·安全技术· 文章编号: 1000-3428(2018)03-0166-05 文献标志码: A 中图分类号: TP393

基于 TPM 2.0 的内核完整性度量框架

王 勇[a,b],张雨菡[a,b],洪 智[a,b],文 茹[a,b],樊成阳[a,b],王 鹃[a,b]

(武汉大学 a. 计算机学院; b. 空天信息安全与可信计算教育部重点实验室,武汉 430072)

摘 要: 针对当前的完整性度量技术无法支持可信平台模块(TPM)2.0 规范的问题,对 Linux 内核完整性度量架构(IMA)进行改进,设计基于 TPM 2.0 的内核完整性度量框架,同时基于 TPM 2.0 芯片实现支持 TPM 2.0 规范的 Linux 可信内核。测试结果表明,IMA 2.0 可以基于 TPM 2.0 对系统关键文件进行完整性检测,同时抵御对内核文件的篡改攻击。

关键词: 可信计算; 可信平台模块; 完整性; 度量; 扩展验证模块; 安全

中文引用格式: 王 勇,张雨菡,洪 智,等. 基于 TPM 2.0 的内核完整性度量框架[J]. 计算机工程,2018,44(3):166-170,177.

英文引用格式: WANG Yong,ZHANG Yuhan,HONG Zhi. Kernel Integrity Measurement Architecture Based on TPM 2.0[J]. Computer Engineering,2018,44(3):166-170,177.

Kernel Integrity Measurement Architecture Based on TPM 2.0

WANG Yong[a,b],ZHANG Yuhan[a,b],HONG Zhi[a,b],WEN Ru[a,b],FAN Chengyang[a,b],WANG Juan[a,b]

(a. School of Computer; b. Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, Wuhan University, Wuhan 430072, China)

【Abstract】 The problem that the current integrity measurement technology cannot support 2.0 specification of the Trusted Platform Module(TPM), it improves the Linux kernel Integrity Metric Architecture(IMA), and designs the kernel integrity metrics framework based on TPM 2.0. At the same time, based on TPM 2.0 chip, the Linux trusted kernel that supports TPM 2.0 specification is implemented. Test results show that IMA 2.0 can detect the integrity of the system key files based on TPM 2.0 and resist the tampering attacks on the kernel files.

【Key words】 trusted computing; Trusted Platform Module(TPM); integrity; measurement; Extended Verification Module(EVM); security

DOI: 10.3969/j.issn.1000-3428.2018.03.028

0 概述

可信计算(Trusted Computing)[1-2]是指在计算和通信系统中广泛使用基于硬件安全模块支持下的可信计算平台,以提高系统整体的安全性。可信[3]的核心就是以安全芯片为基础,建立可信的计算环境,确保系统实体按照预期的行为执行。

在可信计算中,可信平台模块(Trusted Platform Module,TPM)是可信计算技术的核心,也是可信计算技术的硬件载体。TPM 具有向可信平台提供可信测量、可信存储、可信报告的功能,以及向用户证实平台的可信性的功能。

文本通过研究 TPM 2.0,对目前的内核完整性度量框架进行改进,并基于国民技术的 TPM 2.0 芯片和 Linux 系统,实现改进的完整性度量框架。

1 研究背景

可信计算组织(Trusted Computing Group,TCG)在 2001 年发布了 TPM1.1 规范,随即在 2003 年发布了 TPM1.2 规范[4]。在 TPM 规范中 TCG 组织规定了 TPM 的功能、软硬件接口、安全特性和实现方式等,但是这些规范都并不完善,存在很多问题。例如,TPM1.2 中并没有提供关于对称算法的实现,授权方法比较混乱。于是在 2012 年可信计算组织发布了 TPM2.0 规范[5],在规范中提供了更加灵活稳定的 TPM 接口[6]。TPM2.0 规范相比于 TPM1.2 规

基金项目: 国家自然科学基金(61402342); 国家重点基础研究发展计划项目(2014CB340601)。

作者简介: 王 勇(1991—),男,硕士,主研方向为云安全、可信计算、系统安全; 张雨菡、洪 智、文 茹、樊成阳,硕士; 王 鹃(通信作者),副教授、博士。

收稿日期: 2017-02-11 修回日期: 2017-03-12 E-mail: 490003171@qq.com

范有很多改进。在密码算法支持上,TPM2.0 增加了对对称算法的支持[7],同时还支持厂商通过使用 TCG IDs 来增加必要的新算法,例如国内厂商可按需增加 SM2、SM3 和 SM4 算法,灵活度较之前更高。在授权管理上,TPM2.0 的增强型授权方法提供了比 TPM1.2 更多的授权方式以及更灵活的授权方法组合机制[8]。此外,在其他方面 TPM2.0 也进行了多处改进,包括提供多个控制域、更多的签名机制、更快速的密钥加载等。在使用场景方面,TPM2.0 比 TPM1.2 适应的场景更多,TPM1.2 设计的主要场景是 PC 端,未考虑到虚拟化技术的发展因而不支持使用虚拟化技术的场景。TPM2.0 则更适用于网络、云平台、服务器等更加广阔的平台。

继可信计算的概念后,将度量与可信计算结合的思想被提出,形成了以可信平台模块为可信根的可信度量方法。度量方法包含 3 个重要组成要素: 度量的时间节点,度量在系统中实现的层次,度量的对象。

文献[9]提到的 DigSig 系统,选择在可执行文件头部加入安全信息,并且在运行时验证安全信息,以解决 Linux 可执行程序中被非法篡改的问题。但是缺点在于其自身的安全性无法得到保护。文献[10]提到的 Copilot 系统是独立于系统 CPU 之外使用一个 PCI 板卡的协处理器来完成度量工作,确保度量系统本身的安全。其缺点是设计实现复杂,并且需要预先设定度量周期,不能实现真正的动态度量。文献[11]提出了一种使用 Kalman 方法的简化模型,该模型基于贝叶斯网络设计,为了使模型具有一定的动态适应性而引入了衰减及奖惩机制,但在预测的准确性、灵敏性及时间连续性上存在不足,对当前复杂的网络环境适应度不高。文献[12]提出了一种在计算栅格环境下适用的可信度量模型,该模型具有较好的可扩展性,其中重点考虑可靠性和可用性 2 个因素。文献[13]提出了一种动态信任预测认知模型,它符合人类心理认知习惯,构建了自适应的总体信任决策方法直接信任树(Direct Trust Tree,DTT),基于有效历史交互证据窗口,实现全局反馈信息的搜索与聚合,降低了网络带宽开销,并提高了模型的可扩展性。文献[14]提出了一种基于虚拟机监控器的动态完整性度量模型,实现了软件加载后对虚拟机中运行的进程进行监控。虽然该方案不依赖于专有硬件的支持,但同时也失去了硬件级别安全的优势。

IBM 研究院于 2004 年提出了完整性度量体系(Integrity Measurement Architecture,IMA)[15],基于 TPM 实现系统的完整性度量。在操作系统将程序载入到内存中时,对程序文件进行度量。这些研究成果有效推动了可信度量的研究与发展,但是仍然存在一些问题。其中一个主要的问题是,目前的度量方案都是基于 TPM 1.2 的,无法支持新一代 TPM 规范 TPM 2.0。

2　完整性度量架构

IMA 是目前使用最为广泛的完整性度量架构。它通过在内核中增加模块,使得当运行程序运行、内核模块被挂载和动态链接库被加载时,对用到的代码和关键数据(如配置文件和结构化数据)进行一次度量,并将度量结果扩展到 TPM 的 PCR10(默认)中,同时创建并维护一个度量列表(Measurement List,ML)。当远程挑战者发起挑战时,将度量列表 ML 和用 TPM 签名的 PCR10 中的度量值发送给挑战者,挑战者通过对比度量值和基准值来判断平台是否可信。

2.1　IMA 原理

IMA 的实现目标是检测到对文件的任何恶意更改,包括远程攻击、本地攻击,甚至是硬件攻击(如攻击 TPM 芯片)。恶意更改包括对数据和元数据(如文件名)的恶意更改、替代旧文件、删除文件等。

IMA 提供了一个实时的度量列表 ML,从系统启动后就会存在于内核中。IMA 作为内核中的一个安全模块在系统启动模块初始化时会检测 TPM 芯片是否存在以决定后续操作步骤。当可信平台模块 TPM 芯片存在时,IMA 则会将度量列表的完整性值扩展入 TPM 芯片的寄存器 PCR10 中。TPM 芯片的作用是保护度量列表的完整性。当针对 IMA 的度量列表的软件性攻击发生时,攻击会被检测到。如果恶意攻击是通过访问一个恶意文件来实现的,这个恶意文件的度量值会在文件被加载运行前被写入度量列表并扩展入 TPM 芯片的 PCR10 中,即使是恶意代码也无法删除 TPM 芯片中 PCR10 中的度量值。如果恶意软件攻击了远程证明软件,恶意软件也无法掩盖它本身的存在,因为恶意软件无法伪造一个签过名的假度量列表。相反,如果没有 TPM 芯片,恶意代码就很容易伪造一个无法被检测到的伪度量列表。因此,在一个可信引导的系统中,IMA 可以用来证明系统运行时的完整性。IMA 的度量和远程证明功能并不能在攻击发生时保护系统的完整性,而是能检测到系统中是否发生了恶意行为,以便能够及时地修复系统。

TCG 组织规定了一个标准的完整性证明的 XML 格式,其中包括度量列表和通过 TPM 对 PCR10 的值签名后得到的值来验证平台的完整性。无论系统被何种方式入侵,恶意软件都无法伪造一个有效的度量列表和对应的 TPM 签名。因为恶意软件无法访问 TPM 的私有签名密钥,并且在恶意软件入侵前,恶意软件的度量值就已经被扩展入 PCR 中了。恶意软件不能去掉本身的度量值,也不能伪造一个干净的 TPM 签名。

2.1.1 本地文件检查

IMA-appraisal 是 IMA 的基础功能的扩展功能，它会将被评估文件的度量基准值存储在安全扩展属性“security. ima”中，并且在之后打开文件操作时将文件的当前测量值与存储在安全扩展属性中的基准值进行对比，如果值并不匹配，则会拒绝访问该文件。在默认情况下，为了方便文件的经常变化，“security. ima”中存储的是文件的哈希值，而不是签名值。但是这也导致 IMA 无法提供强大的完整性和真实性保护来对抗离线攻击和在线攻击。例如攻击者可以很容易地伪造文件的哈希值来替换原本的安全扩展属性中的值。

2.1.2 签名文件属性

IMA-appraisal 的数字签名扩展功能是 IMA-appraisal 的扩展功能，是对原本存储在文件的安全扩展属性“security. ima”中文件的哈希值签名并再存入安全扩展属性“security. ima”中。这个功能是为不可改变的文件提供 ima-appraisal 缺乏的更加强大的完整性和真实性保护。当只读文件被访问时，需要验证签名并对比文件的当前哈希值。安全扩展属性被签名的文件不可以被修改，但是可以允许删除和替换文件以实现文件的更新。攻击者因为无法得到签名用到的私钥而很难伪造安全扩展属性签名。

IMA-appraisal 的数字签名扩展功能在 IMA-appraisal 验证文件的安全扩展属性“security. ima”时，判断文件的扩展属性是否被签名。若是被签名则会在内核中查找对应的密钥，然后验证签名和哈希值。

2.1.3 扩展验证模块

扩展验证模块(Extended Verification Module，EVM)通过对所有的安全扩展属性计算 HMAC 值，然后存储在文件的安全扩展属性“security. evm”中，提供对所有的安全扩展属性的离线保护，包括:

* security. ima(IMA 度量的文件的哈希值或签名值)
* security. selinux(SELinux 的上下文标签)
* security. smack64(打标签的文件)
* security. capability(可执行文件的能力标签)

由于用于 EVM 用的 HMAC 的算法密钥是非常敏感的信息，因此需要使用可信密钥或者父密钥为可信密钥的加密密钥。并且可信密钥要通过 TPM 芯片和当前实时测量值关联起来。

可信密钥是提供给关键的敏感应用使用的密钥，例如 EVM 的密钥。可信密钥和加密密钥均是可变长度的对称密钥，并且都是在内核中创建的。用户态中可见并存储的只有加密的 blob 数据块。可信密钥需要一个 TPM 芯片来报账密钥的安全性，而加密密钥则是可以在任何系统上使用的。所有的用户级密钥的 blob 数据块都是以二进制的方式显示和存储的。

可信密钥使用 TPM 芯片生成和密封密钥，密钥是在 TPM 芯片中，通过验证特定的 PCR 值来解封的，并且只能在 TPM 中解封。如 EVM 密钥在使用时需要验证关联的 PCR 值，在验证通过后才能被解封用于 EVM 的 HMAC 算法中。

2.1.4 IMA 的策略

IMA 度量和评估的文件是通过策略进行定制的，默认的策略是对 root 用户的文件、内核模块、可执行文件、动态链接库进行度量，对 root 用户的文件进行评估。通过对这些文件进行度量，远程验证方可以检测到关键的系统文件是否被修改或者被恶意软件执行。

用户可以添加额外的 IMA 度量和评估策略，并利用 SElinux 标签等来实现更细粒度的策略。

3 改进的完整性度量框架及其实现

基于 TPM 2.0 规范对上述完整性度量框架进行改进。

3.1 基于 TPM 2.0 的完整性度量方案

针对 IMA 不能支持 TPM 2.0 标准的问题，本文设计和实现了基于 TPM 2.0 的 Linux 内核完整性度量方案。其基本架构如图 1 所示。

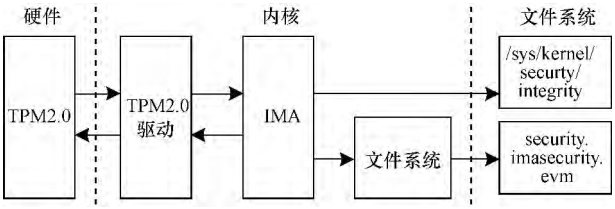

图 1 基于 TPM 2.0 的 Linux 内核完整性度量方案

支持 TPM 2.0 的 Linux 完整性度量架构由三方面构成: 底层硬件，内核模块，上层文件系统模块。底层硬件指 TPM 2.0 芯片。内核模块包括 TPM 2.0 驱动、IMA 2.0 模块和文件系统调用部分。上层文件系统模块包括读取信息的接口和文件的安全扩展属性。

该方案基于 TPM 2.0 标准，可在访问内核文件之前对文件进行度量，并把度量值加入内核常驻列表中。如果 TPM 2.0 芯片可用，则可扩展 IMA PCR，并标记 IMA PCR，允许对度量列表的远程验证。同时可对存储在文件扩展属性上的 Good 值执行本地度量验证，并保护文件的安全扩展属性免受离线攻击。

3.1.1 IMA 模块的改进

IMA 2.0 的主体结构如图 2 所示，包括以下组件: TPM 2.0 驱动，IMA 度量模块，以文件形式存在于文件系统的动态度量列表，以及以文件扩展属性存在的“security. ima”。

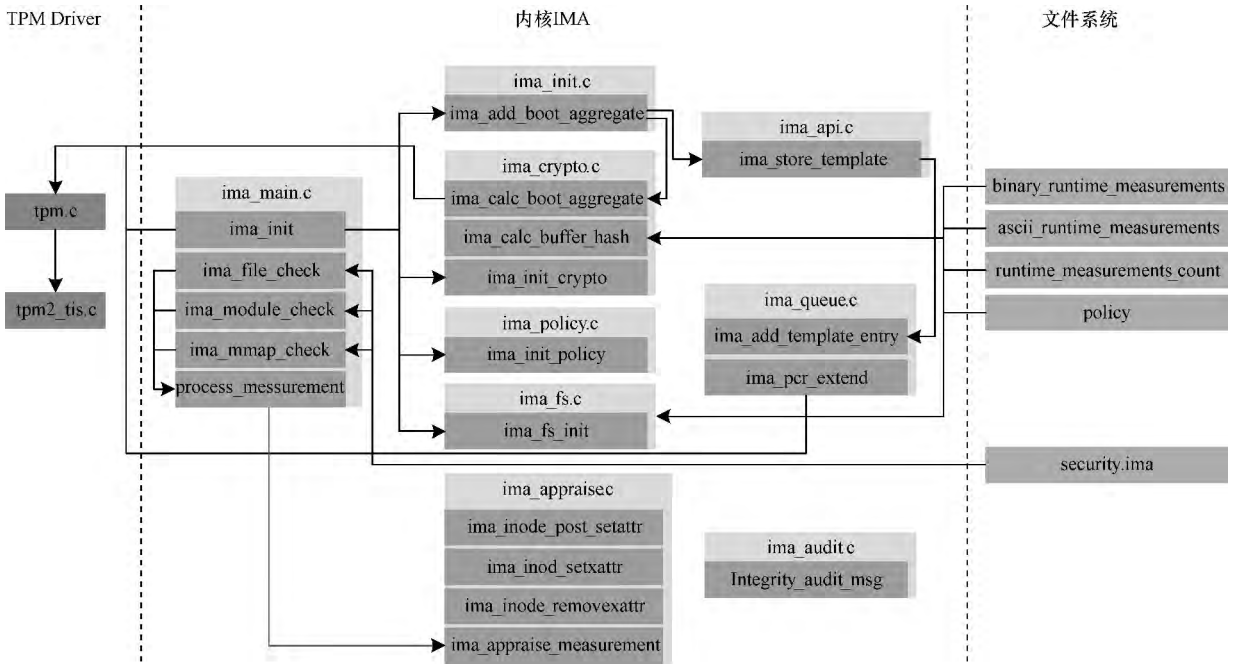

图 2　基于 TPM 2.0 的 IMA 主体结构

TPM Driver 连接了硬件 TPM 2.0 芯片和系统内核,为 IMA 提供了读取和扩展 PCR 的接口。

IMA 是最重要的主体部分,它分为 8 个子模块:

1) ima_main 模块负责调用 ima_init 模块实现初始化,同时实现了提供给内核其他模块的实现度量可执行文件、动态链接库、文件的接口(此处接口是提供给内核中其他模块调用的) 。

2) ima_init 模块负责在 Linux 内核启动时初始化 IMA 模块。为了实现可信链的延续,IMA 将 BIOS 启动和系统启动过程中写入 TPM 芯片中的 PCR 值进行一个哈希计算,即对 TPM 寄存器 PCR0-7 进行 SHA1 计算,然后通过调用 TPM2.0 驱动提供的接口将度量值扩展至 TPM 的 PCR10 中,并将该值作为内核度量列表中的首项,最后依据策略初始化 IMA 的其他模块。

3) ima_policy 模块负责初始化默认的度量策略规则。

4) ima_crypto 模块实现了 ima_calc_file_hash( ) 和 ima_calc_boot_aggregate( ) 接口计算文件和 boot-aggregate 的哈希值(度量值) ,以提供给其他模块使用。

5) ima_fs 模块负责统计 IMA 的相关状态及实时的度量列表并显示在文件系统中。

6) ima_queue 模块负责实现一个存储度量值的队列,并且将度量值扩展到 TPM 芯片的 PCR10 中。其中值得注意的是: 实时的动态列表是只扩展的,在一个启动周期中动态列表的条目只增加不修改也不删除。

7) ima_api 模块实现了一些供其他模块调用的接口。ima_audit 模块负责将 IMA 运行过程中的一些信息输出。

8) ima_appraise 模块实现了设置和删除文件节点的安全扩展属性“security. ima”的接口给内核中负责安全的模块,再由安全模块提供接口为内核中的文件系统设置和删除安全扩展属性。

3.1.2　TPM 2.0 内核驱动

在 IMA 和 EVM 中需要调用 TPM 进行 PCR 扩展和请求密钥等操作,而 Linux 内核中只有 TPM1.2 驱动,因此编写了 TPM 2.0 内核驱动。TPM 1.2 和 TPM 2.0 对外提供的 tpm_tis 接口并没有发生变化,只是在对内的具体实现上发生了改变。所以在编写 TPM2.0 的 TPM2_TIS 的过程中参考了 TPM1.2 中 tpm_tis 的架构,参照其流程,在具体实现上进行修改。

关键部分如下:

1) 驱动程序的加载和卸载入口部分

驱动程序的加载入口 module_init(init_tis) ,init_tis( ) 通过调用 Linux 提供的通用驱动程序注册接口 platform_driver_register( ) 注册 TPM2.0 设备,注册成功后调用 tpm2_tis_init( ) 完成后续的初始化工作。驱动程序的卸载入口是 module_exit(cleanup_tis) ,cleanup_tis( ) 通过调用 tpm_remove_hardware( ) 清除状态并将设备卸载。

2) 初始化驱动程序函数

初始化驱动程序函数 tpm2_tis_init( ) 的主要功能是在注册了 TPM2.0 设备后完成剩下的初始化工作。首先调用 tpm_register_hardware( ) 完成注册的其他工作,如创建文件系统中的属性设备组等,然后调用 ioremap( ) 函数设置 TPM2.0 的虚拟内存地址

空间、默认的等待时间、TPM 内部的各项标志并在日志中输出 TPM2.0 的信息。接着调用 TPM2.0 设备自检函数 tpm2_do_selftest( ),在自检完成通过后建立中断,完成初始化过程。

3) 设备自检函数

tpm2_do_selftest( ) 是设备的自检函数,它会调用 transmit_cmd( ) 将封装的自检命令和自检参数发送到 TPM 2.0 硬件设备中,输出日志信息“TPM2.0 自检…”。当 TPM 自检通过、设备初始化完成后,可以在内核中看到 TPM 设备和设备信息。

4) 编写 tpm_tis 驱动的实体结构

仿照 TPM1.2 编写 tpm2_tis 接口,以及相应的一系列 tpm2_tis 接口功能函数。

5) 编写 TPM2.0 设备属性组

设备属性组 tis_attr_grp 对应了在 TPM2.0 属性目录/sys/class/misc/tpm0/devices 下查看到 TPM 设备的属性和状态。TPM1.2 提供的属性包括: pcrs,查看 TPM 中所有 PCR 中的值; pubek,查看公钥; enabled,查看 enabled 状态; active,查看是否处于激活态; owned,查看是否被拥有; caps,查看 capbility 信息,如固件版本等; timeouts,查看超时信息。在 TPM2.0 驱动中主要实现的几个查看属性的接口是: endorseauth,从上次 tpm2_clear 后背书授权密钥是否被改变; ownerauth,从上次 tpm2_clear 后拥有者授权是否被改变; phenable,平台层次中的对象能否被使用以及平台授权和平台政策能否被用于授权; shenable,存储层次中的对象能否被使用以及拥有者权限和策略能否被用于授权。这些新增加的属性查看接口是针对 TPM2.0 的新特性,尤其是根据其在授权策略管理上的增强而增加的。

6) 编写 IMA 调用部分

作为内核模块,IMA 和 EVM 需要通过调用 TPM 驱动和 TPM 进行交互,读取扩展 TPM 的 PCR 值和发送信息给 TPM。需要编写读取 PCR 值的接口 tpm2_pcr_read( )、扩展 PCR 值的接口 tpm2_pcr_extend( )。

以上这 6 个部分基本涵盖了重编的 TPM2.0 驱动的核心代码,都是结合 TPM2.0 的新特性和当前的需求仿照 TPM1.2 完成的。其中包含的许多通用部分如 TPM 文件操作函数 tpm_write( )、tpm_read( ), TPM 硬件注册函数 tpm_register_hardware( ) 等都采取了重用的方式。

4 系统测试

在 Ubuntu12.04 版本内核上实现了 IMA 2.0 系统,其中 TPM 硬件采用国民技术的 TPM 2.0 芯片。同时对系统进行了功能和攻击测试。

4.1 功能测试

首先开启 IMA 2.0 功能,测试内核文件度量功能,如图 3 所示。系统能够对内核文件进行度量,并显示内核文件的度量值。

```
root@ubuntu:/home/test# head -5 /sys/kernel/security/ima/ascii_runtime_measureme
nts
10 0000000000000000000000000000000000000000 ima 0000000000000000000000000000000000
00000000 boot_aggregate
10 27eb54523dfb51dfb298d10e82e29f70172fc5d3 ima c712dfebbe0603e1f0ac6da77401bd66
59ac5cb0 /sbin/init
10 1b2889ac528dd07640524f1e4d05a80da7cb4709 ima ec511c6d01317d927d7ca01c982664a0
9dae2e14 /lib/i386-linux-gnu/ld-2.15.so
10 1873b704931336b3bfd02ca55d433e67c4f2cb17 ima 8dfcce2803f941bcda0bb9745e556ba1
a57499e9 /etc/ld.so.cache
10 884318a5eb0e04326b055458e50b22a21e69dce5 ima 9153d5024f01719f84a01967b6af165c
02e3d42f /lib/i386-linux-gnu/libnih.so.1.0.0
```

图 3 内核文件度量列表

4.2 攻击测试

当内核文件被修改后,IMA 2.0 可检测出内核文件完整性的改变,并禁止该文件的执行。测试结果如图 4 所示。

```
root@ubuntu:/home/test# ./test.sh
testing!
root@ubuntu:/home/test# getfattr -m . -d test.sh
# file: test.sh
security.ima=0sAYzLWjW4GFWMvza0ATYqt4axX/DC

root@ubuntu:/home/test# evmctl ima_sign --rsa test.sh
root@ubuntu:/home/test# getfattr -m . -d test.sh
# file: test.sh
security.ima=0sAwFz0lpWAAAXNy6wELqAcQEEAITcUHEveeUZY0WPpOgaHk+1NhGVy7S/xZj7toI+g
HAZ5ZY1nWUcFM+vmQNuScbjcpW9BylCxoWG78mChIkXZ2O+OJFYLPsK2sZ/TDO9ukROmlhLRnfL9PhUg
S1bgiAwaEpEf0jBrsVRywefotPpUxi6EKCkkkUgXcmcc3uQsh7Z

root@ubuntu:/home/test# ./test.sh
testing!
root@ubuntu:/home/test# echo "1" >> test.sh
bash: test.sh: Permission denied
```

图 4 内核文件攻击测试

5 结束语

完整性度量技术是当前可信计算领域中至关重要的一项技术,在构造可信计算平台的信任链、完成可信平台的远程验证方面十分重要。针对当前 Linux 完整性度量架构不能支持新一代 TPM 2.0 规范的问题,本文提出了 Linux 内核完整性度量架构改进模型,并基于国民技术 TPM 芯片和 Linux 操作系统实现了改进的 IMA 2.0 框架。功能测试和攻击测试表明,提出和实现的 IMA 2.0 框架能够对系统内核进行度量,同时检测出对内核文件的篡改。

参考文献

[1] 冯登国,秦 宇,汪 丹,等. 可信计算技术研究[J]. 计算机研究与发展,2011,48(8):1332-1349.

[2] WANG Juan,SHI Yuan,PENG Guojun,et al. Survey on Key Technology Development and Application in Trusted Computing[J]. China Communication,2016,13(11):69-90.

[3] 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.

[4] Trusted Computing Group. TCG TPM Specification 1.2[EB/OL]. (2003-08-20). http://www.trustedcomputinggroup.org.

(下转第 177 页)

[5] BANNUR S N,SAUL L K,SAVAGE S. Judging a Site by Its Content: Learning the Textual, Structural, and Visual Features of Malicious Web Pages [C] //Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence. New York, USA: ACM Press, 2011: 1-10.

[6] CANALI D,COVA M,VIGNA G,et al. Prophiler: A Fast Filter for the Large-scale Detection of Malicious Web Pages [C] //Proceedings of the 20th International Conference on World Wide Web. New York, USA: ACM Press,2011: 197-206.

[7] ESHETE B,VILLAFIORITA A,WELDEMARIAM K. Binspect: Holistic Analysis and Detection of Malicious Web Pages [C] //Proceedings of International Conference on Security and Privacy in Communication Systems. Berlin,Germany: Springer,2012: 149-166.

[8] ESHETE B,VENKATAKRISHNAN V N. Webwinnow: Leveraging Exploit Kit Workflows to Detect Malicious URLs [C] //Proceedings of the 4th ACM Conference on Data and Application Security and Privacy. New York, USA: ACM Press,2014: 305-312.

[9] CHEN G,CHOI B. Web Page Genre Classification [C] //Proceedings of ACM Symposium on Applied Computing. New York,USA: ACM Press,2008: 2353-2357.

[10] 贾梦青,王宗敏,陈　刚. 基于用户 HTTP 行为分析的网站分类研究 [J]. 计算机工程与设计,2010,31(3): 491-494.

[11] 王　涛,余顺争. 基于统计学习的挂马网页实时检测 [J]. 计算机科学,2011,38(1): 87-90,129.

[12] 黄华军,钱　亮,王耀钧. 基于异常特征的钓鱼网站 URL 检测技术 [J]. 信息网络安全,2012(1): 23-25,67.

[13] 宋江春,沈钧毅. 一种新的 Web 用户群体和 URL 聚类算法的研究 [J]. 控制与决策,2007,22(3): 284-288.

[14] 李　洋,刘　飚,封化民. 基于机器学习的网页恶意代码检测方法 [J]. 北京电子科技学院学报,2012,20(4): 36-40.

[15] 倪　平,陈正果,欧阳雄弈,等. Web 恶意代码主动检测与分析系统的设计与实现 [J]. 计算机应用,2011,31(S2): 106-108.

[16] JANG J,WOO J,YUN J,et al. Mal-netminer: Malware Classification Based on Social Network Analysis of Call Graph [C] //Proceedings of the 23rd International Conference on World Wide Web. New York, USA: ACM Press,2014: 731-734.

[17] BILGE L,SEN S,BALZAROTTI D,et al. EXPOSURE: A Passive DNS Analysis Service to Detect and Report Malicious Domains [J]. ACM Transactions on Information and System Security,2014,16(4): 1-28.

[18] KOLBITSCH C,LIVSHITS B,ZORN B,et al. Rozzle: Decloaking Internet Malware [C] //Proceedings of IEEE Symposium on Security and Privacy. Washington D. C., USA: IEEE Press,2012: 443-457.

[19] BLONDEL V D, GUILLAUME J L, LAMBIOTTE R, et al. Fast Unfolding of Communities in Large Networks [J]. Journal of Statistical Mechanics: Theory and Experiment, 2008(10): 155-168.

[20] URVOY T, CHAUVEAU E, FILOCHE P, et al. Tracking Web Spam with HTML Style Similarities [J]. ACM Transactions on the Web,2008,2(1): 1-28.

编辑　顾逸斐

(上接第 170 页)

[5] Trusted Computing Group. TCG TPM Specification 2.0 [EB/OL]. (2013-09-26). http://www.trustedcomputinggroup.org.

[6] CHEN Liqun, LI Jingtao. Flexible and Scalable Digital Signatures in TPM 2.0 [C] //Proceedings of the 20th ACM Conference on Computer and Communications Security. New York,USA: ACM Press,2013: 37-48.

[7] 张立强,张焕国,张　帆. 可信计算中的可信度量机制 [J]. 北京工业大学学报,2010,36(5): 586-591.

[8] ARTHUR W,CHALLENER D. A Practical Guide to TPM 2.0: Using the Trusted Platform Module in the New Age of Security [M]. Berlin,Germany: Springer,2015.

[9] APVRILLE A, GORDON D, HALLYN S E, et al. DigSig: Runtime Authentication of Binaries at Kernel Level [C] //Proceedings of the 18th USENIX Conference on System Administration. New York,USA: ACM Press,2004: 59-66.

[10] PETRONI N L, FRASER T, MOLINA J, et al. Copilot——A Coprocessor-based Kernel Runtime Integrity Monitor [C] //Proceedings of Usenix Security Symposium. Berlin,Germany: Springer,2010: 179-194.

[11] MELAYE D,DEMAZEAU Y. Bayesian Dynamic Trust Model [C] //Proceedings of the 4th International Central and Eastern European Conference on Multi-agent Systems and Applications. Berlin, Germany: Springer, 2005: 480-489.

[12] RANGASAMY K, SOMASUNDARAM T S. Trust Management System for Computational Grids [J]. European Journal of Scientific Research,2012,79(1): 15-23.

[13] 李小勇,桂小林. 动态信任预测的认知模型 [J]. 软件学报,2010,21(1): 163-176.

[14] 杨　蓓,吴振强,符湘萍. 基于可信计算的动态完整性度量模型 [J]. 计算机工程,2012,38(2): 78-81.

[15] SAILER R,ZHANG Xiaolan,JAEGER T,et al. Design and Implementation of a TCG-based Integrity Measurement Architecture [C] //Proceedings of Conference on USENIX Security Symposium. Berlin, Germany: Springer,2004: 223-238.

编辑　顾逸斐