面向可信云计算的资源安全管理机制研究

李保珲[1], 李 斌[1], 任 望[1], 杨 光[1], 王永涛[1], 杜宇鸽[1], 张 鹏[2*]

[1] 中国信息安全测评中心 系统评估处 北京 中国 100085

[2] 中国科学院信息工程研究所 信息内容安全国家工程实验室 北京 中国 100093

摘要 数据所有权和控制权的分离对云中的程序和数据构成了严重的安全威胁, 因此, 云计算的可信性是决定其推广和普及程度的关键。本文认为, 云计算资源管理机制对云计算可信性具有关键的影响作用; 在此认识基础上, 本文首先从资源安全管理机制本身及其实现的脆弱性两大方面分析了国内外的相关研究现状; 然后, 经分析得出, 与普通网络环境相比, “共享与隔离”及“安全和性能”这两个矛盾在云计算环境中更为突出, 且这两者的完美解决更加依赖于计算体系结构和计算模式的创新; 最后, 为有效提升云计算可信性, 提出了云计算资源安全管理机制应优先着重关注的五个方面问题, 并给出了相应思考。

关键词 可信云; 资源管理; 脆弱性; 计算体系结构; 计算模式

中图法分类号 TP309 DOI 号 10.19363/j.cnki.cn10-1380/tn.2018.03.06

Research On the Resource Security Management Mechanism for Trusted Cloud Computing

LI Baohui[1], LI Bin[1], REN Wang[1], YANG Guang[1], WANG Yongtao[1], DU Yuge[1], ZHANG Peng[2*]

[1] Chinese Information Technology Securiry Evaluation Center, Department of System Evaluation, Beijing 100085, China

[2] National Engineering Laboratory Of Information Security Technologies, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

Abstract The separation of data ownership and control constitutes a serious security threat to programs and data in cloud. Therefore, the credibility of cloud computing determines its spread and popularity. For the major impact on the trustworthiness of cloud computing influenced by resource management mechanisms, this paper firstly analyses the research status at home and abroad from the from the two aspects of the resource management mechanism itself and its realization vulnerabilities. Then, we come to conclusion that the two contradictions, “sharing and isloating” and “security and performance”, are more prominent in cloud computing environment, comparing with common network environment. And, the perfect solution for the contradictions depends more on the innovation of the computing architecture and computing model. Finally, in order to effectively enhance the credibility of cloud computing, we proposed that five aspects of virtualization resource security management mechanism should be paid more attention to.

Key words trusted cloud computing; resource management mechanism; vulnerability; computing architecture; computing model

1 引言

云计算以资源聚合和虚拟化、应用服务和专业化、按需供给和灵便使用的服务模式, 提供高效能、低成本、低功耗的计算与数据服务。它是万物互联时代信息基础设施与应用服务模式的重要形态, 是新一代信息技术集约化、规模化与专业化发展的必然趋势。国家《“十三五”战略性新兴产业发展规划》(〔2016〕67 号)明确提出云计算等技术“将广泛渗透于经济社会各个领域, 信息经济繁荣程度成为国家实力的重要标志”, 应“推动云计算等技术向各行业全面融合渗透”。

然而, 作为一种新兴的资源利用方式, 云计算模式将租户数据的所有权和控制权分离, 使得云租户对其数据和业务系统的控制能力减弱。这种控制权旁落的状况导致云租户在云中的程序或数据容易

通讯作者: 张鹏, 工学博士, 副研究员, Email: zp@iie.ac.cn。

本课题得到国家自然科学青年基金(NO.61702552)、国家重点研发计划(NO.YFB0801300)、国家自然科学青年基金(NO.61402464)、国家高技术研究发展计划 863 项目“面向可信第三方的云平台可信评测技术及系统”(NO.2015AA016001)资助。

收稿日期: 2017-09-19; 修改日期: 2018-01-13; 定稿日期: 2018-02-05

被其他云内其他租户或云服务商篡改、分析或窃取, 对云租户安全构成极大风险, 文献[1]将这一问题称为云的可信性问题, 本文沿用这一称谓。在这方面, 云租户程序或数据遭受其他云租户或云服务商威胁的事件频发, 仅举几例说明之:

1. 文献[2]中, 云管理员利用其便利条件成功删除或篡改了云租户的虚拟机登录密码; Google 曾经为此类原因辞退了 2 位恶意内部员工[3];

2. 文献[4]基于 Cache 的侧信道攻击方式, 成功获取了位于同一物理机上其他虚拟机的密钥信息;

3. 攻击者利用 Hypervisor 的漏洞, 成功获取了同一物理机上其他虚拟机的控制权[5];

4. 云服务提供商可以在未经云租户授权的情形下, 轻易捕获客户虚拟机的内存以及虚拟寄存器等数据[6]。

早在 2010 年，云安全联盟已经将该类攻击认定为云计算的七大安全威胁之一[7]，另外，在一项调查研究中，76%的受调查者认为，云服务商内部员工执行攻击行为是非常可能且时常发生的[8]。可以看出，云服务的推广和普及力度，很大程度上取决于云计算的可信性。为此，产业界和学术界都在不断地提出相应的安全机制和解决方案，以消除云租户的顾虑，让企业和个人大规模地使用云计算服务。这些方法多数采用一些系统伴随策略，比如采用以流量监测为关键技术的防火墙、入侵检测和病毒防范等附加技术确保安全，然而，这些措施并没有触及云不可信问题的根源，而且以封堵、查杀等为主要手段的措施容易失效；更进一步，与其依赖外部监测等措施将威胁挡在外面，不如提升系统自身机制的安全性，使威胁无计可施，已有工作开始致力于这方面的研究和实践[9]。基于该认识，本文旨在从云计算资源管理机制这一视角出发，对云计算面临的根源性可信威胁、研究现状及解决方案进行一个全面的论述，并通过总结分析，从根本上找出提升云计算可信性的研究方法。

本文的第二部分剖析了云资源共享机制对云计算可信性提出的挑战；紧接着，阐述了围绕这些不可信因素开展的研究工作，并进行了总结分析；然后，给出了从根本上提升云计算可信性的若干建议并进行了展望，最后，总结全文。

2 资源共享形成的云可信威胁分析

云计算的根本特性就是通过资源共享机制满足多租户的弹性可扩展等需求。然而, 本文认为恰恰是当前云计算的资源共享机制是威胁云计算可信性的根本原因之一, 下面予以详细分析。

对目标成功实施攻击的前提是在遵守系统规则的基础上, 借助攻击者和攻击对象之间共享的媒介资源建立接触关系; 反之, 若攻击对象不在攻击者的攻击半径内, 则攻击将无法成功实施。仅举四例说明之: 第一, 以经典的两台物理机通过网线连接的网络攻击场景为例, 攻击者只有通过共享的物理网络资源将探测或攻击代码传输至受害者才能成功实施攻击; 第二, 以用户态的栈缓冲区溢出攻击, 攻击者只有借助共享的内存地址空间, 将程序控制流转移至 shellcode 才能成功实现攻击; 第三, 以内核态的攻击为例, 在这种情况下, 所有的内核代码通过共享的段寄存器等资源, 实现了共享内核地址空间, 因此, 一旦攻击者突破了内核中的某一薄弱点, 便会导致基于段等安全机制的保护功能失效, 从而能够访问内核的剩余部分; 第四, 以云计算的集中运维所形成的恶意内部人员问题为例, 所有的云计算中心在空间共存于云服务提供商的机房中, 且接受相同内部人员的运行和维护, 这就对恶意内部人员作恶形成了便利条件。

需要指出的是, 本文所指的资源共享是广义的, 主要包括两种共享方式: 一种是时间上的复用, 如多个 vCPU 共享同一个物理 CPU、相同内容的内核代码共享同一物理地址的内存空间, 多个进程共享同一 Cache, 多租户共享运维人员在时间的体现; 另一种是空间上的共存, 比如不同程序内容占用不同的物理地址内存空间, 内核态代码共享同一特权空间等。

相比普通网络环境, 云环境中共享资源更多、力度更强, 范围覆盖了网络、寄存器、内存、硬盘等物理资源, 因此, 云中的攻击途径更加丰富多样。表 1 列举了由于资源共享所能进行的攻击。

可以看出, 资源共享是形成安全问题的根本原因, 也是云可信性缺失的根本因素。与此同时, 云计算是在现有虚拟化、高速网络、分布式计算等众多信息技术基础之上整合而成的一种计算模式。虽然, 现有的虚拟化资源管理器满足小规模及相对封闭环境的部署需求, 但令人失望的是, 其并没有针对大规模及相对开放的商业部署环境提供或设计足够的安全特性及机制[10], 直接造成了云的资源共享和集中运维等模式存在严重缺陷。因此, 深入分析当前的虚拟化资源管理器的资源共享机制对于提升云计算的可信性具有重要的理论、指导和实践意义。

表 1 云中资源共享所形成的安全问题举例

Table 1 Problems caused by resource sharing in cloud

共享资源类型	攻击类型举例
网络	**网络攻击**: 攻击扫描与探测、Exploit 代码发送、窃取数据传输等。
寄存器	**基于 Cache 的侧信道攻击**: 由于共享 L1 或 L2 等寄存器, 可形成侧信道攻击, 实现密钥等信息窃取[4]; **突破基于段的保护机制**: 由于整个 Hypervisor 空间共享代码段寄存器, 所以, 攻击者借助 Hypervisor 的任一脆弱点进入该空间后, 便可使基于段的保护机制失效。
内存	**虚拟机逃逸攻击**: 与传统的内核漏洞一样, 虚拟机管理器一系列内存破坏方面的漏洞, 这些针对内存操纵的恶意代码可以改变程序的控制流(比如, 更改函数返回地址)或者关键数据(比如函数指针列表、跳转表等), 达到改变程序执行流程、突破原有的隔离机制以实现虚拟机逃逸等攻击[5]; 这些漏洞的发生位置一般在最高特权级发生, 而由最高特权级至底特权级转移是被允许的, 因此, 和普通网络环境下发生的由底特权级至高特权级的攻击相比, 一旦发生攻击其上的客户虚拟机以及各种应用都将受到更大程度的破坏。 **基于内存的侧信道攻击**: 通过分析并利用 KVM 的内存复用情况, 成功探测到了其他虚拟机上运行程序或打开的文件情况, 如 sshd、Apache2 以及 IE6、Firefox 等程序[11]。
硬盘	**剩余信息泄露**: 当文件系统执行删除操作时仅对 inode 等文件索引执行操作, 对实际存储在硬盘数据扇区的数据实际不执行操作时, 便会形成信息泄露风险[12]。
人力资源	**恶意内部人员**: 云服务提供商具有较高的管理权限, 可以对客户虚拟机实施启动、停止、迁移、克隆、快照等操作, 更为危险的是通过管理虚拟机可以在未经授权的情况下轻易捕获客户虚拟机的内存以及虚拟寄存器等数据。

其实, 从本质上来看, 资源共享形成的安全问题主要表现两个方面:

首先, 为了支持多租户共享 CPU、内存、硬盘等资源, 不得不引入资源管理器等软件来执行资源分配、回收、隔离及调度等功能; 但是, 这使得云软件栈更为复杂, 可信基的扩大增加了脆弱性出现的概率。同时, 资源管理器一般具有较高权限, 因此, 其造成的安全问题威胁也更大, 本文将这类问题称为资源管理机制实现的脆弱性问题。

其次, 如前面分析, 资源共享运行机制本身也能形成一些列的安全问题, 如基于共享 Cache 和内存所造成的侧信道攻击, 再比如恶意内部人员问题, 本文将这类问题称之为资源管理机制本身的问题。

下面, 分别针对这两种问题的相关研究进行分析。

3 国内外研究现状及分析

近年来, 云计算的可信性越来越成为制约云计算快速大规模推广实施的瓶颈, 本文主要就资源管理机制实现的脆弱性问题和资源管理机制本身的问题进行分析。对于后者中的集中运维问题, 因其人为因素比较突出, 本文拟将其单独讨论。故, 下文分别针对资源管理机制本身的问题、恶意内部人员以及资源管理机制实现的脆弱性问题三类不可信安全风险的相关研究工作进行分析。

3.1 研究现状

3.1.1 资源管理机制本身的问题

资源管理主要解决资源的分配、回收、共享和隔离等问题。在资源隔离方面，Hypervisor 和内核提供了多种隔离措施，如在 GDT、LDT 中，均有自己的段界限等属性，同时还设计了基于段的特权级保护模式（CPL、RPL、DPL、IOPL）、并限制某些特殊指令的使用，如 lgdt、lidt 及 cli 等，这是对描述符所描述对象的隔离保护；在分页隔离机制中 PDE 和 PTE 中的 R/W 和 U/S 等提供了页级隔离方法，同时分页机制将线性地址与物理地址加以映射，提供了对物理地址的保护。在资源共享方面，云资源的共享主要包括时间上的复用和空间上的共存两种方式，下面从这两方面进行详细分析。

需要说明的是，作为物理存储逻辑表示方法呈现的存储虚拟化设备一般只是作为单纯的数据存取载体存在、不直接参与运算过程，更进一步的，攻击一般是由运算过程中的控制流非法转移等因素形成，也就是说攻击较少发生在单纯的数据存取载体内。因此，本文不将存储虚拟化作为研究重点。另外，由于 PaaS 和 SaaS 层的资源利用方式一般是基于 IaaS 层的 Hypervisor 或内核的资源管理机制实现的，比如，在阿里 ECS 云虚拟机中安装应用软件，该 SaaS 层服务便主要是利用了底层 Hypervisor 与内核的资源管理接口。综上，囿于篇幅等原因，本文

重点探讨 IaaS 层的计算资源共享问题。

基于时间复用的资源共享问题

时间上复用是实现资源共享的主要方式之一。以内存为例，云服务提供商为了提高物理内存的使用率，会将内容重复的内存部分以共享的形式提供服务[13]，但对这种共享方式来说，这种共享方式对读操作的影响不大，但对于写操作便会执行写时复制[14]等操作；再以共享 CPU 资源为例，CPU 硬件虚拟化技术就是通过上下文切换机制实现的，也就是保存上一虚拟机的运行时状态，然后进入下一虚拟机状态，实现了不同虚拟机之间在时间维度的切换与复用。特别的，由于时间上复用内存或 cache，进程切换等时刻便会执行内容重载操作[15]，一方面，是否存在重载操作可导致执行时间的变化；另一方面，上下文切换时的剩余数据也可导致信息泄露，这两者最终可形成侧信道攻击。比如最近爆发的 Meltdown 和 Spectre 漏洞便是在时间复用上缺少了相应的保护措施而形成的。需要指出的是，在非虚拟化环境中，也存在类似的侧信道攻击，如文献[16, 17]已经在不同进程间分别成功实现了 RSA、AES 等秘密数据的获取。由于对基于内存的侧信道攻击所能获取的信息和案例少，故本文主要对基于 Cache 的侧信道攻击进行分析。

基于 cache 的侧信道攻击可以分为时间驱动和行踪驱动两种，其中，前者对攻击的条件要求较高，如不需要同驻等，但由于噪音数据的来源多且数量大，故其需要采集的数据较多；后者对攻击条件比较严格，如要求同驻等，但由于噪音数据来源少且数量小，故需要采集的数据相对较少。早在 2012 年，文献[18]便在 Amazon 云上分别实现了前述两种类型的侧信道攻击，如跨虚拟机的键盘操作监控、cache 负载测量。HomeAlone[19]也基于类似的机理，设计了验证云租户虚拟机是否运行在物理隔离的处理器上。

对于该类攻击的隔离及防御方法可以分为以下几类。一是在硬件层面设计新型的 cache 结构[20, 21]，实现资源隔离，比如文献[20]引入了分割锁缓存和随机变换缓存以此来增加推理信息的复杂度。然而，由于工艺复杂度和性能消耗大等因素，我们发现这些新兴的 Cache 结构并没有出现在主流的处理中，也没有得到大规模流行。二是针对 AES 等密码算法的攻击，通过在软件层面重构相关密码算法的实现过程，实现了抵御基于 cache 的时间侧信道密码实现算法[22, 23]；但是这种方法的适用面较窄，不具备通用性和可推广性，在近几年并没有获得大规模应用，预计在可预见的未来是不会大规模普及的。三是在现有硬件特性的基础上，通过在软件层面上设计新型的 Cache 利用机制，并以此确保 Cache 的安全利用。如 PLcache[24]通过对每一个 cache 行引入一 bit 标识，实现了动态 cache 划分；CAT[25]利用 cache 划分机制，将 cache 划分为不同的区域，可以有效抵御基于 Last-level cache 的攻击。然而，这种方法无法抵御基于划分后子区域的 time-channel 攻击。为此，CATalyst 在其基础上进行了改进，其利用 CAT 将 cache 划分安全和不安全区域，其中，不安全区域是由硬件管理且可以被任意应用使用，安全区域是由软件管理且只允许 cache-pinned 的安全页内容载入该部分，从而将 cache 做成一个软硬件综合管理的混合体；文献[26]基于 page coloring 技术实现 cache 的安全利用。与之类似，文献[27]提出了依据 cache 的大小，将物理内存划分为不同的区域，然后将不同的物理内存映射至不同的 cache，如此实现物理隔离，但是这种物理隔离的实现方法由于粒度粗、违背了资源共享的初衷，因此，其对性能的影响较大。文献[28]通过动态重映射和延长 cache 索引等技术可有效的抵御基于 Cache 的侧信道攻击，也具有一定的通用性，但是，Cache 的作用就是为了解决 CPU 运算速度和内存读写速度不匹配的问题，这就要求 Cache 的设计在缓存大小有限的前提下实现较低的缺失率、较高的复用率，然而，该方法却在较大程度上降低了 Cache 的使用效率，也因此对整个系统的性能造成很大影响。四是通过作业调度、物理资源隔离等方式，实现程序在时间或空间上的隔离执行，比如，文献[29]基于强制 VM 的确定执行来防止此类攻击的成功发生，但是这种方法需要对云计算中心的改动很大，且延长了任务执行时间，其有效性值得商榷。文献[30]通过虚拟机的放置策略进行限制，比如，通过将物理机上的 CPU 等资源划分为不同的域，与此同时，同类或属于同一部门的虚拟机放置在同一隔离域，以此避免 cache 的共享。但是，这种方法违背了云计算资源共享的特性，降低了资源利用率和计算性能。

基于空间共存的资源共享问题

空间上共存是实现资源共享的另一种主要方式，如多虚拟机共享同一物理内存、多文件共享同一硬盘存储空间。由于多实体共享同一物理设备，隔离机制本身的脆弱性等问题，容易形成隔离失效等问题（如栈缓冲区溢出攻击便实现了临近数据的覆盖），势必会形成共享实体之间的互相访问，导致实体之间的互相干扰，安全和功能便出现问题了。

比如，为了实现同驻物理机上各虚拟机之间内存资源的隔离，Xen 设计了影子页表等机制，达到了为各虚拟机划分独立地址空间的目的；再比如，为了实现硬盘资源的隔离，多数虚拟化机制设计了基于映像文件的方式来隔离不同的虚拟磁盘文件，也就是在宿主操作系统上建立一个文件来保存虚拟机的操作系统等文件，比如，VirtualBox 的 vdi 文件，VMware 的 vmdk 文件等。由于内存历来是攻防关注的重点，故下文重点对各虚拟机或各进程共享的内存资源进行分析。

其实，从本质上来看，系统虚拟化环境中的多虚拟机内存资源复用方式和非虚拟化环境中的多进程内存资源复用方式是相似的，比如，KVM 就是借助 Linux 内核本身的多进程共享机制会虚拟机提供隔离的内存资源；再比如，Xen 的影子页表技术和 Linux 内核中基于页表的地址转换技术并无本质不同，如这两种方式都需要采用基于段和页等机制的隔离措施[31]保证空间共存各部分的隔离。另一方面，分析常见的虚拟机逃逸等攻击方式可知，这类攻击与传统基于内核漏洞的攻击类似，其一般基于虚拟机管理器的缓冲区溢出、竞争条件、释放后复用等内存破坏方面漏洞，可以改变程序的控制流（比如，更改函数返回地址）或者关键数据（比如函数指针列表、跳转表等），达到改变程序执行流程、突破原有的隔离机制以实现对 Hypervisor 或者其他虚拟机的攻击。比如，文献[32]便是基于释放后复用攻击实现逃逸的。因此，本文将 Linux 内核的多进程和系统虚拟化的多虚拟机内存资源共享技术归为一类来进行分析。由于篇幅限制，本文就主要的防御机制进行分析。

栈缓冲区溢出是突破限定区的常见攻击途径之一，其通过覆盖临近区域内容，可以实现代码注入、程序控制流转移等。文献[33]将栈段设计为不可执行，从而在一定程度上阻止了注入代码获得执行权限；然而，这种方式难以组织 return-into-libc[34]的攻击方式。为此，Pax[35]所有数据内存页上代码获得执行权限，并且设计了额外的措施来阻止该种攻击方式。与之不同，文献[36]设计了基于金丝雀的保护措施，也就是在栈中的特定位置植入探测数据，通过检测该数据是否发生变化来判断是否发生了缓冲区溢出攻击。

除了栈缓冲区溢出外，堆缓冲区是黑客的第二类攻击点。为此，Windows XP SP2 采用了不可执行堆、堆 cookie 和堆管理元数据安全断开等机制应该该类问题。格式字符串溢出是黑客的第三类攻击点。FormatGuard[37]和 GUN C 库的 FORTIFY_SOURCE 均提供了相应的保护机制，比如，FormatGuard 通过静态分析输入输出类函数参数等技术降低了格式字符串攻击的成功率。

其实，攻击者在进行攻击之前，需要充分掌握进程空间的地址布局，固定的地址空间布局就使得攻击者容易掌握该类信息。基于此，文献[38]等提出了地址空间随机布局的防御机制，加大了攻击获取该类信息的难度，也因此提升了成功攻击的难度。然而，攻击者设计了堆喷射和动态代码产生喷射等技术[39]来应对这种防御措施。为此，文献[40]提出了新型的防御方法 INSeRT，其通过组合随机化机器指令核心要素以及随机植入特定陷入代码段等机制，实现了该类攻击的有效防御，且性能损耗小于 5%。

3.1.2 恶意内部人员

内部人员能够成功作恶的途径有两个: 一是利用管理软件集中且复杂的缺陷发现安全漏洞实现登录，比如，文献[41]指出，若恶意内部人员获取 Hypervisor 的特权, 可执行内存扫描攻击; 二是利用自身所处地位的优势违规对云资源进行操作, 比如, 文献[41]指出, 若可直接接触云存储服务器, 可篡改镜像模板或未加密快照等。下面分别对这两方面进行分析。

管理软件集中且复杂

对于管理软件栈集中且复杂问题, Disaggregated Xen[42]将 Dom0 中新建客户虚拟机的代码转至独立的虚拟机 DomB(Dom Builder)中, 利用客户操作系统良好的隔离特性将客户虚拟机建立代码中的脆弱点封闭在该虚拟机中, 从而不会对其他虚拟机造成影响。Xoar[43]将 Dom0 分解为 7 类不同功能的虚拟机, 以此达到更好地隔离故障和减小攻击面的目的。与之类似, SSC[44]也是通过将 Dom0 分解为称为“UDom0”的多个管理虚拟机, 并且在这些 UDom0 上设置了隔离策略。然而, 一方面, 这些工作只是简单的对 Dom0 进行功能分解, 并没有对分解后的各子部分仍可能面临的安全风险进行防范, 因此, 对 Dom0 的安全提升作用有限。与前述分割思路不同, 文献[45]一方面简化虚拟机启动程序、降低了虚拟机启动过程中与 Hypervisor 的交互次数, 另一方面, 将集中式操作简化、转变为分布式操作, 既提升了虚拟机的启动速度, 也提升了虚拟机的安全性, 该项工作为解决“隔离与性能”这个矛盾提供了有益参考。

除此之外, 还有部分工作集中在基于云租户可

控的加密等措施保证云租户隐私即便被拿走, 恶意内部人员也无法看懂。在这方面, CryptDB[46]、Mylar[47]以及 ShadowCrypt[48]等提供了对云服务商透明的云租户数据加密机制, 可有效预防恶意内部人员的内部威胁; Over-encryption[49]等设计了依赖云服务商配合的云租户数据加密服务, 在数据到云服务商之前已经过加密, 故可有效预防云服务商内部人员偷窥或窃取用户隐私。但是, 这类方法一方面需要较大的计算资源, 另一方面针对云租户程序的攻击无能为力。

违规操作

CSA 对此在管理层面上提出了若干建议, 比如控制供应链管理、建立良好的违规通报制度, 也有部分安全公司提出利用云堡垒机技术来支撑管理[50], 然而, 对于部分高级别员工存在有利条件越过网络限制, 通过直接接触服务器等实施恶意行为, 不能有效应对“最后一公里”的安全风险。

另外, 研究发现, 恶意内部人员一般伴随着明显的心理学波动或者消极的社会言论, 因此, 部分工作监控并分析内部人员的心理状态变化, 预测其恶意入侵倾向[51, 52]。还有部分工作对内部人员使用云平台服务的日志信息和行为轨迹进行分析, 挖掘其个人特征和使用意图[53, 54]。

3.1.3 资源管理机制实现的脆弱性问题

虚拟机管理器是云进行资源管理的主要部件, 而且其功能也是越来越丰富。以被广泛使用的 Xen 为例, 其提供虚拟机管理、调度、指令模拟、事件通道、内存管理等功能, 其代码量也从 2.0 版本的约 45K 行飙升至约 270K 行, 这样的一个代码量势必会有大量的漏洞[9]。下面分别对主要的 Hypervisor 重构、Hypervisor 完整性验证等方法进行分析。

重构 Hypervisor

针对此类攻击最直接的想法是将不必要的功能分割出去以缩减 Hypervisor 的代码规模, 通过减少可信基来降低脆弱性出现的概率。在这方面, NoHype[55]充分利用 CPU 和 I/O 设备等硬件的特性, 取代了部分由软件支持的 Hypervisor 部分功能, 但是这种设计方式丧失了虚拟化技术便利资源管理的特性; Nova[56]将 Hypervisor 重构为多个独立且运行于用户模式的 Hypervisor, 只有小部分特权指令运行于内核模式, 限制虚拟机只能攻击其所属的 Hypervisor。但是这些方法只支持静态缩减, 不能在虚拟机运行过程中动态缩减。为此, Min-V[57]构建了两类场景: 一类是具有完整 Hypervisor 功能的启动环境, 另一类是关闭部分虚拟功能的生产环境。为了进行动态缩减, 其首先在完整的 Hypervisor 启动环境上启动一个客户虚拟机, 生成这个虚拟机的快照后将其迁移到生成环境中, 实现了 Hypervisor 中无用代码的动态删除。与上述思路不同, Cloudvisor[58]对 Xen 的软件架构进行了重置, 其利用嵌套虚拟化技术将将 Xen 和 Dom0 置于非 root 模式, 将 Xen 及 Dom0 和具有更高权限的 Cloudvisor 划分在不同的特权空间, 这样, 当 Xen 和 Dom0 在执行特权操作时便会陷入 Cloudvisor 执行安全检查。H-SVM[59]和 HyperWall[60]将内存管理和安全保护进行了分离, Hypervisor 不能任意访问所有内存, 比如, 一旦内存页划分给客户虚拟机之后, 通过硬件机制确保在该虚拟机非授权的情形下不能被 Hypervisor 访问。然而, 一方面, 频繁的上下文切换会对性能产生一定程度的影响; 另一方面, 只解决了恶意 Hypervisor 这单个因素对 guest OS 的攻击; DeHype[61]和 HyperLock[62]以为每一个客户虚拟机分配一个 KVM 实例的方式实现隔离, 将客户虚拟机的影响范围仅限于其所属的 KVM 实例中, 但这种方法性能损耗大且占用的资源也较多; 与之类似, 文献[9]通过每个客户虚拟机都分配相应 Hypervisor 运行空间的方法, 限制恶意云租户无法通过 Hypervisor 的脆弱性攻击其他云租户虚拟机, 但其并没有考虑如何保障隔离后各 Hypervisor 子部分的安全性。文献[63]为敏感应用程序开辟出一块受保护内存, 并结合加密措施, 让应用程序的执行受限在该范围内; 当需要和外界程序交互时, 需进行边界安全检查等。

监控 Hypervisor 的完整性

在确保 Hypervisor 完整性方面, 文献[64, 65]基于 TPM 等技术可实现对 Hypervisor 启动时完整性的测量, 但是这类方法对运行时的完整性却无能为力。为此, 部分研究工作基于特定的硬件特性对运行时的 Hypervisor 完整性进行保护。在这方面, HyperGuard[66]和 HyperCheck[67]提供了对运行时 Hypervisor 完整性的测量方法, 但是, 这两者都依赖于 CPU 的 SMM 模式, 因此, 难以获得全部 Hypervisor 运行时的上下文信息; HyperSentry[68]利用系统管理模式(System Management Mode, SMM)来保护 Hypervisor 的控制流, 虽然能够获取完整性所需的全部信息, 但其除了需要 SMM 的支持外, 也需要 IPMI(Interface Platform Management Interface)的支持, 因此, 其对硬件的要求较高。

3.2 讨论和总结

表 2 总结了云计算可信性威胁与代表性思路及工作之间的对应关系。

表 2 问题与相应工作对应关系

Table 2 Relationship between problems and corresponding work

问题			代表性思路及工作
云计算可信性威胁	资源管理机制本身的问题	基于时间复用的资源共享问题	1. 改变 cache 的架构和使用方法: [20] 、[22] 、[23]、[24] 、[27] 、[28]
		基于空间共存的资源共享	1. 增加不可执行的限制: [33] 、[35]; 2. 边计算边防护: [36] 、[37]; 3. 引入随机机制: [38]、[40]
	恶意内部人员	管理软件集中且复杂	1. 改变资源管理机制: [42-45] 2. 加密机制: [46-49]
		违规操作	1. 创新管理制度: [50] 2. 威胁模式挖掘: [51-54]
	资源管理机制实现的脆弱性问题	Hypervisor 脆弱性	1. 减小可信基: [55-57]; 2. 更好的划分和隔离措施: [58-63]、[9]
		破坏 Hypervisor 的完整性	1. 启动时完整性检测: [64-65]; 2. 边计算边防护: [66-68]

由表 2 可以看出, 当前资源管理机制方法主要由四种:

(1) 改变当前的资源复用或管理机制, 弥补现有管理机制的不足, 从根本上去掉时间或空间上资源共享方式所能带来的威胁, 如重构 Hypervisor、改变当前 cache 的复用策略、将管理虚拟机拆分为多个具有最小权限的虚拟机、增加不可执行的限制等。

(2) 打破原有的固定格局, 通过引入随机机制, 利用多样性改变系统相似性、单一性, 利用动态性、随机性改变系统静态性、确定性[69], 最终实现攻击难度的提升并以此确保系统自身安全, 如文献[38]通过引入进程地址空间的随机布局机制极大地提升了攻击难度。

(3) 引入“边计算边防护”的运行模式, 实现系统的自我监测与防护, 如在应对资源管理机制实现的脆弱性问题时引入 Hypervisor 完整性监控机制, 比如文献[36]引入金丝雀等检测方法在运行时实现运行时溢出攻击监测, 文献[33]引入不可执行机制, 在运行时阻止攻击程序获得可执行权限。

(4) 创新人员管理机制, 重视管理和政策法规在信息安全中的地位和作用, 限制内部人员工作方式和途径, 构建新型人力资源共享机制。

4 展望

云计算环境中资源共享的范围更广、力度更大, 故处理好“共享与隔离”这对矛盾, 确保各虚拟机或进程之间的良好隔离是提升云可信性的途径之一。与此同时, 在确保安全的前提下, 还需要兼顾“安全与性能”这一矛盾[70], 从隔离性最高但慢到不实用的模拟执行, 到现代全虚拟化技术所采用的动态二进制翻译与硬件辅助虚拟化, 再到修改虚拟机系统的半虚拟化, 最后到共享内核、基于容器的操作系统级虚拟化, 无一不是在安全隔离和性能的天平上寻找平衡点。因此, 统筹处理、综合考虑“共享与隔离”和“安全与性能”这两对矛盾是提升云计算可信性的关键所在。

结构是功能的物质基础[71]; 同时, 结构决定功能、性能、效能和安全性, 即一个确定功能可以有多种实现结构, 不同的实现结构具有不同的使用性能、效能和安全性[72]。然而, 当前云计算的基础结构通过层层迭代与合成构建在虚拟机上, 并在虚拟机上实现功能和性能, 以此来对应多样化、规模化、无确定模型表征的云计算应用需求, 难以达到高效能的需求; 另一方面, 现有的计算机体系结构出于降低成本等方面的考虑, 去掉了成熟的安全机制, 如存储器的隔离保护机制、程序安全保护机制等, 导致程序可以被随意修改, 系统区域的数据可以随意修改[73]。为此, 本文认为应在深刻把握云中多样的资源共享形态及其在隔离前后不同场景及阶段矛盾特殊性的基础上, “必须借助计算体系结构和计算模式的技术创新去实现相对安全”[74], 设计相应的隔离方法及隔离后各子部分间新型矛盾的应对策略, 以此构建面向可信云计算的新型虚拟化资源安全管理机制。在这方面, 本文抛砖引玉, 认为应首先关注如下问题:

(1) 研究 Hypervisor 安全计算体系结构

云租户之间的攻击一般是利用 Hypervisor 的脆弱性成功实施的, 而大体量的 Hypervisor 共享同一

特权空间(如64位的X86, 共享根模式下的Ring-0特权空间)是其脆弱性较多的根本原因。文献[9]发现, 75.39%的攻击直接与 Hypervisor 相关, 其余 24.61%的攻击多数与 Dom0 中的 QEMU 和工具栈有关。

因此, 应在分析 Hypervisor 内部各模块权限大小、交互关系和现有硬件特性等多方面研究对Hypervisor划分策略和实现方法的影响, 同时注重研究 Hypervisor 安全计算体系结构及相应的安全保障机制, 包括隔离策略和机制、各子部分的交互和通信方式、隔离后的安全保障机制等。

(2) 针对基于痕迹驱动的侧信道攻击设计按需可配置的 Cache 缓存隔离模型和安全复用机制

和时间驱动的攻击相比, 痕迹驱动的攻击具有获取数据精准、噪音数据小、推理简单等优点, 因此, 其具有更大的威胁。文献[75]指出, 破解一个128AES的密码, 痕迹驱动只需要约 500 000 次数据采集, 而时间驱动的攻击却需要 2pow(27.5)次数据采集。造成这种攻击的根本原因之一是因为攻击者可以通过直接接触、分析受害者所使用 Cache 的规律, 推断出受害者相应内存访问规律, 最终将蛮力攻击的搜索空间缩小、实现快速识别。

如果将含有关键信息的内存所对应的 Cache 数据隔离开, 那么切断了攻击者直接接触受害者 Cache 数据的可能, 也致使其无法进行后续推理。基于这种思考, 在现有硬件结构的基础上, 研究融合时间复用和空间共存的新型 Cache 缓存隔离模型和安全使用机制应对基于痕迹驱动的侧信道攻击, 包括 Cache 和内存的映射关系、隔离 Cache 的复用策略及 Cache 缓存一致性问题等。

(3) 构建管理虚拟机组计算模式

在很多云计算虚拟化资源管理系统中都有一个系统超级用户或系统管理员, 拥有对系统全部资源的存取和分配权, 所以它的安全至关重要。如果不加以限制, 有可能由于超级用户的恶意行为、口令泄密、偶然破坏等对系统造成不可估量的损失和破坏, 因此, 有必要对系统超级用户的权限加以分割和限制, 实现权限最小化原则, 并形成权限之间的制约机制。为此, 构建管理虚拟机中各功能之间的交互和依赖关系模型, 并基于此研究管理虚拟机组计算模式及其相应的安全保障机制以应对恶意内部问题, 包括各子部分的交互方式、权限策略和机制、隔离后的安全保障机制等, 最终从权限分割及最小化、互相制约、追溯问责等角度限制内部的恶意人员。

(4) 基于虚拟机自省的可信数据获取技术

审计、取证及问责是提升云计算可行的关键一步。当面临审计、取证、问责时, 云提供商为了逃避责任, 可能故意删除、修改日志, 审计的日志并不是原始记录信息, 从而导致可信证据缺失, 然而, 出现故障或发生安全事件时, 若没有足够的可信证据, 则无法据此进行审计和问责。由此可见, 需设计云计算可信性数据采集、存储等机制以支撑审计、取证及问责, 包括设计云平台可信测评协议, 验证虚拟机从创建到撤销整个生命期中的可信性, 对云平台上收集的各种证据进行可信度量模型。

(5) 构建虚拟化资源安全管理架构的可信评价与验证模型

研究虚拟化资源安全管理架构的可信评价与验证模型, 包括计算效能影响评价模型及可信环境安全性验证方法, 实现安全计算体系结构或计算模式的性能和安全评价及验证、保障虚拟化资源安全管理架构在运行过程中的安全可信状态, 为设计有效的云计算虚拟化资源安全计算体系结构和计算模式提供理论和实践支持。

5 总结

针对现有云计算虚拟化资源管理机制对云可信性的影响, 本文首先分析了当前云计算资源管理机制及其安全隐患; 然后, 从资源管理机制本身的安全问题、资源安全机制的脆弱性问题以及恶意内部人员问题三个方面分析了国内外的研究现状; 紧接着, 分析认为, 计算体系结构和计算模式的创新是解决“共享与隔离”和“安全与性能”这两对矛盾的关键, 并提出了在这方面应优先着重解决的五个问题及若干思考; 最后, 总结全文。

致谢 感谢中国信息测评中心系统评估处的各位同事以及中国科学院信息工程研究所信息内容安全国家工程实验室相关老师和同学提出的有益建议。

感谢审稿专家和编辑部老师对本文提出的有益建议及指导。

参考文献

[1] Fang Binxing, A hierarchy model on the research fields of cyberspace security technology. *Chinese journal of Network and Information Security*. vol 1. no.1, pp. 2-7 (in Chinese), 2015.
(方滨兴, 从层次角度看网络空间安全技术的覆盖领域. *网络与信息安全学报*, 2015, 1(1):2-7.)

[2] Wang Guofeng, Liu Chuanyi, Pan Hezhong, et al. Survey on Insider Threats to Cloud Computing. *Chinese Journal of Computer.* Vol. 40. no.2, pp.296-316 (in Chinese), 2017.
(王国峰, 刘川意, 潘鹤中, 等. 云计算模式内部威胁综述. *计算机*

学报, 2017, 40(2): 296-316.)

[3] 不管是好消息还是坏消息只要泄密 Google 都会开除你. http://jandan.net/2010/11/15/raise-leaker.html, 2010.

[4] Ristenpart T, Tromer E, Shacham H, et al, Hey, you, "get off of my cloud: exploring information leakage in third-party compute clouds", In Proc. *ACM Conference on Computer and Communications Security(CCS'09)*, pp.199-212, 2009.

[5] "毒液"漏洞(CVE-2015-3456)影响全球数百万虚拟机安全. http://www.freebuf.com/news/67325.html, 2015.

[6] Li Baohui, Xu Kefu, Zhang Peng, Guo Li, Hu Yue, Fang Binxing, Research and application progress of virtual machine introspection technology. *Ruan Jian Xue Bao/Journal of Software,* 2016, 27(6): 1384−1401 (in Chinese). http://www.jos.org.cn/ 1000-9825/5006.htm (李保珲, 徐克付, 张鹏, 郭莉, 胡玥, 方滨兴, 虚拟机自省技术研究与应用进展, *软件学报*, 2016,27(6):1384−1401.)

[7] C. S. Alliance, "Top threats to cloud computing, version 1.0," Cloud Security Alliance, Tech. Rep., March 2010. [Online]. Available: http://www.cloudsecurityalliance.org/ topthreats/csathreats.v1.0.pdf, 2010.

[8] "Star-205 cloud security alliance - top threats to cloud computing v2.0.pdf," in RSA Conference Europe, 2010. [Online]. Available: http://365.rsaconference.com/docs/ DOC-2819, 2010.

[9] Lei Shi, Yuming Wu, Yubin Xia, Nathan Dautenhahn, Haibo Chen, Binyu Zang, Haibing Guan, Jingming Li, "Deconstructing Xen". *The Network and Distributed System Security Symposium 2017 (NDSS'17)*, pp.1-15, 2017.

[10] Colp P, Nanavati M, Zhu J, et al, "Breaking up is hard to do: security and functionality in a commodity hypervisor", *In Proc. ACM Symposium on Operating Systems Principles(SOSP'2011)*, pp.189-202, 2011.

[11] SUZAKI, K., IIJIMA, K., YAGI, T., AND ARTHO, C. "Memory deduplication as a threat to the guest OS". *In Proceedings of the Fourth European Workshop on System Security (EUROSEC '11)*, pp.1–6, 2011.

[12] Pereira T E, Brasileiro F, Sampaio L, "File system trace replay methods through the lens of metrology". In Proc. *IEEE MASS Storage Systems and Technologies,* pp.1-16, 2017.

[13] Miło´s, G., Murray, D., Hand, S., and Fetterman, M.A.,Satori: "Enlightened page sharing", *In Proc. USENIX Annual Tech*, pp. 133-156, 2009.

[14] Gupta, D., Lee, S., Vrable, M., Savage, S., Snoeren, A.C., Varghese, G., Voelker, G.M., and Vahdat, A., "Difference Engine: Harnessing Memory Redundancy in Virtual Machines", *In Proc. Operating Systems Design and Implementation (OSDI'2008)*, pp.309–322, 2008.

[15] Zhou Z, Reiter M K, Zhang Y. "A Software Approach to Defeating Side Channels in Last-Level Caches", *In Proc. ACM Sigsac Conference*, pp.871-882, 2016.

[16] C. Percival. "Cache missing for fun and profit", *In Proc. BSDCan 2005*, pp. 123-144, 2005.

[17] D. A. Osvik, A. Shamir, and E. Tromer, "Cache attacks and countermeasures: the case of AES", *In Proc. RSA Conference Cryptographers Track (CT-RSA2006)*, pp.1-25, 2006.

[18] Younis Y A, Kifayat K, Merabti M. "Cache Side-Channel Attacks in Cloud Computing", *In Proc. International Conference on Cloud Security Management*, pp.1-23, 2014.

[19] Zhang Y, Juels A, Oprea A, et al. "HomeAlone: Co-residency Detection in the Cloud via Side-Channel Analysis", *In Proc. IEEE Symposium on Security and Privacy*, pp.313-328, 2011.

[20] Z. Wang and R. B. Lee, "New cache designs for thwarting software cache-based side channel attacks," *in ACM/IEEE International Symposium on Computer Architecture (ISCA'2007)*, pp.1-13, 2007.

[21] L. Domnister, A. Jaleel, J. Loew, N. Abu-Ghazaleh, and D. Ponomarev, "Non-monopolizable caches: Lowcomplexity mitigation of cache side channel attacks," *Trans. Arch. & Code Optimization (TACO),* vol. 8, no. 4, 2012.

[22] R. K¨onighofer, "A fast and cache-timing resistant implementation of AES. In Topics in Cryptology", *In Proc. The Cryptographers' Track at the RSA Conference 2008*, pp.187–202, 2008.

[23] E. K¨asper and P. Schwabe,. "Faster and timing-attack resistant AES-CGM", *In Cryptographic Hardware and Embedded Systems — CHES 2009*, pp.1–17, 2009.

[24] Z. Wang and R. B. Lee, "New cache designs for thwarting software cache-based side channel attacks," *in ACM/IEEE International Symposium on Computer Architecture (ISCA)*, pp.1-14, 2007.

[25] M. Godfrey, "On the prevention of cache-based sidechannel attacks in a cloud environment," Master's thesis, Queen's University, Ont, CA, 2013.

[26] T. Kim, M. Peinado, and G. Mainar-Ruiz, "STEALTHMEM: system-level protection against cache-based side channel attacks in the cloud," *in USENIX Security Symposium*, pp.189–204, 2012.

[27] RAJ, H., NATHUJI, R., SINGH, A., AND ENGLAND, P. "Resource management for isolation enhanced cloud services". I*n Proc. of the 2009 ACM Cloud Computing Security Workshop*, pp.77–84, 2009.

[28] Kalamatianos J. Dynamic remapping of cache lines[J]. 2016.

[29] A. Aviram, S. Hu, B. Ford, and R. Gummadi, "Determinating timing channels in compute clouds", *In ACM Cloud Computing Security Workshop,* pp.103–108, 2010.

[30] Yu S, Gui X, Tian F, et al, "A Security-Awareness Virtual Machine Placement Scheme in the Cloud", *In Proc. IEEE International Conference on High PERFORMANCE Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing*, pp.1078-1083, 2014.

[31] Veen V V D, Dutt-Sharma N, Cavallaro L, et al, "Memory Errors: The Past, the Present, and the Future", *In Proc. International Workshop on Recent Advances in Intrusion Detection,* pp. 86-106, 2012.

[32] Elhage, N., Virtunoid: Breaking out of KVM, Elhage. Black Hat: USA, 2011. http://media.blackhat.com/bh-us-11/Elhage/BH_US_11_Elhage_ Virtunoid_Slides.pdf, 2011.

[33] Designer, S.: Linux kernel patch to remove stack exec permission, 1997.

[34] Designer, S.: Getting around non-executable stack (and fix) , 1997.

[35] The Pax Team: Design & Implementation of PAGEEXEC (2000), 2000.

[36] Cowan, C., Pu, C., Maier, D., Hintongif, H., Walpole, J., Bakke, P., Beattie, S., Grier, A., Wagle, P., Zhang, Q.: "StackGuard: Automatic Adaptive Detection and Prevention of Buffer-Overflow Attacks", *In: Proceedings of the 7th USENIX Security Symposium,* pp.1-23,1998.

[37] Cowan, C., Barringer, M., Beattie, S., Kroah-Hartman, G.: "FormatGuard: Automatic Protection From printf Format String Vulnerabili-

ties", *In Proc. USENIX Security Symposium,* pp.1-23, 2001.

[38] The Pax Team: Design & Implementation of PAGEEXEC, 2000.

[39] Blazakis, D.: "Interpreter Exploitation", In Proc. of the 4th USENIX Conference on Offensive Technologies, pp.1-21, 2010.

[40] Wei, T., Wang, T., Duan, L., Luo, J.: "Secure dynamic code generation against spraying", *In Proc. ACM CCS 2010*, pp. 212-232, 2010.

[41] Nguyen M D, Chau N T, Jung S, et al, "A demonstration of malicious insider attacks inside cloud IaaS vendor", *In Proc. International Journal of Information and Education Technology,* pp.483-499, 2014.

[42] Derek Gordon Murray, Grzegorz Milos, and Steven Hand, "Improving xen security through disaggregation", *In Proc. of the fourth ACM SIGPLAN/SIGOPS international conference on Virtual execution environments,* pp.151–160, 2008.

[43] Patrick Colp, Mihir Nanavati, Jun Zhu, William Aiello, George Coker, Tim Deegan, Peter Loscocco, and Andrew Warfield, "Breaking up is hard to do: security and functionality in a commodity hypervisor", *In Proc of the Twenty-Third ACM Symposium on Operating Systems Principles,* pp.189–202, 2011.

[44] Shakeel Butt, H Andrés Lagar-Cavilla, Abhinav Srivastava, and Vinod Ganapathy, "Self-service cloud computing", *In Proc. of the 2012 ACM conference on Computer and communications security,* pp.253–264, 2012.

[45] Manco F, Lupu C, Schmidt F, et al, "My VM is Lighter (and Safer) than your Container", *In Proc. of the 26th Symposium on Operating Systems Principles. ACM,* pp.218-233, 2017.

[46] Popa R A, Redfield C M S, Zeldovich N, et al, "CryptDB: Protecting confidentiality with encrypted query processing", *In Proc. ACM Symposium on Operating Systems Principles (SOSP 2011)*, pp.85-100, 2011.

[47] Popa, Raluca Ada, Stark, Emily, Helfer, Jonas, et al, "Building web applications on top of encrypted data using Mylar", *In Proc. of USENIX & SAGE,* pp.22-27, 2014.

[48] He W, Akhawe D, Jain S, et al, "ShadowCrypt: Encrypted Web Applications for Everyone", *In Proc. ACM Sigsac Conference on Computer and Communications Security*, pp.1028-1039, 2014.

[49] Vimercati S D C D, Foresti S, Jajodia S, et al, "Over-encryption: management of access control evolution on outsourced data", *In Proc. International Conference on Very Large Data Bases. VLDB Endowment,* pp.123-134, 2007.

[50] Li Huihui, A cloud security solution based on a cloud management system ROS, *Journal of Taiyuan Normal University (Natural Science Edition),* vol.13, no.4, pp. 51-54 (in Chinese), 2014.
(李慧慧, 基于一种云管理系统 ROS 的云安全解决方案, *太原师范学院学报(自然科学版)*, 2014(4):51-54.)

[51] Greitzer F L, Kangas L J, Noonan C F, et al, "Identifying At-Risk Employees: Modeling Psychosocial Precursors of Potential Insider Threats", *In Proc. Hawaii International Conference on System Sciences. IEEE Computer Society*, pp.2392-2401, 2012.

[52] Hashem Y, Takabi H, GhasemiGol M, et al, Inside the Mind of the Insider: Towards Insider Threat Detection Using Psychophysiological Signals. J. Internet Serv. Inf. Secur., 2016, 6(1): 20-36.

[53] Eberz S, Rasmussen K B, Lenders V, et al. Looks like eve: Exposing insider threats using eye movement biometrics. *ACM Transactions on Privacy and Security*, 2016, 19(1): 1.

[54] Kent A D, Liebrock L M, Neil J C. Authentication graphs: Analyzing user behavior within an enterprise network, Computers & Security, 2015, 48: 150-166.

[55] E. Keller, J. Szefer, J. Rexford, and R. Lee, "NoHype: virtualized cloud infrastructure without the virtualization", *In Proc. ISCA,* pp.350–361, 2010.

[56] U. Steinberg and B. Kauer. "NOVA: A microhypervisor-based secure virtualization architecture", *In Proc. Eurosys*, pp.209–222, 2010.

[57] Anh Nguyen, Himanshu Raj, Shravan Rayanchu, Stefan Saroiu, and Alec Wolman, "Delusional boot: securing hypervisors without massive re-engineering", *In Proceedings of the 7th ACM european conference on Computer Systems,* pp.141–154, 2012.

[58] Fengzhe Zhang, Jin Chen, Haibo Chen, and Binyu Zang, "CloudVisor : Retrofitting Protection of Virtual Machines in Multi-tenant Cloud with Nested Virtualization", *In Proc. SOSP 2011,* pp.203–216, 2011.

[59] S. Jin, J. Ahn, S. Cha, and J. Huh, "Architectural Support for Secure Virtualization under a Vulnerable Hypervisor," *in Proc. MICRO,* pp. 201-213, 2011.

[60] J. Szefer and R. Lee, "Architectural support for hypervisor-secure virtualization," *in Proc. ASPLOS,* pp.144-159, 2012.

[61] Chiachih Wu, Zhi Wang, and Xuxian Jiang. "Taming hosted hypervisors with (mostly) deprivileged execution", I*n Proc. NDSS 2013,* pp.23-45, 2013.

[62] Zhi Wang, Chiachih Wu, Michael Grace, and Xuxian Jiang, "Isolating commodity hosted hypervisors with hyperlock", *In Proceedings of the 7th ACM european conference on Computer Systems*, pp.127–140, 2012.

[63] Kuvaiskii D, Oleksenko O, Arnautov S, et al, "SGXBOUNDS: Memory Safety for Shielded Execution", *In Proc. Twelfth European Conference on Computer Systems,* pp.205-221, 2017.

[64] Trusted Computing Group: Trusted Platform Module. http://www.trustedcomputinggroup.org/developers/trusted platform module, 2017.

[65] Intel Trusted Execution Technology. http://www.intel.com/ technology/security/, 2017.

[66] R. Wojtczuk and J. Rutkowska. Xen 0wning trilogy. In Black Hat conference, 2008.

[67] J. Wang, A. Stavrou, and A. K. Ghosh, "HyperCheck: A hardware-assisted integrity monitor", *In Proceedings of the 13th International Symposium on Recent Advances in Intrusion Detection (RAID'10),* pp.22-45, 2010.

[68] Azab A M, Ning P, Wang Z, et al, "HyperSentry: enabling stealthy in-context measurement of hypervisor integrity", *In Proc. ACM Conference on Computer and Communications Security,* pp.38-49, 2010.

[69] Wu Jiangxing, Meaning and Vision of Minmic Computing and Mimic Security Defense. *Telecommunications Science, vol.7,* pp. 1-7(in Chinese), 2014.
(邬江兴, 拟态计算与拟态安全防御原理的愿意和愿景, *电信科学*, 2014, 30(7): 1-7.)

[70] 黄铁军, 强 AI 的"仿真主义"和神经计算机的"五原则", https://baijiahao.baidu.com/po/feed/share?wfr=spider&for=pc&context=%7B%22sourceFrom%22%3A%22bjh%22%2C%22nid%22%3A%22news_3295557000172563055%22%7D, 2017.

[71] Ben-Yehuda M, Xenidis J, Ostrowski M, et al. The price of safety: Evaluating IOMMU performance[C]//The Ottawa Linux Symposium. 2007: 9-20.

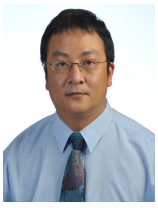
[72] 邬江兴, 基于认知的主动重构云计算环境. http://wireless.it168.com/a2011/0520/1193/000001193203.shtml, 2011.

[73] Zhang Huanguo, Hom Wenbao, Lai Xuejia. A review of network space security. *SCIENTIA SINICA Informations*. vol. 46, no.2, pp.125-164 (in Chinese) 2016.
(张焕国, 韩文报, 来学嘉,等, 网络空间安全综述. *中国科学:信息科学*, 2016, 46(2): 125-164.)

[74] 沈昌祥, 可信计算构筑网络安全, 抢占网安核心制高点, http://www.21ic.com/news/rf/201608/684179.htm, 2016.

[75] OSVIK, D. A., SHAMIR, A., AND TROMER, E, "Cache attacks and countermeasures: the case of AES", *In Topics in Cryptographers Track at the RSA Conference 2006*, pp.1–20, 2006.

李保珲 于 2016 年在北京邮电大学计算机科学与技术专业获得工学博士学位。现任中国信息安全测评中心助理研究员。研究领域为云计算及其安全、网络信息安全。研究兴趣包括: 云安全、信息系统风险评估技术等。Email: delibh@126.com



李斌 工学博士, 现任中国信息安全测评中心研究员。研究领域为风险评估技术。研究兴趣包括: 云计算及其安全评估技术。Email: lib@ itsec.gov.cn

任望 于 2014 年在四川大学软件工程专业获得硕士学位。现任中国信息安全测评中心助理研究员。研究领域为信息系统风险评估技术。研究兴趣包括: 网络信息安全。Email: renw@itsec.gov.cn

杨光 于 2009 年在哈尔滨工程大学计算机应用专业获得工学博士学位。现任中国信息安全测评中心副研究员。研究领域为风险评估理论和技术。研究兴趣包括: 网络空间安全。Email: yangguang@itsec.gov.cn

王永涛 于 2011 年在上海交通大学计算机系统结构专业获得工学博士学位。现任中国信息安全测评中心副研究员。研究领域为信息安全、风险评估技术。研究兴趣包括: 网络空间安全、密码学。Email: wangyt@itsec. gov.cn

杜宇鸽 于四川大学软件工程专业获得硕士学位。现任中国信息安全测评中心助理研究员。研究领域为信息系统风险评估技术。研究兴趣包括: 网络信息安全。Email: duyg@itsec. gov.cn

张鹏 于 2014 年在中科院计算所计算机应用专业获得工学博士学位。现任中科院信息工程研究所副研究员。研究领域为网络数据流处理。研究兴趣包括: 大数据安全、云计算安全。Email: pengzhang@iie.ac.cn