文章编号:1671-9352(2014)09-0001-08 DOI:10.6040/j.issn.1671-9352.2.2014.408

基于扩展 LS^2 的 VMM 动态度量形式化分析

纪祥敏[1,2,3],赵波[1,3*],向骏[1,3],夏忠林[1,3]
(1. 武汉大学计算机学院,湖北 武汉 430072;
2. 福建农林大学计算机与信息学院,福建 福州 350002;
3. 武汉大学空天信息安全与可信计算教育部重点实验室,湖北 武汉 430072)

摘要:虚拟机监控器(virtual machine monitor,VMM)动态度量是保障虚拟化环境安全的重要手段,但是目前VMM动态度量正确性缺乏理论分析。基于VMM动态度量流程,确立了动态度量正确性目标,明确了定义动态度量应满足的重要属性,从操作语法、语义及推理规则方面扩展安全系统逻辑(logic of secure systems, LS^2),据此推导动态度量程序的不变性,验证VMM动态完整性度量应满足的正确性。结论分析表明,应用本文扩展的 LS^2 方法分析得出的动态度量结论与该技术实际应用效果一致,说明扩展的 LS^2 方法有效,可为虚拟化环境安全提供理论参考。

关键词:虚拟机监控器;动态度量;安全系统逻辑
中图分类号:TP309 **文献标志码**:A

Formally analyzing VMM dynamic measurement based on extended LS^2

JI Xiang-min[1,2,3], ZHAO Bo[1,3*], XIANG Shuang[1,3], XIA Zhong-lin[1,3]
(1. Computer School, Wuhan University, Wuhan 430072, Hubei, China;
2. College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou 350002, Fujian, China;
3. Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, Wuhan University, Wuhan 430072, Hubei, China)

Abstract: Dynamic measurement for Virtual Machine Monitor (VMM) is a vital means to guarantee virtualized environments security, but there is currently little theoretical analysis on the correctness of VMM dynamic measurement. Therefore, based on VMM dynamic measurement process, the correctness goal of dynamic measurement is established in this work, which also gave a clear definition of several important properties to be met during dynamic measurement. Meanwhile, Logic of Secure Systems (LS^2) is extended by the operating syntax, semantics and reasoning rules, whereby reasoning several procedure invariances, and then formally verifying the correctness of VMM dynamic integrity measurement. The analysis shows that model and analysis conclusions drawn from the extended LS^2 coincide with practical application effect, and that the extended LS^2 is effective to provide security theoretical support for virtualized environments security.

Key words: virtual machine monitor; dynamic measurement; logic of secure systems

收稿日期:2014-06-24; **网络出版时间**:2014-08-27 10:43
网络出版地址:http://www.cnki.net/kcms/doi/10.6040/j.issn.1671.9352.2.2014.408.html
基金项目:国家"九七三"重点基础研究发展计划项目(2014CB340600);国家自然科学基金重点项目(61332019);国家自然科学基金资助项目(61173138, 61272452);湖北省重点新产品新工艺研究开发项目(2012BAA03004)
作者简介:纪祥敏(1971 -),男,讲师,博士研究生,研究方向为信息系统安全、可信计算. E-mail:jixm168@126.com
*通讯作者:赵波(1972 -),男,教授,博士,研究方向为信息系统安全、嵌入式系统. E-mail:zhaobo@whu.edu.cn

0 引言

作为云计算的关键技术,虚拟化在安全隔离方面具有极大优势[1]。然而,当前虚拟化平台自身面临着前所未有的安全挑战和威胁[2];处于核心层的虚拟机监控器(virtual machine monitor, VMM)完整性保护问题亟需研究解决[3]。研究表明,可信度量是确保 VMM 完整性的重要方法之一。但是,目前采用的静态可信根度量(static root of trust for measurement, SRTM)技术只是在平台启动时才进行一次完整性度量,难以满足 VMM 多次完整性度量需求,并已证明不安全[4]。在这种情况下,Intel 和 AMD 推出的动态可信度量根(dynamic root of trust for measurement, DRTM)[5]技术应运而生,已经成为研究焦点之一[6]。

动态完整性度量的关键是如何保证度量过程的正确性[7]。但是目前缺乏针对 VMM 动态完整性度量正确性的形式化分析。动态度量技术领先于理论,缺乏理论层次的深入研究分析,难以从根源上发掘动态度量之中存在的安全隐患。因此,有必要对动态度量过程进行形式化分析,以发现存在的问题。为此,本文针对 VMM 动态完整性度量的形式化分析需求,从操作语法、语义与推理规则方面对安全系统逻辑(logic of secure systems LS^2)[8]作必要的合理扩展,确立形式化建模与分析的目标与假设,定义应满足的度量程序不变性;在应用扩展的 LS^2 形式化建模过程中,由度量序列事实推导完整性度量的重要属性,据此证明并分析 VMM 动态度量过程的正确性,以确保虚拟化平台的安全性。

1 相关工作

现有关于虚拟化环境安全的研究大部分基于"虚拟机监控器安全可靠"的假设前提。然而,近三年来在 Xen 中发现 127 处安全漏洞[9]。这些漏洞中,一部分可以直接被恶意代码利用、执行恶意程序。更有甚者,虚拟机监控器 rootkits 攻击等都严重威胁 VMM 的完整性。为此,国内外学者从技术手段和理论研究等方面,针对 VMM 完整性度量与保护开展了一系列形式化建模与理论分析研究[10],发掘 VMM 潜在的安全问题。在技术层面,针对虚拟化平台提出的动态可信度量根 DRTM 机制,TrustVisor 利用轻量化 VMM 对进程代码与数据进行完整性保护[4]。文献[11]对 DRTM 进行虚拟化,并实现可信执行环境的细粒度保护。文献[12]建议应用 Hoare 高阶扩展逻辑推理动态加载代码。然而,这些研究工作都局限于已知程序的顺序执行。同时,这些技术方案侧重于具体平台给出的特定度量方法,对于完整性度量抽象层次不足,缺乏统一的虚拟化平台完整性度量的抽象模型,没有经过严格的形式化建模与理论分析,可能存在安全缺陷。

在理论层面,研究学者主要面向可信计算平台进行了形式化建模与分析。文献[12]利用一阶逻辑对可信计算平台引导建模,并分析完整性度量与信任链的建立过程;文献[13]提出并分析了基于 TCG 的远程证明方案;文献[14]研究了一种远程证明协议的形式逻辑;文献[15]采用模型检查研究远程认证协议角色和系统的信任关系。这些研究工作为平台完整性度量提供了理论支撑,但是这些逻辑不执行协议语义,不能直接应用在虚拟化环境分析。在形式化分析工具方面,传统的 BAN 逻辑[16]、π 演算[17]等一般倾向于对传统网络系统通信协议的安全属性进行建模分析,而对于系统内部程序的安全性分析支持不够。GNY 逻辑和 AT 逻辑等大部分模态逻辑抽象性过高,往往缺失系统协议执行过程中的某些状态信息,无法全面反映安全协议运行的整体状态[18-19]。虚拟化环境固有的特性,如 VMM 并发执行、内存隔离等,导致面向传统可信平台的分析方法难以在 VMM 动态度量中直接使用。为此,针对 VMM 动态度量,亟需在理论上研究更合适的形式化方法,对度量过程进行形式化建模与分析,明确动态度量过程应满足的重要属性,验证 VMM 动态度量执行的正确性。

LS^2 基于协议组合逻辑(protocol composition logic,PCL),采用现有的进程演算方法,面向复杂安全应用系统进行功能抽象、安全属性建模,使用不变量属性规则,支持对未知代码的动态加载分析。同现有方法相比,LS^2 逻辑语法简单、语义清晰、扩展性好,便于可信计算环境中的平台配置寄存器 PCR 的扩展操作[7-8],易于检测针对系统完整性的攻击行为,比较适合于 DRTM 机制的验证与分析。

2　VMM 动态度量形式化建模

在分析 VMM 动态度量流程的基础上,对 LS² 进行操作语法、语义以及推理规则合理扩展,并明确定义 VMM 动态度量应满足的程序不变性等若干重要属性,确立动态度量正确性目标与假设,进而利用扩展的 LS² 对动态度量过程形式化建模,据此验证 VMM 动态度量过程的正确性。

2.1　VMM 动态度量基本思路与流程

为了实现对 VMM 动态度量,在动态可信度量根 DRTM 与度量目标 VMM 之间嵌入一个承上启下的完整性度量功能模块,即可信引导度量模块(trusted boot and measurement, TBM),TBM 提供了一个安全可靠的度量环境与代码载入功能,并具体执行 VMM 动态度量。

VMM 动态度量基本思路:结合静态可信度量根 SRTM 技术和动态可信度量根 DRTM 技术优点,以硬件芯片 TPM 为可信存储根 RTS 和可信报告根 RTR,以 DRTM 为 VMM 动态完整性度量的基点,采用信任链传递的方式,经过可信度量模块 TBM 到 VMM,构成一条动态度量的唯一路线;沿着这个度量路线,一级度量一级,一级信任一级,最终确保 VMM 的动态完整性。如图 1 所示,VMM 动态度量流程如下:

1) 当系统接收到一个动态加载指令(SKINIT 为 AMD 的 SVM 指令,SENTER 为 Intel 的 TXT 指令),处理器从当前正在执行的操作系统切换到 DRTM,同时禁止中断、DMA 内存访问和调试访问操作,锁定(写保护)动态平台配置寄存器;

2)通过动态加载机制正确载入 TBM 代码,并进行完整性度量,确保可信执行,由此构建 VMM 的保护层,为 VMM 动态度量提供一个隔离的安全运行环境;

3)在受保护的动态加载会话期间,TBM 作为一个独立的度量程序,按序从内存读取 VMM 代码,进行完整性度量,并把度量值 VMM_outputs 集成扩展到硬件 TPM 的动态平台配置寄存器(dPCR 17-20);

4)生成 dPCR 签名,并将签名连同 VMM_outputs 发到外部验证器 Verifier;

5)由 Verifier 检验 VMM 数据与代码完整性是否受到破坏,将完整性度量结果与正确基准值进行比较,给出完整性度量结论。

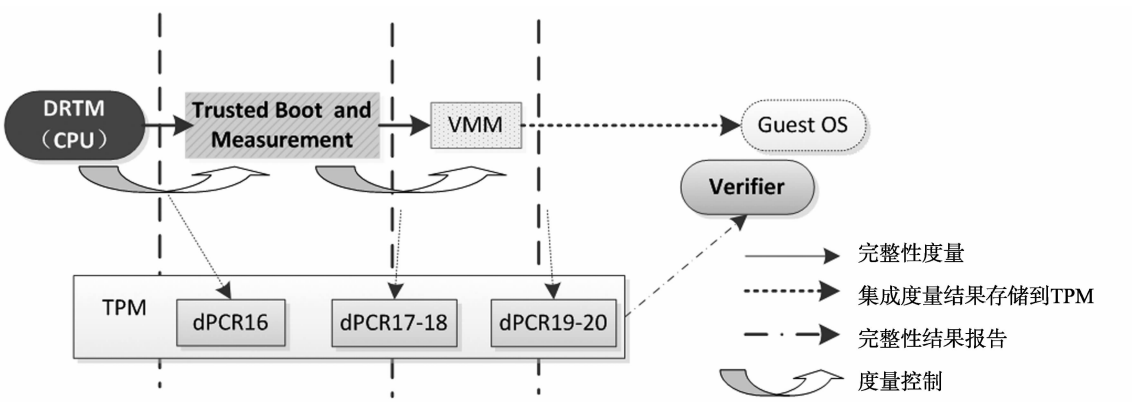

图 1　VMM 动态度量流程

Fig. 1　VMM dynamic measurement process

2.2　LS² 扩展

LS² 基于网络协议和内存保护,着重分析安全系统安全属性,支持可信计算协议分析,包括基于 DRTM 的协议[20]。LS² 主要由三部分构成:编程模型、语义语法和证明系统。其中,编程模型可用于对安全应用系统中的行为、参与实体等进行建模,比如内存控制及程序跳转操作等;语义、语法可用于建立目标系统抽象的安全属性,通常由模态公式$[P]_I^{t_b,t_e}A$来表示:在$(t_b,t_e]$时间内,线程 I 按序执行程序 P 时,不变量属性 A 成立。证明系统主要由一些公理和规则构成,为安全属性提供形式化的推理验证。现有的 LS² 主要针对通用性信息系统的安全属性进行分析,难以对 VMM 完整性度量过程进行精确描述,因而无法满足动态度量的证明环境,所以不能将 LS² 方法直接用于分析 VMM 动态度量属性。本文针对 VMM 动态度量流程,增加

VMM 的操作语义和操作语法,对 LS^2 做必要的合理扩展,如表 1 所示。

表 1 LS^2 的语法、语义扩展

Table 1 Syntax and semantic extension of LS^2

符号	定义	符号	定义
DRTM(vmm)	动态度量可信根	¬ Reset(TBM)	TBM 没有重启
TBM(vmm)	可信引导度量 TBM	TMPvmm	VMM 可信度量属性
INIT(vmm)	VMM 初始化程序	*m*. dpcr. *i*	机器 *m* 中的第 *i* 个动态 PCR
LateLaunch(vmm)	VMM 动态加载程序	TMSvmm	VMM 度量序列
Verifier(vmm)	外部检验器程序	Ver_{vmm}	VMM 度量结果校验
Reset(vmm,I)	线程 I 重启 VMM	VMM_outputs	VMM 度量值
Jump(vmm)	VMM 跳转	eval f_{vmm}, x	由输入 x 计算函数 f_{vmm}
Reset(TBM)	TBM 重启		

证明系统是 LS^2 的核心,其中 Reset、Jump 推理规则直接影响动态度量属性验证过程的正确性验证。为了保证对 VMM 正确动态度量,根据 VMM 动态度量流程,在操作语法上约定 TBM 必须在 VMM 之前启动,同时 VMM 重启或程序跳转时 TBM 保持正常运行(不能重启)。为此,基于 VMM 动态度量环境,进行 $Reset_{vmm}$、$Jump_{vmm}$规则必要扩展。

规则 2.1 ($Reset_{vmm}$扩展规则):

$$\frac{(\forall Q \in IS(INIT(vmm)) \wedge (\neg\ Reset(TBM)\,on(t_b,t_e))) \vdash [Q]_I^{t_b,t_e} A(t_b,t_e)}{\vdash Reset(vmm,I)@t \supset \forall t'.\ (t' > t) \supset A(t',t)}。$$

$Reset_{vmm}$扩展规则针对 VMM 的加载与重启。在 VMM 载入、初始化(INIT(vmm))过程中,如果 TBM 正常运行(没有重启),即(¬ Reset(TBM)on(t_b,t_e)),那么在 VMM 重新加载启动后,之前的完整性动态度量属性(不变量公式 $A(t_b,t_e)$)保持成立;否则,导致完整性度量异常,当前运行的 TBM 就难以确保当前程序序列的正确性。

规则 2.2 ($Jump_{vmm}$扩展规则):

$$\frac{(\forall Q \in IS(P) \wedge (\neg\ Reset(TBM)\ on\ (t_b,t_e))) \vdash [Q]_I^{t_b,t_e} A(t_b,t_e)}{\vdash Jump(I,P)@t \supset \forall t'.\ (t' > t) \supset A(t',t)}。$$

$Jump_{vmm}$扩展规则主要侧重于未知程序的加载。VMM 中的程序 P 跳转后,如果 TBM 没有重启,程序 P 保持之前的可信度量属性 A。由此,在 VMM 完整性度量过程中每次程序跳转时,TBM 必须正确加载,并且提供完整性度量与扩展功能,不然难以确保 dPRC 内度量值的正确性。

要保证 VMM 动态度量执行的正确性,那么度量过程必须满足动态 PCR 锁定属性、LateLaunch(vmm)程序不变性和 TBM(vmm)程序不变性。以下对这三个属性分别定义。

定义 2.1 动态 PCR 锁定属性 存在一个时间点 t 和线程 J,在时间 t 线程 J 发起动态加载 LateLaunch(vmm)时,即锁定 m. dpcr. i,把 dinit 写入 dPCR,在时间点 t 之前不跳转任何程序。将这一定义应用扩展 LS^2 形式化表述为

$$Lock_{dPCR} \equiv \exists t_0,t_1,t_2,J\ \wedge (t_0 < t_1 < t_2) \wedge (LateLaunch_{(vmm,J)}@t_1) \supset$$

$$(Mem(m.\ dpcr.\ i,dinit)@t_2) \wedge (IsLocked(m.\ dpcr.\ i,J))\ on\ (t_1,t_2)) \wedge (\neg\ Jump(J)\ on\ (t_0,t_1))。$$

定义 2.2 LateLaunch(vmm)**程序不变性** 存在一个时间点 t 和线程 J,仅当把 TBM(vmm)整个代码度量并扩展到 m. dpcr. i 之后,线程 J 于时间 t 跳转到程序 TBM(vmm),在这期间锁定 m. dpcri,表述为

$$[LateLaunch_{(vmm)}]_J^{t_b,t_e} \equiv \exists t,J \wedge (t_b < t < t_e) \wedge (Extend(J,m.\ dpcr.\ i,\ TBM(vmm))@t) \supset$$

$$(Jump(J,TBM(vmm))@t)) \wedge (IsLocked(m.\ dpcr.\ i,J)\,on(t_b,t_e)。$$

定义 2.3 TBM(vmm)**程序不变性** 在$(t_b,t_e]$时间内,如果 VMM 没有重启或没有动态加载,在 t_b 时 m. dpcr. i锁定,并且包含序列 seq(dinit, TBM(vmm)),之后包含 seq(dinit,TBM(VMM),N,EOL),则存在一个线程 J 把随机数 N 扩展到 m. dpcr. i,然后进行函数 f_{vmm}计算,接着扩展标志 EOF,结束会话。这样,每个操作按指定的顺序执行一次,并且在程序执行期间锁定 m. dpcr. i,表述为

$$[\mathrm{TBM}_{(\mathrm{vmm})}]_J^{t_b,t_e} \equiv \forall t,x((\neg\ \mathrm{Reset}(\mathrm{vmm})\mathrm{on}(t_b,t_e])\wedge(\neg\ \mathrm{LateLaunch}(\mathrm{vmm})\mathrm{on}(t_b,t_e])\wedge$$
$$(\mathrm{IsLocked}(\mathrm{m.dpcr.i},J)@t_b)\wedge(\mathrm{Men}(\mathrm{m.dpcr.i},\mathrm{seq}(\mathrm{dinit},\mathrm{TBM}(\mathrm{vmm})))@t_b)\wedge$$
$$(t_b<t\leqslant t_e)\wedge(\mathrm{Men}(\mathrm{m.dpcr.i},\mathrm{seq}(\mathrm{dinit},\mathrm{TBM}(\mathrm{vmm}),N,\mathrm{EOL}))@t))\supset$$
$$\exists t_n,\ t_E,t_X.\ (t_b<t_1<t_2<t_3<t)\wedge$$
$$(\mathrm{Extend}(J,\mathrm{m.dpcr.i},\mathrm{N})@t_1)\wedge(\mathrm{Eval}(J,f_{\mathrm{vmm}})@t_2)\wedge$$
$$(\mathrm{Extend}(J,\mathrm{m.dpcr.i},\mathrm{EOL})@t_3)\wedge$$
$$(\mathrm{IsLocked}(\mathrm{m.dpcr.i},\mathrm{J})\ \mathrm{on}\ (t_b,t_e])。$$

2.3　VMM 动态度量形式化建模

为了确保 VMM 动态度量过程正确性,首先需要应用扩展 LS² 确立动态度量目标与假设,从而对度量过程进行形式化建模,据此验证 VMM 动态度量过程正确性。总体来说,VMM 动态度量建模与分析可按如下 3 个步骤进行:

1)确立度量建模目标与假设;

2)动态度量形式化建模;

3)动态度量正确性验证。

2.3.1　目标与假设

本文形式化建模目标是应用扩展 LS² 有效分析 VMM 动态度量的正确性:(1)建模反映 VMM 度量过程中的每个进程沿着预期序列执行,整个度量过程没有完整性缺失,满足完备性;(2)模型便于验证动态度量执行序列与度量结果的正确性。

为此,本文把上述目标抽象为 VMM 完整性的可信度量属性(trusted measurement property, TMP),具体定义如下。

定义 2.4　**VMM 完整性的可信度量属性** $\mathrm{TMP}_{\mathrm{vmm}}$　一个二元组 $\mathrm{TMP}_{\mathrm{vmm}}:=\{\mathrm{TMS}_{\mathrm{vmm}},\ \mathrm{Ver}_{\mathrm{vmm}}\}$,其中,$\mathrm{TMS}_{\mathrm{vmm}}$表示 VMM 动态完整性度量过程中所包含的唯一度量执行序列,即 DRTM→TBM→VMM;$\mathrm{Ver}_{\mathrm{vmm}}$表示对度量执行序列及度量结果的正确性校验。

定义 2.4 蕴含着确保 VMM 动态完整性可信度量正确性的先决条件,即在 VMM 动态度量过程中,从 DRTM 出发,自下而上、先后动态加载并且逐级度量 TBM 和 VMM 的代码完整性,同时度量过程中不加载额外程序,保证度量序列的唯一性与安全可靠。其中 TBM 的正确加载是 VMM 完整性可信度量的关键,而 TBM 的可信度量又强依赖于 DRTM 机制与硬件信任根 TMP。最后由外部检验器 Verifier(vmm)对整个执行序列与度量结果正确性进一步验证,由此逐级保证 VMM 动态完整性可信度量的正确性。

定义 2.5　**VMM 动态度量目标安全性**　在 DRTM 技术支持下,线程 J 首先发起动态加载,载入并度量可信引导度量模块 TBM(vmm);TBM(vmm)获得控制权后,加载并度量 VMM,并把度量值扩展到动态 PCR;最后,线程 J 扩展标志 EOL,完成动态加载会话。在动态加载会话期间均锁定 m.dpcr.i,确保 VMM 动态度量目标的安全性。应用扩展 LS² 把这种动态度量目标安全性形式化表示为 J_{VMM}:

$$J_{\mathrm{VMM}}\equiv[\mathrm{Verifier}(\mathrm{vmm})]_J^{t_b,\ t_e}\exists$$
$$t_0,t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_b,\ t_e,J,N\wedge$$
$$(t_b<t_0<t_1<t_2<t_3<t_4<t_5<t_6<t_7<t_e)\wedge$$
$$(\mathrm{New}(I,N)@t_0)\wedge$$
$$(\mathrm{LateLaunch}(\mathrm{vmm},\ J)@t_1)\wedge(\mathrm{Jump}(J,\mathrm{TBM}(\mathrm{vmm}))@t_2))\wedge$$
$$(\mathrm{Extend}(J,\mathrm{m.dpcr.i},\mathrm{VMM})@t_3)\wedge(\mathrm{Extend}(J,\mathrm{m.dpcr.i},N)@t_4)\wedge$$
$$(\mathrm{Eval}(J,f_{\mathrm{vmm}},\mathrm{input},\mathrm{output})@t_5)\wedge$$
$$(\mathrm{Extend}(J,\mathrm{m.dpcr.i},\mathrm{output})@t_6)\wedge(\mathrm{Extend}(J,\mathrm{m.dpcr.i},\mathrm{EOL})@T_7\wedge$$
$$(\mathrm{IsLocked}(\mathrm{m.dpcr.i},J)\mathrm{on}(t_1,t_7))。$$

为了验证上述 VMM 动态度量的正确性,须对 Verifier(vmm)和 TPM 进行前提假设:首先,验证器 Verifier(vmm)与 TPM 为不同部件;其次,TPM 只执行指定程序 $\mathrm{TPM}_{\mathrm{DRTM}}$(vmm),不泄漏签名私钥,不伪造签

名。前提假设由 T_{VMM} 表示,形式化描述为

$$T_{\text{VMM}} \equiv \{\text{Ver} \neq \text{AIK}(\text{vmm}), \text{Honest}(\text{AIK}(\text{vmm}), \text{TPM}_{\text{DRTM}}(\text{vmm}))\}。$$

上述假设用以保证 Verifier(vmm)和 TPM 自身的安全可靠性,是验证 VMM 动态度量正确性的前提条件。依据这个假定,可以对 VMM 动态度量的正确性进行验证。

2.3.2 动态度量形式化建模

基于上述的形式化建模目标,利用扩展 LS^2,对节 2.1 中的 VMM 动态度量流程进行形式化描述(建模)。参与动态度量过程的主要组件为:动态可信度量根 DRTM(vmm)、动态加载程序 LateLaunch(vmm)、可信引导度量模块 TBM(vmm)、可信平台模块 TPM_{DRTM}(vmm)和外部校验器 Verifier(vmm)。以下应用扩展 LS^2 形式化语法与语义逐一建模。

1)DRTM(vmm)是参与动态度量的第一个组件,由度量的可信根 CPU 执行,接收外部检验器发出的随机数 n',执行动态加载操作,为 TBM 和 VMM 的度量与校验提供一个隔离的可信环境。将这一过程应用扩展 LS^2 形式化表述如下:

$$\text{DRTM}(\text{vmm}) \equiv n' = \text{receive};\ \text{write} = \text{m.nonce}, \text{n}';\ \text{LateLaunch}。$$

2)LateLaunch(vmm)为动态度量组件,由硬件平台执行,通过动态加载技术,从固定地址 m.tbm 读取 TBM(vmm)的二进制文件;同时利用 Hook 技术和 SHA-1 哈希函数的度量方法保持对 TBM 的锁定度量,并将度量结果扩展到 TPM 的动态平台配置寄存器 dPCR,确保 TBM 代码的完整性和真实性。之后,通过跳转指令,程序跳转到 TBM(vmm)。将这一过程应用扩展 LS^2 形式化表述如下:

$$\text{LateLaunch}(\text{vmm}) \equiv \text{TBM} = \text{read m.tbm};\ \text{extend m.dpcr.i}, \text{TBM};\ \text{jump TBM}。$$

3)TBM(vmm)。为了保证完整性度量与校验的新鲜度,TBM(vmm)首先度量随机数 n'',并扩展到 dPCR。在动态加载会话期间,TBM(vmm)对 VMM 代码进行度量,计算函数 f_{vmm},并依次把度量值 VMM_outputs 和符号 EOL 扩展到 dPCR,结束动态加载会话。将这一过程形式化表述如下:

$$\begin{aligned}\text{TBM}(\text{vmm}) \equiv\ & n'' = \text{read m.nonce};\ \text{extend m.dpcr.i}, n'';\ \text{VMM} = \text{read m.vmm};\\ & \text{extend m.dpcr.i}, \text{VMM};\ \text{output} = \text{eval } f_{\text{vmm}}, \text{input};\\ & \text{extend m.dpcr.i}, \text{output};\ \text{extend m.dpcr.i}, \text{EOL}。\end{aligned}$$

4)TPM_{DRTM}(vmm)。完整性报告参与组件,由硬件 TPM 执行,对动态 PCR 值 m.dpcr.i 进行签名,并将结果连同 VMM_outputs 和 AIK(认证身份密钥)证书一起发给校验器。将这一过程形式化表述如下:

$$\begin{aligned}&\text{TPM}_{\text{DRTM}}(\text{vmm}) \equiv o = \text{read m.output};\ w = \text{read m.dpcr.i};\\ &r = \text{sign}(\text{dPCR}(i), w), \text{AIK}^{-1}(\text{vmm});\ \text{send}\ (o, r)。\end{aligned}$$

5) Verifier(vmm)。完整性报告参与组件,由 Verifier(vmm)执行,生成并发送随机数 n,接收签名的度量值,验证签名,保证其来自一个真实的 TPM;然后,校验度量值与(dinit, TBM(vmm), VMM, n, o, EOL)序列的匹配情况,如果二者一致,说明 VMM 代码完整可信,校验器返回一个“允许”标记,即允许 VMM 运行,否则,校验失败,说明该代码已经被篡改,丢弃 VMM_outputs 值,并返回一个“拒绝”标志,阻止 VMM 运行或进行代码恢复操作。将这一过程形式化表述如下:

$$\begin{aligned}&\text{Verifier}(\text{vmm}) \equiv\ n = \text{new};\ \text{send}\ n;\ (\text{o}, \text{sig}) = \text{receive};\ \text{ver} = \text{verify sig},\ \text{AIK}(\text{vmm});\\ &\text{match ver}, (\text{dPCR}(i), \text{seq}(\text{dinit},\ \text{TBM}(\text{vmm}),\ \text{VMM},\ n, o, \text{EOL}))。\end{aligned}$$

2.3.3 完整性度量证明

根据节 2.3.1 中的目标与假设和 2.3.2 中的动态度量形式化建模过程,本节使用 $\text{Reset}_{\text{vmm}}$、$\text{Jump}_{\text{vmm}}$ 扩展推理规则与文献[8]中的 LS^2 证明系统的公理,从分析 Verifier(vmm)的操作开始,由度量序列事实 seq(dinit, TBM(vmm)、VMM, N, output, EOL)和 AIK^{-1}(vmm)推导 dPCR 锁定属性,依次构建 LateLaunch(vmm) 和 TBM(vmm)程序不变性,由文献[7-8]的推导与化简方法可得 VMM 动态度量过程与结果正确性 P_{VMM}:

$$\text{P}_{\text{VMM}} \equiv [\ \text{Verifier}(\text{vmm})\ ]_I^{t_b, t_e}\ \exists$$

$$t_0, t_1, t_2, t_3, t_4, t_5, t_o, t_7, t_e, J, I' \wedge \tag{1}$$

$$(t_b < t_0 < t_1 < t_2 < t_3 < t_4 < t_5 < t_6 < t_7 < t_8 < t_e) \wedge \tag{2}$$

$$(\text{New}(I, N) @ t_0) \wedge \tag{3}$$

$$(I' = \text{AIK}(\text{vmm})) \wedge \tag{4}$$

$$(\mathrm{Sign}(i),\mathrm{dPCR}(i),\mathrm{seq}(\mathrm{dinit},\mathrm{TBM}(\mathrm{vmm}),\mathrm{VMM},N,\mathrm{output},\mathrm{EOL}),\mathrm{AIK}^{-1}(\mathrm{vmm}))@t_8)\supset \tag{5}$$

$$(\mathrm{LateLaunch}_{(\mathrm{vmm,J})}@t_1)\wedge \tag{6}$$

$$(\neg\ \mathrm{LateLaunch}(\mathrm{vmm},J)@(t_1,t_7)\wedge(\neg\ \mathrm{Reset}(\mathrm{vmm})\mathrm{on}(t_1,t_7))\wedge \tag{7}$$

$$(\mathrm{Jump}(J,\mathrm{TBM}(\mathrm{vmm}))@t_2))\wedge \tag{8}$$

$$(\neg\ \mathrm{Jump}(J,\mathrm{TBM}(\mathrm{vmm}))@t_1,t_2))\wedge \tag{9}$$

$$(\mathrm{Mem}(\mathrm{m.dpcr.i},\mathrm{seq}(\mathrm{dinit},\mathrm{TBM}(\mathrm{vmm})))@\mathrm{t}_2)\wedge \tag{10}$$

$$(\mathrm{IsLocked}(\mathrm{m.dpcr.i},J)\ \mathrm{on}\ (t_1,t_2)\wedge \tag{11}$$

$$(\mathrm{Extend}(J,\mathrm{m.dpcr.i},\mathrm{VMM})@t_3)\wedge(\mathrm{Extend}(J,\ \mathrm{m.dpcr.i},N)@t_4)\wedge \tag{12}$$

$$(\mathrm{Eval}(J,f_{\mathrm{vmm}},\mathrm{input},\mathrm{output})@t_5)\wedge \tag{13}$$

$$(\mathrm{Extend}(J,\mathrm{m.dpcr.i},\mathrm{output})@t_6)\wedge(\mathrm{Extend}(J,\mathrm{m.dpcr.i},\mathrm{EOL})@t_7)\wedge \tag{14}$$

$$(\neg\ \mathrm{Eval}(J,f_{\mathrm{vmm}})\ \mathrm{on}\ (t_2,t_5))\wedge(\neg\ \mathrm{Eval}(J,f_{\mathrm{vmm}})\ \mathrm{on}\ (t_5,t_7))\wedge \tag{15}$$

$$(\mathrm{IsLocked}(\mathrm{m.dpcr.i},J)\ \mathrm{on}\ (t_2,t_7)) \tag{16}$$

定理 2.1　基于 DRTM 的 VMM 动态完整性度量满足度量过程完备性。

证明　由完整性度量形式化建模过程可见,VMM 度量过程中的每个进程确实沿着预期序列 DRTM→TBM→VMM 执行,整个度量过程没有完整性缺失。同时,P_{VMM}包含了完整性度量过程需要满足的重要属性:式(8)~(11)为 LateLaunch(vmm)程序不变性,式(7)、(1)~(16)为 TBM(vmm)程序不变性,式(9)~(11)和(16)为 dPCR 锁定属性。由此说明基于 DRTM 的 VMM 动态度量满足度量过程完备性。

定理 2.2　基于 DRTM 的 VMM 动态完整性度量满足 $T_{\mathrm{VMM}}\vdash J_{\mathrm{VMM}}$。

证明　根据定理 2.1,结合节 2.3.1 的动态度量目标安全性 J_{VMM}与 VMM 动态度量过程与结果正确性 P_{VMM}可得

$$J_{\mathrm{VMM}}\subset P_{\mathrm{VMM}}。 \tag{17}$$

同时,基于前提假设 T_{VMM},根据 VMM 动态完整性的可信度量属性 $\mathrm{TMP}_{\mathrm{vmm}}$定义,结合 $\mathrm{Reset}_{\mathrm{vmm}}$、$\mathrm{Jump}_{\mathrm{vmm}}$扩展规则,按照 LS² 证明系统中的可靠性公理[8],可得

$$\mathrm{T}_{\mathrm{VMM}}\vdash P_{\mathrm{VMM}}。 \tag{18}$$

由式(17)和式(18)可得 $\mathrm{T}_{\mathrm{VMM}}\vdash J_{\mathrm{VMM}}$,即基于 DRTM 的 VMM 动态度量结果正确。

综上所述,基于 DRTM 的 VMM 动态度量满足度量过程完备性与结果正确性要求,达到动态度量正确性形式化建模目标。

3　结论分析

VMM 动态度量过程中,首先通过动态加载建立一个度量保护环境,保证度量的全过程均在安全的隔离环境中进行,结合锁定保护机制,保证度量过程执行的完整性;同时,把可信引导度量模块 TBM 和 VMM 的度量值扩展到由硬件 TPM 保护的 dPCR,一旦恶意软件对 TBM 和 VMM 进行恶意篡改,外部校验器将会检测到,由此保证正在运行的 TBM 和 VMM 代码完整性。

经过扩展 LS² 证明系统分析表明,DRTM 提供的动态加载机制有效保持了 LateLaunch(vmm)和 TBM(vmm)程序不变性以及 dPCR 锁定属性,对于保证 VMM 动态度量正确性起着关键性作用。同时,TPM 的签名密钥存储在硬件芯片的非易失性内存,不会泄露,TPM 签名也不会被伪造,由此保证度量内容的真实性,在一定程度上克服了基于传统 SRTM 机制的不足。

在 TCB 方面,VMM 动态度量是建立在保证启动和运行时相关硬件与软件完整性基础上。传统基于 SRTM 的 TCB 包含 TPM(SRTM)、BIOS 与 BootLoader;而基于 DRTM 的 VMM 动态度量 TCB 仅包含 TPM(DRTM)和 TBM,TCB 更短,代码量更小,避免针对 BIOS、BootLoader 的恶意攻击,安全性更高。此外,每一次度量包含一个随机数 N,N 作为身份验证的一部分信息,这使校验器可执行重复检测,抵御并防止重放攻击。

此外,在上述形式化建模与分析过程中发现,在动态度量过程中存在程序稳定性和度量执行性能问题。在动态度量时,需要暂停操作系统运行,禁止中断,系统切换相对频繁,这将影响上层相关软件的流畅运行;同时,在完整性度量过程中因存在一定数量的哈希运算操作,耗时较多,使得动态度量总体性能开销明显

增大。

4 结束语

动态度量是确保 VMM 安全可靠的重要方法。但目前缺乏针对 VMM 动态度量正确性的理论分析,为此,应用扩展安全系统逻辑 LS^2 对 VMM 动态度量进行形式化建模,明确定义度量程序不变性,并由度量过程序列事实验证动态度量的重要属性,据此形式化验证分析 VMM 动态度量过程的正确性。研究表明,应用本文扩展的 LS^2 方法分析得出的动态度量结论与该技术实际应用效果一致,说明扩展 LS^2 形式化建模分析方法有效,可为虚拟化环境安全提供理论参考。今后将在此基础上,着重针对 VMM 动态度量过程中的性能优化问题进一步开展工作。

参考文献:

[1] 赵波,向骍,张焕国,等. 虚拟机环境下并行信任关系研究与实现[J]. 电子科技大学学报, 2013, 42(1):98-104.
ZHAO Bo, XIANG Shuang, ZHANG Huanguo, et al. Research on parallel trust structure in virtualization[J]. Journal of University of Electronic Science and Technology of China, 2013, 42(1):98-104.

[2] RUTKOWSKA J. Security challenges in virtualized environments[C]//Proceedings of RSA Conference. [S. l.]: [s. n.], 2008:78-87.

[3] CHEN C, MANIATIS P, PERRIG A, et al. Towards verifiable resource accounting for outsourced computation[C]//Proceedings of the 9th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments. New York: ACM Press, 2013:167-178.

[4] 沈昌祥, 张焕国, 王怀民, 等. 可信计算的研究与发展[J]. 中国科学:信息科学, 2010, 40(2):139-166.
SHEN Changxiang, ZHANG Huanguo, WANG Huaimin, et al. Researches on trusted computing and its developments[J]. Science China: Information Sciences, 2010, 40(2):139-166.

[5] MCCUNE J M, LI Y, QU N, et al. TrustVisor: efficient TCB reduction and attestation[C]//Proceedings of 2010 IEEE Symposium on Security and Privacy. Los Alamitos: IEEE Computer Society, 2010:143-158.

[6] 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.
LIU Ziwen, FENG Dengguo. TPM-based dynamic integrity measurement architecture[J]. Journal of Electronics & Information Technology, 2010, 32(4):875-879.

[7] 常德显, 冯登国, 秦宇, 等. 基于扩展 LS^2 的可信虚拟平台信任链分析[J]. 通信学报, 2013, 34(5):31-41.
CHANG Dexian, FENG Dengguo, QIN Yu, et al. Analyzing the trust chain of trusted virtualization platform based on the extended LS^2[J]. Journal on Communications, 2013, 34(5):31-41.

[8] DATTA A, FRANKLIN J, GARG D, et al. A logic of secure systems and its application to trusted computing[C]//Proceedings of 30th IEEE Symposium on Security and Privacy. Piscataway: IEEE Computer Society, 2009:221-236.

[9] NIST. National vulnerability database[EB/OL]. [201-02-15]. http://web. nvd. nist. gov/view/vuln/search-results? query = Xen&search_type = all&cves = on.

[10] 程戈, 李聪. 可信计算环境构建机制研究进展[J]. 计算机工程与应用, 2013, 49(13):59-64.
CHENG Ge, LI Cong. Research progress of trusted computing environment[J]. Computer Engineering and Applications, 2013, 49(13):59-64.

[11] DAI Weiqi, JIN Hai, ZOU Deqing, et al. TEE: a virtual DRTM based execution environment for secure cloud-end computing [C]//Proceedings of the 17th ACM Conference on Computer and Communications Security. New York: ACM Press, 2010:663-665.

[12] NANEVSKI A, MORRISETT G, BIRKEDAL L. Hoare type theory, polymorphism and separation[J]. Journal of Functional Programming, 2008, 18(5-6):865-911.

[13] ZHANG Xing, HUANG Qiang, SHEN Changxiang A formal method based on noninterference for analyzing trust chain of trusted computing platform [J]. Chinese Journal of Computers, 2010, 33(1):74-81.

[14] 冯登国,秦宇. 可信计算环境证明方法研究[J]. 计算机学报, 2008, 31(9):1640-1652.
FENG Dengguo, QIN Yu. Research on attestation method for trust computing environment [J]. Chinese Journal of Computers, 2008, 31(9):1640-1652.

(下转第23页)

[3] 游林,杨玲. 基于指纹改进的模糊金库算法[J]. 杭州电子科技大学学报,2012, 32(5):49-52.
YOU Lin, YANG Ling. Modified fuzzy vault scheme based on fingerprints [J]. Journal of Hangzhou Dianzi University, 2012, 32(5): 49-52.

[4] HIEP D V, TRAN Q D, NGUYEN THI H L. A multibiometric encryption key algorithm using fuzzy vault to protect private key in BioPKI based security system[C]// Proceedings of 2010 IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF). Piscataway: IEEE, 2010:1-6.

[5] 滕旭. 基于二维码的证件管理研究与应用[D]. 西安:西安科技大学,2011.
TENG Xu. Research and application of identification management based on two dimensional barcode[D]. Xi′an: Xi′an University of Science and Technology, 2011.

[6] 刘艳涛,游林. 一种改进的随机性模糊金库算法[J]. 科技通报,2011, 27(2):288-292.
LIU Yantao, YOU Lin. An improved randomness fuzzy vault scheme[J]. Bulletin of Science and Technology, 2011, 27(2): 288-292.

(编辑:许力琴)

(上接第 8 页)

[15] CHEN Shuyi, WEN Yingyou, ZHAO Hong. Formal analysis of secure bootstrap in trusted computing [M]//Autonomic and Trusted Computing. Berlin-Heidelberg: Springer, 2007: 352-360.

[16] MILLEN J, GUTTMAN J, RAMSDELL J, et al. Analysis of a measured launch[R]. Mitre Corp Bedford MA, 2007.

[17] BURROWS M, ABADI M, NEEDHAM M R. A logic of authentication[J]. Operating Systems Review, 1989, 23(5):1-13.

[18] ABADI M, FOURNET C. Mobile values, new namesand secure communication[C]//Proceedings of the 28th Symposium on Principles of Programming Languages. New York: ACM Press, 2001: 104-115.

[19] 薛锐,冯登国. 安全协议的形式化分析技术与方法[J]. 计算机学报, 2006, 29(1): 1-20.
XUE Rui, FENG Dengguo. The approaches and technologies for formalverification of security protocols[J]. Chinese Journal of Computers, 2006, 29(1): 1-20.

[20] FENG Wei, QIN Yu, YU Aimin, et al. A DRTM-based method for trusted network connection[C]//Proceedings of 2011 International Joint Conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11. Los Alamitos: IEEE Computer Society, 2011: 425-435.

(编辑:许力琴)