多级安全性政策的历史敏感性*

石文昌+, 孙玉芳

(中国科学院 软件研究所,北京 100080)

History Sensitivity of the Multilevel Security Policies

SHI Wen-Chang+, SUN Yu-Fang

(Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

+ Corresponding author: Phn: 86-10-62555043 ext 31, E-mail: rockee@sonata.iscas.ac.cn

http://www.ios.ac.cn

Received 2001-07-17; Accepted 2001-12-27

Shi WC, Sun YF. History sensitivity of the multilevel security policies. *Journal of Software*, 2003,14(1):91~96.

Abstract: Supporting for the security policy flexibility is one of the goals of modern secure operating systems. The DTOS (distributed trusted operating system) program put forth a concept of security policy lattice, which provides a good way for the research on security policy flexibility. However, it is claimed in the DTOS program's description of security policy lattice that MLS (multi-level security) policies are static policies. First, an enforcement scheme for a MLS policy is constructed theoretically, which shows that MLS policies are of history sensitivity and hence have dynamic characteristics, and so that MLS policies can not be simply taken as static policies. Then, an implementation algorithm for the constructed enforcement scheme is given. It is illustrated that the constructed enforcement scheme is of the same complexity as the ordinary enforcement scheme and so is an applicable scheme. As a result, it can be affirmed that reasonable, flexible and practically feasible schemes are available to make MLS policies to be of history sensitivity. Consequently, the improperness of the assertion that MLS policies are static policies is exhibited.

Key words: multilevel security; secure operating system; security policy flexibility; security policy lattice; history sensitivity

摘 要: 对安全政策灵活性的支持是现代安全操作系统追求的目标之一,DTOS(distributed trusted operating system)项目提出了安全政策格的思想,为安全政策灵活性的研究提供了一种很好的手段.然而,DTOS 项目给出的安全政策的格描述把多级安全性(multi-level security,简称 MLS)政策认定为静态安全政策.首先,从理论上构造 MLS 政策的一个实施策略,说明 MLS 政策具有历史敏感性,从而具有动态特征,不能简单地作为静态安全政策对待.同时,给出所构造的实施策略的实现算法,说明该实施策略与常规实施策略具有相同的复杂度,是一个实用的实施策略.由此证明,可以找到合理、灵活、实用的实施策略,使 MLS 政策具有历史敏感性,从而证明把 MLS 政

* Supported by the National Natural Science Foundation of China under Grant No.60073022 (国家自然科学基金); the National High Technology Development 863 Program of China under Grant No.863-306-ZD12-14-2 (国家 863 高科技发展计划); the Knowledge Innovation Engineering Program of the Chinese Academy of Sciences under Grant No.KGCX1-09 (中国科学院知识创新工程)

第一作者简介: 石文昌(1964－),男,广西浦北人,博士,研究员,主要研究领域为系统软件,计算机安全.

策认定为静态安全政策的不合理性.

关键词: 多级安全性;安全操作系统;安全政策灵活性;安全政策格;历史敏感性
中图法分类号: TP316 文献标识码: A

安全操作系统开发的传统措施主要集中在对特定安全政策的实现上,典型情形是按照诸如 TCSEC(trusted computer system evaluation criteria)[1]等某一标准的要求去实现固定的安全需求,这种做法已经满足不了不断出现的新的应用需要.不同的用户群体、不同的应用环境、不同的安全威胁,要求安全操作系统必须能够灵活地对范围广阔的各种安全政策进行支持.对安全政策灵活性的支持已成为现代安全操作系统追求的目标之一[2,3].由美国国家安全局等联合承担的 DTOS(distributed trusted operating system)[4]、Flask[5]和 SE-Linux[6]等相继延续的项目以安全政策灵活性为主要目标,对安全操作系统进行了富有创新性的探索,并取得了重要的成果.

DTOS 项目属于名为 Synergy[7]的一个庞大的操作系统研究项目的一部分,Synergy 项目的目标是为分布式安全系统开发一个灵活的、基于微内核的体系结构,其根本特点是实现安全政策判定与安全政策实施的分离.DTOS 项目的主要目的是以 Mach 系统[8]为基础,开发安全微内核和安全服务器的原型,实现政策灵活性、Mach 兼容性和性能低损失等目标.Flask 是 DTOS 的后继项目,当 DTOS 项目结束以后,对 Mach 系统的支持被取消,Flask 的原型是以 Fluke 系统[9]为基础建立的.Flask 以 DTOS 的安全体系结构为基础,侧重于动态安全政策的支持,解决了 DTOS 在权限撤回等方面存在的问题.SE-Linux 项目则是 Flask 的安全体系结构在非微内核结构的 Linux 系统中实现的实践.

为了支持安全政策灵活性,首先必须清楚安全政策的特征以及不同安全政策之间的关系.DTOS 项目对安全政策的特征进行了划分,提出了安全政策格的思想[10],通过格的结构描述多种安全政策之间的关系以及系统安全机制对安全政策的支持能力,为安全政策灵活性的研究提供了一个很好的手段.这些思想也是 Flask 的基础[3].强制安全控制是安全操作系统必须提供的基本机制[11],多级安全性(multi-level security,简称 MLS)是强制安全控制的重要内容,各种安全操作系统通常都实现对 MLS 的支持[3,4,6,12~16].

文献[10]借助于格的结构对包括 MLS 政策在内的多种著名安全政策进行了描述,然而,它把 MLS 政策认定为静态安全政策.本文通过安全政策的实施策略来证明这个结论的不合理性.首先,从理论上构造 MLS 政策的一个实施策略,说明 MLS 政策具有历史敏感性(见第 1.1 节的定义),从而具有动态特征,不能简单地作为静态安全政策对待.同时,给出所构造的实施策略的实现算法,说明该实施策略与常规实施策略具有相同的复杂度,是一个实用的实施策略.由此证明,可以找到合理、灵活、实用的实施策略,使 MLS 政策具有历史敏感性,从而阐明把 MLS 政策认定为静态安全政策是不合理的.

1 基础背景

1.1 安全政策格

文献[10]为安全政策定义了输入(数量)、敏感性、撤权和(权限)传递性 4 类特征.输入特征一方面描述一次安全判定所依赖的安全属性个数,另一方面描述安全判定是否需要其他参数.敏感性特征描述安全判定受输入特征以外的其他因素(历史因素、环境因素、自主因素、弃权因素)影响的情况;安全判定不受其他因素影响的安全政策定义为静态安全政策,反之,定义为动态安全政策;受历史因素(如曾经执行过的访问操作)影响的安全政策定义为具有历史敏感性.撤权特征描述撤回已授出权限的能力.传递性特征描述权限在主体间传递的可能性.

一个格由一个集合和一个关系定义.安全政策格中的集合是节点集合,每个节点对应一个由安全政策特征组成的集合.安全政策格中的关系是支配关系.安全政策与格节点的对应关系由安全政策特征确定,安全政策所具有的特征的集合与相应格节点对应的安全政策特征集合相等.安全机制也可对应到格节点上,安全机制所支持的安全政策特征的集合与格节点对应的安全政策特征集合相等.

一个安全政策对应到一个格节点也称为该安全政策位于相应的格节点上,安全机制亦类似.如果把一个安

全机制放到格节点 N 上,则意味着该安全机制能支持位于由节点 N 支配的或等于节点 N 的所有格节点上的安全政策.文献[10]给出的安全政策格描述了多种著名安全政策在格中的位置,其中定位的安全机制是整个 DTOS 系统,由 DTOS 在格中的位置可以确定该系统对安全政策的支持能力.

1.2 多级安全性政策模型

MLS政策模型的主要代表是BLP(Bell&LaPadula)模型[12]和Biba模型[17].虽然它们的内涵不同,但在表现形式上,Biba 模型与 BLP 模型是一致的.本文只讨论 BLP 模型,但得出的结论也适用于 Biba 模型.作为本文讨论的基础,我们对 BLP 模型简要描述如下(详细内容参见文献[12]).

约定 1.1. 设 X 和 Y 是任意集合,记 $P(X)$为 X 的幂集,记 $X^Y:=\{f\mid f:Y\to X\}$.

约定 1.2. $\underline{S}$是主体集合,$\underline{O}$是客体集合,A 是访问方式集合,$\underline{C}$是等级分类集合,$\underline{K}$ 是非等级类别集合,$\underline{S}_T$ 是可信主体集合,非可信主体集合 $\underline{S}'=\underline{S}-\underline{S}_T$.

定义 1.1. 当前访问集合 $B:=P(\underline{S}\times\underline{O}\times A)$.访问矩阵集合 $\underline{M}:=\{M\mid M$ 是矩阵$\wedge M$ 中元素 M_{ij} 是主体 S_i 对客体 O_j 的访问方式集$\}$.敏感标记集合 $\underline{L}:=\{(C,K)\mid C\in\underline{C}\wedge K\in\underline{K}\}$;设 $L_1=(C_1,K_1)\in\underline{L},L_2=(C_2,K_2)\in\underline{L}$,则$(L_1\geq L_2):=(C_1\geq C_2\wedge K_1\supseteq K_2)$.敏感标记函数集合 $F:=\{(f_S,f_O,f_C)\mid f_S\in\underline{L}^S\wedge f_O\in\underline{L}^O\wedge f_C\in\underline{L}^S\wedge(\forall S\in\underline{S}(f_S(S)\geq f_C(S)))\}$,$f_S$ 称为主体敏感标记函数,f_O 称为客体敏感标记函数,f_C 称为主体当前敏感标记函数.客体层次关系集合 $\underline{H}:=\{H\mid H\in[P(\underline{O})]^{\underline{O}}\wedge$性质 1$\wedge$性质 2$\}$,其中,性质 1:设 $O_i\in\underline{O},O_j\in\underline{O}$,若 $O_i\neq O_j$,则 $H(O_i)\cap H(O_j)=\varnothing$;性质 2:不存在这样一个客体集合$\{O_1,O_2,....,O_w\}$,使得对于任意 $r(1\leq r\leq w)$都有 $O_{r+1}\in H(O_r)$,其中 $O_{w+1}=O_1$.状态集合 $V:=\{(b,M,f,H)\mid b\in B\wedge M\in\underline{M}\wedge f\in F\wedge H\in\underline{H}\}$.

约定 1.3. $b(S:\underline{x},\underline{y},....,\underline{z})$表示客体集合$\{O\mid(S,O,\underline{x})\in b\vee(S,O,\underline{y})\in b\vee...\vee(S,O,\underline{z})\in b\}$.

约定 1.4. 定义 $\underline{r}$ 为可读不可写方式,$\underline{a}$ 为可写不可读方式,$\underline{w}$ 为可读且可写方式,$\underline{e}$ 为不可读且不可写(可执行)方式.

公理 1(ss-特性). 在状态 $v=(b,M,f,H)$,若对任意主体 S,以下条件成立,则状态 v 满足简单安全特性(ss-特性):

$$O\in b(S:\underline{r},\underline{w})\Rightarrow f_S(S)\geq f_O(O).$$

公理 2(*-特性). 在状态 $v=(b,M,f,H)$,若对 $\underline{S}'$中的任意主体 S,以下条件都成立,则状态 v 满足相对于 $\underline{S}'$的*-特性:

$$O\in b(S:\underline{a})\Rightarrow f_O(O)\geq f_C(S),\tag{1}$$

$$O\in b(S:\underline{w})\Rightarrow f_O(O)=f_C(S),\tag{2}$$

$$O\in b(S:\underline{r})\Rightarrow f_C(S)\geq f_O(O).\tag{3}$$

公理 3(ds-特性). 在状态 $v=(b,M,f,H)$,若对任意主体 S_i 和任意客体 O_j,以下条件成立,则状态 v 满足自主安全特性(ds-特性):

$$(S_i,O_j,\underline{x})\in b\Rightarrow\underline{x}\in M_{ij}.$$

2 多级安全性政策模型的实施策略

2.1 BLP政策模型实施策略A-BLP的构造

下面从理论上构造 BLP 政策模型的一个实施策略,我们称其为 A-BLP 实施策略.

定义 2.1. $\underline{L}$是敏感标记集合,设 $L_1=(C_1,K_1)\in\underline{L},L_2=(C_2,K_2)\in\underline{L},L_3=(C_3,K_3)\in\underline{L}$,函数 $\lambda:\underline{L}\times\underline{L}\to\underline{L}$定义为 $L_3=\lambda(L_1,L_2)$由以下原则确定:

if $(C_1!\leq C_2)$ then $C_3=C_2$ else $C_3=C_1$;

if $(K_1!\subseteq K_2)$ then $K_3=K_1\cap K_2$ else $K_3=K_1$.

定义 2.2. $\underline{L}$ 是敏感标记集合,设 $L_1=(C_1,K_1)\in\underline{L},L_2=(C_2,K_2)\in\underline{L},L_3=(C_3,K_3)\in\underline{L}$,函数 $\gamma:\underline{L}\times\underline{L}\to\underline{L}$ 定义为 $L_3=\gamma(L_1,L_2)$由以下原则确定:

if $(C_1!\geq C_2)$ then $C_3=C_2$ else $C_3=C_1$;

if $(K_1!\supseteq K_2)$ then $K_3=K_1\cup K_2$ else $K_3=K_1$.

约定 2.1. $rq(S_i,O_j,\underline{x})$表示主体 S_i 对客体 O_j 的 $\underline{x}$ 访问请求.

约定 2.2. $L_{RH}\in\underline{L}$和$L_{WL}\in\underline{L}$表示在一个进程的生存期间的两个判定参考量,它们反映以前的授权对敏感标记的影响情况,L_{RH}的初值为系统最小敏感标记值,L_{WL}的初值为系统最大敏感标记值.

规则 2.1. 在(b,M,f,H)状态,$\underline{r}\in M_{ij}\in M$,$f_C(S_i)!\geq f_O(O_j)$时,对$rq(S_i,O_j,\underline{r})$的处理:

IF $(f_S(S_i)\geq f_O(O_j)$ and $f_O(O_j)\leq L_{WL})$ THEN

① 构造f^*,使得$f_S^*=f_S,f_O^*=f_O,f_C^*(S_i)=\gamma(f_C(S_i),f_O(O_j))$,当$S\neq S_i$时,$f_C^*(S)=f_C(S)$.

② $L_{RH}=\gamma(L_{RH},f_O(O_j))$.

③ 对$rq(S_i,O_j,\underline{r})$授权,构造$b^*$,使得$b^*=\{(S_i,O_j,\underline{r})\}\cup b$,进入$(b^*,M,f^*,H)$状态.

ELSE 拒绝$rq(S_i,O_j,\underline{r})$.

规则 2.2. 在(b,M,f,H)状态,$\underline{a}\in M_{ij}\in M$,$f_C(S_i)!\leq f_O(O_j)$时,对$rq(S_i,O_j,\underline{a})$的处理:

IF $(f_O(O_j)\geq L_{RH})$ THEN

④ 构造f^*,使得$f_S^*=f_S,f_O^*=f_O,f_C^*(S_i)=\lambda(f_C(S_i),f_O(O_j))$,当$S\neq S_i$时,$f_C^*(S)=f_C(S)$.

⑤ $L_{WL}=\lambda(L_{WL},f_O(O_j))$.

⑥ 对$rq(S_i,O_j,\underline{a})$授权,构造$b^*$,使得$b^*=\{(S_i,O_j,\underline{a})\}\cup b$,进入$(b^*,M,f^*,H)$状态.

ELSE 拒绝$rq(S_i,O_j,\underline{a})$.

规则 2.3. 在(b,M,f,H)状态,$\underline{w}\in M_{ij}\in M$,$f_C(S_i)\neq f_O(O_j)$时,对$rq(S_i,O_j,\underline{w})$的处理:

IF $(f_S(S_i)\geq f_O(O_j)$ and $f_O(O_j)\leq L_{WL}$ and $f_O(O_j)\geq L_{RH})$ THEN

⑦ 构造f^*,使得$f_S^*=f_S,f_O^*=f_O,f_C^*(S_i)=f_O(O_j)$,当$S\neq S_i$时,$f_C^*(S)=f_C(S)$.

⑧ $L_{RH}=\gamma(L_{RH},f_O(O_j))$,$L_{WL}=\lambda(L_{WL},f_O(O_j))$.

⑨ 对$rq(S_i,O_j,\underline{w})$授权,构造$b^*$,使得$b^*=\{(S_i,O_j,\underline{w})\}\cup b$,进入$(b^*,M,f^*,H)$状态.

ELSE 拒绝$rq(S_i,O_j,\underline{w})$.

规则 2.1、规则 2.2 和规则 2.3 组成了本文给出的 A-BLP 实施策略.这 3 条规则不作单独实施用.它们是一个整体,同时实施,不同的规则针对不同的访问请求进行相应的处理.规则中的L_{RH}和L_{WL}是指进程S_i的参考量.每个进程都有自己的参考量.

可以从理论上证明[18],规则 2.1、规则 2.2 和规则 2.3 是满足 BLP 公理的,因而 A-BLP 实施策略能够满足 BLP 公理的要求,是 BLP 政策模型的一个正确的实施策略.

2.2 A-BLP实施策略的意义

A-BLP 策略是以公理 2 为中心进行构造的,公理 2 依据主体的当前敏感标记$f_C(S)$和客体的敏感标记$f_O(O)$这两个安全属性进行访问授权判定.在常规的安全操作系统实现中,进程(主体)的当前敏感标记一旦确定之后,在进程的整个生存期内是不会改变的[13].A-BLP 策略的特点是允许进程的当前敏感标记在不违反安全原则的条件下进行合理的调整.

在常规实施策略的访问授权判定中,当检查到公理 2 中的相应条件得不到满足时,就作出拒绝访问的判定.A-BLP 策略也和常规策略一样进行判定,但当第 1 次检查发现公理 2 的相应条件不满足时,并不立即作出拒绝访问的判定,它将试图调整进程当前敏感标记的值,检查能否满足条件再下结论.组成 A-BLP 策略的 3 条规则的主要思想是设计一套在公理 1 和公理 3 得到满足而公理 2 得不到满足的情形下访问判定的处理措施,3 条规则假设的前提都是满足公理 1 和公理 3 的条件而不满足公理 2 的条件.

为了在第 1 次判定不满足公理 2 的条件下有可能修改当前敏感标记的值但又不违反安全原则,需要考虑进程的历史访问行为,因而引入了 L_{RH} 和 L_{WL} 这两个反映进程历史访问行为的参考量.常规策略局限于在一次判定中确定进程是否满足 BLP 公理,而 A-BLP 策略则着眼于确保进程在整个生存期内满足 BLP 公理.

与常规策略相比,A-BLP 策略为实际应用提供了较大的灵活性.无论不管采取什么方法,在给进程的当前敏感标记赋值时,很难预测进程以后的资源访问行为,所赋的值极有可能无法满足合法的资源访问要求.允许系统根据进程活动的实际场景对当前敏感标记进行合理的动态调整,可解决赋值不准确和合法访问受拒绝的问题.

显然,按照 A-BLP 策略,在进行安全判定时,除了以$f_C(S)$和$f_O(O)$这两个安全属性为依据以外,还要考虑 L_{RH}

和 L_{WL} 这样的历史因素.所以,MLS 政策具有历史敏感性,不能简单地认定为静态安全政策.

3 A-BLP 实施策略的实现算法

本节设计 BLP 模型对 $rq(S_i,O_j,\underline{x})$访问请求进行判定的两个实现算法,目的是验证 A-BLP 实施策略应用于实际系统的可行性,并说明该实施策略的系统实现的复杂度是很低的.算法中给出的 $L_{RH}i$ 和 $L_{WL}i$ 中的 i 用于强调相应的参考量是进程 S_i 的参考量.

算法 1. 常规实施策略的安全判定.

```
L01:IF   (x 不在 Mij 中)   THEN   return(拒绝访问);   /*公理 3*/
L02:SWITCH   (x) {   /*根据请求类型分别处理*/
L03:CASE r:   /*只读*/
L04:   IF   fS(Si)!≥fO(Oj)   THEN   return(拒绝访问);   /*公理 1*/
L05:   IF   fC(Si)≥fO(Oj)   THEN   return(授权访问);   /*公理 2*/
L06:   ELSE   return(拒绝访问);
L07:CASE a:   /*只写*/
L08:   IF   fO(Oj)≥fC(Si)   THEN   return(授权访问);   /*公理 2*/
L09:   ELSE   return(拒绝访问);
L10:CASE w: /*读写*/
L11:   IF   fS(Si)!≥fO(Oj)   THEN   return(拒绝访问);   /*公理 1*/
L12:   IF   fO(Oj)=fC(Si)   THEN   return(授权访问);   /*公理 2*/
L13:   ELSE   return(拒绝访问);}.
```

算法 2. A-BLP 实施策略的安全判定.

```
L14:IF   (x 不在 Mij 中)   THEN   return(拒绝访问);   /*公理 3*/
L15:SWITCH   (x) {   /*根据请求类型分别处理*/
L16:CASE r:   /*只读*/
L17:   IF   fS(Si)!≥fO(Oj)   THEN   return(拒绝访问);   /*公理 1*/
L18:   IF   fC(Si)≥fO(Oj)   THEN   {LRHi=γ (LRHi,fO(Oj));return(授权访问);}   /*公理 2*/
L19:   ELSE   IF   fO(Oj)≤LWLi   THEN   /*规则 2.1*/
L20:              {fC(Si)=γ (fC(Si),fO(Oj)); LRHi=γ (LRHi,fO(Oj)); return(授权访问);}
L21:          ELSE   return(拒绝访问);
L22:CASE a:   /*只写*/
L23:   IF   fO(Oj)≥fC(Si)   THEN   {LWLi=λ (LWLi,fO(Oj)); return(授权访问);}   /*公理 2*/
L24:   ELSE   IF   fO(Oj)≥LRHi   THEN   /*规则 2.2*/
L25:              {fC(Si)=λ (fC(Si),fO(Oj)); LWLi=λ (LWLi,fO(Oj)); return(授权访问);}
L26:          ELSE   return(拒绝访问);
L27:CASE w:   /*读写*/
L28:   IF   fS(Si)!≥fO(Oj)   THEN   return(拒绝访问);   /*公理 1*/
L29:   IF   fO(Oj)=fC(Si)   THEN   /*公理 2*/
L30:      {LRHi=γ (LRHi,fO(Oj)); LWLi=λ (LWLi,fO(Oj)); return(授权访问);}
L31:   ELSE   IF   (fO(Oj)≤LWLi 且 fO(Oj)≥LRHi)   THEN   /*规则 2.3*/
L32:              {fC(Si)=fO(Oj); LRHi=γ (LRHi,fO(Oj)); LWLi=λ (LWLi,fO(Oj)); return(授权访问);}
L33:          ELSE   return(拒绝访问);}.
```

算法 1 是常规实施策略的一个实现算法,算法 2 是 A-BLP 实施策略的实现算法.显然,算法 2 是通过对算法 1 进行适当扩展后得到的.L06,L09 和 L13 分别扩展为 L19~L21,L24~L26 和 L31~L33.值得注意的是,L05,L08 和 L12 也分别扩展为 L18,L23 和 L29~L30,可以证明,这后一类扩展也是确保系统不违反安全原则所必须的.

观察易知,算法 1 和算法 2 的最大执行步数都是常数,其时间开销的增长率[19]都是 $O(n^0)$,即 $O(1)$,算法 2 的空间开销也只是增加了两个占用空间很小的判定参考量,因此,算法 2 与算法 1 的复杂度是相同的,即 A-BLP 实施策略的实现算法与常规实施策略的实现算法具有相同的复杂度.

4 结束语

本文从理论上给出了 MLS 政策的一个实施策略(A-BLP 策略),并通过实现算法说明了 A-BLP 策略实际应用的可行性,实现算法表明,A-BLP 策略的复杂度与常规实施策略相同,不会增加系统的复杂性.A-BLP 策略表

明 MLS 政策具有历史敏感性,阐明了文献[10]在安全政策格的描述中所给出的 MLS 政策是静态政策的结论的不合理性.所幸的是,出于对其他安全政策的支持考虑,文献[10]把 DTOS 系统放在安全政策格的一个能支持历史敏感性的节点上,否则,是难以确保 DTOS 系统能完全支持 MLS 政策的.A-BLP 策略在我们开发的一个安全操作系统[20]中得到了实施,该系统通过了公安部的安全产品检测,满足中国国家安全标准[21]第三等级的要求.这从实践上进一步验证了 A-BLP 实施策略的可行性和本文得到的结论的正确性.

References:

[1] Department of Defense. Department of defense trusted computer system evaluation criteria. DoD 5200.28-STD, Washington, DC, 1985.

[2] Olawsky D, Fine T, Schneider E, Spencer R. Developing and using a policy neutral access control policy. In: Proceedings of the New Security Paradigms Workshops. New York: ACM Press, 1996. 60~67.

[3] Spencer R, Smalley S, Loscocco P, Hibler M, Anderson D, Lepreau J. The flask security architecture: system support for diverse security policies. In: Proceedings of the 8th USENIX Security Symposium. Berkeley, CA: USENIX Press, 1999. 123~139.

[4] Secure Computing Corporation. DTOS lessons learned report. Technical Report, No.CDRL-A008, Secure Computing Corporation, 1997.

[5] Secure Computing Corporation. Assurance in the Fluke microkernel: final report. Technical Report, No.CDRL-A002, Secure Computing Corporation, 1999.

[6] Loscocco P, Smalley S. Integrating flexible support for security policies into the Linux operating system. Technical Report, NSA and NAI labs, 2001.

[7] Saydjari OS, Turner SJ, Peele DE, *et al*. Synergy: a distributed, microkernel-based security architecture. Technical Report, No.R231, INFOSEC Research and Technology, 1993.

[8] Loepere K. OSF mach kernel principles. Technical Report, Carnegie Mellon University, 1993.

[9] Ford B, Hibler M, Lepreau J, Tullmann P, Back G, Clawson S. Microkernels meet recursive virtual machines. In: Proceedings of the Symposium on Operating Systems Design and Implementations. New York: ACM Press, 1996. 137~151.

[10] Secure Computing Corporation. DTOS generalized security policy specification. Technical Report, No.DTOS-CDRL-A019, Secure Computing Corporation, 1997.

[11] Loscocco PA, Smalley SD, Muckelbauer PA, Taylor RC, Turner JS, Farrell JF. The inevitability of failure: the flawed assumption of security in modern computing environments. In: Proceedings of the 21st National Information Systems Security Conference. Gaithersburg, MD: NIST Press, 1998. 303~314.

[12] Bell DE, LaPadula LJ. Secure computer system: unified exposition and Multics interpretation. Technical Report, No.MTR-2997-Revision 1, The MITRE Corporation, 1976.

[13] Gligor VD, Chandersekaran CS, Chapman RS, *et al*. Design and implementation of secure Xenix. IEEE Transactions on Software Engineering, 1987,SE-13(2):208~221.

[14] Flink II CW, Weiss JD. System V/MLS labeling and mandatory policy alternatives. AT&T Technical Journal, 1988,(5-6):53~64.

[15] Grenier GL, Holt RC, Funkenhauser M. Policy vs mechanism in the secure TUNIS operating system. In: Proceedings of the 1989 IEEE Symposium on Security and Privacy. Los Alamitos, CA: IEEE Computer Society Press, 1989. 84~93.

[16] Waldhart NA. The army secure operating system. In: Proceedings of the 1990 IEEE Computer Society Symposium on Research in Security and Privacy. Los Alamitos, CA: IEEE Computer Society Press, 1990. 50~60.

[17] Biba KJ. Integrity considerations for secure computer systems. Technical Report, No.ESD-TR-76-372, Electronic Systems Division, Air Force Systems Command, 1977.

[18] Shi WC, Sun YF, Liang HL. An adaptable labeling enforcement approach and its correctness for the classical BLP security axioms. Journal of Computer Research and Development, 2001,38(11):1366~1372 (in Chinese with English abstract).

[19] Lewis HR, Papadimitriou CH. Elements of the theory of computation. 2nd ed., Prentice-Hall, Inc., 1998.

[20] Shi WC, Sun YF, Liang, HL, *et al*. Design and implementation of secure Linux kernel security functions. Journal of Computer Research and Development, 2001,38(10):1255~1261 (in Chinese with English abstract).

[21] National Quality and Technologies Supervision Bureau. Classified criteria for security protection of computer information system. GB17859-1999, Beijing: China Standards Press, 1999 (in Chinese).

附中文参考文献:

[18] 石文昌,孙玉芳,梁洪亮.经典 BLP 安全公理的一种适应性标记实施方法及其正确性.计算机研究与发展,2001,38(11):1366~1372.

[20] 石文昌,孙玉芳,梁洪亮,等.安全 Linux 内核安全功能的设计与实现.计算机研究与发展,2001,38(10):1255~1261.

[21] 国家质量技术监督局.计算机信息系统安全保护等级划分准则.GB17859-1999,北京:中国标准出版社,1999.