改进增强型高安全等级操作系统 SECIMOS 的设计与实现*)

袁春阳[1,2] 李 琳[3] 石文昌[1] 梁洪亮[1] 贺也平[1]
(中国科学院软件研究所北京 8718 信箱 北京 100080)[1]
(中国科学院研究生院 北京 100049)[2] (青岛大学信息工程学院 青岛 266071)[3]

摘 要 本文介绍了以 Linux 系统为基础,通过改进/增强法来开发高安全等级操作系统 SECIMOS,给出了系统的实现框架,全面论述了其中的核心技术,如访问控制、追责机制、保证方法等。分析了改进/增强法存在的问题,指出应探讨更适合今后开发最高安全等级操作系统的方法。

关键词 安全操作系统,访问控制,保证,TCSEC

The Design and Implementation of High-level Secure Operating System SECIMOS by the Way of Improvment/Enhancement

YUAN Chun-Yang[1,2] LI Lin[3] SHI Wen-Chang[1] LIANG Hong-Liang[1] HE Ye-Ping[1]
(Institute of Software, Chinese Academy of Science, Beijing, P. O. Box 8718, 100080)[1]
(Graduate School of the Chinese Academy of Sciences, Beijing 100049)[2]
(Information Engineering College of Qingdao University, Qingdao 266071)[3]

Abstract SECIMOS, a high-level secure operating system based on Linux is developed by the way of improve/enhance. Its design and implementation is discussed in this paper. The system framework and key techniques, including access control, accountability and assurance methods, are given. The problems of the "improve/enhance" approach are analyzed. It indicates new suitable approach to develop the highest-level secure operating system should be researched further.

Keywords Secure operating system, Access control, Assurance, TCSEC

1 引言

早期开发安全操作系统时,主要是针对不同的具体威胁而设计和实现的。自从安全评估标准 TCSEC[1] 颁布以后,越来越多的研发工作采用了面向安全评估标准的方法,目标是取得更高安全等级的操作系统。这主要是因为高安全等级操作系统已经可以从理论上证明其安全性。具体开发安全操作系统时,常用的方法是仿真法和改进/增强法。由于改进/增强法不会破坏原系统的体系结构,开发代价小,并能很好地保持原操作系统的用户接口和系统效率,因而得到更广泛的采用。

我们结合安全操作系统的特点,按照国家信息安全保护等级划分准则 GB17859-1999[2],利用开源代码的优势,在 Linux 系统的基础上,采用改进/增强法,研制和开发出了目标为符合第四级要求(等同于 TCSEC 标准 B2 级)的高安全等级操作系统 SECIMOS。

本文第 2 节概述了 SECIMOS 系统及其整体框架。第 3 节给出了包括自主访问控制、强制机密性访问控制、受控特权框架和特权用户的访问控制机制的策略及实现。第 4 节讲述了追责策略(如用户身份标识与鉴别、审计、可信路径)的关键技术。第 5 节重点介绍了保证方法中的形式化规范与验证、隐蔽信道分析以及残余信息保护。第 6 节分析了相关工作、改进/增强开发方法存在的问题。最后,总结全文。

2 SECIMOS 系统概述和整体框架

在 SECIMO 系统中,除了标识与鉴别、以访问控制列表 ACL(Access Control List)实现的自主访问控制[3]、以改进的 BLP 模型[4] 为基础的机密性强制访问控制、安全审计[5]、安全管理(特权用户)、可信进程(受控特权框架 CPF[6])和采用安全注意键 SAK(Security Attention Key)的方法实现的可信路径机制等功能之外,还提供形式化规范与验证[7]、隐蔽信道分析和残余信息保护等安全功能和保证,从而构造了高保证、高可信的安全操作系统。

为了防止安全策略机制被绕过,我们以 LSM(Linux Security Module)[8] 为基础实现访问控制模块,以便在内核中进行决策裁决。同时,随着多种安全需求的不断增加,目前对安全操作系统的体系结构的要求,也由支持单一安全策略发展到要支持多安全策略(如 DGSA、DTOS 体系结构)和动态安全策略(如 Flask、GFAC 体系结构)。因而,在整体结构框架设计中,对 SECIMOS 系统的开放性、可扩展性和兼容性提出了更高的要求。内核部分对 LSM 机制进行了扩充,在系统的设计和实施中充分实施了模块化思想,各种不同的安全访

*)本课题得到国家 863 高技术研究发展计划项目(2002AA141080)、国家自然科学基金项目(No. 60073022)和 No. 60373054)和中国科学院研究生院创新资金的资助。**袁春阳** 博士研究生,CCF 会员,主要研究方向为系统软件与信息安全;**李 琳** 硕士,讲师,硕士生导师,主要研究方向为网络通信与信息安全;**石文昌** 博士,研究员,硕士生导师,主要研究方向为系统软件和计算机安全;**梁洪亮** 博士,副研究员,硕士生导师,主要研究方向为计算机安全与系统软件;**贺也平** 博士,研究员,博士生导师,主要研究方向为系统软件安全技术、安全协议的设计与分析。

问控制模型能够以 Linux 可加载内核模块的形式实现，用户可以根据其需求选择适合的安全模块加载到 Linux 内核中[9]，从而大大提高了各个安全策略之间的独立性和灵活性。SECIMOS 安全操作系统的整体结构如图 1 所示。

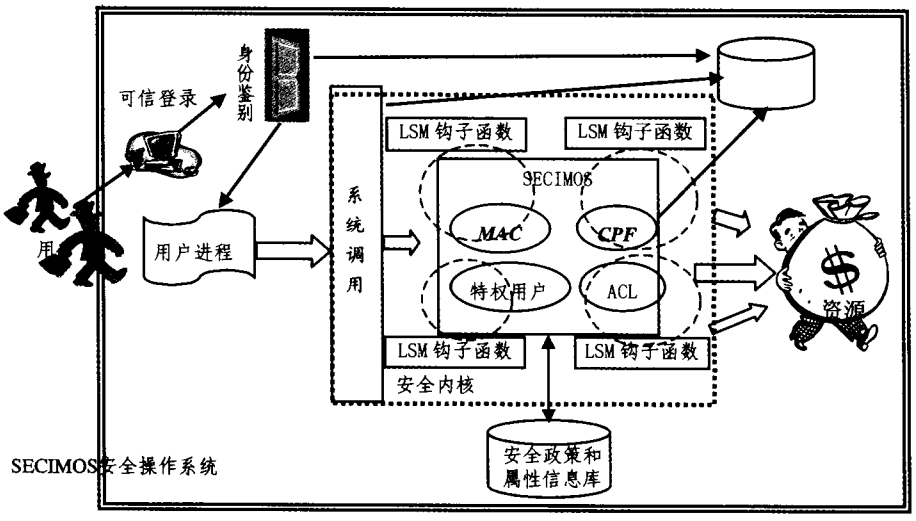

图 1　SECIMOS 整体结构

3　访问控制策略和机制

访问控制是安全操作系统进行安全防护的基本方法，主要有两种方式：自主访问控制和强制访问控制。前者由客体属主或授权用户决定某一主体对文件、目录等客体的访问权限；权限一经确定，将作为以后判断该主体对客体是否具有以及具有什么权限的有效依据。后者实施了全局性的控制策略。为了系统的可用性，通过受控特权框架实现了可信进程机制。为责任分离原则实施了一种简化的 RBAC 模型，划分出不同的用户角色。

3.1　自主访问控制

SECIMOS 系统的自主访问控制系统结构组件是在传统 Linux 的自主访问控制的基础之上，通过加入访问控制表（ACL），对自主访问控制进一步强化[3]。每一个文件类客体都有一个与之相联的访问控制列表，该列表记录了哪些用户、用户组以及未指定的用户所具有的对该客体的访问权限。当用户要访问该客体时，将根据 ACL 判断其是否有访问客体的相应权限。

自主访问控制功能能够阻止非授权用户对客体的访问。通过提供完善的系统资源授权设置，将访问控制的粒度细化到单个用户。对没有访问权限的用户，只能允许由授权用户指定对客体的访问权限。普通用户只能够改变其所拥有的客体的自主属性，管理员能够改变任何客体的自主属性。自主访问控制模块对主体、客体的权限加以细化。将 Linux 中原有的三种访问权限“读、写、执行”扩充为 12 种，即读、写、执行、创建、删除、重命名、硬连接、符号连接、读属性、修改属性、加载和卸载。为保持同现有 Linux 中自主访问机制的兼容，实现时遵循 POSIX. 1e 标准[10]。对客体的访问控制信息的存储采用“分散式”存放的方式，利用扩展属性（Extended Attribute）机制存储客体的访问控制列表。

3.2　强制机密性访问控制

强制机密性访问控制模块实施了基于改进的 BLP 模型[4]的强制访问控制策略。实施强制访问控制的基础是主客体的敏感标记（包括等级分类和非等级类别），敏感标记由 256 个等级分类和 64 种非等级类别组成。可信计算基 TCB（Trusted Computing Base）提供了两个敏感标记之间的比较函数。当主体的敏感标记高于客体的敏感标记时，主体不能读取客体的信息。当主体的敏感标记低于客体的敏感标记时，主体不能向客体写入信息。对于网卡客体，设定其敏感基本范围，只有在该范围内的数据报才可传输，并通过 CIPSO 协议[12]在数据报的 IP option 中添加安全标签来传输敏感标记。

3.3　受控特权框架

可信进程负责进行安全数据库维护、超越全局安全策略的必要操作和系统管理及维护等安全相关操作，系统安全正常的运行依赖于这些操作。错误或恶意地执行这些操作，会危及到整个系统的安全。例如，对安全数据库的错误配置会使得安全核做出错误的访问决策，将导致泄密或关键信息被破坏等严重后果。由于这些操作的重要性，系统通常要求进行这些操作的主体要具有相应的特权。

SECIMOS 系统通过实现受控特权框架 CPF（Controlled Privilege Framework）来实施可信进程机制[6]。受控特权框架将围绕最小特权原则，限制进程的特权。引入状态机制，将进程特权与进程功能、进程用户、进程状态（当前工作）三者关联在一起。以进程（程序）逻辑为中心进行特权控制；以进程特权相关属性为依据，对进程生命周期进行划分；进程在各个阶段中根据进程逻辑分配不同的特权，用户特权属性仅作为一种全局性的约束。

3.4　特权用户

为了限制 Linux 操作系统的超级用户的权限，定义了安全管理员（维护系统安全政策，创建、更新和维护系统内核等）、系统管理员（维护用户账号、文件和内核模块）、系统操作员（负责系统日常维护操作，例如系统开启和关闭、装载和卸载文件系统）、审计管理员和审计操作员等特权用户管理角色。同时保证这些特权用户之间不会通过 su 等系统调用进行用户身份转变，以防止同一个用户同时拥有多个特权角色。

4　追责策略和机制

自用户登录系统后，为系统分配了唯一标识。利用可信路径机制在用户与 TCB 之间的初始登录与鉴别会话建立可

信的通信路径，防止特洛伊木马模仿应用进程来欺骗用户或窃取用户信息。通过安全审计机制记录用户的操作，以便进行事后分析，并追踪到用户身份。

4.1 用户身份标识与鉴别

用户身份识别和鉴别的数据主要包括用户名、用户标识UID和用户口令。UID唯一标识了用户，并保证其在系统生命周期内不会发生变换，以便为访问控制、安全审计提供依据，进行访问裁决和用户追责。通过插件式认证模块(Pluggable Authentication Module)[11]来限制用户口令长度和时效，增强口令强度。利用智能卡，进一步保证用户登录的安全性。

4.2 审计机制

总共审计94个系统调用，这些系统调用涵盖进程控制、文件系统控制、扩展属性控制、IPC控制、用户管理等几大类。同时包括了与隐蔽信道相关的系统调用。每个审计事件主要记录以下四个方面的内容。

- 审计事件基本信息：时间、审计号、审计类别、审计记录大小等；
- 审计事件主体信息：主要是事件发出的用户；
- 客体信息：事件作用的客体(可以是文件、进程等)、安全级别；
- 事件特殊信息：参数等。

审计共设定了五种审计规则。

- 无条件审计：审计员不能设定或改变的审计事件，包括与审计子系统有关的所有审计，如审计打开、审计关闭、审计配置文件修改等。
- 全局审计：审计员设定适用于系统内所有主、客体的审计事件集合。
- 主体审计：为某些主体分别设定审计事件集合，包括根据主体的标识、安全级别、组标识等。
- 客体审计：为某些客体分别设定审计事件集合，包括根据客体标识和安全级别来设定。
- 安全级别审计：根据主体和客体安全基本设定审计条件。

这五种审计规则的优先次序从高到低，主体审计、客体审计、安全级别审计是对全局审计的补充。

存放审计记录文件时需要自主访问控制和强制访问控制策略的共同安全保护。同时，设定只能由授权的特权用户才能够修改和查看审计记录，防止非授权用户破坏和篡改审计记录。为审计功能专门设置两个被授权的特权用户：审计管理员主要负责关闭审计、修改和加载审计配置；审计操作员能够以只读方式查看审计记录和关闭审计。

4.3 可信路径

SECIMOS系统中，可信路径机制能够为用户与TCB之间的初始登录与鉴别会话建立可信的通信路径，防止特洛伊木马模仿应用进程来欺骗用户或窃取用户信息。为了确保可信路径只能由用户激活，采用了安全注意键SAK(Secure Attention Key)机制。SAK是终端驱动程序能够检测到的特殊键的组合。该机制的基本思想是：当且仅当用户按下SAK时，在用户与TCB之间建立才可信的通信路径。我们通过修改初始启动程序，并对终端的读写加以控制，从而实现了终端和X Windows方式下的可信路径机制。

5 安全保证及其他

安全保证说明安全操作系统开发是如何达到其安全目标的，是保证安全功能能够满足预期目标的重要手段，也是开发高安全等级操作系统的必备条件。本文主要介绍其中最重要的两个保证措施：形式化规范与验证、隐蔽信道分析。SECIMOS系统实现的强制访问控制策略和完整性访问控制策略都使用了形式化的安全策略模型[7]，采用共享资源矩阵(SRM)法，按照资源耗尽类、事件计数类、强制访问控制相关类等隐蔽信道划分方法对系统中的存储隐蔽信道进行了全面的分析，并进行了相应的带宽测算，同时进行了相应的处理。而客体重用控制确保系统的存储介质以某种方式在系统主体之间分配或者重新分配时，在排除明确的共享之外防止敏感信息的泄漏。

5.1 形式化规范与验证

形式化安全策略模型FSPM(Formal Security Policy Model)是高等级安全操作系统的基础和前提，它为系统设计与开发提供一个精确的、无二义性的系统安全策略描述。我们为SECIMOS系统建立了一个明确定义的形式化安全策略模型FSPM[7]，并对FSPM进行了内部一致性的形式化证明，即证明其所定义的操作规则(安全如何实施的)确实使所定义的模型安全，满足关键的安全需求("安全"的定义)。对系统实现是否确实实施了形式化安全策略模型，采用了非形式化的方式进行解释说明，即表明系统实现与形式化模型要求是一致的。如图2所示。

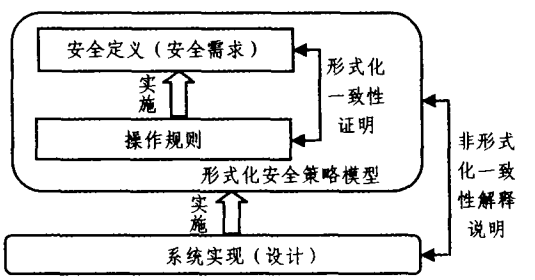

图2 形式化规范与验证

建模工具选用了形式化建模语言——Z(Zed)语言。采用的形式化方法是面向模型的方法，基于一阶谓词逻辑和集合论。计算模型为非限定的状态转换机。我们为强制访问控制子系统建立的安全策略模型如图3所示。

5.2 隐蔽信道分析

隐蔽信道是指按常规不会用于传送信息但却被用来泄漏信息的信息传送通道，一般分为两种：存储隐蔽信道和时间隐蔽信道。根据安全保护等级划分准则第四级的要求，我们重点研究了存储隐蔽信道，对存储隐蔽信道的标识、带宽计算和处理方法如下。

标识隐蔽信道的主要目的是确定隐蔽信道。存储隐蔽信道的特点是：利用系统中存在的共享资源，一个进程来改变它的值，而另一个进程观察这种改变从而传递信息。根据此特点，SECIMOS系统使用SRM矩阵描述系统资源和行为。矩阵的行代表系统行为，矩阵的列代表用户可以共享的系统资源，矩阵中的元素代表操作是否能够读写相应的资源。构造SRM矩阵时，分析系统中所有的系统调用，并适当地选择某些系统调用作为系统行为。忽略需要特权才可以调用的系统调用，如关机等。根据标识出的存储隐蔽信道的特点，将其分为三类：资源耗尽类(10个)、事件计数类(6个)和MAC冲突类(4个)。

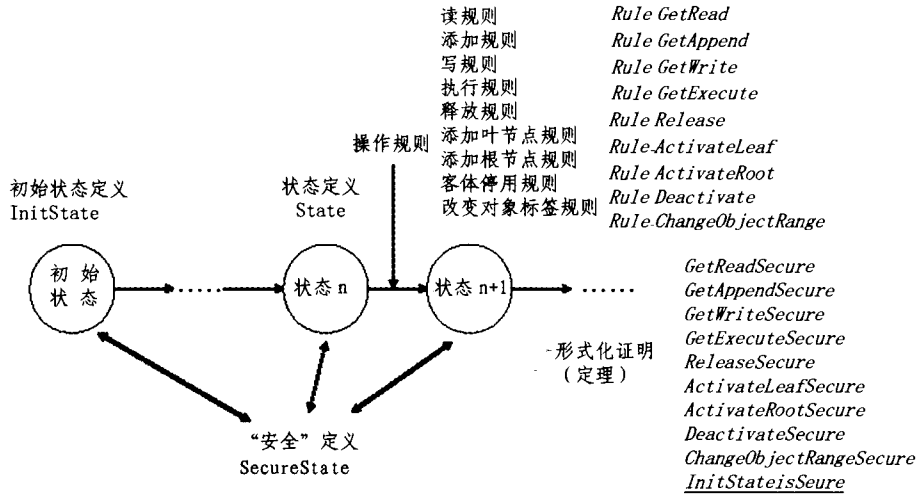

图 3 强制访问控制子系统的安全策略模型

带宽计算是理论估算和工程计算出隐蔽信道传输信息的最大带宽。我们采用工程计算的方法，即分别使用 GNU 的 time 程序和内核提供的 profile 功能。测量构成存储隐蔽信道的系统调用在被调用多次的情况下所用时间总和，计算平均值，得出隐蔽信道的最大带宽。

对于隐蔽信道的处理要考虑到系统所处的安全环境、要保护的数据本身的特征和系统本身的特征，采取了消除隐蔽信道、限制带宽和审计使用的方法。如：通过修改内核代码，随机分配 pid，消除 pid 相关的事件计数类存储隐蔽信道；通过限制调度周期内进程最小运行时间，将最大带宽限制在 2.5bps以下；给安全审计子系统提供相应审计点，审计所有可能引起存储隐蔽信道的系统调用。

5.3 残余信息保护

残余信息保护功能为 SECIMOS 系统提供客体重用控制。在 SECIMOS 系统中，TCB 在其空闲存储客体空间中对客体初始指派、分配或再分配给一个主体之前，撤销客体所含信息的所有授权。当主体获得对一个已被释放的客体的访问权时，当前主体不能获得原主体活动所产生的任何信息。我们分别对文件、目录、管道、设备、信号量、共享存储区、消息队列、访问控制表、外部介质、内存、寄存器等客体进行分析设计，确保解决这些客体的重用问题，最终确保 SECIMOS 客体的所有原有信息，都在客体分配或者回收时被清除。

6 相关工作和讨论

目前以 Linux 系统为基础的安全操作系统中，最高安全级别为 RedHat 公司开发的 RedHat Linux 企业版，达到 CC 标准 EAL4＋级要求（基本等同与 TCSEC 标准 C2 级）[13]。我们一直试图使用改进/增强法来开发更高安全等级的操作系统，并探讨其中的关键技术和方法，如本文所述 SECIMOS 的目标为 TCSEC B2 级。而且这种开发方法，亦有成功之举。目标为实现 TCSEC 标准的 B2-A1 安全操作系统。IBM 公司以 SCO XENIX 系统为原型而开发和设计的 Secure XENIX 系统[14]，后又更名为 Trusted XENIX。Trusted XENIX 3.0 于 1992 年被美国国家安全局正式评估为 B2 级产品[15]。但是尽管如此，这种以 Linux 系统为基础的改进/增强法仍存在如下问题：

(1)形式化验证问题。Linux 作为宏内核的典型代表，其内核过于庞大复杂，为形式化验证工作带来了很多困难，无法对包括内核在内的整个 TCB 进行验证。为此，我们只能从增加的安全策略模型出发，进行模型规范和模型内在的一致性验证。

(2)隐蔽信道问题。Linux 内核中存在过多共享变量，为存储隐蔽信道创造了有利条件。

(3)系统编写语言问题。C 语言本身非安全语言，存在安全隐患，会导致系统其他安全漏洞，如缓冲区溢出。如目标为达到 TCSEC 标准 B3 级的安全 TUNIS 系统[16]采用了可验证的强类型语言 Turing Plus 编写。A1 级系统 TRW 公司的 ASOS（Army Secure Operating System）系统[17]使用了更严谨的 Ada 语言来实现。

(4)保证过程问题。高安全等级操作系统的开发除了对安全功能的要求增加外，更多地侧重了安全保证的要求，而保证贯穿于从系统设计初始到系统生命周期终结的整个过程。由于 Linux 开发的初始目的并未侧重安全性，因而改进/增强法不能从开发的初始阶段对系统安全性进行保证。

(5)底层硬件问题。在划分可信计算基时，除了软件部分外，还应包括提供其支撑的硬件。而如何确保硬件的可信性，也是开发高保证安全操作系统的关键问题之一。

可以看出，以 Linux 为基础的改进/增强的开发方法很难开发出满足 GB17859 标准最高级（第五级）要求的安全操作系统。所幸的是，微内核（如 Minix3、L4 等）、虚拟监视机（如 VMWare，Virtual Server，Xen 等）提供了更小、更易验证的内核，而可信计算可保证硬件可信性。所有这些条件都为今后最高安全操作系统的开发提供了基础。

结论 本文论述了如何从操作系统的核心结构层出发，按照国家安全标准，开发高安全等级的操作系统，以及其中的关键技术和实现方法。但在以 Linux 系统为代表的开源软件为基础的改进/增强型开发方法中，存在无法避免的问题，很难满足最高安全等级操作系统的需求。探讨一条适宜最高安全等级操作系统的开发方法是今后我们研究的重点。

参 考 文 献

1 CSC-STD-001-83. Department of Defense Standard. Department of Defense Trusted Computer System Evaluation Criteria. DoD Computer Security Center. August 1983

2 GB17859-1999. 计算机信息系统安全保护等级划分准则. 1999

（下转封四）

(上接第 292 页)

3 孙亚楠，石文昌，梁洪亮，等. 安全操作系统基于 ACL 的自主访问控制机制的设计与实现. 计算机科学，2004，31(7)：153～155

4 Bell D. Secure Computer Systems: A Network Interpretation. In: Proceedings of 2nd Aerospace Computer Security Conference, McLean, Virginia, 1986. 32～39

5 陈慧，石文昌，梁洪亮. 通过扩展 LSM 框架构建审计支持机制. 计算机科学，2005，32(3)：144～147

6 梁彬，孙玉芳，石文昌，等. 一个受控特权框架的设计与实现. 软件学报，2004，15(增刊)：74～82

7 Zhou Z Y, Liang B, Jiang L, et al. A Formal Description of SECIMOS Operating System. In: Third International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security (MMM-ACNS 2005), St. Petersburg, Russia, LNCS, Vol 3685. Sep. 2005. 286～297

8 Wright C, Cowan C, Morris J, et al. Linux security modules: General security support for the linux kernel. In: Proceedings of the 11th USENIX Security Symposium

9 Wu Y J, Shi W C, Liang H L, et al. Security On-Demand Architecture with Multiple Modules Support. In: The First Information Security Practice and Experience Conference (ISPEC 2005), Singapore, Lecture Notes in Computer Science, Vol 3439. Springer-Verlag, Apr. 2005. 121～131

10 IEEE Standard Department. Portable Operating System Interfaces (Security APIs) (Posix. 1e). 1998

11 Morgan A G, Kukuk T. The Linux-PAM Module Writers' Guide. The Linux-PAM Guides. http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/Linux-PAM MWG. html. 2006

12 IETF CIPSO Working Group. Commercial IP Security Option (CIPSO 2.2). July 1992

13 List of evaluated products. http://www.commoncriteriaportal.org/public/consumer/index.php? menu=4. 2006

14 Gligor V D, Burch E L, Chandersekaran C S, et al. On the Design and the Implementation of Secure XENIX Workstations. In: Proceedings of the 1986. IEEE Symposium on Security and Privacy, Apr. 1986. 102～117

15 National Computer Security Center. Final Evaluation Report TIS Trusted XENIX version 3.0. Report No. 92/001. April 1994

16 Grenier G L, Holt R C, Funkenhauser M. Policy vs. Mechanism in the Secure TUNIS Operating System. IEEE Symposium on Security and Privacy, 1989. 84～93

17 Waldhart N D. The Army Secure Operating System. In: IEEE Computer Society Symposium on Research in Security and Privacy, 1990. 50～60

计算机科学

(1974 年 1 月创刊)
第 34 卷第 8 期 (月刊)
2007 年 8 月 25 日出版

国际标准连续出版物号 ISSN 1002－137X
国内统一连续出版物号 CN50—1075/TP

定价：30.00 元　国外定价：5 美元
邮发代号：78—68
发行范围：国内外公开

主管单位：国家科学技术部
主办单位：国家科技部西南信息中心
编辑出版：《计算机科学》杂志社
重庆市渝北区北部新区洪湖西路 18 号　邮政编码：401121
电话：(023) 63500828　E-mail：jsjkx@swic.ac.cn
网址：www.jsjkx.com
社　长：牟炳林
总　编：彭　丹
主　编：朱宗元
主编助理：徐书令
印 刷 者：重庆科情印务有限公司
总发行处：重庆市邮政局
订 购 处：全国各地邮政局
国外总发行：中国国际图书贸易总公司 (北京 399 信箱)
国外代号：6210—MO