处理的方式,并保持策略的一致性,则复杂系统的安全问题处理上便变得可行.

6　现实的系统安全

上面研究的复杂系统,要求系统的组成是确定的,即组成系统的部件在系统生命期中的持续运行阶段不会增减. 但在实际情况下,系统往往呈开放特性,即系统的组成会随时间的推移而变化,也就是存在系统组成部件的增减. 更确切地说,就是系统中的安全域有可能扩张或收缩,系统中也有可能增加新的安全域或减少安全域. 所有这些变化都要求系统的安全策略能随这些变化而调整,以便全面地反映变化后的系统的安全要求. 然而,对于规模庞大的复杂系统而言,保持整个系统安全策略完备、正确和一致并不是一件容易的事情,因此对于具有"伸缩"特性的开放的复杂系统而言,针对策略描述与处理方法[10~12]的研究就显得尤为重要.

参考文献:

[1] David F. Ferraiolo, John F. Barkley, and D. Richard Kuhn. A role based access control model and reference implementation within a corporate intranet[J]. ACM Transactions on Information Systems Security, February 1999, 2(1): 34 - 64.

[2] John McLean. Security Models and Information Flow[A]. Proceedings of the IEEE Symposium on Security and Privacy[C]. Oakland, California, IEEE Computer Society Press, 1990. 180 - 187.

[3] John McLean. Security Models, Encyclopedia of Software Engineering [M]. Marciniak J (ed.). Wiley & Sons, 1994.

[4] John McLean. A comment on the "basic security theorem" of bell and LaPadula[J]. Information Processing Letters, 1985, 20(2): 67 - 70.

[5] D E Bell, L J La Padula. Secure Computer System: Unified Exposition and Multics Interpretation [R]. The MITRE Corporation, Massachusetts, USA, ESD-TR-75 - 306, 1976.

[6] Andrew C. Myers Barbara Liskov. A Decentralized Model for Information Flow Control[A] Proceedings of the 16th ACM Symposium on Operating Systems Principles[C]. Saint-Malo, France: October 1997. 129 - 142.

[7] Ravi S. Sandhu. Lattice-based access control models[J]. IEEE Computer, November 1993, 26(11): 9 - 19.

[8] Carl E. Landwehr, Constance L. Heitmeyer and John D. McLean. A security model for military message systems[J]. ACM Trans, Comput, Syst, 1984, 2(3): 198 - 222.

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

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

[11] 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, 2000. 172 - 183.

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

作者简介:

李守鹏　(见本卷第 7 期第 980 页)

孙红波　(见本卷第 7 期第 980 页)