可信操作系统研究*

谭 良[1]，周明天[2]
(1. 四川师范大学 计算机学院，成都 610066；2. 电子科技大学 计算机科学与工程学院，成都 610054)

摘 要：简要回顾了安全操作系统的发展历史，指出了安全操作系统当前存在的主要问题；在此基础上提出了可信操作系统的概念，分析了可信操作系统的特点、内涵以及与安全操作系统的关系；最后提出了可信操作系统需要解决的问题，为下一步将要开展的工作奠定基础。

关键词：安全操作系统；可信操作系统；可信计算
中图分类号：TP311　　**文献标志码**：A　　**文章编号**：1001-3695(2007)12-0010-06

Research of trusted operating system

TAN Liang[1], ZHOU Ming-tian[2]
(1. *School of Computer, Sichuan Normal University, Chengdu* 610066, *China*; 2. *School of Computer Science & Engnineering, University of Electronic Science & Technology of China, Chengdu* 610054, *China*)

Abstract: The developing history of the SOS(security operating system) was reviewed in brief, some problems in the SOS were pointed out. Moreover, the conception of the trusted operating system(TOS) was put forward, and the characters, the connotation of the TOS and the relationship between the TOS and the SOS were addressed. Finally, presented and discussed some requirements for the TOS, which needed to be resolved.

Key words: security OS(SOS); trusted OS(TOS); trusted computing

0 引言

根据计算机软件系统的组成，软件安全可划分为应用软件安全、数据库安全、操作系统安全和网络软件安全。在数据库中，DBMS 通常是建立在操作系统之上的，若没有操作系统安全机制的支持，数据库就不可能具有存取控制的安全可信性。在网络环境中，网络的安全可信性依赖于各主机系统的安全可信性；而主机系统的安全性又依赖于其上操作系统的安全性。因此，若没有操作系统的安全性就没有主机系统的安全性，从而就不可能有网络系统的安全性。计算机应用软件均建立在操作系统之上，它们均是通过操作系统完成对系统中信息的存取和处理。因此，若没有操作系统的安全性，就不可能有应用软件信息处理的安全性。操作系统安全是计算机系统软件安全的必要条件[1~5]。

操作系统实质是一个资源管理系统，管理处理机、存储器、设备、文件和作业等计算机资源，用户通过它获得对资源的访问权。安全操作系统的目的是保证它所管理资源的安全性，包括机密性、完整性和可用性等。信息的保密性是为了防止信息在非授权情况下的泄露；信息的完整性是为了保护信息不被非法窜改或破坏。

1972 年，作为承担美国空军的一项计算机安全规划研究任务的研究成果，J. P. Anderson 等人在一份研究报告[6]中提出了引用监控机(reference monitor)、引用验证机制(reference validation mechanism)、安全核(security kernel)和安全建模(modeling)等重要概念，并提出了开发安全操作系统总的指导思想(原则)。J. P. Anderson 等人指出，要开发安全系统首先必须建立系统的安全模型。安全模型给出安全系统的形式化定义，正确地综合系统的各类因素。这些因素包括系统的使用方式、使用环境类型、授权的定义、共享的客体(系统资源)、共享的类型和受控共享思想等。这些因素应构成安全系统的形式化抽象描述，使得系统可以被证明是完整的、反映真实环境的、逻辑上能够实现程序的且受控执行的。完成安全系统的建模之后，再进行安全核的设计与实现。这一原则表明，要开发安全操作系统必须完成两大任务，即访问控制框架的建立和安全模型的建立。四十多年来，安全操作系统的开发一直遵循这一原则，在访问控制框架和安全模型方面取得了丰硕的成果。在访问控制框架方面有基于政策描述语言的 FAM(flexible authorization manager)[7]和企业间多协调框架[8]、基于安全属性的 GFAC 框架[9~12]、基于统一模型的数据库 FMP[13]、RBAC[14]、Flask[15,16]框架。在安全模型方面包括 BLP、HUR、UNIX system V/MLS、BIBA、信息流的格模型、不干扰模型、CW 模型、中国墙模型、RBAC 和 DTE 等[17]。

纵观安全操作系统将近四十年的发展历史可以发现，安全操作系统的主要应用范围仍然是在国防和军事领域，在商用和

收稿日期：2006-11-19；修返日期：2007-01-29　基金项目：国家"863"计划资助项目(863-104-03-01)；2003 年度四川省科技攻关资助项目(03GG007-007)

作者简介：谭良(1972-)，男，四川泸县人，博士，主要研究方向为信息安全、中间件(tanliang2008cn@126.com)；周明天(1939-)，男，广西容县人，教授，博导，主要研究方向为网络计算、信息安全、分布并行处理.

民用领域尚未有成熟的安全操作系统出现[7]。迄今为止,整个国际上安全操作系统的实际应用并不成功。在实际应用中发挥作用的操作系统绝大部分不是安全操作系统。文献[18,19]认为,安全操作系统在商业和民用领域的不成功,主要是因为安全操作系统缺少灵活性和兼容性,降低了系统性能和效率,应发展专用安全操作系统。文献[17,20~24]认为,当前安全操作系统不成功的本质原因是安全操作系统存在诸多不完善的地方,如对多安全政策的支持;对动态多安全政策的支持,包括政策切换、权限撤销等方面;对环境适应性的支持等。

1　安全操作系统的发展历史

早在 20 世纪 60 年代,对操作系统安全的研究就引起了众多机构(尤其是美国军方)的重视。至今人们已在这个领域付出了几十年的努力,开展了大量的工作,取得了丰硕的成果,逐渐建立起了比较完善的安全操作系统理论体系。安全操作系统的研究大致可分为四个阶段[20]:第一阶段开始于 1967 年,标志是 Adept-50 系统的启动,这一时期是基本思想、技术和方法的探索时期,创建了安全操作系统的基本理论;以可信计算机系统评估准则(TCSEC)的颁布为标志的第二阶段,开始于 1983 年,安全评估标准的颁布,对安全操作系统的设计和评估起到了很大的指导作用;1993 年进入以多政策为特点的第三阶段,这一时期研究的重点是如何实现多种安全策略的共存问题;目前,安全操作系统的研究已发展到第四阶段——动态策略时期。为了支持多种策略的灵活配置需要进一步研究新的体系结构。

2　安全操作系统存在的问题

2.1　对用户身份的鉴别过程容易受攻击

当前,一些主流的操作系统通过简单的口令来确认用户身份。这种鉴别是单向的和不安全的。例如,对于 UNIX 或 Linux 操作系统,每个用户有一个注册名和一个口令,每个注册名对应一个用户身份标志号。在系统内部使用的是用户身份标志号,它可标志进程启动者和文件拥有者的身份。用户的注册名和口令保存在系统的口令文件中,在注册过程中,系统将用户提供的口令用加密算法(一般是 DES)进行加密,然后与口令文件中的相应口令密文作对比。如果口令正确则用户身份得以鉴别通过,注册进程便为用户启动口令文件相应项中指定的初始 shell。又如,对于 Windows 操作系统,用户名和口令存在 SAM 文件中,尽管它在系统运行时受操作系统保护,即使是超级用户也无法直接打开,但仍然有四种获取 SAM 数据的方法:a)在另外一个操作系统中对目标系统分区进行操作,再把 SAM 文件拷贝到软盘中;b)拷贝 Windows 磁盘修复工具(命令 RDISK)SAM 文件拷贝;c)从 SAM 中直接抽取口令密码的哈希值;d)涉及网络用户名/密码交互的窃听。黑客一旦获得 SAM 文件,通过破解就可以获得用户的口令。

对于一些安全计算机,在开机和用户登录方面加强了鉴别力度,采用了双因素认证,包括智能卡、USBKey,甚至还采用了指纹、虹膜等认证方式。采用这些方式有两个共同的特点:密钥或特征码是放在操作系统的文件系统中;采用的是单向认证。因而,与普通操作系统一样,存在着相同的安全隐患。

2.2　传统访问控制技术不能解决开域授权问题

传统的访问控制理论表现为一种关口控制的概念,如图 1 所示。不让不符合条件者进去,但是一旦取得进入的资格和权利,在范围内的活动行为就无法监管了,进入后想做什么就做什么。究其原因,主体对客体的访问和行为是根据预定的授权和身份识别来决定的。授权一旦确定,不看主体的表现,也不考查主体行为的可信性,直到另外一次授权的改变。对重要信息的授权人操作行为的忽视往往是信息流失事件多发的根源,即使专用业务系统有一定的流程控制和系统审计措施,对信息目标的流动管理往往在授权之后却没有通用的控制方法。这是由授权机制本身所限制的。

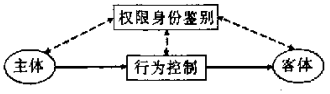

图 1　传统的行为控制

一般来讲,一个具备操作权限的人对信息的流动性操作过程,往往可分解为以下步骤:a)授权人行为主体身份权限的获取;b)授权人对行为主体的控制;c)行为主体对行为目标的权限获取;d)行为动作的执行;e)行为结果。

以 Windows 操作系统中的重要文件拷贝行为为例,其具体过程为:内部人员通过操作系统的身份认证获得系统应用的使用权(系统核心服务提供的应用如 Explorer 应用程序激活)、重要信息的文件读权限(系统文件目录服务对 Explorer 应用程序的权限开放)、目标文件拷贝载体(如移动 U 盘)的访问与权限;当这些权限均被操作者(经过认证后的操作系统当前用户)获得后,操作者利用这些权限实施重要文件向移动 U 盘的拷贝。在通常情况下,这种授权权限在操作系统身份认证后,往往不再对当前用户进行资源访问的限制,即便一些专用应用进行了文件加密的控制、文件目录的权限控制,但对于已获取控制权的人而言,拷贝的行为以及拷贝的目标存储介质往往是不受约束的,可以称这种授权为开域授权。目前基于 Windows 操作系统之上的应用系统,往往难以控制这类开域授权后的行为结果。这就是各类信息流失事件的行为模式。

2.3　操作系统不能确认自身的完整性

操作系统自身的身份和完整性是否可信也需要得到确认。比如现在有很多病毒和黑客可以直接通过硬件对操作系统进行破坏。由于提供服务的操作系统和代表用户运行的进程共享计算机系统的主存,操作系统需要自我保护防止用户对其有意或无意的破坏。自我保护意味着一方面操作系统软件自身是可信的,不能含有可能被恶意用户利用的安全漏洞导致软件运行偏离预期的轨迹;另一方面操作系统不能在用户使用的过程中被恶意用户窜改。

2.4　操作系统的安全政策不能适应环境的变化

操作系统必须应用于真实环境才可以发挥其作用。真实

安全环境有两个特征:a)安全威胁是多种多样的,有木马、病毒、蠕虫、拒绝服务攻击和缓冲区溢出攻击等。它们可能威胁信息的机密性、完整性和可用性。因此要求安全操作系统支持政策的多样性,满足多种安全目标。b)安全环境是变化的:一种是周期性的变化,如银行下班前与下班后的安全政策不一样,下班后对资源访问的控制更严格;一种是环境的突发性变化,如发生了黑客入侵事件、火灾、地震或战争等。另外,对于系统内信息的访问许可在某些情况下,可能会随着时间的变化而变化。例如某报社第二天将要出版的新闻稿件在第二天早上9:00之前是敏感信息,而在9:00之后它就是公共信息;又如对于战场上的军事情报,在决策的某一时刻前是敏感信息,过后就是普通信息了等。因此,操作系统必须能及时响应环境变化,将适合于新环境的安全政策立即有效地实施。

2.5 操作系统中的部分主体与客体之间缺少相互验证

安全操作系统中的主体是能够发起行为的实体,如进程等,主体发出访问操作,是存取要求的主动方。客体是主体行为的承担者,通常意义的客体包括文件、目录、共享内存、消息、信号量、管道、存储器、缓冲器、磁盘和外部设备等。客体的概念还可以推广到所有的对象概念上,客体甚至还可以是进程。进程本身也可以被操作,作为行为的受体。

值得注意的是,大部分安全操作系统在处理客体时,没有区分具体的客体类型,将不同类型的客体采用同一方法标志和处理。在传统的访问控制技术条件下,部分主体对某类客体的访问还存在一些问题。一方面,部分主体在访问某类客体时需要验证客体的身份和完整性。传统的访问控制技术显然不能办到。例如在Windows操作系统中,有许多DLL,DLL文件是Windows的基础,因为所有的API函数都是在DLL中实现。DLL没有程序逻辑,由许多功能函数构成,它并不能独立运行,一般由进程加载并调用。当作为主体的进程调用某DLL文件中的功能函数时,如果主体不对该DLL文件进行身份认证和完整性检验,则黑客或攻击者可以采用如下两种方法实施攻击:用一个伪造的DLL文件替换原DLL文件;伪造一个接口相同但包含恶意代码的功能函数供主体调用。这两种情况该主体均不能辨别,如图2所示。

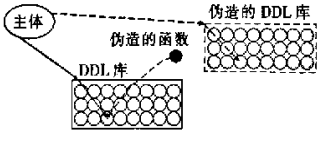

图2 主体访问伪客体

另一方面,客体均是静态客体。静态客体具有如下特征:a)客体仅仅是主体的行为对象,一旦主体拥有权限,客体只能完全"接受"主题的行为,客体没有根据自身的实际情况与主体的访问进行协商的权利。b)客体不能验证主体的身份。例如,不管是Windows、UNIX还是Linux,管道都是一种进程(主体)之间进行通信的有效机制。当主体1将用户名和密码通过管道客体传给主体2时,黑客或攻击者可以伪装为主体2接收主体1传过来的机密信息而不被察觉。如果管道客体在传递信息之前对主体的身份进行认证,那么黑客或攻击者就会被发现,如图3所示。在Windows中,黑客或攻击者通过RunAs服务进程利用命名管道来提升权限就是属于这种情况的典型案例。

图3 主体与伪主体之间的通信

2.6 用软件方式实现的安全核不能满足安全操作系统发展的需要

安全操作系统一个重要的特征是安全核。安全模型的实施及相关接口和配置数据均在安全核中。到目前为止,绝大部分的安全操作系统,包括SE-Linux等均是用软件的方式实现安全核的,并要求满足如下原则:a)必须具有自我保护能力;b)必须总是处于活跃状态;c)必须设计得足够小,以便于分析和测试,从而能够证明它的实现是正确的。第一个原则保证引用验证机制即使受到攻击也能保持自身的完整性;第二个原则保证程序对资源的所有引用均得到引用验证机制的仲裁;第三个原则保证引用验证机制的实现是正确的和符合要求的。

在安全操作系统发展的初期,安全政策少、安全模型小、安全核和内核交互少,因而安全核小,满足这三个原则相对容易。但是随着安全操作系统的发展和威胁的增多,现在的安全操作系统不仅要支持多个安全政策,而且还需要安全政策之间的动态切换,并适应环境变化。此时的安全核复杂、庞大。从软件工程的角度看,尽管可以采用模块化和面向对象的设计方法、CGFA或微内核体系结构等来降低代码实现中存在的错误,但错误不能避免。总之,安全核越复杂、越庞大,就越不能保证自身的正确性和完整性,黑客和攻击者就更容易攻击安全核。Greg Hoglund的论文(http://phrack.org/phrack/55/P55-05)提出了一种通过修改可信计算基(trusted computing base,TCB)来违反安全引用监视器(SRM)的访问控制,并给出了一个具体的NT rootkit实现(http://www.megasecurity.org/ Tools/Nt_rootkit_all.html)。

在TCSEC标准中,将所有实现安全核的软件和硬件统称为TCB。实际上,安全核中的引用监控机、引用验证机制、安全建模等关键部分均是用软件来实现的。将对安全核的信任建立在软件的基础上显然是有缺陷的。因为软件总存在错误,软件不能自己保护自己(存在悖论)。安全操作系统需要增加一个新的、用硬件实现的信任源或其他机制,对安全核实施正确性和完整性验证。

3 可信操作系统的概念

从上面的分析可以看出,在安全操作系统面临的这些问题中,安全操作系统不适应环境变化这一问题是可以在安全操作系统的框架内解决的,如文献[17,25]对该问题进行了有益的探索并提出了新的访问控制框架。其余的问题在安全操作系统框架内是不容易解决的,如对于安全操作系统的开域授权问题。因为安全操作的访问控制模型只能解决一次行为是否安全的问题,而不能解决用户整个行为是否可信。安全操作系统

的目的是为用户提供一个安全的环境,而不能保证处于该安全环境中的用户行为是否可信。目前,随着新的安全技术的兴起,可信计算正成为人们关注的热点[26-32]。可信计算[33]一般是指通过对计算机硬件的加强,以允许软件利用某一安全特征堵塞来自硬件的独立操作。典型地,可信计算包括加密硬件以允许检测(通过用户和其他人)对软件未授权的修改以及加密系统组件的通信信息。一个可信计算平台应该提供至少三个基本特征,即保护能力、完整性测量和完整性报告。从可信计算的概念及其特征可以发现,安全操作系统面临的这些问题是可以用可信计算技术来解决的。

在给出可信操作系统的概念之前,首先解释什么是计算环境中的可信。本文沿用文献[34]提出的观点。文献[34]还提出了可信计算的体系结构,包括可信终端、终端可信应用、可信网络连接、可信网络服务器、可信交易以及可信评测方法和管理方法等。

定义 1　一个可信实体(包括组件、系统或过程)的行为在任意操作条件下是可预测的,并能很好地抵抗应用程序、病毒以及一定的物理干扰造成的破坏。

定义 2　可信操作系统是能够通过支持多种安全政策来适应环境变化,并保证在系统中的本地用户或远程实体的行为总是以预期的方式和意图发生的。客体内容是真实、保密和完整的,以及自身完整性的操作系统。

TCB 安全核的概念对于操作系统和其上建设的数据库管理系统(DBMS)以及应用系统均是有意义和有效的。在操作系统的安全观念中,也按照角色(如管理员、超级用户、普通用户等)给予一定的引用系统服务的权限。也就是说,在 TCSEC 中提出的可信概念实际上是一种特权概念。只要是系统的管理员或超级用户,就认为是可信的。

与 TCB 比较起来,可信操作系统具有三个特点:主体行为的可信性;客体内容的保密性、完整性和可信性;自身的完整性。

对于主体行为的可信性,可以从两个方面进行理解:a)指行为历史记录反映主体行为是否违约、违规、违法、超越权限以及超范围等统计特性,如违约概率、错误概率等。这样的行为可信性可以划分级别。行为可信性是可信认证的基础,它在传递中通常要受到损害。b)行为可信性还可以定义为考察行为的预期性。也就是说,不可信行为对于从事某项工作来说不是必要的或预期的。从某种意义上说,可以把破坏行为的预期性理解为破坏行为的可信性或对行为的可信性产生怀疑。从理论上讲,一个业务应用系统的终端系统的功能范围应该仅仅是该业务应用终端所必需的功能。但实际上,现代计算机应用系统的终端大量地利用个人计算机,一个服务器中通常要存放多个应用系统。应用服务器或数据服务器提供的功能远远超过了应用系统所要求的,也即一个专用的应用系统建立在一个通用的网络系统上。作为一个终端,在使用应用系统时,各项服务都是通过设计的规定服务实现的。这时终端是相对安全的,它不多做一件事情。但是终端不仅仅是应用系统终端,它使用 Windows 操作系统;而作为一个 Windows 操作系统用户,采用同一台计算机、同样的计算机识别、同样的系统域网络授权,对系统的服务器进行访问,它可以做任何想做的事情。从这个意义上讲,这台计算机是非常不安全的。从计算机终端来看,同一个操作人员和同一台计算机的行为包括业务应用系统规定的操作,也包括非业务系统规定的操作,甚至还包括犯罪的操作。对于这样的可信计算机终端,采用通常意义上的访问控制、授权、口令、识别等方法将不再有效。因此必须采用行为监管的方法。

客体内容的机密性(confidentiality)是指防止信息泄露给未授权的用户;完整性(integrity)是指防止未授权用户对客体内容的修改;客体内容可信性研究的是客体内容的真实性。在现实社会中,内容的真实性是通过内容与现实的一致性或言语的证据来考察的。内容真实性的考察是一项困难的工作,需要花费大量的人力和物力来进行调查研究或取证。但是在计算机世界中,实现内容的真实性不可以向现实世界一样去考察。简单地说,在计算机世界中,判断内容是否可信,首先判断信息来源是否可信,如果信息来源可信并真实,那么对信息进行处理的行为本身也必须是可信的;即使行为可信,也不一定表明内容进行增加、删减或修改一定是正确的。但是,如果行为监控充分细致,这种增加、删减和修改也是有可信来源的,包含在行为的输入条件中,那么便满足了内容变化的可信要求。

自身的完整性就是操作系统自己不被非法窜改,如现在有很多病毒和黑客可以直接通过硬件对操作系统进行破坏。由于提供服务的操作系统和代表用户运行的进程共享计算机系统的主存,操作系统需要自我保护,防止用户对其有意或无意的破坏。自我保护意味着一方面操作系统软件自身是可信的,不能含有可能被恶意用户利用的安全漏洞导致软件运行偏离预期的轨迹;另一方面操作系统不能在用户使用的过程中被恶意用户窜改。因此系统需要有一个硬件支持的无条件信任的根来代表可信的第三方,从最底层的硬件开始,对系统中每一个启动和运行实体的身份进行鉴别。从系统启动的最初就保证是可信的,并且在每一次运行实体控制权转移过程中系统均是可信的,这样才能保证操作平台是值得信任的。系统需要建立起一条操作系统—硬件—用户的完整的信任链关系。在信任传输的作用下,实现安全机制的整体性检查,从而确保了各环节的可信性,进而保证了整个系统的可信性。

归根结底,可信操作系统必须确保用户身份、主体行为的可信性,这是对使用者的信任;确保信息存储、处理、传输的机密性、完整性、真实性,确保服务及应用程序的完整性,体现了客体内容的可信;确保密钥操作和存储的安全;确保系统具有免疫能力,从根本上防止病毒和黑客。

4　可信操作系统与安全操作系统的关系

可信与安全之间是有联系的。对于行为主体来说,可信的行为一定是安全的行为,但安全的行为不一定是可信的行为。可信与安全之间也是有区别的。对于行为主体,一次行动可以得到安全或者不安全的结论,但是得不到信任或者不信任的结论;信任只有通过经常的行动才能体现。因此可以说,安全是

行动的结果,信任则是行为的结果。下面说明可信操作系统与安全操作系统的关系。

a)可信操作系统与安全操作系统是有联系的。安全操作系统是可信操作系统的基础,如图4所示,没有安全操作系统就没有可信操作系统。可信操作系统不是主观臆造的,是在安全操作系统的基础上发展起来的。安全操作系统中的安全政策、访问控制体系结构及其实现方式仍然适合可信操作系统。

b)可信操作系统是安全操作系统的进一步发展。目前,安全操作系统面临诸多问题,如用户身份鉴别、安全属性即时撤销(或安全授权即时撤销)、环境适应性、主体行为可信和客体内容可信、自身完整性等。在这些问题中,安全属性即时撤销(或安全授权即时撤销)、环境适应性等仍然可以沿着安全操作系统的研究路线,在安全操作系统的安全框架内加以解决。但其他问题的解决,如用户身份鉴别、主体行为可信、客体内容可信、自身完整性等,必须转换观念,跳出传统安全操作系统的研发思路,找到新的解决办法。从本质上说,传统安全操作系统的可信是角色的特权,而可信操作系统中的可信是一种信誉,这本身就是一种进步,如图5所示。

图4 安全操作系统是可信操作系统的基础

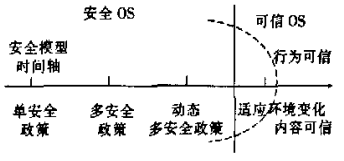

图5 可信操作系统是安全操作系统的进一步发展

c)可信操作系统与安全操作系统是有区别的。直观地说,可信操作系统研究的是如何为用户提供一个可信的计算环境;安全操作系统研究的是如何为用户提供一个基础安全平台。可信的内涵和外延均包括了安全。因而,可信操作系统的研究方法与思路不能完全沿用于安全操作系统,必须要创新和拓展。从研究的内容来看,安全操作系统主要研究在操作系统中实现单安全模型、多安全模型、动态多安全模型的方法及体系结构等;可信操作系统主要研究操作系统中的主体行为可信和客体可信以及自身的完整性。安全操作系统的目标是保证信息的机密性、完整性和可用性;可信操作系统除了这些之外,还包括主体行为的可预测性和可控性、客体真实性和自身完整性等。它们之间的关系如图6所示。

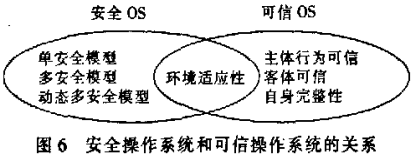

图6 安全操作系统和可信操作系统的关系

5 实现可信操作系统需要解决的问题

需求1 可信操作系统的体系结构

要实现可信操作系统,必须建立可信操作系统的体系结构。该体系结构既可以满足安全操作系统的需要,又可以为用户身份鉴别、自身的完整性、主客体身份鉴别和环境适应性提供保障。为了实现这些功能,必须扩展安全核,增加信任根。但是安全核如何扩展、信任根如何增加、需要什么样的体系结构等问题是摆在可信操作系统面前的重要问题之一。

需求2 自身完整性

操作系统是一组控制和管理计算机硬件及软件资源、合理地对各类作业进行调度以及方便用户的程序集合。随着黑客技术的发展,破坏操作系统内核程序、修改核心数据结构、替换操作系统关键代码等事件时有发生。如果操作系统都不能保证自身的正确和完整,如何为主体提供可信的计算环境?所以,操作系统自身的完整性是操作系统发挥有效作用的前提和基础。

需求3 用户身份鉴别

用户要使用操作系统,首先必须登录。登录前,必须由管理员事先建立用户账户。一般包括用户名和密码,保存在系统内。用户在登录时,要求用户输入用户名和密码,然后将用户输入的用户名和密码与先前的用户名和密码进行比较。如果正确,用户登录进入系统享用计算机资源;否则,拒绝用户登录。不管是本地登录,还是远程登录,其思想基本如此。但这种登录方式存在两个致命的缺陷:a)认证的单向性,即只能一方鉴别另一方,不能实现双向认证;b)不安全。用户的账户信息可以用加密的技术来增强其安全性,但是操作系统始终是采用软件的方式来管理和保护这些敏感信息,因此为黑客们提供了可乘之机,它们可以通过各种手段窃取这些敏感信息。口令认证方式如此,IC卡认证、生物认证也是如此。

需求4 环境适应性

操作系统要提供可信计算环境。它必须首先感知环境,然后根据当前环境面对的威胁,适当地启用对应的安全策略进行应对。对于用户来说,在不同的时间,所需要的可信计算环境可能不同。因此,可信操作系统对环境的适应性是必要的。

需求5 主体行为可信

保证主体行为可信是可信操作系统与安全操作系统最大的区别之一。传统的访问控制技术只能保证主体的一次行为是否安全,不能保证主体行为轨迹是否可信。例如,对于开域授权问题,传统的访问控制技术就无能为力,只有在安全核中增加监控代理监督主体的行为,才能保证主体的行为不会泄密、不会破坏客体、不会造成危害。

需求6 可信客体

客体是主体的行为承受者。在操作系统中,有各种各样的主体,包括文件(文件中还分为文本文件和二进制文件)、目录、管道、消息等。但传统的安全操作系统并没有区分客体的多样性。将这些客体用统一的方法表示和标志,这为黑客的入侵带来了可乘之机。另一方面,客体内容的真实性、保密性和完整性也是可信操作系统必须关注的问题。当然,这里所说的真实性与现实世界中的真实是不一样的。这里所指的是客体内容的改变和变更必须留下主体的行为印记。

6 结束语

随着网络技术的不断发展和Internet的日益普及,人们对Internet的依赖也越来越强。WWW、BBS、e-mail等名词均已为

大众所熟悉。互联网已经成为人们生活的一部分。然而,Internet 是一个面向大众的开放系统,网上攻击与破坏事件层出不穷,包括窃取机密、非法访问、恶意攻击、计算机病毒、不良信息资源和信息战,甚至于社交工程。面对如此严峻的安全形式,更需要新的思路来解决目前遇到的问题。

参考文献:

[1] ABBOTT R P, CHIN J S, DONNELLEY J E, *et al*. NBSIR 76-1041, Security analysis and enhancement of computer operating systems[S]. Gaithersburg, MD:[s. n.],1976.

[2] HOLLINGWORTH D, GLASEMAN S, HOPWOOD M. Security test and evaluation tools: an approach to operating system security analysis,1-5298[R]. Santa Monica:Calif Rand Corp,1974.

[3] LOSCOCCO P A, SMALLEY S D, MUCHELBAUER P A, *et al*. The inevitability of failure: the flawed assumption of security in modern computing environments[C]//Proc of the 21st National Information System Security Conference. Gaithersburg, MD: NIST Press, 1998: 396-407.

[4] LEPREAU J, FORD B, HIBLER M. The persistent relevance of the local operating system to global applications[C]//Proc of SIGOPS European Workshop. New York:ACM Press,1996:187-192.

[5] BAKER D B. Fortresses built upon sand[C]//Proc of UCLA Conference on New Security Paradigms Workshops. Lake Arrowhead, CA: [s. n.], 1996:148-153.

[6] ANDERSON J P. Computer security technology planning study *volume* Ⅱ[M]. *Bedford, MA:HanscomAir Force Base*, 1972.

[7] JAJODIA S, SAMARATI P, SUBRAHMANIAN V, *et al*. A unified framework for enforcing in multiple access control policies[C]//Proc of ACM SIGMOD Int Conf on Management of Data. New York:ACM Press, 1997:474-485.

[8] GALIASSO P, HALE O B J, SHENOI S, *et al*. Policy mediation for multi-enterprise environments [C]//Proc of the 16th Annual Computer Security Application Conference. New Orleans:[s. n.],2000: 100-106.

[9] ABRAMS M, LAPADULA L, EGGERS K, *et al*. A generalized framework for access control:an informal description[C]//Proc of the 13th National Computer Security Conference. Bedford, MA:[s. n.], 1990:134-143.

[10] ABRAMS M, HEANEY J E, KING O, *et al*. Generalized framework for access control: towards prototyping the ORGCON policy[C]// Proc of the 14th National Computer Security Conference. Washington DC:[s. n.],*1991*:1-4.

[11] ABRAMS M, LAPADULA L, LAZEAR M, *et al*. Reconciling a formal model and prototype implementation-lessons learned in implementing the ORGCON policy[M]. Bedford:Mitre Corporation, 1991.

[12] LAPADULA L. Rule-set modelling of trusted computer system[C]// ABRAMS M, JAJODIA S, PODELL H. *Information security*:an integrated collection of essays. Cambridge:IEEE Computer Society Press, 1995.

[13] BERTINO E, JAJODIA S, SAMARATI P. Supporting multiple access control policies in database systems[C]//Proc of IEEE Symposium on Security and Privacy. Washington DC:IEEE Computer Society,1996: 94.

[14] OSBORN S, SANDHU R, MUNAWER Q. Configuring role-based access control to enforce mandatory and discretionary access control policies[J]. ACM Trans on Information and System Security, 2000, 3(2):85-105.

[15] Secure Computing Corp. Assurance in the fluke microkernel, TR 55113[R]. [S. l.]:Fluke Microkernel Roseville,1999.

[16] Secure Computing Corp. Assurance in the fluke microkernel: formal top-level specification[EB/OL]. (1999-02). http://www.cs.utah.edu/flux/flask/.

[17] SHAN Zhi-yong. Research on the framework for multi-policies and practice in secure operating system[D]. Beijing:Institute of Software, Chinese Academy of Sciences,2002.

[18] 施军, 朱鲁华, 沈昌祥, 等. 专用安全操作系统[J]. 计算机研究与发展,2003, 39(5):561-567.

[19] 沈昌祥. 可信计算平台与安全操作系统[J]. 网络安全技术与应用,2005,4(4):7-9.

[20] SHI Wen-chang. Research on and enforcement of methods of secure operating systems development[D]. Beijing: Institute of Software, Chinese Academy of Sciences, 2001.

[21] LIANG Hong-lia. Research on and enforcement of secure operating system supporting multiple security policy[D]. Beijing: Institute of Software, Chinese Academy of Sciences, 2003.

[22] LIU Ke-long. A formal model and implementation of secure Linux operating system and secure Web system[D]. Beijing: Institute of Software, Chinese Academy of Sciences,2001.

[23] CHEN Ze-mao. Research on security architecture of secure operating system for malicious code defending[D]. Xi'an:Naval University of Engineering,2005.

[24] LIANG Bin. Research on trusted process mechanism and related problems[D]. Beijing: Institute of Software, Chinese Academy of Sciences,2004.

[25] CARNEY M, LOE B. A comparison of methods for implementing adaptive security policies[C]//Proc of the 7th USENIX Security Symposium. Minnesota: Michael Carney Secure Computing Corporation,1998:1-14.

[26] 侯方勇, 王志英. 可信计算研究[J]. 计算机应用研究, 2004,21(12): 1-4.

[27] 刘鹏, 刘欣. 可信计算概论[J]. 信息安全与通信保密, 2003, 7(11): 17-19.

[28] 周明辉, 梅宏. 可信计算初探[J]. 计算机科学, 2004, 31(7): 5-8.

[29] 屈延文. 软件行为学[M]. 北京:电子工业出版社,2005.

[30] 林闯, 彭雪梅. 可信网络研究[J]. 计算机学报, 2005, 28(25): 751-758.

[31] 谭良, 余堃, 周明天. CRL 分段—过量发布新模型[J]. 电子学报, 2005, 33(2): 227-230.

[32] 谭良, 周明天. CRL 增量—过量发布新模型[J]. 计算机科学, 2005,32(4): 133-136.

[33] Trusted computing group[EB/OL]. (2001). http//www. trustedcomputinggroup. org.

[34] 周明天, 谭良. 可信计算及其进展[J]. 电子科技大学学报,2006, 35(4):686-697.

更正启事

本刊 2007 年第 11 期 176 页《面向公安应用的智能移动翻译软件的研究与实现》一文中的图 3 应为

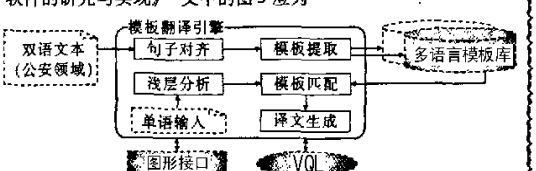

图 3 嵌入式多语言模板翻译引擎

特此更正。

《计算机应用研究》编辑部

2007 年 12 月

可信操作系统研究

作者： 谭良， 周明天， TAN Liang， ZHOU Ming-tian
作者单位： 谭良,TAN Liang(四川师范大学,计算机学院,成都,610066)， 周明天,ZHOU Ming-tian(电子科技大学,计算机科学与工程学院,成都,610054)

刊名： 计算机应用研究 ISTIC PKU
英文刊名： APPLICATION RESEARCH OF COMPUTERS
年，卷(期)： 2007,24(12)
被引用次数： 3次

参考文献(34条)

1. SHI Wen-chang Research on and enforcement of methods of secure operating systems development 2001

2. ABRAMS M;LAPADULA L;EGGERS K A generalized framework for access control:an informal description 1990

3. GALIASSO P;HALE O B J;SHENOI S Policy mediation for multi-enterprise environments[外文会议] 2000

4. JAJODIA S;SAMARATI P;SUBRAHMANIAN V A unified framework for enforcing in multiple access control policies 1997

5. LIU Ke-long A formal model and implementation of secure Linux operating system and secure Web system 2001

6. LIANG Hong-lia Research on and enforcement of secure operating system supporting multiple security policy 2003

7. ABRAMS M;LAPADULA L;LAZEAR M Reconciling a formal model and prototype implementation-lessons learned in implementing the ORGCON policy 1991

8. ABRAMS M;HEANEY J E;KING O Generalized framework for access control:towards prototyping the ORGCON policy 1991

9. ABBOTT R P;CHIN J S;DONNELLEY J E NBSIR 76-1041.Security analysis and enhancement of computer operating systems 1976

10. ANDERSON J P Computer security technology planning study volume Ⅱ 1972

11. BAKER D B Fortresses built upon sand 1996

12. LEPREAU J;FORD B;HIBLER M The persistent relevance of the local operating system to global applications 1996

13. LOSCOCCO P A;SMALLEY S D;MUCHELBAUER P A The inevitability of failure:the flawed assumption of security in modern computing environments 1998

14. HOLLINGWORTH D;GLASEMAN S;HOPWOOD M Security test and evaluation tools:an approach to operating system security analysis,1-5298 1974

15. 周明天;谭良 可信计算及其进展 2006(04)

16. 查看详情 2001

17. 谭良;周明天 CRL增量-过量发布新模型[期刊论文]-计算机科学 2005(04)

18. 谭良;余堃;周明天 CRL分段-过量发布新模型[期刊论文]-电子学报 2005(02)

19. 林闯;彭雪梅 可信网络研究[期刊论文]-计算机学报 2005(25)

20. 屈延文 软件行为学 2005

21. 周明辉;梅宏 可信计算初探[期刊论文]-计算机科学 2004(07)

22. 刘鹏;刘欣 可信计算概论[期刊论文]-信息安全与通信保密 2003(11)

23. 侯方勇;王志英 可信计算研究[期刊论文]-计算机应用研究 2004(12)

24. CARNEY M;LOE B A comparison of methods for implementing adaptive security policies 1998

25. LIANG Bin Research on trusted process mechanism and related problems 2004

26. CHEN Ze-mao Research on security architecture of secure operating system for malicious code defending 2005

27. 沈昌祥 可信计算平台与安全操作系统[期刊论文]-网络安全技术与应用 2005(04)

28. 施军;朱鲁华;沈昌祥 专用安全操作系统[期刊论文]-计算机研究与发展 2003(05)

29. SHAN Zhi-yong Research on the framework for multi-policies and practice in secure operating system 2002

30. Secure Computing Corp Assurance in the fluke microkernel:formal top-level specification 1999

31. Secure Computing Corp Assurance in the fluke microkernel[TR 55113] 1999

32. OSBORN S;SANDHU R;MUNAWER Q Configuring role-based access control to enforce mandatory and discretionary access control policies 2000(02)

33. BERTINO E;JAJODIA S;SAMARATI P Supporting multiple access control policies in database systems 1996

34. LAPADULA L Rule-set modelling of trusted computer system 1995

引证文献(3条)

1. 陆阳. 王强. 张本宏. 诸葛战斌 计算机系统容错技术研究[期刊论文]-计算机工程 2010(13)

2. 方炜炜. 杨炳儒. 周长胜. 杨君 基于EFI的可信计算平台研究[期刊论文]-计算机应用研究 2009(8)

3. 谭良 可信操作系统中可信客体的研究[期刊论文]-计算机应用 2008(5)