doi:10.3969/j.issn.1000-436x.2013.05.023

ICFF：一种 IaaS 模式下的云取证框架

谢亚龙[1,2]，丁丽萍[1]，林渝淇[1,2]，赵晓柯[1,2]
(1. 中国科学院 软件研究所 基础软件国家工程研究中心，北京 100190；2. 中国科学院 研究生院，北京 100190)

摘 要：分析了云取证技术所面临的挑战，提出了一种基础设施即服务(IaaS)云模型下的取证框架 ICFF，并在开源 IaaS 云平台 Eucalyptus 中进行了实现，最后通过实验的方法对 ICFF 进行了验证分析。实验结果表明，该框架能够有效并快速地获取云平台中的证据数据。

关键词：云计算；数字取证；云取证；IaaS 模式
中图分类号：TP391 **文献标识码**：A **文章编号**：1000-436X(2013)05-0200-07

ICFF: a cloud forensics framework under the IaaS model

XIE Ya-long[1,2], DING Li-ping[1], LIN Yu-qi[1,2], ZHAO Xiao-ke[1,2]
(1. National Engineering Research Center of Fundamental Software, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China;
2. Graduate University of Chinese Academy of Sciences, Beijing 100190, China)

Abstract: Technical challenges of cloud forensics was summarized and a forensics framework under an infrastructure as a service(IaaS) cloud model called ICFF was proposed. Then, this framework on the open source IaaS cloud platform Eucalyptus was implemented. Finally, a test case to demonstrate the effectiveness of ICFF was designed. Experiments show that the framework can obtain evidence data in cloud platform effectively and efficiently.

Key words: cloud computing; digital forensics; cloud forensics; IaaS model

1 引言

随着云计算技术的迅猛发展，云环境下的取证技术成为了当前的一个研究热点。与传统的数字取证技术不同，云取证所面对的系统结构更加复杂、数据规模更加巨大。就拿一个小规模的云来举例，假设该系统中有 2 000 个节点，每个节点磁盘空间为 320 GB，RAM 为 2 GB，则总的磁盘取证空间是 640 TB，RAM 取证空间为 4 TB，即使不考虑节点间操作系统及文件系统的不同，如此庞大的数据量是传统取证技术无法胜任的。因此，如何从云中获取完整可靠的证据数据是当前云取证研究的难点问题[1]。具体而言，云取证主要面临以下 4 大技术难题。

1) 云中数据物理存放地点范围太大。云数据中心由成千上万台拥有大容量存储设备的 PC 组成，云系统屏蔽了下层数据存储的实现细节，只对用户提供一个逻辑上单一的数据存放地址，因此，要想获得云中数据的物理存放地址并非易事[2]。

2) 云应用产生的逻辑上相关的数据可能被分散存放。云应用所产生的数据被统一存储在逻辑上连续的地址空间中，而这些数据的物理存放地址可能并不连续，甚至可能被分散在好几个不同的存储设备中。

3) 待取证数据规模大，而真正与犯罪相关的信息很少。云中存储着海量数据，从如此大规模的数据中提取证据信息有如大海捞针。就以 IaaS 模型举例，一个虚拟机镜像小则几 GB，大则几十至几百 GB，而存储在这些镜像中的关键证据信息可能就只

收稿日期：2012-08-17；**修回日期**：2012-12-25
基金项目：国家科技重大专项基金资助项目（2010ZX01036-001-002, 2010ZX01037-001-002）；中国科学院知识创新工程基金资助项目（KGCX2-YW-125, KGCX2-YW-174）；国家自然科学基金资助项目（61170072）
Foundation Items: The National Science and Technology Major Project (2010ZX01036-001-002, 2010ZX01037-001-002); The Knowledge Innovation Key Directional Program of Chinese Academy of Sciences (KGCX2-YW-125, KGCX2-YW-174); The National Natural Science Foundation of China (61170072)

有几 KB。

4) 云的弹性扩展机制要求取证能及时适应系统规模的变化，即实现弹性取证[3]。弹性扩展是云系统的关键特征，它能在云应用运行期间实现支撑云应用的虚拟机实例个数的动态增加或者减少。当虚拟机实例个数减少时，若不能及时提取出残留在该虚拟机实例中的证据信息，则这些证据信息很可能会丢失，且难以恢复[4]。

与传统的数字取证技术相比，云取证技术面临的挑战主要包括 2 个方面。首先，没有成熟的云取证工具供调查人员使用，云取证活动主要依靠调查人员掌握的传统取证技术及相关经验，若调查人员操作不当很可能会破坏原始证据信息，从而导致取证失败。其次，证据信息的获取难度及方法会随着云服务模型及部署模型的不同而不同[5]。例如，相对于公有云而言，私有云架构更加清晰、云数据的存放节点地点固定，因此私有云模式下的取证难度远小于公有云；相对于软件即服务 (SaaS, software as a service)模型而言，IaaS 模型能提供更多底层的数据供调查人员取证分析，因此 IaaS 模型下的取证难度会小于 SaaS 模型。

目前，国内外学者对于云取证的研究主要包括 3 个方面：1) 研发应用于用户端的证据提取工具，该工具主要用于获取用户使用云应用时所产生的证据数据；2) 探索特定的云平台在遭受不同攻击时的取证方法；3) 将云系统中的虚拟机实例作为主要取证分析对象，解决虚拟机实例迁移过程中的数据保全问题。上述研究中，第 1 种方法所获取的证据数据非常有限，得进一步获取云服务端的证据数据；第 2 种方法通用性较差；第 3 种方法在虚拟机实例被恶意破坏时就会失效。

本文给出了一个 IaaS 云模型下的取证框架 ICFF，并在开源 IaaS 云平台 Eucalyptus[6,7]中进行了实现。通过实验研究证明了该框架能够有效解决云取证中的 4 大技术难题。

2　相关工作分析

云计算的推出带来了云安全问题，云安全成了云计算发展的瓶颈。因此，作为与云安全相对应的云取证也成了关注的焦点。在 CSA 发布的《云计算关键领域安全指南 V3.0》上把云取证作为一项关系云计算发展的重大问题所提出。Keyun 等人对 150 多位数字取证专家进行了采访，列举出了他们对云取证领域的一些关键问题的看法，如云取证技术面临着哪些挑战、带来了哪些机遇，有哪些有价值的研究方向等[8]。Birk 等人从技术的角度剖析了云取证所面临的技术难题[5]，而 Reilly 等人则从法律的角度分析了云取证技术所面临的法律障碍[9]。

OWADE[10]宣称是第一个云取证工具，由斯坦福大学的 Elie Bursztein 教授于 2011 年在黑帽(black hat)大会上提出。Elie Bursztein 认为云服务的接入端应当包含有大量证据信息，因此他设计的 OWADE 工具主要用于用户端的证据提取。该工具目前仅支持 Windows 系统，能对主流的浏览器如 Chrome、Internet Explorer、Firefox 及 Safari 进行证据提取和分析，并能获得当前主流即时通信软件如 Skype、Google Gtalk 及 MSN 的本地聊天记录。虽然该工具在用户端能提取到用户使用诸如 Skype 等云服务的证据信息，但是这些信息非常有限且很容易遭到犯罪分子的恶意破坏，因此还需与 CSP 端提取到的证据一起组成完整的证据链。

Zafarullah 等人针对开源云平台 Eucalyptus 环境下的取证技术进行了研究[11]。首先，他们在云系统内部部署了 2 台攻击源主机；然后，利用上述主机对云控制器 (CLC, cloud controller) 节点发起 DoS/DDoS 攻击，耗尽 CLC 节点的 CPU、内存及网络带宽等资源，致使 Eucalyptus 无法开启新的虚拟机实例，无法及时对终端用户的请求进行响应；最后，从 Syslog、Snort 等日志记录中查找本次攻击的来源。Zafarullah 总的研究思路就是先对 Eucalyptus 进行攻击，然后针对该类型的攻击寻找相应的取证方法。虽然作者提出的方法对特定类型攻击的取证调查有较强的借鉴意义，但还存在 2 点不足：首先，网络攻击的方式多种多样且不断变化，对每一种攻击都提出一种取证方案的可行性不大且没有必要；其次，对于那些符合云系统安全规则所产生的数字证据，作者并没有提出有效的提取方法。

华中科技大学的周刚博士提出了一种以现场迁移技术为基础的云取证方法，该方法将虚拟机实例视为取证分析对象[12]。当有取证需求时，将待取证虚拟机实例迁移至本地，在迁移过程中，对虚拟机实例的内存映射、网络连接等易失性数据进行了保全，然后将该虚拟机实例在本地进行加载，最后利用一些传统的取证工具在虚拟机实例中进行取证。该方法虽然能有效地从正常运行的虚拟机实例中获得证据数据，但当虚拟机实例被用户恶意破坏

无法加载时，该方法就会失效。

本文对云计算及其安全和取证进行了全面的分析研究。首先，对虚拟机技术进行了研究，剖析了其实现已有的安全模型、面临的安全威胁和安全防护及取证等机制；其次，对针对 Xen 虚拟机的入侵技术进行了研究，特别是 Blue Pill 等较为致命的入侵技术及其防范技术进行了深入研究，提出了采用 DMA 技术进行防范的方法，实现了对 Xen 的超级调用的审计；第三，研究了基于虚拟机的隐蔽信道通信机制并提出了隐蔽信道的标识方法[13,14]。

因此，基于对虚拟机的研究，本文提出的方法和国内外其他相关研究相比有以下优势。

1) 具有完整性保护能力。在取证框架中设计了一系列的安全保护机制，能够有效应对恶意用户或恶意软件对该框架发起的干扰，有效防止恶意用户篡改、删除证据数据，从而保证了证据的完整性。此外，当某个虚拟机(VM, virtual machine)被恶意破坏且无法修复时，能保证对该 VM 的取证不受影响。

2) 具备弹性扩展能力。实现了取证力度伴随 VM 的规模变化而变化。将“证据抓取器”与 VM 绑定在一起，当 VM 的数量增加或减少时，“证据抓取器”能及时地将 VM 中的证据数据转移到专用的取证虚拟机中。

3) 取证效率高。训练出的“证据抓取器”实时抓取 VM 中的证据数据，避免了对 VM 中所有数据及云中所有 VM 的取证。能对“证据抓取器”进行静态、动态配置，限定其数据抓取范围，从而避免了对 VM 中的常规系统文件等无关数据的取证，提高了取证效率。

3 ICFF 的设计与实现

3.1 Xen 简介

虚拟化技术是云计算的关键技术。当前一些主流的云平台均基于 Xen 进行实现，如 Amazon EC2、Eucalyptus、Xen Cloud Platform 等。本文提出的取证框架 ICFF 也是基于 Xen 进行设计实现的。

Xen[15]是一个开源虚拟机监控器，其体系结构如图 1 所示。由图 1 可知，Xen 直接运行在物理硬件之上，并向上层操作系统提供虚拟化环境。每一个运行在 Xen 之上的虚拟机均被称为一个虚拟域，虚拟域又可分为特权域 Domain0 及非特权域 DomainU，特权域拥有硬件设备驱动并提供非特权域的管理接口。Xen 同时支持半虚拟化和全虚拟化，半虚拟化需要修改客户操作系统(guest OS)源码，而全虚拟化则不需修改系统源码，但需有诸如 Intel VT、AMD-V 等硬件虚拟化技术的 CPU 的支持。

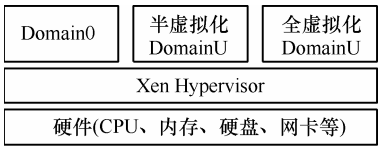

图 1 Xen 体系结构

3.2 整体架构

如图 2 所示，整个取证框架由 4 个部分组成：位于取证虚拟机(FVM, forensic virtual machine)中的证据保护及分析模块、证据提取模块，位于虚拟服务器中的证据抓取器(EC, evidence crawler)，位于 Xen Hypervisor 中的实时取证模块。其中，FVM 为云平台服务商所有，并不向公众提供服务，仅用来保存及分析收集到的证据数据并提供查询接口；虚拟服务器为云平台服务商所提供的租用对象，供个人或企业租用。

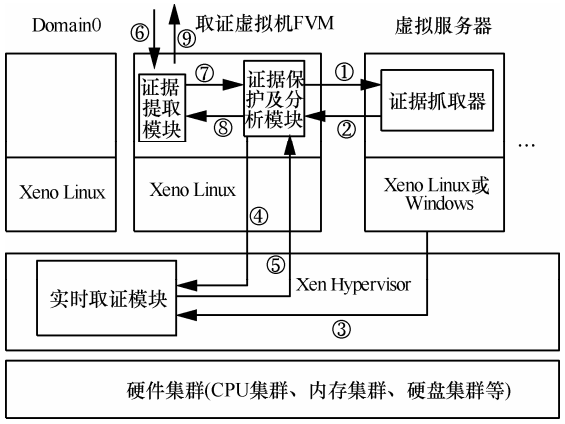

图 2 取证框架系统架构

1) 证据保护及分析模块：本模块向 EC 下达证据收集命令(图 2 中的①)，分析现有证据并根据分析结果通知实时取证模块获取与虚拟服务器相关的实时证据数据(图 2 中的④)。

2) 证据抓取器：收集虚拟服务器中的证据数据并发送给 FVM(图 2 中的②)。

3) 实时取证模块：截获虚拟服务器中的 VM Exit 指令，获取其系统中正在运行的进程信息(图 2 中的③)，将获取的上述数据发送给 FVM(图 2 中的⑤)。

4) 证据提取模块：供调查人员使用，是 FVM 对外的唯一接口。负责接收调查人员的证据提取命令(图 2 中的⑥)，将上述命令转换为统一规则的查询语句后发送给证据保护及分析模块(图 2 中的⑦)，等待分析模块回传符合需求的证据数据(图 2 中的⑧)，将证据数据的副本传输给调查人员(图 2 中的⑨)。

3.3 证据抓取器

EC 的任务是从虚拟服务器中识别并抓取证据数据，然后将证据数据传输至 FVM 中。EC 结构如图 3 所示。

1) 激活

在抓取证据之前需先将 EC 激活，激活方式如下。①定时激活，在某个时间点或每隔一个时间段将抓取器激活；②通过远程命令激活；③事件激活，当虚拟服务器系统出现特定事件(如有用户登录)时激活。在 VM 未受到恶意攻击之前，EC 所抓取的证据数据尚未被修改，能被法庭所认可。而当 VM 被黑客攻破后，黑客能任意篡改该 VM 中的数据，此时获取的证据数据已不再具有法律效力。因此，需在黑客尝试攻击 VM 阶段及时激活抓取器，完成证据转移工作。

黑客要想对某个 VM 发起攻击，其首先得发现该 VM 中系统的现有漏洞，然后针对该漏洞使用专门的攻击方法。在黑客进行漏洞扫描、远程入侵过程中均会引发一系列的系统事件，如特定网络端口扫描事件、应用程序缓冲区溢出错误事件、系统用户登录事件等。因此，对这些事件进行监听并及时激活 EC，能实现对黑客入侵全过程的取证。

2) 证据识别及抓取

证据数据的智能识别需结合人工智能技术来实现，本文根据日常取证经验，对 Windows 及 Linux 平台中证据数据的存在形式、存放路径进行了归纳，并将这些知识形成了统一的规则提供给 EC。

证据数据的抓取范围可以通过以下 2 种方式来设定：①通过配置文件指定证据数据的路径；②通过远程命令实时指定待抓取数据的类型、路径、特征等。第 1 种方式常用于提取常规证据数据，如操作系统日志、应用软件(如 Web 服务器、数据库)日志、非日志形式的敏感文件等；第 2 种方式则是第 1 种方式的一个补充，实时接收并完成来至其他模块的取证需求。

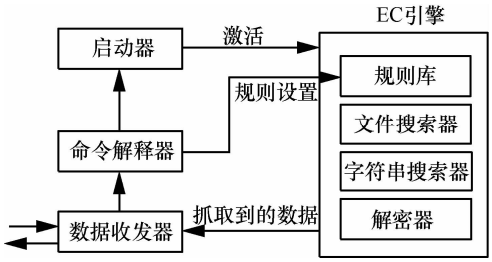

图 3　EC 结构

3.4 实时取证模块

实时取证模块随着 Xen 启动而启动，负责全程记录 VM 中进程的执行情况，所记录的信息包括时间戳、VM 的 ID 号、进程名、进程 ID 号。上述记录信息可以作为黑客入侵行为的有效证据，同时对黑客入侵方法及入侵过程的分析也极具参考意义。

对于那些使用 Linux 内核的 Guest OS，可以将进程监控模块嵌入到其内核源码中，然后注册一个事件通道号，用于 Guest OS 与 Xen 之间通信，最后通过共享内存的方式将获取的进程记录信息传递到 Xen 中。

若不想修改 Guest OS 内核源码，则需使用硬件虚拟化技术。实现原理如图 4 所示，Guest OS 中任一进程切换时，都需将新进程的页表基地址写入到特权寄存器 CR3，此时 CPU 会发生 VM Exit，陷入到 Xen 中。具体实现步骤如下。

1) 在 Xen 中捕获 CPU 嵌入指令 VM Exit。

2) 设置 Hook 函数获取 CR3 寄存器的值。

3) 从相应 VM 的堆栈中获取进程描述符(PD, process descriptor)。

4) 从 PD 中获取进程 ID 及进程名，将上述进程信息及 Xen 中统一时间戳写入记录表。

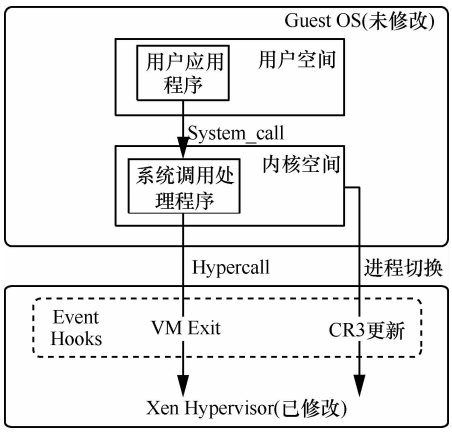

图 4　实时取证模块原理

3.5 证据保护及分析模块

该模块负责接收由 EC 传输过来的证据数据，然后对这些数据进行校验、保护及分析。

3.5.1 证据保护

云计算环境下的不安全因素较多，有来至云系统内部的安全威胁，也有来至云系统外部(如互联网)的安全威胁。为了提高该取证框架的抗干扰能力，保障证据数据的完整性及机密性，特设计了以下安全机制。

1) 证据数据的传输校验机制

设计传输校验机制主要有 3 个目的：①确保证据数据传输过程中不被篡改；②确保证据数据的内容不会被非法用户获取；③确保证据数据不能被伪造。

在本取证框架中，引入了数字证书机制，由云平台服务商给 FVM 及 EC 颁发证书。EC 抓取到证据数据后，首先利用 MD5 算法生成校验值，然后将上述证据数据及校验数据组装成证据数据分组，随后利用该 EC 的数字证书对数据分组进行签名并加密，最后通过网络将其发送至 FVM。

FVM 收到 EC 传输过来的数据分组时，首先将其解密，然后获取该数据分组的数字签名，若该签名与 VM 中 EC 的证书不符，则说明该数据分组是由 VM 中其他恶意程序所伪造，进而表明该 VM 已被恶意用户控制。数字签名验证通过后，再对证据数据进行校验，若校验不通过，则说明证据数据分组传输过程中存在分组丢失现象，需将该证据数据分组丢弃并要求 EC 重传。整个流程如图 5 所示。

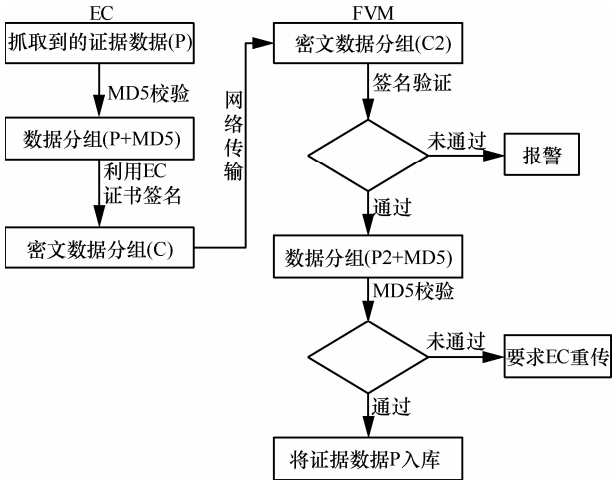

图 5 证据数据传输校验流程

2) 对 EC 的保护机制

EC 的角色至关重要，只有它正确运行才能保证 FVM 能源源不断地获得证据数据。因此，需设计一种检测机制，用于判定 EC 是否被其他恶意程序所干扰。干扰方式有 2 种：一是阻止 EC 运行；二是阻碍 EC 抓取及传输证据数据。相应的检测机制如图 6 所示。

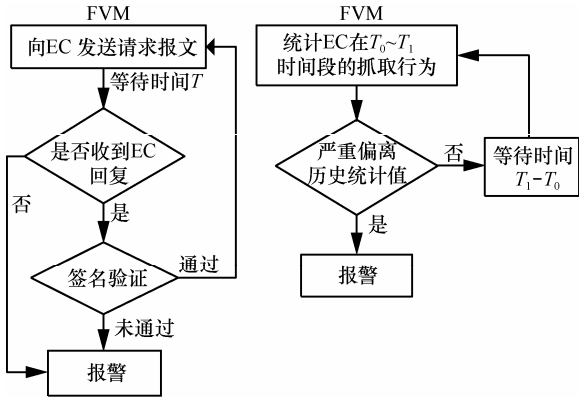

图 6 EC 保护流程

FVM 定时向 VM 中的 EC 发送请求报文，EC 收到请求后需及时响应，并利用其数字证书对响应数据进行签名。若 FVM 未收到响应数据，或从响应数据中提取的签名与 EC 本身的证书不符，则可认为 EC 程序已被恶意程序所终止。

FVM 中的数据接收程序对每个 EC 的证据抓取行为进行了记录，并按照时间段对 EC 的抓取次数、抓取数据量进行了统计，当 EC 在某个时间段的行为严重偏离了统计值时，认为有恶意程序阻碍其抓取或传输证据数据。

3) FVM 中的证据保护机制

FVM 中存放着大量的证据信息，为了确保这些证据数据不被恶意用户篡改，需设计有效的保护机制抵御以下 2 种威胁：①来至云系统内部其他 VM 的威胁；②来至 FVM 系统内部其他应用程序的威胁。Xen 系统中，Hypervisor 层实现了 VM 间的安全隔离机制，某一 VM 中的恶意程序无法篡改其他 VM 中的数据，这就有效抵御了第 1 种威胁，这也正是本文设置 FVM 并将证据保护及分析模块安插在 FVM 中的主要原因。对于第 2 种威胁，采用了以下抵御策略：在 FVM 中对强制访问控制机制进行了配置，只有证据接收进程拥有证据数据的写操作权限，除证据接收、分析、提取进程外，其他进程均不能读取证据数据。

3.5.2 证据分析

证据数据收集完成后，可利用数据挖掘技术

对这些数据进行分析，实时发现云系统中出现的一些安全问题[16,17]。证据的智能分析技术较为复杂，为了尽可能地精简本取证框架，只实现了对来自云系统内部的拒绝服务式攻击的智能检测，并预留了证据智能分析接口。智能分析作为未来工作去完成。

4　实验验证

4.1　实验环境

为了验证本文提出的方法，本文搭建了开源云平台 Eucalyptus，之所以选择该平台，是因为 Eucalyptus 是一个与 Amazon EC2 兼容的 IaaS 系统，具有较强的代表性。整个云系统由 5 台相同配置的 PC 机构成，每台 PC 均配有 Intel Core i7 860@2.8GHz CPU、4 GB 内存、500 GB 硬盘，运行 32bit 的 Fedora 12 操作系统。其中，1 台 PC 作为 Eucalyptus 的前端节点，运行着云控制器 CLC、集群控制器 CC、存储控制器 SC 及 walrus 组件，其余 4 台则为实际的计算节点，运行着节点控制器 NC 组件。

Eucalyptus 平台上共运行 7 个虚拟机实例：1 个 FVM，3 个搭载 Windows Server 2003 操作系统的 VM，3 个搭载 Fedora 10 操作系统的 VM。除 FVM 外，其他 VM 均向外提供 Web 服务。

4.2　攻击方法

Windows Server 2003 系统中存在一个设计上的漏洞(CVE-2008-4250) [18]，恶意用户通过构造一个特殊的远程 RPC 请求，则可远程执行其构造的任意代码。在实验过程中，利用该漏洞对云中的 VM 发起攻击，具体步骤如下。

1) 利用渗透工具 Metasploit 构造特殊 PRC 请求，获得目标机 VM01 的 System 用户权限并开启 4444 端口。

2) 远程登录 VM01，开启命令行程序 cmd.exe。

3) 从 VM01 中登录事先搭建好的 FTP 服务器，从该服务器中下载漏洞扫描工具、“灰鸽子”木马。

4) 在 VM01 中执行漏洞扫描工具，获取云平台中其他 VM 可能存在的系统漏洞信息。

5) 在 VM01 中安装“灰鸽子”木马，删除其系统日志。

4.3　证据提取

在上述攻击过程中，黑客最终删除了 VM01 中的系统日志并安装了木马，将该机器变成了“肉机”，并可以将该机器作为跳板对整个云系统发起攻击。黑客的上述操作严重破坏了 VM01 中的证据数据，致使传统的取证方法无法从该机器中获得完整及可靠的证据信息。

本文所述的取证框架能实时获取 VM01 中的证据数据，在取证虚拟机 FVM 中，与 VM01 相关的部分证据数据如表 1 所示。由表 1 可知：黑客构造的 RPC 请求会致使浏览器服务意外终止；攻击源主机的 IP 地址为 192.168.1.110；攻击源主机所属的源工作站为 VvvWoTb5JVSKTJD。

表 1　　EC 获取的部分证据数据

时间	来源	事件	描述
2012-6-13 15:51:36	VM01-Browser	8032	浏览器服务已很多次无法正常启动
2012-6-13 15:51:46	VM01-Security	680	源工作站: VvvWoTb5JV
2012-6-13 15:51:46	VM01-Security	529	源网络地址: 192.168.1.110
2012-6-13 15:52:14	VM01-Security	592	已经创建新的过程

实时取证模块中的部分证据数据如表 2 所示，由该表可以得到黑客入侵系统后所进行的操作：首先启动 cmd.exe，然后执行 ftp.exe 下载部分恶意工具，其次执行漏洞扫描 sfind.exe，最后安装木马 G_Server.exe。

表 2　　实时取证模块中部分证据数据

时间	来源	进程 ID	进程名
2012-6-13 15:50:54	VM01	1964	mshta.exe
2012-6-13 15:50:56	VM01	1032	mmc.exe
2012-6-13 15:52:14	VM01	2348	cmd.exe
2012-6-13 15:53:10	VM01	1152	ftp.exe
2012-6-13 15:54:31	VM01	1604	sfind.exe
2012-6-13 15:56:03	VM01	3372	G_Server.exe

由上述证据数据可知，本文不但获得了黑客所属工作站、IP 地址等关键信息，还通过实时取证模块得到了其入侵系统后所运行的进程信息，由这些进程信息所推导出的黑客攻击步骤与上一节中所述的攻击步骤完全一致，也进一步证明了 ICFF 框架的有效性。

4.4　证据数据的规模

ICFF 框架所获取证据数据的规模与用户的具体配置相关，在默认配置条件下，框架只自动抓取 VM 系统的日志及审计信息。通过实验发现，每个 VM 每天平均产生 2 239 KB 的日志及审计数据。在

本节所述的实验中，就是通过分析上述 2 MB 左右的证据数据获得了黑客的 IP 地址等关键信息，从而避免了对整个虚拟机镜像(每个镜像平均 8.23GB)的取证，提高了取证效率。

5　结束语

本文分析了云环境下数字取证技术所面临的挑战和难题，提出了一种 IaaS 云服务模型下的取证框架 ICFF。在该框架中，训练了一种在 Windows 及 Linux 平台中正常工作的智能证据抓取器，该抓取器能够自动地从虚拟机中抓取证据数据；设置了一个专门用来存储、分析证据信息的取证虚拟机 FVM；设计了一套安全保护机制，能有效提高该取证框架的抗干扰能力，保障证据数据的完整性及机密性。将该框架在开源 IaaS 云平台 Eucalyptus 中进行了实现，并设计实验进行了验证分析。实验表明，该框架能够有效并快速地获取云平台中的证据数据，并能对取证系统和证据数据进行有效保护。在下一步工作中，将增强实时取证模块的功能，增加对 VM 内存的实时取证，完成对证据数据的智能分析。另外，还进一步研究 PaaS 及 SaaS 云服务模型下的取证方法。

参考文献：

[1] SIMSON L G. Digital forensics research: the next 10 years[J]. Digital Investigation, 2010, (7): 64-73.

[2] WOLTHUSEN S D. Overcast: forensic discovery in cloud environments[A]. The 5th International Conference on IT Security Incident Manage ment and IT Forensics[C]. Stuttgart, 2009. 3-9.

[3] KEYUN R, JOE C, TAHAR K, *et al*. Cloud forensics: an overview[A]. 7th IFIP Conference on Digital Forensics[C]. Florida, USA, 2011. 35-46.

[4] BIGGS S, VIDALIS S. Cloud computing: the impact on digital forensic investigations[A]. International Conference for Internet Technology and Secured Transactions[C]. London, Engloud, 2009. 1-6.

[5] BIRK D, WEGENER C. Technical issues of forensic investigations in cloud computing environments[A]. IEEE 6th International Workshop on Systematic Approaches to Digital Forensic Engineering[C]. Oakland, 2011. 1-10.

[6] Eucalyptus[EB/OL]. http://www.eucalyptus.com/.

[7] NURMI D, WOLSKI R, GRZEGORCZYK C, *et al*. The eucalyptus open-source cloud-computing system[A]. IEEE/ACM International Symposium on Cluster Computing and the Grid[C]. Shanghai, China, 2009. 124-131.

[8] KEYUN R, IBRAHIM B, JOE C, *et al*. Survey on cloud forensics and critical criteria for cloud forensic capability: a preliminary analysis[A]. Proceedings of the ADFSL Conference on Digital Forensics, Security and Law[C]. Virginia, USA, 2011. 105-121.

[9] REILLY D, WREN C, BERRY T. Cloud computing: forensic challenges for law enforcement[A]. International Conference for Internet Technology and Secured Transactions[C]. London, Engloud, 2010. 1-7.

[10] ELIE B, IVAN F, MATTHIEU M, *et al*. Doing forensics in the cloud age OWADE: beyond files recovery forensic[A]. Black Hat[C]. Las-Vegas, USA, 2011. 1-23.

[11] ZAFARULLAH Z, ANWAR F, ANWAR Z. Digital forensics for eucalyptus[A]. Frontiers of Information Technology[C]. Islamabad, 2011. 110-116.

[12] 周刚. 云计算环境中面向取证的现场迁移技术研究[D]. 武汉:华中科技大学, 2011.
ZHOU G. Research on Scene Migration of Computer Forensics in Cloud Computing Environment[D]. Wuhan: Huazhong University Science and Technology, 2011.

[13] WU J Z, DING L P, WANG Y J, *et al*. Identification and evaluation of sharing memory covert timing channel in xen virtual machines[A]. IEEE 4th Conference on Cloud Computing[C]. Washington DC, USA, 2011. 283-291.

[14] WU J Z, DING L P, LIN Y Q, *et al*. Xenpump: a new method to mitigate timing channel in cloud computing[A]. IEEE 5th Conference on Cloud Computing[C]. Hawaii, USA, 2012. 678-685.

[15] PAUL B, BORIS D, KEIR F, *et al*. Xen and the art of virtualization[A]. Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles[C]. New York, USA, 2003. 164-177.

[16] XU W, HUANG L, FOX A, *et al*. Detecting large-scale system problems by mining console logs[A]. Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Priciples[C].New York, USA, 2009. 117-132.

[17] HU Z B, SU J, SHIROCHIN V P. An intelligent lightweight intrusion detection system with forensics technique[A]. 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications[C]. Dortmund, 2007. 647-651.

[18] CVE-2008-4250[EB/OL]. http://www.cve.mitre.org/cgi-bin/cvename. cgi?name =CVE-2008- 4250.

作者简介：

谢亚龙（1988-），男，湖南邵阳人，中国科学院软件研究所硕士生，主要研究方向为云计算、数字取证与系统安全。

丁丽萍（1965-），女，山东青州人，博士，中国科学院软件研究所研究员、博士生导师，主要研究方向为数字取证、系统安全与可信计算。

林渝淇（1988-），男，山东潍坊人，中国科学院软件研究所博士生，主要研究方向为隐蔽信道、云计算与系统安全。

赵晓柯（1989-），男，河南禹州人，中国科学院软件研究所硕士生，主要研究方向为系统安全。