文章编号:1671-8860(2008)10-1034-04　　文献标志码:A

SmartMK:基于 TPM 的可信多内核操作系统架构

陈文智[1]　黄　炜[1]

(1　浙江大学计算机学院,杭州市浙大路 38 号,310027)

摘　要:提出了一个多内核架构 SmartMK 来支撑不同安全等级和类别的应用。基于 TPM 和新的 CPU 安全技术,实现了多内核之间的强隔离与安全通信机制,以软硬件协同保护的方式实现安全的操作系统运行环境。在 SmartMK 架构上提出了分层次的强制访问控制方模型,进一步降低复杂环境中的访问控制复杂度。性能测试和实际应用都表明,SmartMK 能够有效加强系统的安全性,同时很好地保证了系统的运行效率。

关键词:TPM;多内核;可信操作系统;分层次强制访问控制;可信计算基

中图法分类号:TP316

安全特性随着操作系统对实时程序和其他更多应用的支持,如数字版权保护[1]、3G 手机安全等问题而备受关注。对系统安全内核的研究主要有独立安全内核、分离内核[2](Separation Kernel, SK)以及最小权限分离内核[3](least privilege separation kernel, LPSK)三种。双内核技术[4]属于分离内核的一种特别形式,其目的是用于提高 Linux 实时性,但系统的稳定性并不可靠。虚拟化技术通过系统管理程序来简单灵活地实现安全策略,而虚拟化平台本身的安全又成了另一个问题。可信硬件如可信平台模块[5](trusted platform module, TPM)和新的 CPU 技术,如 SVM 和 TXT[6]则提供了新的基础支持。相关研究包括:IBM 提出的 TCG-Linux[7]安全启动架构、McCune 等人提出的 SEA 架构[8]。

1　SmartMK 的组成

SmartMK 具有 3 个运行于内核态的内核和多个运行于用户态的内核,如图 1 所示。在典型情况下 V-Kernel 和 C-Kernel 共同管理运行于虚拟化平台上的 T-Kernel 和 Linux;其中运行 T-Kernel 内核的线程具有较高的优先级,而由 C-Kernel 和 D-Kernel 支撑的线程则具有居中的优先级。在 SmartMK 中,C-Kernel 提供了内核的各种基本机制,因此,在平台启动过程中首先启动;D-Kernel 借助于 C-Kernel 的基础启动,并建立相应的数据流框架;最后,支持虚拟化平台的 V-Kernel 从 C-Kernel 和 D-Kernel 获得支持,完成虚拟化平台 SmartVP 的启动。

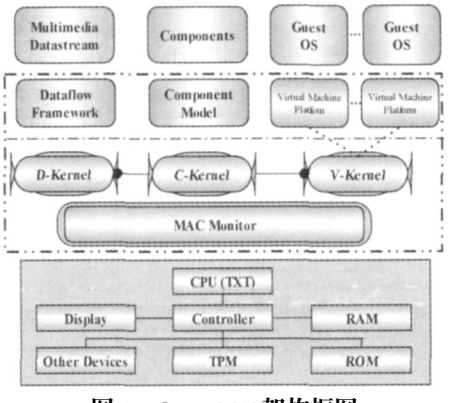

图 1　SmartMK 架构框图

Fig. 1　Structure of SmartMK

1.1　支持构件模型的内核 C-Kernel

C-Kernel 类似于一个实时的微内核,其基本功能是:支持线程调度、地址空间管理和同步消息传递。对 C-Kernel 而言,每一个构件的运行形态都用(线程,空间)组合表示。C-Kernel 通过实现了 4 个基本功能以支持构件化模型:①可抢占的多线程;②快速的线程间通信;③保护的共享地址空间;④反射式的线程调度和空间管理。C-Kernel 基于构件间通信技术以及共享地址空间机制等,实现高效的远程方法调用;它提供线程调

收稿日期:2008-08-28。

项目来源:国家 973 计划资助项目(2007CB310900);国家基础科研资助项目(A142008190)。

度和地址空间访问的反射机制,支持构件模型的灵活定制和扩展。C-Kernel 的基础运行形态是实时的多线程调度以及线程间通信。

1.2　支持数据流框架的内核 D-Kernel

D-Kernel 以及它所支持的数据流框架,实现构件之间新的交互方式,提高构件化的数据流应用的实现效率。D-Kernel 采用了 3 个控制器的结构化设计:事件控制器、任务控制器和模型控制器,分别对事件、元件行为和其他与数据流模型相关的工作进行调度。就优先级而言,事件控制器总是抢占任务控制器,任务控制器总是抢占模型控制器。D-Kernel 采用的并发计算模型,对构件之间的交互关系增加了更强的约束性和调度性;同时它主要关注构件在数据流处理中的输入输出行为,可以帮助构件具有更好的独立性和重用性。

1.3　支持虚拟化平台的内核 V-Kernel

V-Kernel 通过对计算平台的硬件资源的虚拟化形成一个虚拟化平台,支持在处理器的用户态下多种运行标准的操作系统,从而支持大量通用应用程序。笔者基于 V-Kernel 实现了一个 SmartVP[9] 的原型,并且在 Arm 平台上实现了对 T-Kernel 和 Linux 两个系统的并发支持。

2　多内核隔离与安全通信机制

TPM 和相关 CPU 安全技术的出现从硬件层面为内存保护和指令安全执行提供了基础,本文基于 TPM 和 TXT 技术来实现多内核隔离。SmartMK 的多内核隔离利用了 TPM 提供的认证和密封存储机制,认证的过程可以认为是 TPM 利用自己的 Hash 函数对一个事件 m 进行签名,并对 TPM 的特定平台配置寄存器的值进行扩展,$v_{t+1} = \mathrm{Hash}(v_t \| m)$ 表示连接操作。TPM 的 v1.2规范允许静态和动态 PCRs,只有重新启动系统才可以设置静态 PCR 价值,而动态 PCR17 在特定的硬件命令下也能被 Reset。

2.1　SmartMK的内核空间分布

为了让处于 TXT 保护下的各个内核仍然能正常处理中断,对 SmartMK 的多内核进行内核空间的分布如图 2 所示。

三个内核由两份相同的中断服务例程进行分隔,同时要满足以下要求:①三个内核的物理内存起始地址与页框对齐;②两份中断服务例程与三个内核的物理内存起始地址与页框边界对齐,并且在系统启动时中断服务例程页框属性标识为只读;③系统对 D-Kernel 和 V-Kernel 的 Hash 操作应该包含相邻的中断服务例程,即 $v_{t+1} = \mathrm{Hash}(v_t \| m_{D\text{-}\,\mathrm{Kernel+Int}})$ 及 $v_{t+1} = \mathrm{Hash}(v_t \| m_{V\text{-}\,\mathrm{Kernel+Int}})$,对 C-Kernel 的 Hash 操作为 $v_{t+1} = \mathrm{Hash}(v_t \| m_{\mathrm{Int}+C\text{-}\,\mathrm{Kernel+Int}})$。

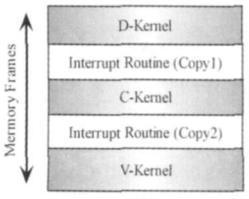

图 2　SmartMK内核空间分布

Fig. 2　Kernel Address Space of SmartMK

2.2　SmartMK的通信机制

在 SmartMK 中,D-Kernel、C-Kernel 与 V-Kernel 都运行在内核态中,C-Kernel 提供相应的内核间通信机制和基于地址空间的保护机制,如图 3 所示。

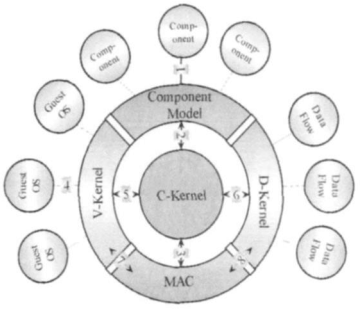

图 3　可信通信机制示意图

Fig. 3　Secure Communication Mechanism

SmartMK 内核间高效通信如下:①V-Kernel 将上层 Guest OS 的内核请求交由 C-Kernel 通过访问强制访问控制监视器(MAC monitor)来验证通信请求是否合法,如果合法则根据访问安全等级直接发送请求到目的地或者由 C-Kernel 为其建立安全 I/O 通道(由 TPM 硬件支持)并发送请求,否则拒绝服务,通信路径如图 3 中 4→7→5→8→4;②构件模型组件按照类似于 V-Kernel 的机制建立与上层 Component 相关的通信通道,通信路径如图 3 中 1→2→8→1;③D-Kernel 直接与 C-Kernel 交互,通信路径如图 3 中 6→8→→6。跨内核的调用都需要经过 MAC 的检查。

3　分层次强制访问控制

以 SK 和 LPSK 为基础,SmartMK 采用分层次强制访问控制模型。在 SmartMK 的多内核架构中需要支持虚拟化平台,如果直接将每个虚拟机中应用进程作为主体或者资源放进 MAC 策略

中则有以下缺陷。

1) 新增加一个进程资源后,MAC 矩阵进行扩大,变化量 $\Delta=(m+1)^2-m^2\approx 2m$,即在虚拟化平台支撑多个操作系统时 MAC 矩阵膨胀很快。

2) 虚拟化平台支持的 Guest OS 之间常常没有任何访问权限,在 MAC 矩阵中值存在大量的空白入口,造成空间浪费。

因此,SmartMK 把 C-Kernel、D-Kernel 和 V-Kernel 作为自然分区,同时把每个内核控制的资源聚类,再次形成相应的从属分区。每个运行在虚拟化平台之上的 Guest OS 也将作为一个独立的分区存在,属于 C-Kernel 的从属分区,则 SmartMK 具有一个分区流矩阵(partition flow matrix, PFM),表示出分区之间的最大访问权限。当一个主体 S 需要对资源 O 进行某种形式 m 的访问的时候,则需要同时满足分区流矩阵和主体-资源访问矩阵所允许的方式,即

$$m\in \mathrm{PFM}(S,Q)\ \&m\in \mathrm{SRM}(S.\mathrm{partition},.O.\mathrm{Partition})$$

经过分区流矩阵和主体-资源访问矩阵的约束,原有的网状 MAC 结构变成了层次 MAC 结构。

4 测试结果与分析

对 C-Kernel 的关键接口进行测试基于一些典型的应用,包括基于闪存的视频播放、基于以太网的浏览等,测试结果如表 1 所示。

表 1 C-Kernel 主要接口测试结果

Tab. 1 Testing Results of Main Interfaces of C-Kernel

操作	最小系统调用	简单 IO 调用	用户态实时响应	远程方法调用	简单的同步消息传递	创建线程	删除线程	创建进程
耗时/μs	1	3	18	29	9	23	54	617

C-Kernel 支持的构件化模型的构件加载时间成本与其他的构件化模型进行了比较,结果如图 4 所示。其中,采用引用调用的形式则时间成本只有 COM 的 11.4%、omniORB 的 7.7%和 ORBacus 的 4.3%。

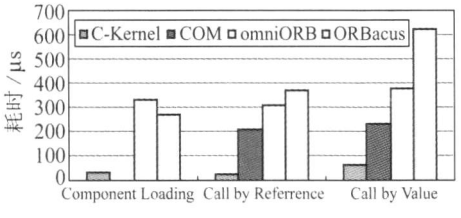

图 4 构件加载时间

Fig. 4 Loading Time for Components

利用视频对话(video over IP,VoIP)对 SmartMK 的 D-Kernel 内核及其相应的数据流框架进行测试的结果如图 5 所示,并发的应用越多,D-Kernel 的性能优势越显著。

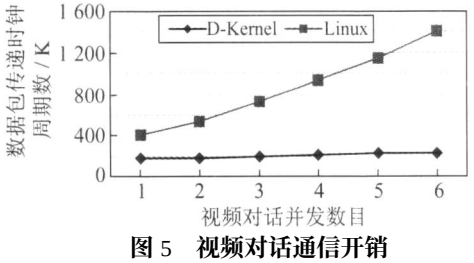

图 5 视频对话通信开销

Fig. 5 Communication Cost of VoIP

5 结 语

目前,可信硬件的发展还不够成熟,如 TPM 的 Seal 和 Unseal 等操作的耗时仍然很长,在多核环境中,使得单一时刻只能有一个进程对 TPM 进行访问,从而造成 SmartMK 性能的损失。随着 TPM 和其他的安全硬件技术的不断改善,由于硬件的固有开销造成的影响会逐渐减少。而在多核环境下,对 SmartMK 采用不同的线程调度策略来提高系统性能将是有待研究的问题。

参 考 文 献

[1] Popescu C B, Crispo B, Tanenbaum S A. Support for Multi-level Security Policies in DRM Architectures[C]. The 2004 Workshop on NSP, Nova Scotia, Canada, 2004

[2] Greve D, Wilding M, Vanfleet W M. A Separation Kernel Formal Security Policy[C]. ACL2 Workshop, Boulder, Colorado, 2003

[3] Levin T E, Irvine C E, Nguyen T D. A Least Privilege Model for Static Separation Kernels[EB/OL]. http://cisr.nps.edu/-downloads/nps_cs-_05_003.pdf, 2007

[4] Yodaiken V, Barabanov M. A Real-Time Linux[J]. Linux Journal, 1997, 3(4):19-23

[5] Trusted Computing Group. Trusted Platform Module Main Specification. Version 1.2[EB/OL]. http://www.trustedcomputinggroup.o-rg/groups/

tpm/ , 2008

[6] Intel Corporation. Trusted Execution Technology Preliminary Architecture Specification and Enabling Considerations[EB/OL]. http://download.intel.com/technology/security/downloads/315168.pdf, 2007

[7] Sailer R, Zhang Xiaolan, Jaeger T. Design and Implementation of a TCG-based Integrity Measurement Architecture[C]. The 13th Conference on USENIX Security Symposium, CA, USA 2004

[8] McCune J M, Parno B, Perrig A, et al. An Execution Infrastructure for TCB Minimization[J]. ACM SIGOPS Operating Systems Review, 2008, 42(4): 315-328

第一作者简介:陈文智,副教授,主要研究方向为嵌入式实时系统、虚拟化技术。

E-mail:chenwz@zju.edu.cn

SmartMK:TPM-based Trusted Multi-Kernel Operating System Architecture

CHEN Wenzhi[1]　*HUANG Wei*[1]

(1 College of Computer Science, Zhejiang University, 38 Zheda Road, Hangzhou 310027, China)

Abstract: The emergence of general security hardware provides operating system and electronic equipment with a hardware-based security protection, but there were few studies about using the hardware to provide system-level security protection directly. A multi-kernel structure SmartMK was proposed to support applications of different security levels and different types; based on the trusted platform module(TPM) and the new CPU security technology, the strong separation and secure communications mechanisms between multi-kernel were realized and the security of the operating system operating environment was achieved by the hardware and software together. A mandatory access control model was offered to the SmartMK reduce the complexity of access control. Performance testing and application of SmartMK showed that it can effectively strengthen the system security while guaranteeing the system's efficiency.

Key words: TPM; multi-kernel; trusted operating system; layered mandatory access control; trusted computing base

About the first author: CHEN Wenzhi, associate professor. He is mainly engaged in the research on real-time embedded operation systems and virtualization technology.

E-mail: chenwz@zju.edu.cn