TCG 下可信度量机制与模型分析

张 策[1,2], 崔 刚[1], 傅忠传[1]

(1. 哈尔滨工业大学 计算机科学与技术学院,150001 哈尔滨;
2. 哈尔滨工业大学(威海) 计算机科学与技术学院, 264209 山东 威海)

摘 要: 针对可信计算中可信度量的研究现状和所面临的问题,对依据可信需求如何表征对象的可信性问题进行了深入研究. 分析了可信度量有关的基本概念、主要任务和预期目标;剖析了基本度量模型体系结构,给出了完整的度量过程及其形式化描述;综合考虑各种度量因子,依据度量对象、度量代理与实现策略的差异,选取典型模型进行评述. 在此基础上,对可信度量模型进行了归类比较总结. 研究分析表明在可信度量中引入信息流策略、访问控制,建立有效的度量模型,并注重实体行为的度量是可信度量后续研究的关键.

关键词: 可信计算;可信度量;度量机制
中图分类号: TP309 **文献标志码**: A **文章编号**: 0367 - 6234(2013)01 - 0072 - 06

Overview of trustworthiness measurement mechanism and model in TCG

ZHANG Ce[1,2], CUI Gang[1], FU Zhongchuan[1]

(1. School of Computer Science and Technology, Harbin Institute of Technology, 150001 Harbin, China;
2. School of Computer Science and Technology, Harbin Institute of Technology at Weihai, 264209 Weihai, China)

Abstract: As one of the main bottle necks in Trusted Computing going forward, the technical core of trustworthiness measurement is how to characterize object trustworthiness in accordance with requirement of the trust. This paper states the current state of research and the confronted problems. Firstly, the basic concepts, main tasks and expected objectives of measurement are summarized and presented. Secondly, the framework of measurement, complete measurement process and formal description are described. With considering all factors, in the light of the difference of objects, measurement agents and implementation, some representative model of measurement are discussed. And on that basis, the latest models are classified and summarized respectively, and the directions for continuing research are discussed.

Key words: trusted computing; trustworthiness measurement; measurement mechanism

TCG(Trusted Computing Group)[1]提出的基于TPM(Trusted Platform Module)安全芯片的可信计算技术使得网络与信息安全技术从"软安全技术"到"硬安全技术"过渡成为可能[2-3]. 作为可信计算理论研究与技术实现的基础性问题,可信度量研究的主要是如何采取合适的策略对不同的实体对象进行以可信性为主的全面度评.

本文结合可信度量研究现状,以度量策略实施为主线进行综合评述,首先介绍可信度量有关概念、主要任务,并给出基本度量机制及形式化描述,然后选取与分析一些典型度量模型,进行归类比较,进而指出后续研究方向.

收稿日期: 2011 - 12 - 09.
基金项目: 国家自然科学基金资助项目 (90818016).
作者简介: 张 策(1978—),男,讲师;博士研究生;
崔 刚(1949—),男,教授,博士生导师.
通信作者: 张 策, zhangce@ hitwh. edu. cn.

1 基本概念与主要任务

1.1 基本概念

定义 1 可信(Trustworthiness)是对实体综合特性及行为与预期符合程度的一种描述.

定义 2 可信度(Trustworthiness). m 是度量结果,对实体可信性的定量表述.

定义 3 可信度量(Trustworthiness Measurement, TM)由度量代理(Measurement Agent, MA)在预定的

度量策略支持下采用适当的算法对实体对象进行以可信性为主的表征过程.

定义 4　度量代理也被称为度量模块或可信引擎(Trusted Engine),制定度量机制与策略,采用特定算法对被度实体进行可信性评判.

定义 5　度量因子(Measurement Factor, MF)包括度量时间点 t(加载、运行、卸载时)、对象类型 p(代码、内核、上层应用及 Hypervisor 等)、粒度 o(文件、信息流、系统调用、堆栈、参变量等)等等.

1.2　主要任务

可信计算首先要解决的就是可信度量问题,只有定量地描述可信,才能对可信有更加准确的认识. 可信度量就是对各分类实体的可信性进行综合表征,给出可信度. 可信度量的主要任务包括:

1)确定预期断言(Expection Claim). 对实体所期望的状态、功能、行为与结果进行恰当的表述. 期望断言描述可有多种形式,其决定了可信标准的准确性程度,可由被多方认可的机构或实体长期观测经验得到.

2)制定度量策略(Measurement Policy). 综合考虑度量影响因素,制定可行的度量方法对实体进行可信性判定,是度量实施的核心. 在实施上,需要定义一组策略与准则,将度量技术应用到度量对象上,并得到一组度量值.

3)进行可信评判(Trustworthiness Judgment). 根据所评判对象对可信性需求程度的不同,采用合适的评判量化手段对获得的可信支持数据与预期进行匹配,给出可信度.

2　可信度量基本体系结构

度量是衡量对象的一种手段,是对实体属性的一种量化表示. 度量就其本意是指,根据预定的规则,采用可行的多种技术手段对某实体对象相关属性进行测试、收集、分析、计算的持续性定量化和与标准值匹配的过程. 基本度量体系结构模型如图 1 所示.

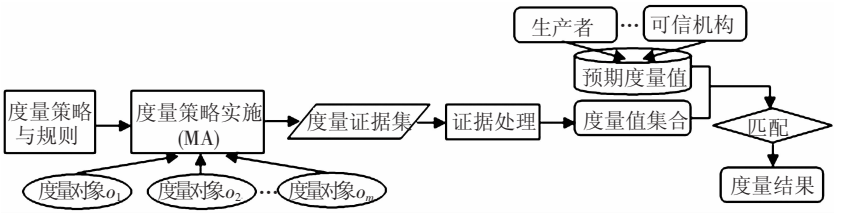

图 1　基本度量体系结构模型

具体度量过程如下:

1)建立度量环境变量. O、R 与 E 分别为度量对象、策略与规则及获得的证据集合,满足 $o \times 2^R \rightarrow 2^E$;$MV$ 与 EV 分别为度量值与预期标准值集合,满足 $2^E \rightarrow 2^{MV} \rightarrow 2^{EV}$;$M$ 为度量结果,满足 $2^{MV} \times 2^{EV} \rightarrow M$.

2) 进行度量实施.

$MA(o_i, \{r_1, \cdots, r_m\}) = \{e_1, \cdots, e_n\}, o_i \in O$; $r_j \in R; e_k \in E; 1 \leqslant j \leqslant m, 1 \leqslant k \leqslant n$.

3) 对获得的度量证据集进行处理,获得可比较的度量值为

$$\mathrm{Proc}(\{e_1, \cdots, e_n\}) = \{mv_1, \cdots, mv_n\},$$
$$mv_t \in MV; 1 \leqslant t \leqslant n.$$

4) 根据预期阈值进行度量匹配,获得对象可信度为

$\mathrm{Mat}(\{mv_1, \cdots, mv_n\}, \{e_1, \cdots, e_n\}, \varepsilon) = m$, $m \in M$, 其中 ε 为期望的阈值要求,给出可信结果.

3　可信度量机制研究及典型模型评述

可信度量侧重的是对实体对象可信性的度量,要实现对可信性度量,必须为度量制定合适的度量策略. 可信度量已由 TCG 的单一的完整性度量逐渐向可信多属性的综合度量转变[4-5],但各属性间的依赖或涌现关系尚不明晰. 本文依据度量对象、度量时间点的不同,按照具体度量策略的差异,并选择较为典型的度量模型进行分析与评述.

3.1　IMA 可信静态度量

在可信静态度量方面,IMA (Integrity Measurement Architecture)[6] 是其中的典型代表. IMA 由 IBM 研究院实现,其度量的关注点是基于 TCG 的完整性度量思路,侧重对各种软件代码和文件的完整性度量,度量时刻是在加载到内存时进行的,并通过 TPM 来进行保护. IMA 度量架构由度量机制、完整性挑战机制与完整性验证机制

3个组件构成,其完整性属性是数据的二进制属性,故可以检测到受破坏与否. 度量实施中实体度量链表节点包括对象名、Hash值和Dirty标记.

其中:

1)预期断言,是内核预加载信任链上各实体的度量列表以及存储在预定义的TPM PCR中的启动过程Hash值;

2)度量策略采用SHA1;

3)可信评判采用标准的Trusted/Untrusted二值逻辑.

IMA是最先基于TCG可信概念实现从BIOS一直到应用层可信链度量的系统. 其主要优点是:1)实现了TCG的可信度量机制,能够抵御大部分对完整性进行破坏的非法攻击,比如Rootkit[7]等;2)度量策略与实现简单(IMA修改了Linux内核与运行时系统(<1 000 Lines)),且扩展性较好;3)由于采用Cache机制并且只有在实体发生改变时才度量,可减少开销.

IMA存在着明显不足为:1)IMA没有对运行中的可信属性进行度量,易造成TOC-TOU[8]问题导致泄露隐私;2)度量范围过大,IMA对所有载入系统的程序均进行度量,这是低效率的;3)没有采用分类度量,对于非结构化数据无法度量.

3.2 PRIMA信息流度量

针对静态加载时的度量不能准确反应运行时的动态可变行为[9-11],且需要全部度量对象已知,导致效率较低,与之相对提出了可信动态度量. PRIMA(Policy-Reduced Integrity Measurement Architecture)结合CW-lite模型,利用SELinux策略,基于IMA实现了原型系统. PRIMA定义主客体的完整性等级,度量进程间信息流主客体完整性等级,依据访问控制策略来判定信息流动,可验证经典的Biba完整性模型和Clark-Wilson策略. 其信息流情况如图2所示.

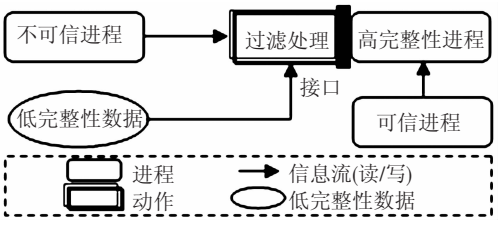

图2　PRIMA中信息流动情况

S为MAC策略中定义的主体集合,T为可信主体集合,F为过滤主体集合,U为不可信主体集合,$T \subset S$,$F \subseteq S$,$F \cap T = \varnothing$,$U = S - (T \cup F)$. 可信主体s对应的代码和使用的静态数据需要被度量. 度量记录$m_e = (c \| s \| r)$,其中:c为代码或数据的摘要值,s为主体,r为角色标示. $M_{i+1} = M_i \| m_e$构成度量列表. 每个过滤主体与一个可信主体相关联,过滤主体被授权可访问低完整性输入,并通过过滤界面对其进行必要处理,为了保护高完整性进程,过滤界面对低完整性输入进行升级或丢弃处理. 为了使得挑战方能够验证CW-Lite完整性,PRIMA度量对象包括IMA中的代码和静态数据、MAC策略、可信主体T、代码与主体间的映射. 系统启动时,MAC策略与可信主体被首先度量,挑战方可据此构造信息流图G并验证运行中的信息流. 运行中需要度量与目标应用有关的代码,以及代码与可信主体间的映射,其他的均假定为不可信.

PRIMA模型的优点为:1)将IMA的静态二进制度量扩展为信息流完整性度量;2)效率较高,因为不需要度量与目标应用无关的进程和不可信主体的代码. PRIMA模型的不足主要为:1)信息流动态度量依赖于特定的MAC策略;2)可用性受到影响,这是由于依赖MAC区分各种主体,以及没有对不可信主体进行度量造成的.

3.3 虚拟度量

文献[12-13]认为,真正的完整性依赖于内存中软件的状态,针对TCG完整性度量无法抵御运行时攻击的不足[14],提出了在Intel协处理器的TXT技术与Xen hypervisor虚拟环境下针对Linux内核的完整性监视器LKIM. 度量模型由度量对象、度量代理MA和决策体构成,LKIM在启动时产生部分度量,在运行中能够响应系统事件或命令,产生与Linux内核安全相关数据结构的详细状态记录. 其基本架构和工作流程如图3所示.

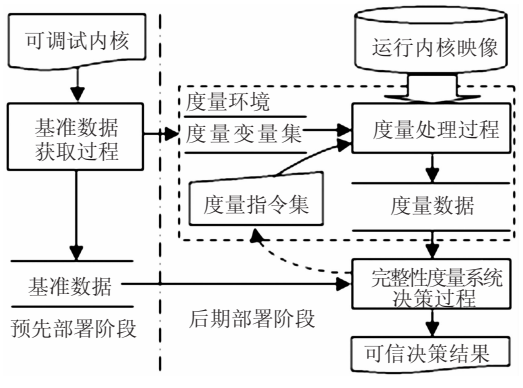

图3　LKIM结构与数据流

静态度量基准值可以通过图3中虚线左侧的编译方式获得,而可扩展的动态基准值用以实现

度量目标的变更. 在具体实施上,LKIM 利用上下文语境检查技术度量内核组件:对内核中代码段、系统调用表、中断描述表、全局描述符表等静态区代码与数据采用哈希计算;对其他非结构化区则进行目标对象和各种内核动态数据结构相关的有效监测遍历. 上下文监测可提供运行中 Linux 内核可执行路径的识别度量,获得可信度量证据,最终来表征运行内核的操作完整性. LKIM 依据度量变量集将被度量进程拆分成一系列分散的度量元,该度量元以先分组后分层方式组织,可被独立监测. 度量过程通过预定的度量指令(Measurement Instruction)指示运行中哪些度量元应被监测,度量先从顶层元变量开始,不断递归至下层,当达到预定的完整性要求时,则停止.

LKIM 的优点为:1)扩展了 TCG 的哈希度量为上下文相关度量,粒度更细;2)LKIM 可提高安全性,并具备隐私保护和探测 Rootkit 攻击能力. 其不足为:1)由于度量粒度过细开销不易控制;2)由于 LKIM 在度量期间暂停了目标域的执行,因而会引起度量中出现拒绝服务现象;3)对如何获得动态的基准值描述不够.

在 Hypervisor 完整性度量方面,文献[15]提出了硬件机制支持下对 Hypervisor 进行秘密的语境度量模型 HyperSentry. HyperSentry 利用现有资源构造可信计算基 TCB(Trusted Computing Base),采用独立于 Hypervisor 的软件组件来实现完整性度量,如图 4 所示.

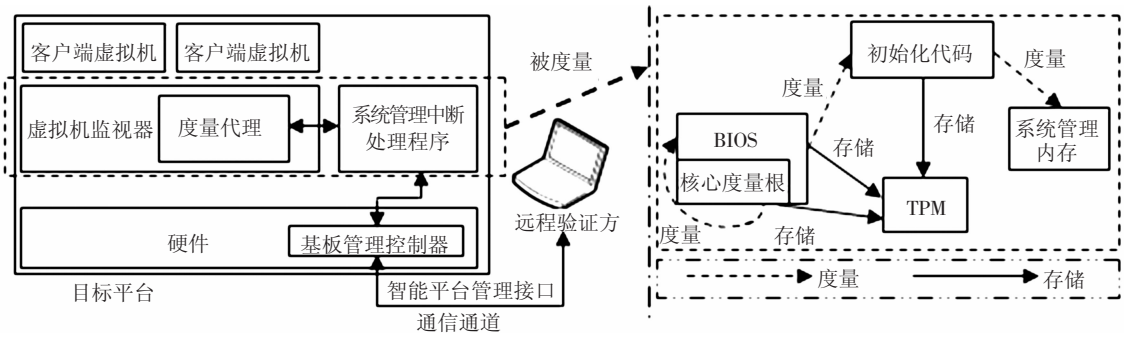

图 4　HyperSentry 架构

在度量触发的机密问题上,采用独立于 CPU 与系统最高特权软件控制的带外通道技术来触发度量代理. 带外通道技术应用智能平台管理接口 IPMI 通过激活系统管理中断 SMI 来触发 HyperSentry 度量. 整个度量触发机制采用如下形式来进行 Remote verifier→IPMI→BMC→SMI→MA,用以确保度量机密性要求. Hypervisor 与 SMI 处理程序的静态可信通过图 4 右侧虚线框的可信启动过程来实现;度量代理 MA 则是由 SMI 处理程序通过计算其哈希值来完成的. 在语境度量实现上,将 CPU 设定为所需语境,并且为度量代理提供可验证与受保护的执行环境. HyperSentry 并不关注度量采取何种具体机制,依具体应用而定,但在其实例中度量的是代码完整性和客户 VM 间被隔离内存的完整性.

HyperSentry 模型的优点为:1)与 LKIM 一样采用分离式度量;2)解决了度量代理触发问题(引起的 scrubbing 攻击). 其不足为:1)需要较大程度硬件支持[16],实现技术难度大;2)该模型较难应用在 PC 上.

3.4　度量中 TOC-TOU 与攻击问题

TOC-TOU(Time-of-Check Time-of-Use)问题是度量值描述的状态与运行或验证中的当前状态不一致/不同步现象,从而造成安全隐患. 针对 TOC-TOU 问题,文献[17]中提出了采用加载时度量与卸载时度量差值的办法来描述系统当前状态,并据此来解决 TOC-TOU 问题,但由于假设访问控制模块等的完整性不被破坏,且没有对实现方法的安全性与隐私性进行严格验证,较难实际应用. 文献[18]提出了一种操作系统度量架构 DIMA,其度量对象实现了细粒度化,可对系统中当前进程和模块进行动态实时度量,因此不存在 TOC-TOU 问题,但同样在安全性措施、正确性验证上有所不足,由于度量对象较多,当规模过大时耗时负担加重.

可信计算支持加载时完整性度量技术,但无法阻止和检测运行时的攻击(堆与栈缓冲区溢出等). 文献[19]中提出了一种动态完整性度量与验证架构 DynIMA,用以“缩短”静态加载时度量与动态运行时度量的差距,并可以解决 ROP(Return-oriented programming)攻击,其基本架构和工作流程如图 5 所示.

DynIMA 采用加载时度量与动态跟踪相结合的方式来实现.

1）预期断言.

对于所有的程序二进制,跟踪事件已提前进行了定义,存储在跟踪策略模块中.

2）度量策略. 如图5右侧虚线框所示,程序加载时,加载器采用IMA方式度量程序的静态二进制代码,同时相应地追踪测量组件重写代码以包含跟踪代码. 程序执行时跟踪代码将所获得的数据存储在数据段中,并持续动态地进行更新数据. PIM利用跟踪代码执行完整性相关的运行时检查和监视,跟踪测量可以采用Taint Tracking技术和Dynamic Tracing技术[20]. 为此,需要程序加载器应包括静态完整性度量部分和能够重写程序代码以包含监视运行中动态事件和维护跟踪数据的特殊跟踪代码.

该模型的优点为:1）DynIMA本质上是应用代码重写技术的动态完整性监测技术,不仅可以实现度量同时具备动态跟踪;2）可监测ROP攻击. 其不足主要表现在:1）实现较复杂;2）性能不高.

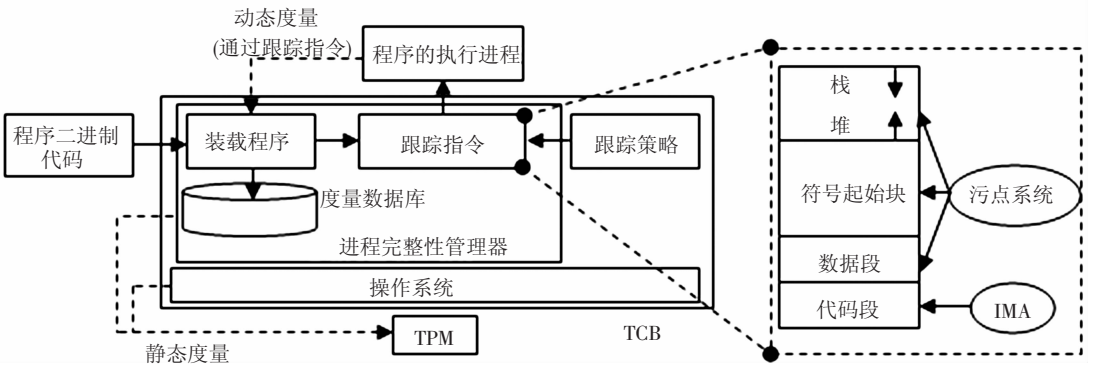

图5　DynIMA模型架构和工作流程

3.5　各种可信度量模型的比较

表1是对各种可信度量模型的比较,其中指标1－5是对度量模型有直接影响的主要因素,为了更好的刻画模型,增加了指标6－10.

表1　典型可信度量模型的比较

模型	IMA	PRIMA	LKIM	DynIMA	DIMA	HyperSentry
对象类型	代码	信息流	内核	磁盘文件、代码	应用程序 、内核模块	Hypervisor
时间点	加载时	加载及运行时	加载及运行时	加载及运行时	运行时	运行时
粒度	中	细	细	细	细	用户自定义
属性内容	二进制完整性	二进制完整性＋信息流完整性	操作完整性	执行完整性	完整性	完整性
度量机制	SHA1	SHA1＋MAC	Hash＋上下文检查	Hash＋跟踪技术	Hash	Hash＋用户自定义
可扩展性	好	较好	好	差	较好	好
健壮性	好	好	好(Xen下)	较差	好	好
实时性	差	较好	好	好	好	好
隐私保护	较差	较好	好	最差	差	差
开销	小	较大	较大	较大	10 ms量级	<1.3%

4　结　论

1）从2001年TCG提出TPM1.1标准以来,可信计算取得了长足发展,可信度量机制与模型问题已成为可信计算科学范畴内的重要问题和研究热点.

2）由本文可见,可信度量研究已从TCG的静态完整性扩展到动态的涵盖访问控制与信息流的可信性度量.

3）可信度量涉及到一个完整的计算机系统的各个层次及平台间信任关系,尚需要建立有效的度量模型,并侧重基于实体行为的可信性全面度量.

参考文献

[1] Trusted Computing Group. TCG Specification Architecture Overview [EB/OL]. https://www.trustedcomputinggroup.org/.

[2] CAMENISCH J. Better privacy for trusted computing platforms [J]. Lecture Notes in Computer Science, 2004, 3193: 73 – 88.

[3] PEARSON S. Trusted Computing Platforms, the Next Security Solution [R]. Bristol UK: Trusted E-Services Laboratory, HP Laboratories, 2002.

[4] SHI E, PERRIG A, Van DOORN L. Bind: a fine-grained attestation service for secure distributed system [C]//Proceedings of the 2005 IEEE Symposium on Security and Privacy (S&P'05). Oakland, California: IEEE, 2005: 1 – 15.

[5] BAIARDI F, CILEA C, SGANDURRA D, *et al.* Measuring semantic integrity for remote attestation [J]. Lecture Notes in Computer Science, 2009, 5471: 81 – 100.

[6] SAILER R, ZHANG Xiaolan, JAEGER T, *et al.* Design and implementation of a TCG-based integrity measurement architecture [C]//Proceedings of the 13th conference on USENIX Security Symposium. CA: USENIX Association Berkeley, 2004: 223 – 238.

[7] HOFMANN O S, DUNN A M, KIM S, *et al.* Ensuring operation system kernel integrity with OSck [C]// Proceedings of the Sixteenth International Conference On Architectural Support For Programming Languages And Operating Systems. Newport Beach, California, USA: ASPLOS, 2011: 279 – 290.

[8] AZAB A M, PENG Ning, SEZER E C, *et al.* HIMA: a hypervisor-based integrity measurement agent [C]// Proceedings of the 2009 Annual Computer Security Application Conference. Honolulu, HI: IEEE, 2010: 461 – 470.

[9] JAEGER T, SAILER T, SHANKAR U. PRIMA: policy-reduced integrity measurement architecture [C]// Proceedings of the Eleventh ACM Symposium On Access Control Models and Technologies. New York, NY: ACM, 2006: 19 – 28.

[10] XU Ziyao, HE Yeping, DENG Lingli. An integrity assurance mechanism for run-time programs [J]. Lecture Notes in Computer Science, 2009, 5487: 389 – 405.

[11] LIU Changping, FAN Mingyu, FENG Yong, *et al.* Dynamic Integrity measurement model based on trusted computing [C]//Proceedings of 2008 International Conference on Computational Intelligence and Security. Washington, DC: IEEE Computer Society, 2008: 281 – 284.

[12] LOSCOCCO P A, WILSON P W, PENDERGRASS J A, *et al.* Linux kernel integrity measurement using contextual inspection [C]//Proceedings of the 2007 ACM Workshop on Scalable Trusted Computing. New York: ACM, 2007: 21 – 29.

[13] THOBER M, PENDERGRASS J A, McDONELL C D. Improving coherency of runtime integrity measurement [C]//Proceedings of the 3rd ACM Workshop on Scalable Trusted Computing. New York, NY: ACM, 2008: 51 – 60.

[14] 李博, 李建欣, 胡春明, 等. 基于 VMM 层系统调用分析的软件完整性验证 [J]. 计算机研究与发展, 2011, 48(8): 1438 – 1446.

[15] AZAB A M, NING Peng, WANG Zhi, *et al.* Hypersentry: enabling stealthy in-context measurement of hypervisor integrity [C]//Proceedings of the 17th ACM Conference on Computer and Communications Security. New York: ACM, 2010: 38 – 49.

[16] WANG Jiang, STAVROU A, GHOSH A. HyperCheck: a hardware-assisted integrity monitor [J]. Lecture Notes in Computer Science, 2010, 6307: 158 – 177.

[17] 张谦, 贺也平, 孟策. 解决度量 – 验证时间差的一种差值证明方法 [J]. 通信学报, 2009, 30(10A): 43 – 50.

[18] 刘孜文, 冯登国. 基于可信计算的动态完整性度量架构 [J]. 电子与信息学报, 2010, 32(4): 875 – 879.

[19] DAVI L, SADEGHI A R, WINANDY M. Dynamic integrity measurement and attestation: towards defense against return-oriented programming attacks [C]// Proceedings of the 2009 ACM Workshop on Scalable Trusted Computing. New York, NY: ACM, 2009: 49 – 54.

[20] LIN Hui, LEE G. Micro-architecture support for integrity measurement on dynamic instruction trace [J]. Journal of Information Security, 2010, 1(1): 1 – 10.

（编辑　张　红）