软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(6):1402−1416 [doi: 10.13328/j.cnki.jos.005009]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

KVM 虚拟化动态迁移技术的安全防护模型*

范 伟[1,2], 孔 斌[3,4], 张珠君[1], 王婷婷[1,2], 张 杰[1,2], 黄伟庆[1,4]

[1](中国科学院 信息工程研究所,北京 100093)

[2](中国科学院 研究生院,北京 100049)

[3](国家保密科技测评中心,北京 100044)

[4](北京交通大学,北京 100044)

通信作者: 张珠君, E-mail: zhangzhujun@iie.ac.cn

摘 要: 虚拟机动态迁移技术是在用户不知情的情况下使得虚拟机在不同宿主机之间动态地转移,保证计算任务的完成,具有负载均衡、解除硬件依赖、高效利用资源等优点,但此技术应用过程中会将虚拟机信息和用户信息暴露在网络通信中,其在虚拟化环境下的安全性成为广大用户担心的问题,逐渐成为学术界讨论和研究的热点问题.从研究虚拟化机制、虚拟化操作系统源代码出发,以虚拟机动态迁移的安全问题作为突破点,首先分析了虚拟机动态迁移时的内存泄漏安全隐患;其次结合 KVM(kernel-based virtual machine)虚拟化技术原理、通信机制、迁移机制,设计并提出一种基于混合随机变换编码方式的安全防护模型,该模型在虚拟机动态迁移时的迁出端和迁入端增加数据监控模块和安全模块,保证虚拟机动态迁移时的数据安全;最后通过大量实验,仿真测试了该模型的安全防护能力和对虚拟机运行性能的影响.仿真结果表明,该安全防护模型可以在 KVM 虚拟化环境下保证虚拟机动态迁移的安全,并实现了虚拟机安全性和动态迁移性能的平衡.

关键词: KVM 虚拟化;动态迁移;安全防护模型;混合随机变换

中图法分类号: TP309

中文引用格式: 范伟,孔斌,张珠君,王婷婷,张杰,黄伟庆.KVM 虚拟化动态迁移技术的安全防护模型.软件学报,2016,27(6):1402−1416. http://www.jos.org.cn/1000-9825/5009.htm

英文引用格式: Fan W, Kong B, Zhang ZJ, Wang TT, Zhang J, Huang WQ. Security protection model on live migration for KVM virtualization. Ruan Jian Xue Bao/Journal of Software, 2016,27(6):1402−1416 (in Chinese). http://www.jos.org.cn/1000-9825/5009.htm

Security Protection Model on Live Migration for KVM Virtualization

FAN Wei[1,2], KONG Bin[3,4], ZHANG Zhu-Jun[1], WANG Ting-Ting[1,2], ZHANG Jie[1,2], HUANG Wei-Qing[1,4]

[1](Institute of Information Engineering, The Chinese Academy of Sciences, Beijing 100093, China)

[2](Graduate University, The Chinese Academy of Sciences, Beijing 100049, China)

[3](National Secrecy Science and Technology Evaluation Center, Beijing 100044, China)

[4](Beijing Jiaotong University, Beijing 100044, China)

Abstract: Live migration of virtual machines is the transfer of running virtual machines from one host server to a new host server to ensure computing tasks completed without notifying the owners of virtual machines, which has many beneficial characteristics such as load balancing, hardware independent, and high efficiency utilization of resource. However, live migration of virtual machines exposes information of virtual machines and their users to the network, making its security in the virtualized environment a serious problem that

* 基金项目: 国家自然科学基金(61502486)
Foundation item: National Natural Science Foundation of China (61502486)
收稿时间: 2015-08-15; 修改时间: 2015-10-09; 采用时间: 2015-12-05; jos 在线出版时间: 2016-01-21
CNKI 网络优先出版: 2016-01-22 11:20:10, http://www.cnki.net/kcms/detail/11.2560.TP.20160122.1120.019.html

concerns many users becomes a hot issue in the industry and academia. This article focuses on researching the mechanism of virtualization and the source code of virtualization operating system, and explores breakthrough in security problems of live migration. Firstly the article analyzes potential memory-leak security threat of live migration. Then it designs and puts forward a new security protection model based on hybrid random transform coding method. Combined with KVM (kernel-based virtual machine) virtualization structure, communication mechanism and migration mechanism, the model adds monitor module and security module at source and destination of live migration, ensuring the data security while the virtual machines are migrating. Finally, a series of experiments are designed to simulate and test the security protection capability of the model and its impact to virtual machine's performance. The simulation results show that the proposed model can ensure the security of live migration in the KVM virtualization environment, as well as balance the security of virtual machines and performance of live migration.

Key words: KVM virtualization; live migration; security protection model; hybrid random transformation

虚拟化技术是实现云计算为用户提供灵活增减 IT 资源、按需付费等特色服务的基础核心,在虚拟化环境下,云计算才成为可能,所以虚拟化技术无疑成为了研究热点.虚拟化架构主要由宿主机、虚拟化层和虚拟机组成,虚拟机动态迁移技术是在保持虚拟机为用户提供持续服务的状态下,从一个虚拟平台宿主机迁移到另一个虚拟平台宿主机中,为资源的整合和动态调度带来了巨大便利,虚拟平台厂商都各自提出了自主的动态迁移技术,目前流行的 KVM 虚拟化技术是基于硬件虚拟化的 Linux 全虚拟化解决方案,发展迅速,迁移技术日趋完善,但忽略了对安全问题的重视,存在一些安全威胁.

Oberheide[1]在 2008 年就从 3 个方面对虚拟机动态迁移的安全性进行了实验与分析,分别是 VMM(virtual machine monitor,虚拟机监控器,等同于 Hypervisor,是虚拟化操作系统的核心)控制层、数据层和迁移模块层.在 VMM 控制层,由 VMM 实施启动及管理在线迁移,所以它的通信机制必须经过验证并抗篡改,黑客可能会通过 VMM 控制层来影响在线虚拟机迁移,进而控制客户虚拟机.在数据层,虚拟机的内存信息必须得到保护,防止被嗅探和篡改.在迁移模块层,如果黑客根据迁移模块的漏洞破坏 VMM,那么他可能完全控制 VMM 和其他所有虚拟机.针对数据层的攻击,黑客通常采用地址解析协议(address resolution protocol,简称 ARP)欺骗、域名系统(domain name system,简称 DNS)污染以及路由劫持等手段,通过监视在线迁移通道,黑客可以从迁移的内存信息中提取出用户账户、口令等敏感信息.Oberheide 利用 Xensploit 工具(即:Oberheide 设计的工具可以在动态迁移过程中实施中间人攻击)对源 Domain0 和目的 Domain0 之间迁移的 DomainU 中部分内存信息进行了篡改,验证了数据层的安全威胁.Yamunadevi[2]在 KVM 虚拟化平台上进行了动态迁移实验,也证实了被迁移虚拟机内存上的信息容易被黑客截获.另外,本团队[3,17]从 2012 年开始就一直对动态迁移过程中内存泄漏安全问题进行研究,可还原出虚拟机迁移过程中的内存泄漏数据.黄休平[4−7]等人可以根据特征值恢复源文本信息,增加了虚拟机迁移过程中的安全隐患.

由于虚拟机动态迁移期间传输的数据是不安全的,Sulaiman 等人提出通过使用 IPsec(Internet protocol security,网络安全协议集)实现数据在宿主机之间安全传输[8],并通过调整 MTU(maximum transmission unit,最大传输单元)和 MSS(maximum segment size,最大段大小)的值来提高迁移时的表现.Pati 提出使用 RSA 算法(一种公钥解密算法)和 SSL(secure sockets layer,安全套接层协议,一种常用加密算法)来保证数据安全[9],并使用 MOSIX 软件(集群软件)将内存迁移确保负载均衡.Nagin 等人提出基于 SSH 通道的在线迁移方式[10],保证迁移过程的安全,这些方法虽然能够有效防止监听者窃取传输的虚拟机数据,在一定程度上保证迁移虚拟机实例的传输安全,但是它无法保证虚拟机是否被迁移到一个可信的服务器中,这样容易造成虚拟机的失控,而且也未采取任何方式的完整性保护措施,无法确认传输的虚拟机实例是否被篡改.并且这种方式明显增加了动态迁移中虚拟机的宕机时间,性能大大下降.

可信平台模块(trusted platform module,简称 TPM)是由可信计算组(trusted computing group)指定的规范,使用 TPM 芯片可以搭建一个安全且可信赖的运算平台,通过虚拟化技术可以让每个虚拟机拥有一个 TPM(实际上使用物理机上的 TPM),此 TPM 为 VTPM(虚拟 TPM)[11],通过 VTPM 迁移协议可以验证源和目的 VMM 的完整性并在动态迁移前创建一个安全的连接通道,张新方等人则对基于可信通道的 VTPM 迁移协议进行改进[12],改进的协议使用加密、完整性检验和随机数保证迁移数据的完整性、机密性和及时性.但是基于 VTPM 的迁

移不适合动态迁移的环境.

同样,基于可信令牌(trusted token)[13]的迁移方式,它是由制定策略、实现迁移策略和审计迁移组件组成.用户的策略包含对虚拟机迁移中目的云平台的可信保证等级 TAL(trust assurance level).当目的宿主机上可信令牌中的 TAL 值处于用户迁移策略中的 TAL 值范围时,迁移才可以发生.基于角色/策略的迁移方式[14]保护迁移过程,它包含认证服务(attestation service)、密封存储(seal storage)、策略服务(policy service)、迁移服务(migration service)和安全 Hypervisor(secure hypervisor)组件.虽然基于这两种方式的安全迁移在主机间认证有一定优势,但它需要相关硬件的支持,不适合动态迁移的环境.

NSE-H[15](network security engine-hypervisor,网络安全引擎)是对 Hypervisor 的一种扩展.它具有防火墙、入侵检测等防御系统功能来抵御对虚拟网络的入侵,Chen[16]等人针对 NSE-H 提出一种实时迁移框架 CoM.NSE-H 可以在迁移过程中保证虚拟机的安全上下文,如防火墙、入侵检测系统等.但是 NSE-H 不能满足虚拟机迁移数据层的安全需求.

本文的研究重心在于对 KVM 虚拟化平台上的动态迁移安全问题进行探讨,进而提出安全防护模型.第 1 节将对动态迁移过程中的安全隐患进行总结归纳.第 2 节对动态迁移过程中的内存泄漏问题进行分析,并设计实验截获数据并还原信息.第 3 节在基于 KVM 虚拟化系统底层源代码和动态迁移机制的前提下提出一种基于混合随机编码方式的安全防护模型.第 4 节用实验仿真的方式对安全防护模型的防护效果进行验证和分析.最后总结全文并指出下一步研究方向.

1 动态迁移安全隐患

虚拟机是虚拟化系统中一种特殊的信息资产,存在发生不期望事件而造成损害的潜在可能性,实际应用中面临多种多样的安全隐患,下面主要对其在动态迁移过程中的安全隐患进行讨论,主要包括以下 4 个方面.

(1) 不安全的通信通道.迁移的数据可能遭到监听甚至修改.一方面,攻击者通过监听源宿主机与目标宿主机之间的网络,获得迁移过程中传送的全部数据,从而可以确定迁移虚拟机的大小、迁移时间、甚至虚拟机操作系统的用户口令、应用数据等敏感信息.另一方面,攻击者通过修改迁移的内存数据为特定数据,进而控制虚拟机.

(2) 缺少访问控制.例如允许未经授权的用户启动和迁移虚拟机.攻击者可以通过启动大量虚拟机迁移到一个目标宿主机上,耗费宿主机资源使其拒绝服务.或者被迁移的虚拟机带有恶意软件、木马、恶意代码等,通过迁移到其他安全宿主机使其可以攻击目标宿主机或其他虚拟机.

(3) 平台缺少完整性验证.虚拟机可能被迁移到不可信平台中,攻击者在虚拟机迁移前,已经通过某种手段控制了目的平台,一旦虚拟机被迁移过来,攻击者就会对其进行攻击破坏,获取虚拟机权限.

(4) 虚拟化系统迁移模块本身存在漏洞.攻击者可能在 VMM 上的迁移模块寻找漏洞,比如整数符号问题的栈溢出,内存分配程序问题的堆溢出,这些漏洞允许攻击者获得特权代码执行权限并危及到 VMM 和宿主机安全.

2 动态迁移内存数据泄漏分析

2.1 原 理

动态迁移实际上是把虚拟机的配置封装在一个文件中,然后通过高速网络,把虚拟机配置和内存运行状态从一台物理机迅速传送到另外一台物理机上,期间虚拟机一直保持运行状态,并在迁移结束后在迁入端继续虚拟机提供的服务,对于一次动态迁移而言,攻击者可以利用的不安全点包括迁出端、迁入端、网络通信链路、网络文件系统服务端,如图 1 所示.

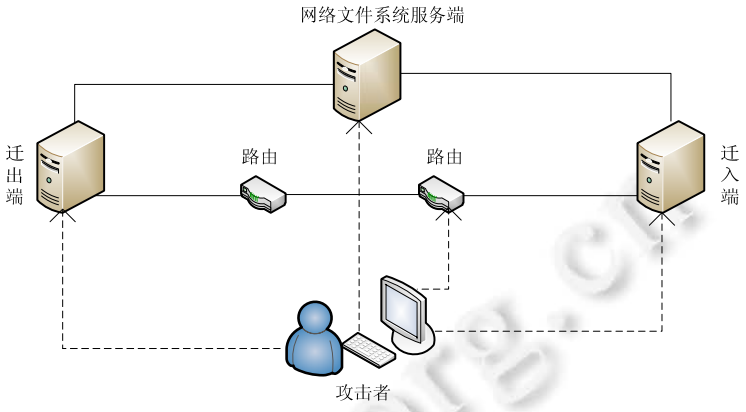

Fig.1 The attack points of virtual machine live migration

图 1 虚拟机动态迁移攻击点

在迁出端或迁入端,如果攻击者已经成功攻陷主机或者得到了主机高级权限,就可以控制虚拟机的迁移、监控其上虚拟机的内存甚至修改被迁移虚拟机的数据达到预期的目的.网络文件系统服务端,又称为共享存储,主要存放虚拟机文件如虚拟机配置文件、虚拟机的虚拟磁盘文件.为了保证迁移过程的快速和减小迁移带来的停机时间,常常建立共享存储来存储虚拟机的硬盘数据,这样动态迁移发生时,只需迁移虚拟机的内存数据[17].如果网络文件系统服务端被病毒破坏或遭受木马入侵等,这些都会对虚拟机的数据完整性、机密性和可用性造成威胁.

攻击者经常采用控制网络通信链路的方式,通过监听网络通信可以得到虚拟机的信息,这种方法是较为简单有效的方法,也是用户难于防控的攻击方式.本节将从攻击者角度通过网络通信链路实施攻击,通过嗅探软件监听迁移时的通信内容,进而分析内存数据,还原出敏感信息.

2.2 实验方案

本实验方案结构为:两台资源配置一致的宿主机应用 KVM 虚拟化系统,两台宿主机上的虚拟机硬盘存储基于共享存储.将其中一台宿主机上的客户虚拟机动态迁移到另一台宿主机时,通过交换机端口镜像功能,利用网络嗅探类工具从交换机上截获宿主机间的传输数据,用于后续的数据分析.

2.3 截获数据分析

实验场景 1. Linux 系统的终端模式下用户开机后选择相应账号和口令,并使用一些简单命令,此时进行迁移,迁移时虚拟机状态如图 2 所示,该虚拟机状态 1 的原始口令为 centospassword,Linux 系统界面下不显示输入的口令.

实验结果. 此时抓取的数据包可以清晰地看到用户的口令信息如图 3 所示,可以清晰看到用户的原始口令已经被还原出来,而且虚拟机界面的内容也存在于数据包中.

分析. 在此模式下,输入的命令、用户的密码等重要信息都会被存放到内存中,而当前界面所显示的信息会存放在显存中,所以同样会出现在迁移的数据包中.

实验场景 2. Linux 系统中虚拟机状态 2 下打开已保存过的文本文档,不做任何编辑操作,其内容包括:字母、数字、中文、符号等,即:2015_中国科学院信息工程研究所---CAS-iie.此时迁移虚拟机,通过软件抓取数据包,迁移时虚拟机状态如图 4 所示.

实验结果. 可以在抓取的数据包中找到其编辑的相应内容,实验结果如图 5 所示.

分析. 虚拟机运行的文本文档进程和内容都会存储在迁移的内存中,在虚拟机迁移时捕获内存数据就有机会还原出这些敏感信息.

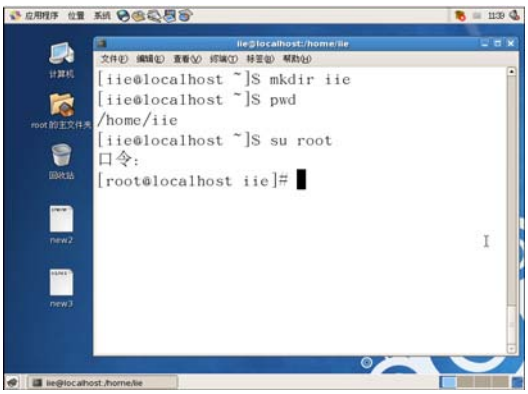

Fig.2 The VM state 1

图 2 虚拟机状态 1

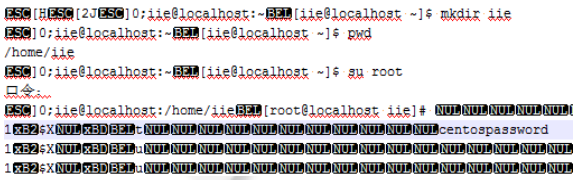

Fig.3 The experimental results 1

图 3 实验结果 1

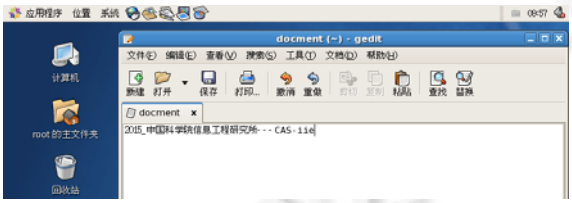

Fig.4 The VM state 2

图 4 虚拟机状态 2

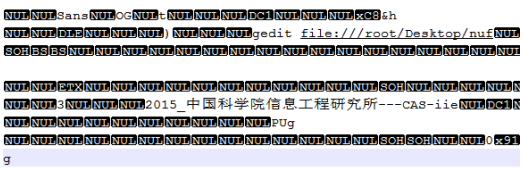

Fig.5 The experimental results 2

图 5 实验结果 2

3 安全防护模型建立

3.1 安全防护模型来源

虚拟机是指通过软件模拟出与正常的操作系统具有完整硬件系统功能的、运行在一个完全隔离环境中的完整计算机系统,在其中运行的进程与正常的操作系统一样都具有虚拟内存空间,而 Linux 系统将其分为内核空间和用户空间两部分,Windows 系统将其分为用户空间和系统空间两部分.内核空间(系统空间)的区域,用户不能访问,无论哪个用户进程处于活动状态,虚拟地址空间内核部分的内容总是同样的.经过第 2 节的分析,动态迁移过程是存在安全隐患的,而内核中的数据大都相同,不会涉及到敏感信息,大多数关键敏感的数据都属于用户,因此应着重保护用户进程使用的数据,本论文基于此防护角度提出安全防护模型.安全防护模型结构如图 6 所示.

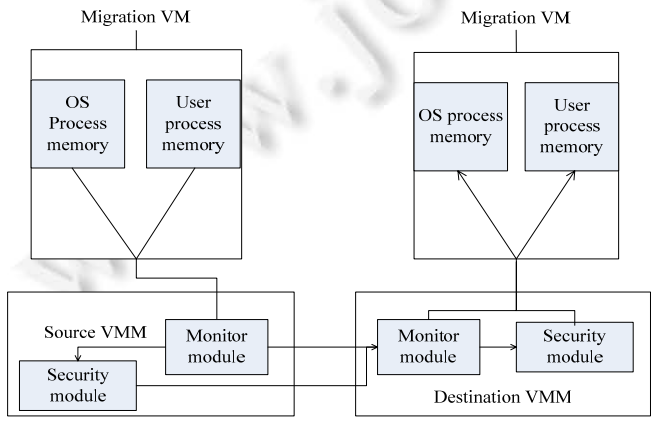

Fig.6 Security protection model of virtual machine live migration

图 6 虚拟机动态迁移安全防护模型

该安全防护模型的思想主要是在传统迁移机制的基础上,迁出端增加了发送数据监控模块(monitor module)和安全模块(security module),当虚拟机迁移时,对将要迁移的数据进行监控,将重要的信息内容在未打包发送到网络之前,通过安全模块重新编码,从而保证迁移数据的安全性.在迁入端同样增加了接收数据监控模块和安全模块,迁入端通过监控模块监听接收的数据,并通过安全模块将收到的迁出端安全模块编码的数据包解码为源虚拟机数据.迁出端的安全模块主要实现将迁移数据重新编码,打破数据原有的存在规律,使得攻击者即便通过某种手段截获了迁移数据,也无法得到想要的文本特征,从而保证数据的机密性、完整性、可用性.此安全防护模型的实现是基于可信的 VMM 进行改进,在上面加固的监控模块和安全模块的安全性与该 VMM 安全性等价,并不会带来新的攻击面.

3.2 算法设计

1) 迁出端数据监控模块实现

通过对 KVM 虚拟化系统的源代码分析,KVM 虚拟机迁移时采用 pre-copy 算法,内存复制过程如下.

(1) 预复制.将虚拟机的全部内存页面从源宿主机复制到目的宿主机;

(2) 迭代复制.将上一轮过程中被修改过且到目前为止本轮复制过程中没有被修改过的页面迭代复制到目的宿主机;

(3) 停机复制.将虚拟机剩余少量没有同步的内存页面和虚拟机系统运行信息复制到目的宿主机.

在原算法基础上进行改进,加入监控模块,其算法结构如图 7 所示,其中,migration_bitmap 标记内存是否为脏页面,迁移时根据 migration_bitmap 找到需要发送的内存页,在内存页面没发送到目的宿主机前,加入一个判断机制,判断当前复制的页面是否属于用户,如果属于,则调用安全模块编码,如果不属于,则将其内容复制到 QEMU 文件中,等待发送.

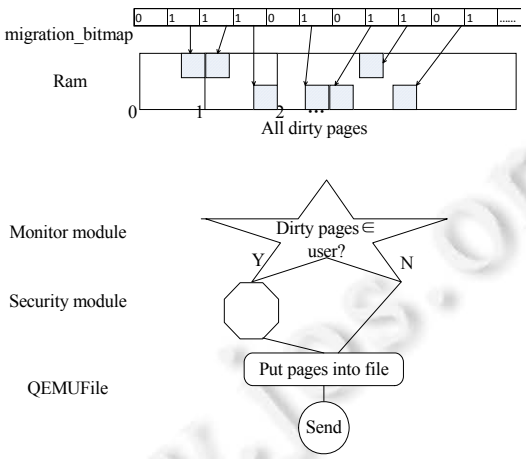

Fig.7 The algorithm structure of the monitor module at source

图 7 迁出端数据监控算法结构

2) 迁出端安全模块设计

迁出端安全模块主要对源宿主机内存页进行重新编码,需要设计一种较好的编码算法防止被攻击者破解.现假设源虚拟机正在看一个 word 文档,攻击者会监听通信过程,并通过抓取的数据包找到 word 特征值,进而找到 word 文档内容,为此设计以下 4 种编码算法:① 移位代换;② 线性变换;③ 混合随机变换;④ RC4 变换.

(1) 移位代换的基本思想是将每一个字符都移动相同的距离,或者设置一个表,根据表中数据进行相应位置的移位.常规利用 ASCⅡ码(american standard code for information interchange,美国标准信息交换代码)进行字符编码,一个字符一个字节,占 8 位.若每个字符都移动相同的距离,那么所有编码数据变换的可能性为 256 种,很容易被穷举攻击,若根据表进行移位代换,假设表长度为 m,则每 m 个数据都有 256^m 种变换(m 最大为 4 096),

若原特征值长度为 n,则原特征值有 256^n 种可能($n<m$),当 n 的值足够大是不可以被破解的.

(2) 线性变换的基本思路是对任意字符进行一定规律的线性变换,用得到的值代替原来的字符,如对任意字符 x,y 为重新编码后的字符,则对应的变化如式(3-1)所示.王防修[18]等人利用此种方法实现了文件在网络传输中的安全性,但此方式虽然实现了非明文传输,但是因为线性变化规律是一定的,所以同一个字符经过的变换都是相同的,而且此时 a 有 2^7 种选择,b 有 2^8 种可能,所以有 2^{15} 种线性变换规律,是不能抵御攻击者根据特征值进行穷举攻击的.

(3) 混合随机变换,是结合上述两种方法的优势实现,其算法基本结构如图 8 所示.

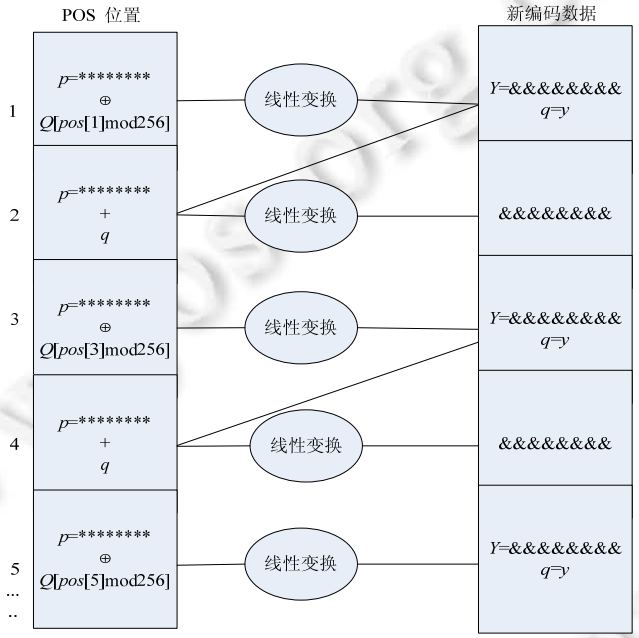

Fig.8 The algorithm structure of the security module at source

图 8 迁出端安全模块算法结构图

设源字符为 p,中间缓冲字符 q,当此数据为已编码个数的奇数位时,对 p 进行如式(3-2)变换,得到 x,其中 $pos[p]$为字符 p 所处已编码字符的位数,Q 为安全模块的设置的移位表,表 Q 暂设长为 256.

再对 x 进行线性变换,对应变化如式(3-1),y 为重新编码后的字符,令 q 等于此次奇数编码的数据即 y.

当此数据为已编码个数的偶数位时,则令 $x=p+q$,其中,q 为上次奇数位的新编码数据,再对 x 进行如式(1)的线性变化,y 为重新编码后的字符,线性变化有 2^{15} 种可能,且同时因为加入 q 缓冲字符、pos 和 Q 增加了不确定因素,其中,Q 有 256!种可能.若已知这种编码机制,那么特征值会有 2^{1707} 种可能,但由于是混合变换方式,奇数位和偶数位变换机制不同,若只用同一种变换机制的思想去穷举破解是很困难的.

$$y \equiv e(x) \equiv ax+b \mod 256, a,b \in Z_{256}, gcd(a,256)=1 \tag{1}$$

$$x \equiv p + pos[p] \mod 256 \tag{2}$$

(4) RC4 算法[19,20]是一个典型的基于非线性数组变换的序列密码.其由 Rivest 提出,常应用于 SSL 以保护因特网的信息流,它以足够大的数组为基础,对其进行非线性变换,产生非线性的密钥序列,一般把这个大数组称为 S 盒,RC4 的每个输出都是这个 S 盒中的元素.其算法主要通过密钥随机调度算法(key scheduling algorithm,简称 KSA)用来设置 S 的初始化排列,伪随机生成算法(pseudo random generation algorithm,简称 PRGA)用来选取随机元素与源数据异或,同时修改 S 的原始排列顺序,此种加密方式使源数据有 2^{1700} 种变换方式.

RC4 变换方式的密钥随机调度算法计算出的密钥比混合随机变换算法的随机性更高,但是混合随机变换理论上的变换可能性更多,更难破解.因此本设计最终选择改进后的混合随机变换算法,为迁出端和迁入端安全

模块的算法.

3) 迁入端数据监控模块实现

迁入端监控模块算法结构如图 9 所示,在迁入端重建源虚拟机时,首先会将接收到的迁移数据存储在QEMU 文件中,然后重建虚拟机时将从QEMU 文件中读取需要的数据,监控模块则负责在重建虚拟机之前,判断将要重建的数据是否是用户的内存页,如果是则调用安全模块对相应内存页解码,否则不做任何处理.

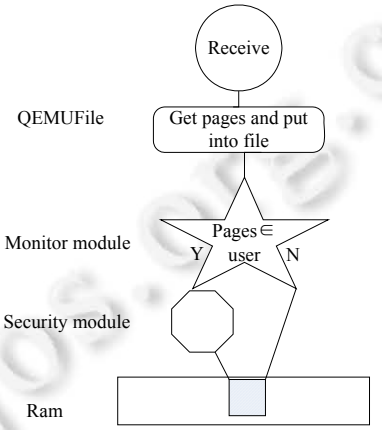

Fig.9 The algorithm structure of the monitor module at destination

图 9 迁入端数据监控算法结构

4) 迁入端安全模块设计

与迁出端安全模块相对,将虚拟机迁入的数据,根据迁出端安全模块的设计,在迁入端执行逆过程,安全模块算法结构如图 10 所示.

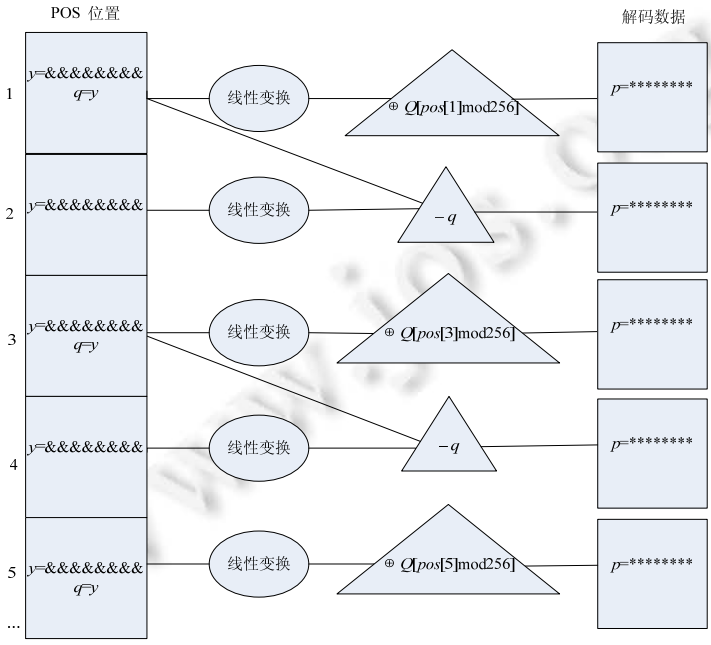

Fig.10 The algorithm structure of the security module at destination

图 10 迁入端安全模块算法结构图

设接收到数据为 y,若 y 字符所处位置是已经解码个数的奇数位,则令 $q=y$,解码时需要编码时 a 在 Z_{256} 上的乘法逆元 $a^{-1}\in Z_{256}$,对应变化如式(3),此时的 $e(x)$为 y.

$$x=d(e(x))\equiv a^{-1}(e(x)-b)\ \bmod\ 256 \tag{3}$$

得到 x 后进行如式(3-4)变化,得到源字符 p,当 y 字符所处位置是已经解码个数的偶数位,则只进行线性变换,再减去上一个奇数位收到的字符 q,得到源字符 p.

$$p\equiv x-pos[p]\ \bmod\ 256 \tag{4}$$

3.3 预期效果

该安全防护模型主要从两个方面对传统迁移机制进行修改,一是从防护对象上,二是从安全算法上.

传统的防护方式只是利用数据加密技术将虚拟机的所有数据加密再传输,或是在传输层、网络层等对整个传输通道进行加密,防止窃听和修改,而本文由于在内存页加了标志位,将需要保护的数据范围缩小,在保护数据的同时,减少了迁移需要的时间.

攻击者需要通过窃听得到大量的网络通信包,进而要将多个数据包组合分析获得一些特征值,得到虚拟机的一些重要信息,安全模块算法如果使用传统的算法复杂的 AES、DES 等加密方法,这样的算法过于繁琐,安全性固然提高,但是迁移时的加解密时间开销会很大,停机时间长,严重影响虚拟机的迁移时间和虚拟机的性能.本文设计一种安全防护模型,在 VMM 层进行防护,可以监控虚拟机内存,在迁出端和迁入端可以针对特定内存数据编码和解码,本模型设计的改进混合随机变换编码方式,其具体应用过程如下.

1) 假设特征值为“*efg*”这 3 个位置上连续的字符,线性变换中 a 为 3,b 为 5,缓冲字符 q 为 x,现有两种情况,一是 $pos[e]$为奇数位,此时设 e 的位置为 1;二是 $pos[e]$为偶数位,则设 e 的位置为 2,编码过程如下.

第 1 种情况 $pos[e]$=1,则 $pos[g]$为 3,$Q[pos[1]]$=3,$Q[pos[3]]$=5.经过重新编码后“*efg*”变为“7*Lb*”,具体过程如下所示.

$$e_1\equiv a(e\oplus Q[pos[e]])+b\ \bmod\ 256\rightarrow '7', q_1=e_1;$$
$$f_1\equiv a(f+q_1)+b\ \bmod\ 256\rightarrow 'L';$$
$$g_1\equiv a(g\oplus pos[g])+b\ \bmod\ 256\rightarrow 'b'; q_1=g_1;$$

第 2 种情况 $pos[e]$=2 时,则 $pos[f]$为 3,$pos[g]$为 4,$Q[pos[f]]$=15.经过重新编码后“*efg*”变为“£@·”,具体过程如下所示.

$$e_2\equiv a(e+q)+b\ \bmod 256\rightarrow '\mathcal{L}';$$
$$f_2\equiv a(f\oplus Q[pos[f]])+b\ \bmod\ 256\rightarrow '@',\ q_2=f_2;$$
$$g_2\equiv a(g+q_2)+b\ \bmod\ 256\rightarrow '\cdot';$$

2) 设特征值是 3 个相同的字符“*eee*”位置上连续,同样假设线性变换中 a 为 3,b 为 5,缓冲字符 q 为 x,有两种情况,一是第 1 个 e 的位置 $pos[e]$为奇数位,此时设 e 的位置为 1,则第 3 个 e 的位置是 3;二是第 1 个 e 的位置 $pos[e]$为偶数位,则假设 e 的位置为 2,那么第 2 个 e 的位置为 3,编码过程如下.

第 1 种情况第 1 个 e 的位置为 1,则第 3 个 e 的位置为 3,Q 和上面 1)中第 1 种相同,经过重新编码后“*eee*”变为“7*I*%”,具体过程如下所示.

$$e_3\equiv a(e\oplus Q[pos[e]])+b\ \bmod\ 256\rightarrow '7'; q_1=e_3;$$
$$e_4\equiv a(e+q_1)+b\ \bmod\ 256\rightarrow 'I';$$
$$e_5\equiv a(e\oplus Q[pos[e]])\ \bmod\ 256\rightarrow '\%'; q_1=e_5;$$

第 2 种情况若第 1 个 e 的位置为 2,则第 2 个 e 的位置为 3,Q 与上面 1)中第 2 种相同,重新编码后“*eee*”变为“£*C*┌”,具体过程如下所示.

$$e_6\equiv a(e+q)+b\ \bmod\ 256\rightarrow '\mathcal{L}';$$
$$e_7\equiv a(e\oplus Q[pos[e]])+b\ \bmod\ 256\rightarrow 'C'; q_2=e_7;$$
$$e_8\equiv a(e+q_2)+b\ \bmod\ 256\rightarrow '\ulcorner';$$

由以上可知,当特征值的位置不同时,编码的结果也不同,并且即使是连续的相同字符也很难观察出其规律

性,所以被破解的几率很小.

本模型设计的改进混合随机变换编码方式,将需要保护的数据重新编码,并利用源数据的不确定性扩大下一个数据的不确定性,打乱原有数据的规律,其编码的线性变换过程是通过线性同余方程实现的,若要解码必须知道源线性变化过程,才能推导出解码所用的线性变换方式,仅仅通过截取的数据包这种被动攻击的方式是很难破解的,同时因为是针对特定的内存页采用重新编码的方式,在编码的页面加入标志位,接收端 VMM 层的监控模块可以监控这个标志位来决定是否解码内存页面,如果是监听则无法找到这个特征,从而也就无法确定哪些数据是重新编码的,也就无法找到破解方式,所以可以保证数据的安全性.同时此编码机制中有 q 这个中间缓冲字符的存在,若攻击者修改迁移过程中的数据,那么迁移将会不成功,因为此时会解码错误,导致恢复虚拟机状态失败,从而保证了迁移过程中的数据完整性.

因此,本安全防护模型理论上可以实现在保证迁移虚拟机安全性的同时,保证迁移的时间在一个可接受范围,即实现动态迁移安全性和效率的平衡.

4 仿真验证

根据设计的安全防护模型,基于 KVM 虚拟化底层动态迁移机制,修改 KVM 源码,编译,进行虚拟机迁移,测试结果.

4.1 功能测试

在 Linux 系统下编辑文本文档,保存后迁移,查找截获的数据包,找不到原文档内容.编译设计的代码后多次重复第 2 节的截获敏感数据实验,发现找不到预期的数据.证明该安全防护模型和算法完全实现了预期的安全功能.

本防护方案假设攻击主要是监听网络间的通信数据,再进行破解,可归类为唯密文攻击,理论上纯暴力破解是不能实现的.为了更好论证此方案的安全性,现假设攻击者不仅通过监听得到被迁移虚拟机的所有数据,同时也知道会被重新编码传输的信息串,令此信息串长度为 n,因算法是基于线性变换和移位代换进行改进,而线性变换仅仅将源字符经过线性变换为另一个字母,每次变化规律相同,并未对字母出现概率隐藏,所以可以利用语言的规律性来分析破解.现有统计分析方法针对移位代换的有效统计分析方法主要分 3 步,第 1 步是确定密钥的长度,常用方法是 kasiski 测试法和重合指数法;第 2 步是确定密钥,常用的方法是重合指数测试法,第 3 步是根据第 2 步确定的密钥恢复出明文.

针对线性变换安全分析,传统线性变换只是单纯将明文加密经过一种线性变化,因此字符的出现规律没有改变,同一个字符线性变换后还是一样,所以比较容易被分析攻击.而本安全防护模型中的算法在线性变换之前对源字符进行了移位处理,相同的字符经过此种变换变得差异甚远,而且每次的移位也是随机的,所以相同的字符在虚拟机迁移过程中线性变换的结果都不同,因此通过统计分析字符规律是不可行的.

针对移位代换的安全分析,kasiski 测试法的基本原理是:若用给定的密钥字长度 m 的密钥周期地对明文字母加密,则当明文中有两个相同字母组在明文序列中间隔的字母数为 m 的倍数时,这两个明文字母组对应的密文字母组必然相同.反过来,若密文中出现两个相同的字母组,他们所对应的密文字母组未必相同,但相同的可能性极大.可以通过将密文中相同的字母组找出,并对其间隔的字母数进行综合研究,找出它们间隔字母数的最大公因子,就有可能得到密钥长度 m 的值.此种方法虽然理论上确实可以获得密钥长度 m 的值,但是本文所提安全算法缩小了数据保护范围,仅将重要的数据进行重新编码,而将重新编码和未重新编码的数据统一通过网络传送给接收端,因此依靠监听时获得的数据包统计间隔字母数是不易实现的,统计的间隔也是不准确的.如果继续假设攻击者确定的间隔是准确的,得到了密钥长度为 m,则密钥 $K=(k_1,k_2,...,k_m)$.下一步利用重合指数测试法得到密钥.设密文总长度为 n,令 $f_0,f_1,....,f_n$ 分别表示密文子串 c_i 中字母 $A,B,...,Z$ 出现的频率,再令 $N=n/m$ 表示子串 c_i 的长度,则 26 个英文字母在 c_i 中出现的概率依次是(实际密文不仅仅是 26 个英文字母):

$$\frac{f_0}{N},\frac{f_1}{N},...,\frac{f_{25}}{N}.$$

因为密文子串 c_i 是由对应的明文子串的字符移动 k_i 个位置得到的,所以移位后的概率分布为

$$\frac{f_{k_i \bmod 26}}{N},\frac{f_{(k_i+1) \bmod 26}}{N},\ldots,\frac{f_{(k_i+25) \bmod 26}}{N}.$$

假设 $0\leqslant j\leqslant 25$,定义数值:

$$M_j=\sum_{i=1}^{25}\frac{pif(i+j) \bmod 26}{N} \tag{5}$$

如果 $j=k_i$,则由重合指数的定义可知:

$$M_j=\sum_{i=1}^{25}pi^2\approx 0.065 \tag{6}$$

如果 $j\neq i$,则 M_j 一般与 0.065 相差较大,对任意的 $1\leqslant i\leqslant m$ 都可以使用这种方法来确定 k_i 的值.

对于本文所提的安全算法,此种方式分析破解是不成立的,因为本文的移位代换机制和传统不相同,仅仅按照传统的分析方法是无法破解的.本文的算法主要通过使明文的统计特性与密文的统计特性不一样,统计分析方法在这里不是很有效果.基于数学的发展,任何算法都有被攻破的可能,比如 SSL 中 RC4 算法机制的漏洞,已经被发现多年,安全性并不像理论上那么可靠;本文所提的算法并不交换密钥,不会如 SSL 机制一样出现攻击者控制网络通信链路监听通信数据得到密钥并破解被保护的数据情况.本文提出的算法是否存在逻辑漏洞,作者暂时没有发现,同时将该安全算法放在 VMM 层,尽量保证算法的保密性,减少攻击者分析破解的机会.

4.2 性能测试

在虚拟机负载相同的情况下,分别多次进行原始迁移(normal)、修改源码(modified)安全迁移以及加密通道迁移(ssl),衡量虚拟机迁移的性能指标[21],如总体迁移时间、停机时间、对应用程序的性能影响等方面进行对比,如图 11~图 16 所示.

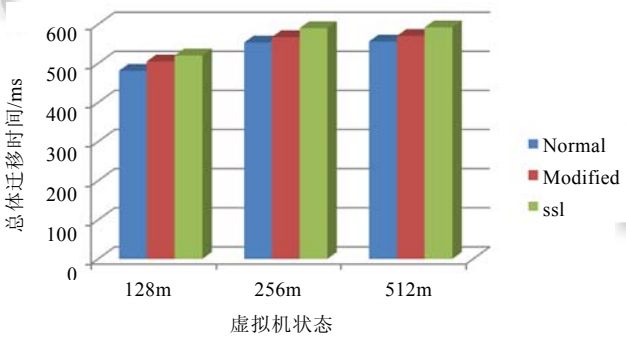

Fig.11 Total migration time of writing intensive VM

图 11 写密集型虚拟机总体迁移时间

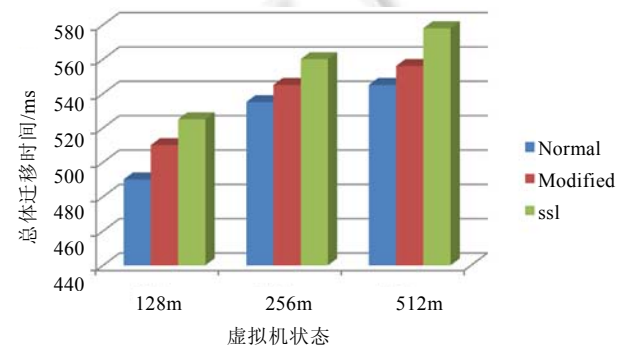

Fig.12 Total migration time of reading intensive VM

图 12 读密集型虚拟机总体迁移时间

总体迁移时间是指从源宿主机开始迁移直到迁移结束的时间,迁移过程中两台机器状态必须同步,时间过长可能会影响稳定性,所以总体迁移时间越短越好.由图 11、图 12 可见,写密集型情况下加入修改源码安全迁移方式在一定程度上使迁移时间增长,但比加密通道迁移(ssl)方式迁移的总时间短,而在读密集型虚拟机上迁移,修改源码安全迁移方式相对于加密通道迁移方式缩短了更多的时间,表现更好,迁移效率更高.

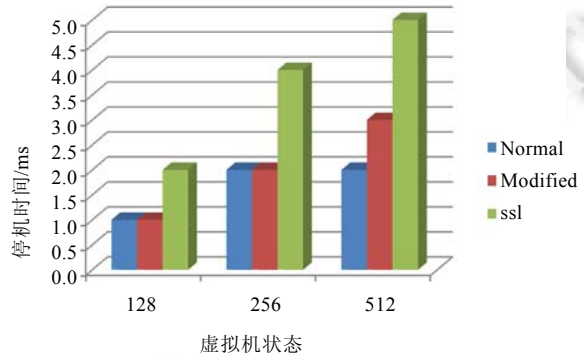

Fig.13 Migration downtime of writing intensive VM

图 13 写密集型虚拟机迁移停机时间

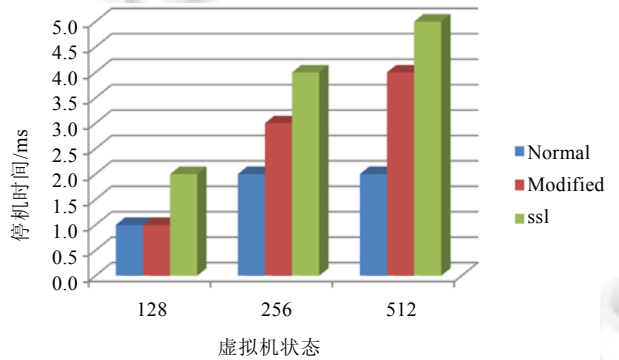

Fig.14 Migration downtime of reading intensive VM

图 14 读密集型虚拟机迁移停机时间

停机时间是指迁移过程中,源宿主机和目的宿主机的虚拟机都不可用的时间,此时无法为用户提供服务,所以迁移时应使停机时间最短,最大限度为用户提供服务.由图 13、图 14 可以看出,由于停机拷贝数据量相对较小,所以虚拟机停机时间较短,虽然此时这 3 种方法时间差别很小,但也能看出修改源码安全迁移方式对虚拟机的停机时间影响比使用加密通道迁移方式迁移时间更短,对用户的影响更小.

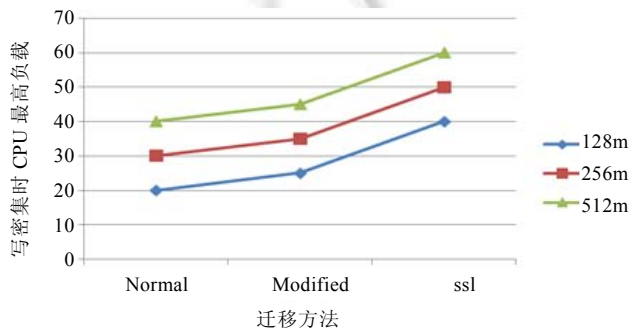

Fig.15 Peak CPU usage of writing intensive VM

图 15 写密集型虚拟机迁移 CPU 最高负载

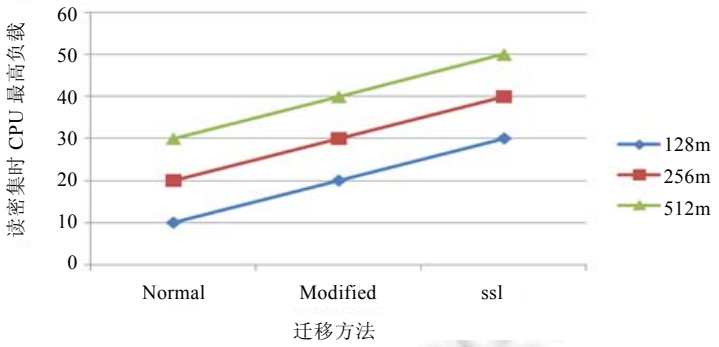

Fig.16 Peak CPU usage of reading intensive VM

图 16 读密集型虚拟机迁移 CPU 最高负载

对应用程序的性能影响是指迁移功能运行时对于被迁移宿主机上运行其他服务性能的影响程度,迁移过程应保证不会通过资源争夺来干扰正在活动的服务,CPU 是计算机重要的计算资源,计算机所有的操作都由CPU 控制,而由图 15、图 16 可以看出无论虚拟机是在写密集还是读密集的环境下进行迁移,用修改源码安全迁移方式迁移的 CPU 占有率较原始迁移方式高,但相对于加密通道迁移方式更低,可以说修改源码安全迁移方式相对于加密通道迁移方式对虚拟化平台上的其他虚拟机或其他服务的影响更小.

从表 1 可以看到,虚拟机迁移时,被重新编码的数据大约为总迁移数据的 75%左右,相对于加密通道迁移方式在传输层对网络进行加密的方式明显减少了数据量,并且从上述写密集和读密集两种条件下进行的实验测试结果中可以看到,虽然此文设计的安全防护模型在总体迁移时间上、停机时间上比普通没有安全措施的迁移方式时间长,但是比用加密通道迁移方式的时间短,对虚拟平台上其他应用的影响相对于加密通道迁移方式更小,可以说从衡量虚拟机迁移效率的三方面出发,此安全防护模型都要优于加密通道迁移方式,时间上可以让使用者更加容易接受,安全性也达到了一个可以应用的级别.

Table 1 Contrast between migration ram pages and total migration data of virtual machine for size 128MB

表 1 128MB 内存虚拟机迁移内存页与总数据量对比

实验次数	内存页数/4096B	迁移总数据/B	占总迁移数据百分比(%)
1	9 560	54 594 028	75.23
2	9 712	54 956 455	75.74
3	9 938	51 890 006	75.23
4	9 925	51 837 185	75.23

虚拟化动态迁移机制的整体安全防护效果不是只靠算法就可以实现的.算法防护能力的优越也需要人为管理和政策的配合,针对此问题,需要对管理员的权限重新规划并明确其职责范围,还要加入用户访问权限和访问记录管理等安全功能,降低非法用户访问非授权的文件,降低算法模块被攻击者获取的风险.此安全防护模型主要在原有 VMM 层加入新的安全模块,对动态迁移机制进行安全加固,而此 VMM 层被认为是整个虚拟化系统中最难以攻克的,可以说是最可信安全的.实际中 VMM 的高特权地位以及自身存在的漏洞和缺陷,使其很容易成为恶意攻击的对象.VMM 一旦被俘虏,将可能被利用来对宿主机系统和客户虚拟机进行窥探和破坏,这将严重威胁虚拟化系统的安全.现有的针对虚拟化系统安全的研究,大都集中于对虚拟机监控器进行加固和改良.本文的侧重点是针对 KVM 虚拟化系统动态迁移机制的安全性进行加固,VMM 安全方面的研究将作为后续研究部分.

5 结束语

本文首先对虚拟机动态迁移过程的安全隐患进行分析,根据其安全漏洞设计截获敏感信息的实验并对截

获数据进行还原,然后针对此安全漏洞,根据 KVM 虚拟化系统动态迁移机制,提出一种基于混合随机变换编码方式的安全防护模型,最后通过理论和实验,仿真验证了其可以实现对动态迁移过程中的安全防护,并实现了安全性和性能的相对平衡,达到了比较满意的效果.而且,本文的研究方法为未来研究虚拟化环境中相关技术的安全性提供了方法借鉴.

但本设计还有一些可以优化之处,如可以同时对传输的数据按内存块进行压缩,进一步减少迁移时间.

对于 KVM 虚拟化动态迁移的其他安全隐患防护归根到底还是 VMM 监控器的安全问题,对每个虚拟化版本及时打补丁和防护虚拟机逃逸可以解决迁移模块本身的安全问题,在 VMM 上添加访问控制模块和完整性验证模块,可以保证动态迁移过程的安全访问和平台完整性.这些都是本文安全防护模型完善和优化的地方,也是未来虚拟化 VMM 安全研究的方向.

致谢 在此,我们向对本文的工作给予支持和建议的同行表示感谢,尤其感谢审稿专家提出的宝贵意见.

References:

[1] Oberheide J, Cooke E, Jahanian F. Empirical exploitation of live virtual machine migration. In: Proc. of the BlackHat DC Convention. 2008.

[2] Yamunadevi L, Aruna P, Sudha D D, Priya N. Security in virtual machine live migration for KVM. In: Proc. of the 2011 Int'l Conf. on Process Automation, Control and Computing (PACC). IEEE. 2011. 1−6. [doi: 10.1109/PACC.2011.5979008]

[3] Han Y, Fan W, Liu C, Lu B, Wang RQ. Risk discovery and study on the virtual machine memory leak. Secrecy Science and Technology, 2012,2:19−23 (in Chinese with English abstract).

[4] Huang XP, Chen L. A method of extracting text of word from Windows XP physical image. Computer and Modernization, 2013,1(8): 165−167 (in Chinese with English abstract). [doi: 10.3969/j.issn.1006-2475.2013.08.041]

[5] Hu M, Yang JY, Jiang W. Data recovery algorithm based on file feature on windows platform. Journal of Computer Applications, 2011,31(2):527−529 (in Chinese with English abstract). [doi: 10.3724/SP.J.1087.2011.00527]

[6] Zhang XF, Huang ZW. Coding-Based document recovery technology. Netinfo Security, 2011,(9):156−158 (in Chinese with English abstract). [doi: 10.3969/j.issn.1671-1122.2011.09.049]

[7] Chen L, Jing K, Dong ZX. Searching physical memory method based on EPROCESS characteristics. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2013,25(1) (in Chinese with English abstract). [doi: 10.3979/j.issn.1673-825X.2013.01.020]

[8] Bin Sulaiman NA, Masuda H. Evaluation of a secure live migration of virtual machines using IPSEC implementation. In: Proc. of the IIAI 3rd Int'l Conf. on Advanced Applied Informatics (IIAIAAI). IEEE, 2014. 687−693. [doi: 10.1109/IIAI-AAI.2014.142]

[9] Patil VP, Patil GA. Migrating process and virtual machine in the cloud: Load balancing and security perspectives. Int'l Journal of Advanced Computer Science and Information Technology, 2012,1(1):11−19.

[10] Nagin K, Hadas D, Dubitzky Z, Glikson A, Loy I, Rochwerger B, Schour L. Inter-Cloud mobility of virtual machines. In: Proc. of the 4th Annual Int'l Conf. on Systems and Storage. New York: ACM, 2011. [doi: 10.1145/1987816.1987820]

[11] Berger S, Cáceres R, Goldman KA, Perez R, Sailer R, Doorn LV. vTPM: Virtualizing the trusted platform module. In: Proc. of the 15th Conf. on USENIX Security Symp. 2006. 305−320.

[12] Wan X, Zhang X F, Chen L, Zhu JX. An improved vTPM migration protocol based trusted channel. In: Proc. of the 2012 Int'l Conf. on Systems and Informatics. 2012. 871−875. [doi: 10.1109/ICSAI.2012.6223146]

[13] Wang W, Zhang Y, Lin B, Wu XX, Miao K. Secured and reliable vm migration in personal cloud. In: Proc. of the IEEE 2nd Int'l Conf. on Computer Engineering and Technology. 2010. 705−709. [doi: 10.1109/ICCET.2010.5485376]

[14] Aslam M, Gehrmann C, Bjorkman M. Security and trust preserving vm migrations in public clouds. In: Proc. of the IEEE 11th Int'l Conf. on Trust, Security and Privacy in Computing and Communications. 2012. 869−876. [doi: 10.1109/TrustCom.2012.256]

[15] Shetty J, Anala MR, Shobha G. A survey on techniques of secure live migration of virtual machine. Int'l Journal of Computer Applications, 2012,39(12):34−39. [doi: 10.5120/4875-7305]

[16] Chen XQ, Wan H, Wang SM, Long X. Seamless virtual machine live migration on network security enhanced hypervisor. In: Proc. of the IEEE 2nd Int'l Conf. on Broadband Network & Multimedia Technology. 2009. 847−853. [doi: 10.1109/ICBNMT.2009.5347800]

[17] Fan W, Huang WQ, Jiang F, Liu C, Lu B, Wang RQ. Research on the virtual machine memory leak in live migration. In: Proc. of the 24th Information Security Conf. 2014. 12−17 (in Chinese with English abstract).

[18] Wang FX, Zhou K. Application of affine password to the file encryption. Journal of Wuhan Polytechnic University, 2010,29(3): 62−64 (in Chinese with English abstract). [doi: 10.3969/j.issn.1009-4881.2010.03.014]

[19] Gu LZ, Zhen SH, Yang YX. Modern Cryptography Course. Beijing: Beijing University of Posts and Telecommmunications Press, 2009. 166−169 (in Chinese).

[20] Forouzan BA. Cryptography and Network Security. New York: McFraw-Hill, 2008. 219−222.

[21] Huang D, Ye D, He Q, Chen J, Ye k. Virt-LM: A benchmark for live migration of virtual machine. In: Proc. of the 2nd ACM/SPEC Int'l Conf. on Performance Engineering. New York: ACM, 2011. 307−316. [doi: 10.1145/1958746.1958790]

附中文参考文献:

[3] 韩奕,范伟,刘超,吕彬.虚拟化内存泄漏的风险探知及研究.保密科学技术,2013,2:19−23.

[4] 黄休平,陈龙.一种从内存镜像中获取 Word 文本的方法.计算机与现代化,2013,1(8):165−167.

[5] 胡敏,杨吉云,姜维.Windows 下基于文件特征的数据恢复算法.计算机应用,2011,31(2):527−529.

[6] 张雪峰,黄志炜.基于编码方式的文档恢复技术.信息网络安全,2011,(9):156−158. [doi: 10.3969/j.issn.1671-1122.2011.09.049]

[7] 陈龙,敬凯,董振兴,田庆宜.基于 EPROCESS 特征的物理内存查找方法.重庆邮电大学学报:自然科学版,2013,25(1). [doi: 10.3979/j.issn.1673-825X.2013.01.020]

[17] 范伟,黄伟庆,姜放,刘超,吕彬,王冉晴.虚拟化技术中动态迁移的内存泄漏安全问题研究.见:第 24 届全国信息保密学术会议(IS2014)论文集.2014.12−17.

[18] 王防修,周康.仿射密码在文件加密中的应用.武汉工业学院学报,2010,29(3):62−64. [doi: 10.3969/j.issn.1009-4881.2010.03.014]

[19] 谷利泽,郑世慧,杨义先.现代密码学教程.北京:北京邮电大学出版社,2009.166−169.

范伟(1984－),男,北京人,博士生,工程师,CCF 会员,主要研究领域为云计算安全,虚拟化安全.

孔斌(1973－),男,高级工程师,主要研究领域为网络安全.

张珠君(1987－),女,工程师,主要研究领域为物联网安全,云计算安全.

王婷婷(1992－),女,博士生,主要研究领域为云计算安全,虚拟化安全.

张杰(1990－),男,硕士生,主要研究领域为云计算安全,虚拟化安全.

黄伟庆(1972－),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为云计算安全,虚拟化安全,大数据安全.