

虚拟化安全：机遇，挑战与未来

刘宇涛，陈海波

(上海交通大学并行与分布式系统研究所，上海 200240)

摘要：随着云计算的流行，虚拟化安全问题受到了广泛的关注。通过引入额外的一层抽象，虚拟化技术为整个系统提供了更强级别的隔离机制，并为上层软件提供了一系列自底向上的安全服务。另一方面，抽象的引入所带来的复杂性提升和性能损失，都对虚拟化安全的研究带来了巨大的挑战。介绍了上海交通大学并行与分布式系统研究所近几年来在虚拟化安全领域所做的一系列具有代表性的工作，包括利用虚拟化提供可信执行环境、虚拟机监控、域内隔离等一系列安全服务，以及对虚拟化环境的可信计算基和跨域调用等方面进行优化的成果，并对当前和未来虚拟化安全领域的问题和探索方向进行了总结。

关键词：虚拟化安全；可信执行环境；虚拟机自省；域内隔离；可信计算基；跨域调用

中图分类号：TP309.2

文献标识码：A

doi: 10.11959/j.issn.2096-109x.2016.00091

Virtualization security: the good, the bad and the ugly

LIU Yu-tao, CHEN Hai-bo

(Institution of Parallel and Distributed Systems, Shanghai Jiaotong University, Shanghai 200240, China)

Abstract: The virtualization security has increasingly drawn widespread attention with the spread of cloud computing in recent years. Thanks to another level of indirection, virtualization can provide stronger isolation mechanisms, as well as bottom-up security services for upper-level software. On the other side, the extra indirection brings complexity and overhead as well, which poses huge challenges. A series of recent representative work done by the institute of parallel and distributed system shanghai jiaotong university, including providing security services of trusted execution environment, virtual machine monitoring, intra-domain isolation, as well as optimizing trusted computing base and cross-world calls in the virtualization environment. Finally the problems and directions in the space of virtualization security were summarized.

Key words: virtualization security, trusted execution environment, virtual machine introspection, intra-domain isolation, trusted computing base, cross-world call

1 引言

虚拟化的定义有广义与狭义之分。广义而言，几乎任何计算机的实体都可以被虚拟化，包括计算机、存储与网络等，相应地有系统虚拟化、存储虚拟化与网络虚拟化等。而狭义上的虚拟化一般均指系统虚拟化或者硬件虚拟化，是指在计算机上创建出多个虚拟计算环境以运行系统软件

栈。本文主要关注后者。

系统虚拟化这个概念在 20 世纪 70 年代就被提出并且流行起来，当时它的主要目的是方便多个用户共享一台物理主机，在该场景下，虚拟化环境中的用户数目少，平台管理人员可信程度高，系统的攻击面小，攻击的经济收益低，因此虚拟化安全并未受到广泛关注。直到 21 世纪，随着云计算的流行，虚拟化技术重新回到人们的视野之

收稿日期：2016-08-25；修回日期：2016-09-27。通信作者：陈海波，haibochen@sjtu.edu.cn

中,在这个时期,虚拟化被广泛应用的主要目的变成了简化部署成本,提高硬件资源利用率。其通过服务整合等方式将多台物理机整合为运行于一台物理机的多个虚拟机,从而显著提高IT利用率。近年来,系统虚拟化成为支撑云计算等计算模型的核心技术,虚拟机也成为当前各种计算平台的主要载体,如从2009年起全球新增的虚拟机数目已经超过新增的物理机数目,亚马逊2012年就在45万台物理主机上运行了超过300万台虚拟机,并预计在2016年底达到800万台虚拟机的数目。在规模如此巨大的云计算平台和虚拟化环境中,用户数量急剧增多,攻击成功所造成的经济损失被放大,整个系统的防护难度也不可避免地变大。IDC(international data corporation)在2008年的一份调查报告中就已经指出:安全性是影响云计算被广泛接受的最大的一个挑战。其中,虚拟化安全扮演了一个非常重要的角色。

近十几年来,特别是当Intel和AMD等各大硬件厂商在自家的处理器中增加了对虚拟化的硬件支持之后,各大公司和高校的研究机构都对虚拟化安全进行了大量深入的研究,并从各个角度提出了可能存在的问题以及相应的解决方案。上海交通大学并行与分布式系统研究所从早期(2007年)开始就在虚拟化安全领域进行了深入的探索,已经有一系列具有代表性的工作发表在国际顶级的会议和期刊上,并且获得了业界同行的广泛认可。本文将从以下3个方面对这一系列工作进行系统的介绍。

首先,利用虚拟化技术为操作系统和上层应用程序提供了一系列自底向上的安全服务。虚拟化技术为传统的软件行业带来了很多机遇,PaaS(platform as a service)、IaaS(infrastructure as a service)等服务平台为云计算带来了清晰的运营模式和巨大的用户数量。在这些新型的服务架构中,新的安全问题也随之而来,如如何在不可信的虚拟机操作系统上提供可信的安全执行环境?如何在外部监控虚拟机内部的执行环境等。虚拟化技术通过引入一层新的抽象,为这些问题的解决提供了新的思路。

其次,在虚拟化安全领域中,对高可信和高效能2个方面进行了深度的思考和探究,并通过

软硬件相结合的方法,提出一系列相应的解决方案。虚拟化技术通过新的抽象层的引入,固然能解决大量的安全问题,但随着虚拟化的发展,虚拟化层复杂度的提高,安全可信基和系统性能都会受到不可忽视的影响,如何克服这2个方面所带来的挑战,必将很大程度上影响虚拟化安全技术在实际系统部署中的可靠性和可行性。在该方向上的研究是极具必要性的。

最后,根据近几年学术前沿的研究内容,结合自身的经验和经历,总结当前和工作比较相关的其他研究,提出了一些虚拟化安全领域中潜在的探索方向。可以说虚拟化安全已经进入了一个新的时期,随着该领域中研究的进行,也同时出现了许多新的问题,本文尝试着列举出其中较为重要的部分,以及相应的可能的解决方案和探究方向。

图1描述了传统虚拟化环境中单个物理主机的系统架构。一般意义上,会有一个虚拟化层,处于虚拟机和硬件设备中间,该虚拟化层在功能上主要负责系统的资源管理、虚拟机的调度等。在安全机制上,由于虚拟化层具有较高的运行权限,因此可以对上层的虚拟机提供安全服务,包括攻击检测、实时监控等。一般来说,虚拟化层的可信基远小于操作系统,但是随着功能的日趋复杂,其可信基也变得越来越来大。本文以下所介绍的几个系统,根据不同的威胁模型会有不同的安全假设,从而决定层级之间相互的依赖关系和信任链。

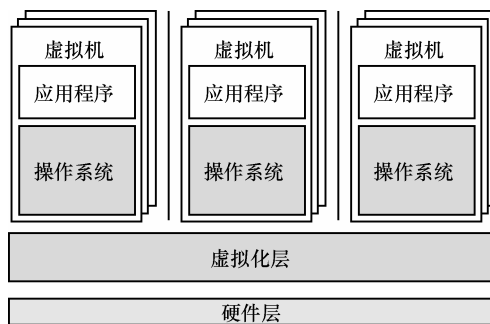


图1 传统虚拟化平台系统架构

本文首先介绍3个系统,它们以虚拟化层作为可信基,为上层虚拟机提供安全执行环境(CHAOS)、域外监控(TxIntro)和域内隔离(SeCage)机制。除此之外,将嵌套虚拟化层

(CloudVisor) 和安全处理器 (HyperCoffer) 作为可信基, 为上层虚拟机提供更加安全的保障。最后, 利用硬件特性为虚拟化环境提供一个更加高效的跨域调用机制 (CrossOver) (如图 2 所示)。

2 机遇：自底向上的安全服务

计算机领域中有一句名言, “任何计算机的问题都可以通过额外一层抽象解决”^[1]。在计算机发展的历史中, 有许多例子说明了这一点, 包括虚拟内存、文件系统、分布式系统等。虚拟化也是这些成功案例的一种。除了资源整合、虚拟机隔离等功能之外, 虚拟化技术本身通过新的这一层抽象, 就可以更好地为上层虚拟机提供一系列自底向上的安全服务, 为创造安全可信的执行环境提供了一个极为有效的道路。

随着包括 Linux、Windows 在内的操作系统功能日趋完善, 代码结构也日益复杂, 从而漏洞存在的可能性也逐渐增大。根据 CVE (common vulnerabilities and exposures) 的统计, 从 1999 年至今, Linux 内核的漏洞达到 1 437 个, 其中近 5 年的漏洞数目更是达到顶峰; Windows 的内核同样脆弱, 单单 Windows 7 这一个操作系统, 从 2009 年至今就有 577 个漏洞被披露, 更别说那些还未被披露的零日漏洞; 除此之外, 各种服务软件、运行库等代码量都极其庞大, 中间存在可能被攻击者利用的漏洞数不胜数。这些日趋复杂和不安全的运行环境使在云计算平台中运行安全敏感的应用和服务变得极为困难。虚拟化技术为此提供了一个巨大的机

遇。首先, 相比于操作系统层, 虚拟化层本身的复杂度急剧降低, 可信度也随之提高; 其次, 新引入的虚拟化层可以对上层软件的关键操作进行拦截, 同时对 CPU、内存、I/O 等资源进行统一管理和调配, 从而对整个系统具有唯一的支配权, 可以通过预先制定的规则对上层虚拟机进行有效的管理; 最后, 随着各大硬件厂商对虚拟化提供越来越多的硬件支持, 虚拟化层能够为上层软件提供更为细粒度和高效能的安全服务。

本文介绍其中 3 个具有代表性的工作, 分别为上层应用程序提供了可信执行环境, 细粒度的域内隔离, 以及为虚拟机提供高效的域外监控服务。

2.1 安全执行环境

CHAOS^[2,3]作为早期提出在不可信操作系统之上建立安全可信执行环境的系统, 开拓了利用虚拟化层保护虚拟机应用层的研究方向。CHAOS 的目的在于解决操作系统巨大的可信基所带来的安全隐患, 它的主要贡献在于引入了一层轻薄的虚拟化层, 为上层应用程序提供包括内存隔离、数据加解密、关键操作拦截在内的一系列安全服务, 如当安全敏感的应用程序需要往磁盘中写敏感数据, CHAOS 虚拟化层会拦截相关操作, 对数据进行加密和完整性保护, 从而防止恶意的操作系统和其他程序窃取或者篡改隐私数据。

图 3 描述了 CHAOS 的总体框架, 假设在虚拟化层上运行的虚拟机操作系统内核是不可信的, 它可能会窃取关键应用服务的敏感数据, 破

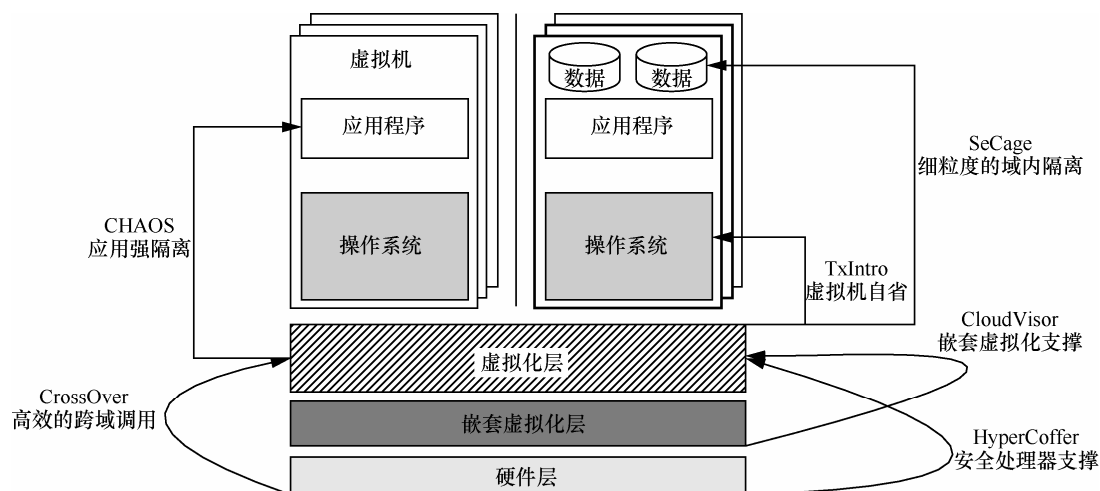


图 2 本文所描述的虚拟化相关安全解决方案

坏数据的完整性等。为了在这种场景下为关键应用提供一个安全的执行环境，引入了安全 VMM (virtual machine monitor) 的概念。CHAOS 是基于 Xen-3.0.2 实现的，它主要保护 3 种类型的敏感数据：CPU 上下文、内存数据和 I/O 数据，因此相应地采用了 3 项技术。第一，拦截关键操作，CHAOS 通过二进制重写技术将安全服务进程中的系统调用入口从原来的 `int 0x80` 改成了 `int 0x81`，从而使其进行系统调用的时候下陷到虚拟化层，另外，通过修改 IDT (interrupt descriptor table)，CHAOS 可以拦截所有安全服务进程用户态和内核态的转换操作，对 CPU 上下文进行存储和恢复，并且将一些敏感的寄存器信息保护起来，防止泄露给恶意的操作系统。第二，内存隔离，CHAOS 通过内存映射机制，将安全服务进程的页表从操作系统的映射中去除，并且对安全服务进程内存页的访问进行跟踪，防止未被授权的读取和修改，从而保证安全服务进程内存数据的隔离。第三，I/O 数据封装，CHAOS 通过 I/O 系统调用拦截和页表跟踪，对端口 I/O 和内存映射 I/O 进行加解密的操作，在数据写入磁盘之前进行加密操作和完整性元数据的更新，读出磁盘并传送给安全服务进程之前进行完整性的检查和解密操作，从而防止恶意操作系统对磁盘数据的窃取和篡改。

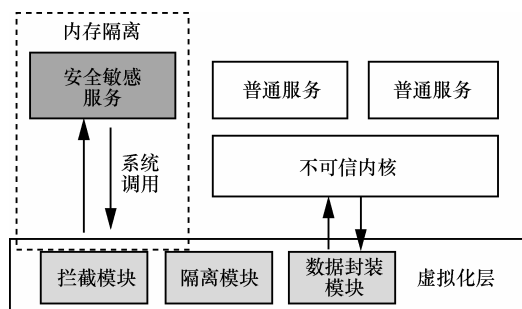


图 3 CHAOS 系统架构

2.2 高效的虚拟机域外监控

在增强计算机安全的诸多技术中，VMI (virtual machine introspection) 是一种新兴的技术。总的来说，虚拟机自省是一种域外监控的机制，其利用 VMM，从虚拟机外部增强虚拟机自身的安全。通过直接扫描虚拟机的运行时内存、磁盘、监控网络行为等操作，VMM 可进行杀毒、

网络防火墙等安全操作。相比传统的运行在虚拟机内部的安全软件（如杀毒软件等），虚拟机自省的优势在于，其安全性不依赖于虚拟机本身。例如在传统平台，恶意软件一旦感染主机成功，首先会禁止安全软件的功能，甚至将其直接卸载；而使用虚拟机自省技术，安全软件本身运行在虚拟机外部，因此，即使恶意软件已经感染了虚拟机，也无法干扰安全软件的执行。这得益于虚拟化平台对 VMM 与虚拟机的执行隔离能力。

然而，VMM 与虚拟机的强隔离也带来了语义鸿沟的问题，即 VMM 无法获知虚拟机的内部语义。例如，VMM 虽然可以读取虚拟机的所有内存，却无法直接获知内存数据的语义信息，这为安全软件的开发带来了很大的挑战。因此，当前的虚拟机自省系统均尝试通过不同的方法，尽可能缩小语义鸿沟。本文经过分析发现，现有的虚拟机自省系统普遍存在以下 3 个问题：①没有一套机制可以及时地触发虚拟机自省；②在虚拟机自省过程中需要暂停虚拟机，造成性能下降；③即使暂停了虚拟机，仍然会发生虚拟机数据读取不一致的现象。

因此，文献[4]提出了一套新的系统 TxIntro，它是一种利用硬件事务性内存进行高效的、安全的虚拟机自省的机制。事务内存是在程序并发控制领域提出的一套完整的解决方案，它是 OCC (optimistic concurrency control) 的一种解决方案，假设在大多情况下，并发的任务不会相互影响，只有在实际发生冲突的情况下才会发生事务的回滚。2014 年，Intel 在其第 4 代 Core 处理器（代号 Haswell）上装配了硬件支持的 TSX (transactional synchronization extensions) 机制，其中的一套软件接口 (RTM, restricted transactional memory) 成为市面上首个为程序员提供的硬件事务内存指令集。TxIntro 是首个利用 RTM 增强虚拟机自省技术的系统。

如图 4 所示，针对以上提到的 3 个问题，通过内存硬件虚拟化中 EPT (extended page table) 的映射机制，TxIntro 将虚拟机内核中的一些关键数据结构（如系统调用表）放入管理虚拟机用户进程中的 RTM 的读集合中，任意对虚拟机关键

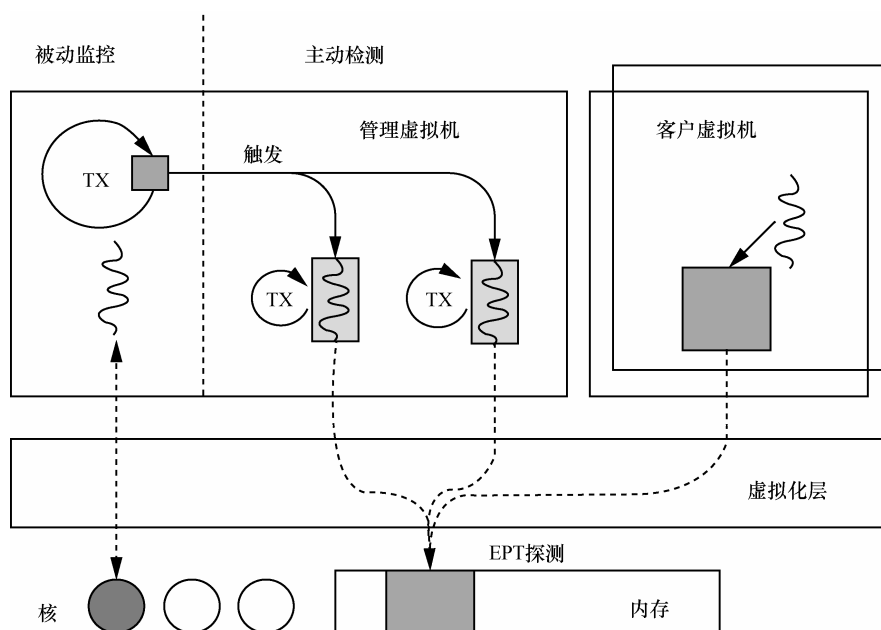


图 4 TxIntro 系统架构

数据结构的恶意修改都会及时触发虚拟机自省。另外，它将自省过程置于事务之中，这样即使不暂停虚拟机，也能防止虚拟机自省过程中数据读取的不一致，因为如果虚拟机在该过程中对相关数据进行了修改，RTM 就会中止这个事务，使其重新执行。

不过，RTM 本身具有一定的局限性，如某些系统事件（如中断、系统调用等）会无条件中止执行中的事务；另外，RTM 维护的读写集合也是有限的，经过测试，RTM 维护了 4 MB 的读集合和 32 kB 的写集合，当事务中的数据超过了读写集合的大小，事务同样会被中止。这些缺陷造成了传统的虚拟机自省过程无法在事务内存中完成。为此，TxIntro 提出了 2 个创新的优化技术：①虚拟机内部的核植入机制，即在虚拟机内部植入一个和虚拟机其他部分完全隔离的进程，该进程映射了虚拟机内核的内存，从而可以直接读取内存，避免了大量由于额外遍历页表翻译地址所造成的读写操作；②两阶段 VMI 复制机制，即将虚拟机自省过程中大量的读写操作移到事务外部执行，而在事务内部只是进行一致性的检查，从而大幅度减小了事务内存的读集合，甚至将写集合直接降为 0。

笔者在 Intel Haswell 机器上基于 Xen 实现了

TxIntro，并将它应用于虚拟机 rootkit 的检测，实验结果表明，TxIntro 可以高效地检测出现有的 rootkit，同时，它使虚拟机自省技术满足了及时性、同步性和一致性，并最小化了对虚拟机的性能影响。

2.3 细粒度的域内隔离

2014 年 4 月，被广泛运用于互联网上公钥认证和数据加密的开源库 OpenSSL 被发现有一个非常严重的安全漏洞，被称为“心脏出血”（heartbleed）。该漏洞被标识为 CVE-2014-0160，产生该漏洞的原因在于其支持的 TLS 协议中心跳（heartbeat）扩展的处理过程漏掉了一步边界检测的步骤，这个漏洞使攻击者能够从客户端与服务器端连接的内存泄露中任意读取多达 64 kB 的数据，也就是说，无需任何特权信息或身份验证，攻击者就可能从服务器中读取包括 X.509 证书的私钥、用户名、密码等在内的关键数据。另外，对 CVE 数据进行分析统计，发现无论是用户态还是内核态，隐私内存泄露是一个非常严重的问题，从 2000~2015 年，共计有 388 个已经被披露的相关漏洞，通过这些漏洞，应用程序关键数据（如 Nginx 服务器的私钥）在外部攻击或者恶意操作系统的威胁模型中很容易被泄露。

笔者发现攻击之所以成为可能,是因为关键数据和整个应用程序的软件栈处于同一个地址空间,该应用程序中任意一个漏洞都可能对关键数据的安全造成威胁。另外,操作系统内核可以映射整个用户态的内存,操作系统过于庞大的可信计算基使关键数据被暴露在更大的威胁模型上。细粒度的域内隔离技术是针对这个问题最有效的手段之一,然而如何高效地对大型软件进行解耦,并且防止在隔离的过程中造成过大的性能损失,是亟需解决的 2 个问题。

为了实现细粒度且高效可行的域内隔离机制,笔者引入虚拟化环境,并且利用 Intel 硬件虚拟化中的新特性,提出一套完整的解决方案。SeCage^[5]是一种针对应用程序中关键数据和相关代码的高效且细粒度的隔离方法,它通过混合分析技术,对应用程序进行解耦,将操作关键数据的关键代码从主程序中分离出来,放入一个通过 Intel 硬件虚拟化内存隔离出来的安全运行环境中。同时利用 Intel 硬件虚拟化提供的 VMFUNC 特性,来提升程序执行流的性能和安全性。

如图 5 所示,SeCage 隔离方法主要包含了 3 个步骤:①通过静态和动态分析相结合的方式,得到预定义好的关键数据的数据流,找到该数据可能流向其他内存区域的数据合集,以及和这些

数据合集相关的函数代码,并且通过自动化的方法和 GCC(gnu c compiler)提供的 section attribute 属性,将关键数据和相关代码从应用程序中解耦出来,放入连续的内存段中;②在虚拟化环境 EPT 内存管理的基础上,通过内存隔离机制,为关键数据和相关代码创建独立的扩展页表,使一般代码无法访问关键数据;③利用 CPU 处理器硬件虚拟化模块提供的特性 EPTP Switching,在运行时通过跳板函数实现高效的执行环境的切换。其中,EPTP Switching 的功能是在不产生虚拟机下陷的情况下改变 EPT 最高级页表指针的值。由于在第 2 步时创建了两套相互隔离的 EPT 页表,将它们对应的最高级页表指针的值填入 EPTP(EPT pointer)数组中,并将该数组的首地址写入数据结构 VMCS(virtual machine control structure)的一个特定的域中。因此,在客户虚拟机的跳板函数中调用 VMFUNC 相关指令,并将寄存器设为相应的值,就可以切换不同的 EPTP,从而达到在不下陷的情况下高效地将当前的 EPTP 切换到之前配置好的任意一个 EPTP 的目的。

SeCage 是基于 KVM,实现在 Intel Haswell 机器上的,笔者将该机制应用在 Nginx+OpenSSL, OpenSSH 和 CryptoLoop 上,成功阻止了 HeartBleed、rootkit 以及相关内核内存泄露的攻

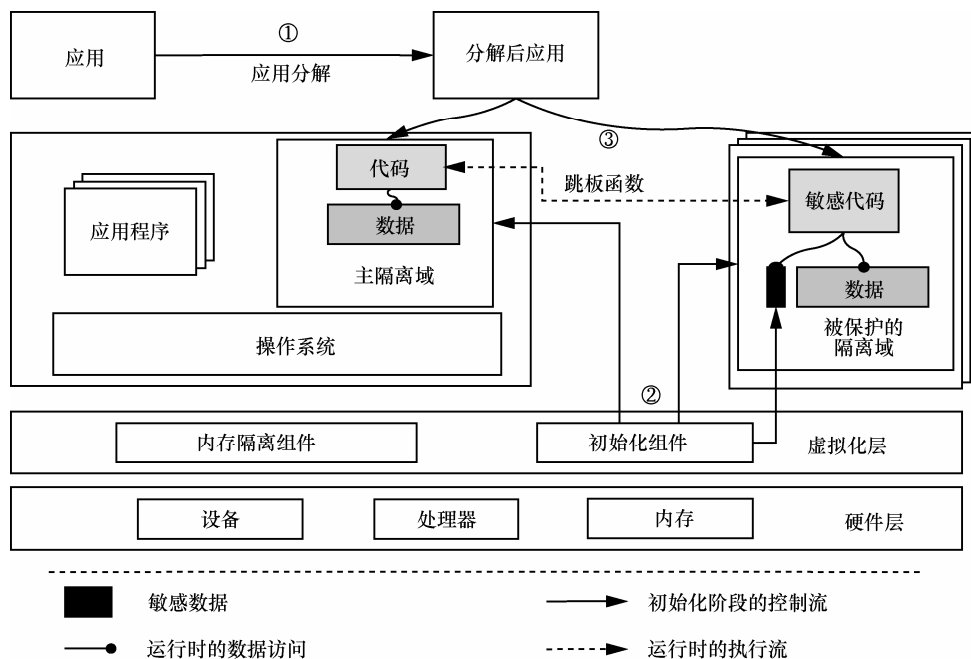


图 5 SeCage 系统架构

击，保护了包括私钥在内的关键数据不被泄露，达到了很好的域内隔离的效果。另外性能测试结果显示，SeCage 的域内隔离保护机制对 Nginx 吞吐量和延迟的影响较小，是一个可行的基于虚拟化的安全解决方案。

3 挑战：可信基与性能的考量

虚拟化通过引入额外的一层抽象可以有效地解决一系列的安全问题，对上层软件提供切实的安全服务，从而提高整个系统的可靠性。然而，虚拟化环境下安全性的保证依赖于 2 个重要的考量因素：可信基的大小和对性能的影响。第 2 节中提到的利用虚拟化技术提供自底向上的安全服务的方法建立在一个极为重要的前提上：即虚拟化层是可信的，整个系统的可信基包含了虚拟化层的所有软件栈。然而，随着虚拟化层功能的日趋复杂，其代码行数也越来越多，出现漏洞的可能性也在逐渐增大，如根据 CVE 的统计，截止到 2015 年，当前主流的 3 大虚拟化平台 VMWare、Xen 和 KVM 分别被披露了 234 个、135 个和 89 个已知的漏洞。另外，随着安全事故的发生（如 2010 年 Google 公司的雇员窃取用户的隐私数据），云平台中恶意的管理人员也成为了一个潜在的威胁。因此，学术界和工业界都在考虑将虚拟化层移除出可信计算基，并且通过更为精简的底层软件系统或者硬件特性来保证整个系统的安全性。在这个研究方向上，本文提出了两套解决方案，分别采用嵌套虚拟化和安全处理器的特性来增强虚拟化环境中客户虚拟机的安全等级。

另一方面，如果利用虚拟化技术实现的系统安全解决方案对原生系统的性能造成过大的影响，也会使相应方案无法被接受。一个不可避免的现象是，额外的抽象一定会带来额外的上下文切换，造成不必要的性能损失，而如何减少这些上下文切换，在降低性能损失的同时保证安全性不受影响，这也是该领域必须解决的问题之一。本文的另一项工作在这个方向上前进了一步，利用现有处理器中的硬件特性对一系列极具代表性的系统进行了优化，在很大程度上减少了它们所带来的额外的上下文切换和相应的性能损失。

3.1 基于软件的嵌套虚拟化

随着多租户云计算平台中虚拟化层软件栈的日益庞大和复杂，整个系统的可信基也逐渐增大，租户在虚拟机中存储的隐私数据面临着来自外部和内部共同的威胁。为了增强虚拟化平台的安全性，避免过大的攻击面，提出将虚拟化监控器移除出可信计算基。因而发现在虚拟化环境中，虚拟机隐私和完整性的保护和其他功能性的逻辑是相互独立的，因此提出 CloudVisor^[6] 系统，它采用嵌套虚拟化技术，将安全保护机制的逻辑从整个虚拟化层剥离出来，由嵌套虚拟化层通过额外的一层抽象对虚拟机监控器进行监控，从而保护虚拟机，防止虚拟机中用户的隐私数据被虚拟化层无意或者恶意地窃取。

如图 6 所示，在 CloudVisor 的架构中，租户向云平台提供加过密的虚拟机镜像和相应的元数据文件，该镜像由 CloudVisor 加载，磁盘数据的

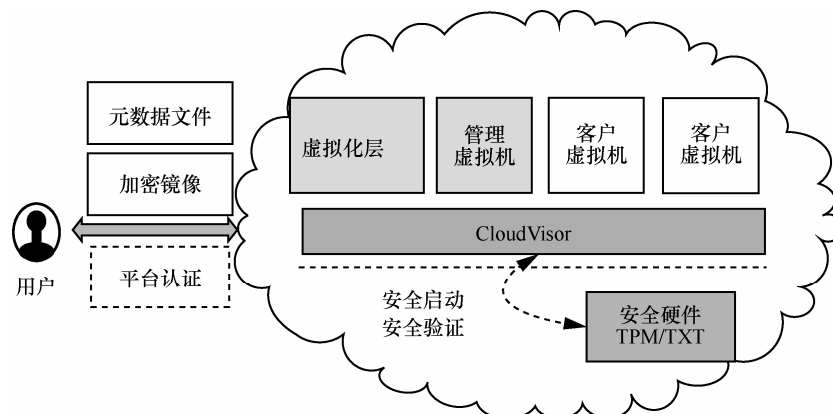


图 6 CloudVisor 系统架构

写入和读取也通过 CloudVisor 进行加解密。虚拟机监控器将被置于非特权指令集中, 并使用 EPT 表的隔离机制防止租户虚拟机数据被非法访问。虚拟机监控器对租户虚拟机进行的所有敏感操作都将下陷到 CloudVisor 中接受安全检查。另外, CloudVisor 利用安全硬件提供的 TXT (trusted execution technology) 功能为系统提供安全启动的支持, 利用 TPM (trusted platform module) 技术为租户提供安全验证支持。

CloudVisor 基于 Xen 实现, 该架构很好地将安全性与功能性解耦合, 在保证功能完备的同时尽量减小了可信计算基的大小, 有利于对可信基进行进一步的正确性验证。CloudVisor 中可信基仅有 5.5 kB 行代码, 相比于庞大的虚拟机监控器具有更加简单的逻辑及更高的可信性, 通过嵌套虚拟化的方式可以较好地控制权限的隔离。

3.2 基于安全处理器的虚拟机隔离

面对当前日益庞大和复杂的虚拟化层软件栈, 基于嵌套虚拟化技术的 CloudVisor 和类似的工作虽然能够在很大程度上缓解这个问题, 但是它仍然存在 2 个问题: ①额外层级的引入使系统上下文切换的数量增多, 性能影响比较大; ②基于软件的方法不能防止包括总线窥探和冷启动在内的物理攻击。因此提出利用安全处理器来代替嵌套虚拟化层的纯软件保护。根据研究经验表明, 防篡改的 CPU 芯片是安全的, 但是, 一般安全处理器没有足够的表达能力, 无法用于处理复杂的

高层次的语义, 如虚拟机和虚拟机监控器以及外部环境之间的数据交互; 此外, 纯粹依靠基于硬件的保护, 由于受限制的功能和有限的片上存储的可扩展性, 无法支持数量众多的虚拟机。因此, 笔者提出 HyperCoffer^[7,8]系统, 这是一个软件/硬件相结合防止恶意虚拟化监控器窃取和篡改上层虚拟机敏感数据的系统。它提出一种新的方法, 让安全处理器提供安全性增强的机制, 而将语义鸿沟交由一个小的软件层进行语义处理。

如图 7 所示, HyperCoffer 利用并扩展了传统的虚拟化环境的安全处理器技术。安全处理器在过去的 10 年已被广泛研究, 其中最主要的 2 项技术是在处理器中加入了用于实现数据机密的 AISE (address independent seed encryption) 算法和用于完整性保护的 BMT (bonsai merkle tree) 算法的逻辑。HyperCoffer 利用安全处理器中的 AISE 加密逻辑, 保证每个虚拟机的数据在芯片外部 (包括内存中) 是处于加密的状态, 只有在被加载到芯片内部时才变成明文, 并且不同的虚拟机通过使用不同的密钥被相互隔离; 另外, 通过 BMT 的逻辑为所有数据生成一个用于完整性保护的 Merkle 树, 将树的最高层父节点保存在芯片内部的数据结构中, 从而保证数据被篡改会被马上检测出来。为了处理虚拟机特定的语义, 保留上层软件的透明性, HyperCoffer 为每一个虚拟机实例提供了一个受保护的软件层, 被称为虚拟机垫片层。虚拟机垫片层处理客户虚拟机和虚拟机

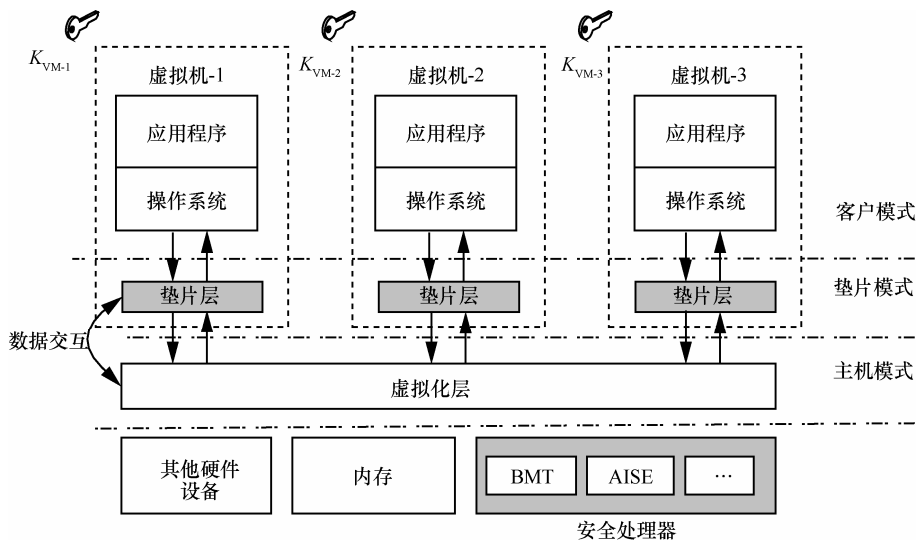


图 7 HyperCoffer 系统架构

监控器必要的交互，包含 CPU 上下文、I/O 数据和辅助信息。该垫片机制是高度可扩展的，可以支持任意数量的虚拟机实例。它作为一个重要手段，用于尽量减少所需的硬件变化和消除虚拟机和安全处理器之间的语义差距。

笔者在基于 Qemu 的全系统仿真环境以及 Xen 虚拟化环境中实现了 HyperCoffer 的原型，并且利用虚拟机垫片机制运行了未经修改的 Windows 和 Linux 虚拟机。Qemu 的仿真结果和虚拟机垫片机制的实际测量结果表明，HyperCoffer 的性能开销很小，系统的可信计算基和安全性都得到了全面的提升。

3.3 虚拟化平台中跨域调用的优化技术

利用虚拟化技术为上层软件栈提供安全服务的另外一个重要的挑战是如何降低安全服务所造成的性能损失。现有的系统架构一般都由多个层级的不同软件域构成，不管是服务的提供还是安全的保护机制都不可避免地会出现跨域的调用，而当前的硬件体系结构直接支持的跨域调用指令是有限的（如 syscall、hypercall 等），因此其他的跨域调用就需要通过拦截的方式进行模拟，使单个跨域调用实际上会有更多数目的额外的跨域上下文切换。通过研究发现，现有的一些利用虚拟化环境提高系统安全性的方法会造成 3~5 倍额外的跨域调用的数目增长，这种模拟的方式不但会造成性能的损失，也不可避免地会给整个系统带来巨大的复杂性。

针对这个问题，Li 等^[9]提出 CrossOver 机制来对当前处理器这种频繁的多层级跨域调用进行优化。CrossOver 的主要思想是将认证（authentication）和授权（authorization）分开，

将跨域间通信的认证交给硬件执行而将具体的授权交由软件执行。具体来说，在初始阶段，调用者与被调用者需要请求虚拟机监控器为它们在硬件上注册一个无法伪造的 ID，之后 2 个域之间的相互调用只需要通过一条新的指令即可直接调用，而不需要经过其他任何权限级和软件层次。CrossOver 通过 Intel 的硬件虚拟化 VMFUNC 扩展初步实现了该机制，从而保证相互间不信任的调用者与被调用者能够高效地调用，同时各自还运行在隔离的地址空间中。

图 8 描述了一个利用 CrossOver 跨虚拟机进行系统调用的例子，在进行这个系统调用之前，2 个虚拟机需要向虚拟机监控器进行注册认证，虚拟机监控器会为它们维护好相应的元数据，在这个场景中就是配置好相应的 EPT 指针。在进行系统调用时，分发管理模块会拦截该系统调用，并跳转至之前映射好的共享代码页，之后会加载一个辅助页表，保存当前上下文信息，并准备好在另一个虚拟机中需要的调用信息，然后执行 VMFUNC 指令切换 EPT 到被调用虚拟机中。被调用虚拟机中的分发管理模块会执行实际的系统调用，并将调用结果返回。将 CrossOver 实现在 Intel Haswell 机器和 Qemu 上，并且利用它优化了包括 Proxos 在内的 4 个现有的系统，极大地减小了它们跨域调用的成本，提高了系统的性能。

4 未来：新的问题与探索

关于虚拟化安全的研究在继续，除了上文提到的为上层的软件栈提供更加可信的安全执行环境和更细粒度的隐私数据保护，为虚拟机提供更为高效和精确的虚拟机监控服务，保证更加精小

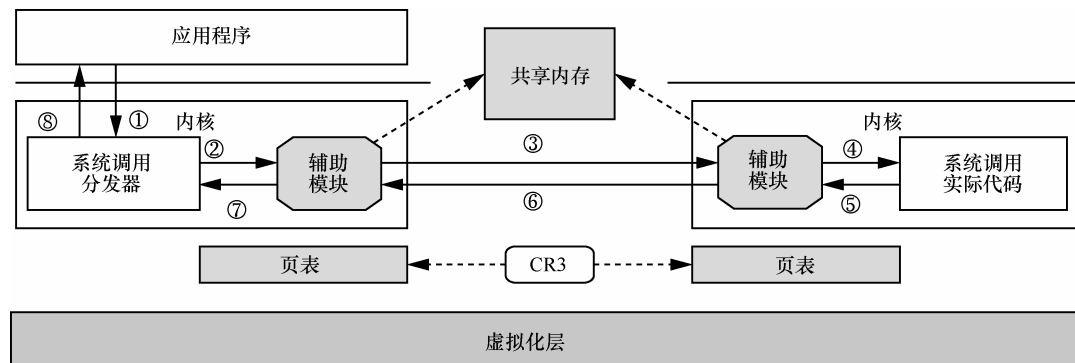


图 8 利用 CrossOver 跨域进行系统调用的例子

的可信计算基, 提供性能更好的安全机制之外, 还有一些其他问题同样值得探究, 在这节中会列举一些相关的研究, 并且说明在这些研究领域还存在哪些问题需要更好的解决方案, 以及具有哪些可能的探索方向。

在利用虚拟化技术提供安全可信执行环境这个领域, OverShadow^[10]是和 CHAOS 同个时期的系统, 它是基于影子页表的内存虚拟化技术, 通过截获虚拟机中用户态和内核态的上下文切换, 加上基于页表操作的被称为“隐身”的技术, 来保护应用程序的内存, 为操作系统和应用程序呈现不同状态(密文和明文)的内存。同时, 其引入了“垫片”的概念, 来保证整个系统的向前兼容性。但是在这一系列的工作中, 应用程序都会通过系统调用来向操作系统申请服务, 因此都无法绕过一种被称为 Iago Attack^[11]的攻击。InkTag^[12]解决了这个问题, 同时, 它是基于 EPT 内存硬件虚拟化。另外, InkTag 还简化了操作系统的验证机制, 并且提供了访问控制和文件系统的一致性的特性。最后, 还有一个被称为虚拟机回滚的攻击用于窃取虚拟机内部的隐私数据, XIA 等^[13]的研究方案在一定程度上解决了这个问题。总之, 该领域经过了长期的研究, 已经趋于成熟, 还存在的问题就是对上层应用无法进行更为细粒度的保护, SeCage 在一定程度上解决了这个问题。另一方面, 硬件提供的越来越多的特性也对高效地保护虚拟机内部安全提供了更多的可能性。笔者之前的一些工作, 如利用硬件提供的预测执行机制来增强信息流追踪技术^[14], 同时做到更好的控制流混淆^[15], 以及利用性能计数器来增强 CFI(control flow integrity)^[16,17]等, 这些技术都可以被扩展到虚拟化平台中用于保护虚拟机内部的安全。

在监控虚拟机这个研究领域中, 自从 2003 年提出虚拟机自省^[18]这个概念以来, 有大量的研究工作^[19,20]致力于解决语义鸿沟的问题, TxIntro 提出虚拟机自省中的一致性问题的利用硬件事务内存较彻底地解决了这个问题。然而在虚拟机自省防范恶意 rootkit 的领域中还有一个开放性问题, 即如何检测基于 DKOM (direct kernel object manipulation) 技术的 rootkit, 当前已经有一些基于内存快照比对的初步方案, 但是并不能彻底解决

该问题, 同时还会造成比较大的性能损失, 因此还是一个待解决的问题。

为了减小虚拟化平台中系统的可信基, 研究者们提出了各种不同的虚拟化系统架构, 除了基于嵌套虚拟化的 CloudVisor 之外, 其他的基于软件的方法还包括 NOVA^[21], 一个微内核的虚拟机监控器, 把单个整体的虚拟化层分装成一个基于组件的系统, 并通过引入不同组件之间基于组件功能的访问控制, 达到提高虚拟化层安全性的目的; DeHype^[22]将虚拟化层的大部分功能放在用户态的虚拟化层, 而在内核态维护一个称为 HyperLet 的微型虚拟机监控器, 用于处理事件通知和硬件 I/O 虚拟化; HyperLock^[23]利用内核态中内存隔离的机制, 将虚拟化层的功能在内核态进行实例的分割与隔离; Xoar^[24]通过分割特权级虚拟机的服务, 将其划分成多个可重启的非特权级服务虚拟机, 从而减小系统的可信计算基, 达到较好的错误和权限隔离。然而, 纯软件的方法都还没有被广泛应用, 除了性能和复杂度的考虑之外, 安全性也无法超越基于硬件的方法。在虚拟化平台系统重构的研究领域中, 设计出一套新的基于硬件支持的虚拟化框架将会是一个非常有吸引力的探索方向。

虚拟化层的错误隔离和恢复也是一个比较重要并且充满挑战性的研究方向, TAN 等提出的安全解决方案 TinyChecker^[25]在一定程度上解决了这个问题, 它借助了嵌套虚拟化的技术, 在虚拟化层遭遇错误的时候能够保证虚拟机快速恢复。然而错误隔离的问题并没有得到根本性的解决, 特别是在当前的虚拟化环境中, 虚拟化层的单点错误就能够造成整个系统的崩溃, 影响所有运行于其之上的虚拟机, 因此, 如何实现一个高效细粒度的虚拟化层错误隔离机制是一个非常有意义的探索课题。

除此之外, 虚拟化环境中的旁路攻击在近几年进入研究人员的视野, 大量的研究^[26-28]表明虚拟化平台中的旁路攻击已经不再只是理论上的攻击手段, 它们可以被用来窃取租户隐私等。针对这个问题, Catalyst^[29]等系统通过重新设计 CPU 缓存来缓解这个问题, 而 Duppel^[30]等系统通过引入一些随机变量来增加攻击难度。然而旁路攻击这个问题并没有得到彻底的解决, 针对 LLC (last

level cache) 的旁路攻击, 目前还没有有效的软件防御手段。现有的系统一方面需要修改硬件或者利用了最新的硬件特性, 难以在所有商用平台上推广使用, 另一方面会造成巨大的性能开销。因此, 如何设计出一个兼容性强, 性能高的旁路攻击防御系统依然是一个开放的问题。

最后, 移动平台的安全牵涉到每一个终端用户的隐私安全, 笔者之前的研究通过终端与云的相结合技术^[31]来保护终端隐私数据的安全, 并且利用基于 TrustZone 的技术^[32]来增强终端线上广告的安全机制。随着硬件的不断发展, ARM 也逐渐在移动终端加入了虚拟化技术的支持。然而移动端与服务器端的虚拟化技术相比, 在架构、应用场景等方面都具有很大的差别, 如何利用移动虚拟化平台构造一个更为可信的终端环境, 将会是一个很有前景的研究方向。

5 结束语

随着云计算的普及和资源整合的迫切需求, 虚拟化平台必然在整个软件架构中占据重要地位, 如何合理应用虚拟化层这一额外的抽象层, 为上层软件提供进一步的功能和安全的相关服务, 以及如何克服虚拟化在安全领域中所带来的问题, 是每一个云计算厂商都需要思考的问题。本文介绍了上海交通大学并行与分布式系统研究所近几年来在虚拟化安全领域做出的一系列贡献, 包括利用虚拟化为上层软件提供自底向上的安全服务, 以及探索虚拟化环境中可信计算基和性能的优化, 并且总结了现在和未来虚拟化安全领域还存在的问题和可能的探索方向。相信这些会对该领域的发展产生一定的借鉴意义。

参考文献:

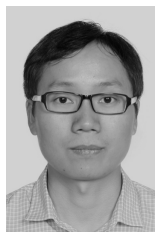
- [1] SPINELLIS D. Another level of indirection. Beautiful code[M]. CA: O'Reilly. 2007.
- [2] CHEN H, CHEN J, MAO W, et al. Daonity-grid security from two levels of virtualization[J]. Information Security Technical Report, 2007, 12(3): 123-138.
- [3] CHEN H, ZHANG F, CHEN C, et al. Tamper-resistant execution in an untrusted operating system using a virtual machine monitor[R]. FDUPPITR-2007-0801.
- [4] LIU Y, XIA Y, GUAN H, et al. Concurrent and consistent virtual machine introspection with hardware transactional memory[C]// 2014 IEEE 20th International Symposium on High Performance Computer Architecture. 2014: 416-427.
- [5] LIU Y, ZHOU T, CHEN K, et al. Thwarting memory disclosure with efficient hypervisor-enforced intra-domain isolation[C]//The 22th ACM Conference on Computer and Communications Security. 2015.
- [6] ZHANG F, CHEN J, CHEN H, et al. CloudVisor: retrofitting protection of virtual machines in multi-tenant cloud with nested virtualization[C]//The 23rd ACM Symposium on Operating Systems Principles. 2011: 203-216.
- [7] XIA Y, LIU Y, CHEN H. Architecture support for guest-transparent VM protection from untrusted hypervisor and physical attacks[C]// IEEE 19th International Symposium on High Performance Computer Architecture. 2013: 246-257.
- [8] XIA Y, LIU Y, GUAN H, et al. Secure outsourcing of virtual appliance[J]. IEEE Transactions on Cloud Computing, 2015: 1.
- [9] LI W, XIA Y, CHEN H, et al. Reducing world switches in virtualized environment with flexible cross-world calls[C]//The 42nd Annual International Symposium on Computer Architecture, ACM. 2015: 375-387.
- [10] CHEN X, GARFINKEL T, LEWIS E C, et al. Overshadow: a virtualization-based approach to retrofitting protection in commodity operating systems[C]//ASPLOS. 2008.
- [11] CHECKOWAY S, SHACHAM H. Iago attacks: why the system call API is a bad untrusted RPC interface[C]//The 18th International Conference on Architectural Support for Programming Languages and Operating Systems. 2013.
- [12] HOFMANN O S, KIM S, DUNN A M, et al. Inktag: secure applications on an untrusted operating system[C]//ASPLOS. 2013.
- [13] XIA Y, LIU Y, CHEN H, et al. Defending against VM Rollback Attack[C]//The 2nd International Workshop on Dependability of Clouds Data Centers and Virtual Machine Technology. 2012.
- [14] CHEN H, WU X, YUAN L, et al. Practical and efficient information flow tracking using speculative hardware[C]//The 35th International Symposium on Computer Architecture. 2008.
- [15] CHEN H, YUAN L, WU X, et al. Control flow obfuscation with information flow tracking[C]//The 42nd International Conference on Microarchitecture. 2009.
- [16] YUAN L, XING W, CHEN H, et al. Security Breaches as PMU deviation: detecting and identifying security attacks using performance counters[C]//ACM SIGOPS Asia-Pacific Workshop on Systems. 2011.
- [17] XIA Y, LIU Y, CHEN H, et al. CFIMon: detecting violation of control flow integrity using performance counters[C]//The 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks. 2011.
- [18] GARFINKEL T, ROSENBLUM M, et al. A virtual machine introspection based architecture for intrusion detection[C]//NDSS. 2003.
- [19] PAYNE B D, DE CARBONE M, LEE W. Secure and flexible monitoring of virtual machines[C]//ACCSAC. 2007: 385-397.
- [20] SRINIVASAN D, WANG Z, JIANG X, et al. Processout-grafting: an efficient out-of-vm approach for fine-grained process execution

- monitoring[C]/CCS. 2011: 363-374.
- [21] STEINBERG U, KAUER B. NOVA: a micro-hypervisorbased secure virtualization architecture[C]/The 5th European Conference on Computer Systems, ACM. 2010:209-222.
- [22] WU C, WANG Z, JIANG X. Taming hosted hypervisors with (mostly) depriveled execution[C]/NDSS. 2013.
- [23] WANG Z, WU C, GRACE M, et al. Isolating commodity hosted hypervisors with hyperlock[C]/The 7th ACM European Conference on Computer Systems. 2012: 127-140.
- [24] COLP P, NANAVALI M, ZHU J, et al. Breaking up is hard to do: security and functionality in a commodity hypervisor[C]/The 23rd ACM Symposium on Operating Systems Principles. 2011:189-202.
- [25] TAN C, XIA Y, CHEN H. TinyChecker: transparent protection of VMs against hypervisor failures with nested virtualization[C]/The Second International Workshop on Dependability of Clouds Data Centers and Virtual Machine Technology. 2012.
- [26] OSVIK D A, SHAMIR A, TROMER E. Cache attacks and countermeasures: the case of aes[C]/Topics in Cryptology-CT-RSA 2006. 2006:1-20.
- [27] YAROM Y, FALKNER K. Flush+ reload: a high resolution, low noise, l3 cache side-channel attack[C]/The 23rd USENIX Security Symposium (USENIX Security 14). 2014:719-732.
- [28] SHI J, SONG X, CHEN H, et al. Limiting cache-based side-channel in multi-tenant cloud using dynamic page coloring[C]/The 7th Workshop on Hot Topics in System Dependability. 2011:194-199.
- [29] LIU F, GE Q, YAROM Y, et al. CATalyst: Defeating last-level cache side channel attacks in cloud computing[C]/IEEE Symposium on High-Performance Computer Architecture. 2016.
- [30] ZHANG Y, REITER M K. Düppel: retrofitting commodity operating systems to mitigate cache side channels in the cloud[C]/The 20th ACM Conference on Computer and Communications Security. 2013: 827-837.
- [31] XIA Y, LIU Y, TAN C, et al. TinMan: eliminating confidential mobile data exposure with security-oriented offloading[C]/The 10th European Conference on Computer Systems. 2015.
- [32] LI W, LI H, CHEN H, et al. AdAttester: secure online advertisement attestation on mobile devices using trustzone[C]/The 13th International Conference on Mobile Systems, Applications, and Services. 2015.

作者简介:



刘宇涛(1989-), 男, 福建龙岩人, 上海交通大学博士生, 主要研究方向为操作系统、虚拟化安全和手机安全。



陈海波(1982-), 男, 湖南邵阳人, 上海交通大学教授、博士生导师, 主要研究方向为系统软件、系统结构与系统安全。