计算机研究与发展 ISSN 1000-1239/CN 11-1777/TP
Journal of Computer Research and Development 47(3): 493－499, 2010

VFRS:一种面向虚拟计算环境的入侵容忍方法

赵 峰[1,2,3] 金 海[1,2,3] 金 莉[1] 袁平鹏[1,2,3]
[1](华中科技大学计算机科学与技术学院 武汉 430074)
[2](服务计算技术与系统教育部重点实验室 武汉 430074)
[3](集群与网格计算湖北省重点实验室 武汉 430074)
(zhaof@hust.edu.cn)

VFRS: A Novel Approach for Intrusion Tolerance in Virtual Computing Environment

Zhao Feng[1,2,3], Jin Hai[1,2,3], Jin Li[1], and Yuan Pingpeng[1,2,3]
[1](*School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan* 430074)
[2](*Key Laboratory of Services Computing Technology and System of Ministry of Education, Wuhan* 430074)
[3](*Provincial Key Laboratory of Cluster and Grid Computing of Hubei, Wuhan* 430074)

Abstract With the emergence of multi-core processor, virtualization technology has attracted attention and developed rapidly in recent years. Virtual computing environment based on virtual machine becomes a hot topic in the field of network computing. Virtual computing environment is open, complex and dynamic, which has brought new challenges to system security, especially to intrusion tolerance. In this paper, VFRS method is proposed in order to protect sensitive data from intrusion in virtual computing environment. Firstly, a probability computing model is constructed to present system call sequences and the SCSFA algorithm is designed to predict the attempt of intrusion and to determine what need to protect, which is based on the analysis of system call sequence in virtual computing systems; Secondly, the sensitive data protected are divided into a number of film data, and for the goals of random errors tolerance, each tablet data are redundant backup based on Byzantine fault tolerance; Then, the redundant data are distributed to different virtual machines. VFRS method can predict the anomaly intrusion and well tolerate the complicated errors in virtual computing environment. The experimental results show that VFRS is effective and of high performance compared with related work. Some key issues of the VFRS method are also discussed and analyzed in detail.

Key words intrusion tolerance; virtual computing; system security; sequence forecast; system call

摘 要 虚拟计算环境的开放性、复杂性和动态性向入侵容忍提出了新的挑战,提出VFRS方法以解决虚拟计算环境中数据对入侵的容忍问题.设计SCSFA算法分析虚拟计算环境的系统调用行为序列,以识别虚拟计算环境下的入侵企图,预测敏感数据的高危区域;其次,将要保护的数据划分成若干片数据,并以容忍虚拟计算环境随机错误为目标对每个片数据冗余备份;然后将冗余片数据分散到不同虚拟机上.VFRS方法能有效预测虚拟计算环境下的异常入侵,并能较好地容忍虚拟计算环境下的复杂性错误.对VFRS方法实现的关键问题进行了详细的讨论和分析.

关键词 入侵容忍;虚拟计算;系统安全;序列预测;系统调用
中图法分类号 TP393

收稿日期:2008-07-15;修回日期:2009-06-17
基金项目:国家自然科学基金项目(60803114);国家"九七三"重点基础研究发展计划基金项目(2007CB310900)

近年来,多核技术的出现使得计算系统的处理能力得到迅猛提升,计算环境呈现出多核化、虚拟化、开放化的特点,虚拟机(virtual machine, VM)技术重新得到重视并得以快速发展,基于虚拟机的虚拟计算环境(virtual computing environment)也成为当前网络计算领域的前沿课题之一.在计算环境虚拟化普及于各种领域带来诸多便利的同时,针对虚拟机安全漏洞的攻击也越来越严重,使得运行其上的信息承受潜在的巨大威胁.因此,创造安全的虚拟计算环境就显得至关重要,也逐步成为虚拟计算中的核心问题和新的研究热点.

入侵容忍系统(intrusion tolerance system, ITS)是第三代安全技术——信息生存技术——中的核心内容,它主要考虑在攻击存在的情况下系统的生存能力,所关注的是攻击造成的后果而不是攻击的原因,要解决在攻击存在的情况下系统的生存性问题[1-2].在无法预知虚拟计算环境所有未知形式的攻击和无法杜绝安全漏洞出现的情况下,入侵容忍就成为创建安全虚拟计算环境的一条重要途径.

虚拟计算环境具有开放性、复杂性和动态性的特点[3-4],传统入侵容忍方法体系结构复杂、可扩展性差,且硬件资源需求很大,无法满足虚拟计算环境可伸缩体系架构、资源灵活管理与按需配置等的需求.因此,针对虚拟计算环境特性,提出新的入侵容忍方法已迫在眉睫.鉴于此,本文在研究现有入侵容忍方法的基础上,提出VFRS(virtual fragmentation-redundancy-scattering)方法,以解决虚拟计算环境数据对入侵的容忍问题.研究结果表明,VFRS方法能有效对抗虚拟计算环境下的异常入侵,并能较好地容忍因入侵导致的复杂性错误.

1 相关工作

Clarkson大学的Matthews等人研究了虚拟计算环境下文件服务器上私有数据的保护机制,以及遭受恶意攻击后数据依据可信检查点的快速恢复机制[5].在这种机制下当虚拟机遭受恶意攻击时,它能自动重启并将系统恢复到一个已知的完好状态,同时也不丢失私有数据最新的更改.但这种机制的运行是以较大的工作负载作为代价的,它写操作的工作负载达到了24%. 文献[6]提出了面向HPC的容错方法,主要利用虚拟机的实况迁移机制实现容错,如果一台虚拟机发生故障,系统就自动地将正在执行的MPI程序动态迁移到其他虚拟机上.

Reiser等人利用虚拟化技术提出了面向网络服务容错和容侵的体系架构VM-FIT[7].它利用虚拟机的隔离性实现客户虚拟机上的构件可信,进而容忍入侵.VM-FIT的弊端在于它的可靠性和效率仍有待提高.台湾国立中央大学的Sun等人提出了VMITN的方法用于解决快速传播攻击的入侵容忍.它利用的OOB(Out-of-Band)网络和虚拟机结合检测入侵[8].经过测试,VMITN方法下有86%的服务和50%的主机可以容忍蠕虫病毒的攻击.

日本Keio University大学的Nguyen等人从数据一致性的角度研究了虚拟计算环境下非法攻击对数据篡改的检测与监控技术,并设计了面向Xen虚拟计算环境的XenFIT和XenRIM系统[9-10].XenFIT和XenRIM侧重于提高数据完整性检测的实时性和系统的易开发性,对单虚拟计算系统中数据篡改的检测有较好的效果,但其对攻击检测的效率依赖于选定的HIDS,同时它们也未讨论数据的保护问题.

2 VFRS容侵技术

尽管容侵系统的设计与实现方法多式多样,采用的技术和手段各不相同,但从对抗入侵的角度上讲,入侵容忍的目标在于保障系统服务的可用性以及保护系统数据的机密性和一致性,最终实现则体现于对物理数据的保护.国际研究成果中,最具代表性的是文献[11]提出的分布式环境的入侵容忍方法——“分片—冗余—发散”方法(fragmentation-redundancy-scattering, FRS).FRS方法描述了分布式环境下针对应用数据访问系统容忍入侵的方法,侧重于保护敏感数据的完整性和可用性,并被国际上最具影响的入侵容忍项目ITUA和SITAR项目所采用[12-13].

但FRS方法存在如下缺陷:1)FRS方法的实现需要大量的物理资源以支持冗余,同时需要配备服务器用于用户的注册、认证和目录索引,成本太高;2)入侵容忍系统一个最基本的假设就是系统潜在的缺陷是可被预测的,而FRS方法忽略了这一点,用认证/授权机制进行数据访问限制,随机考虑敏感数据保护以实现入侵容忍,客观上降低了容侵的性能,影响了系统后续处理的效率;3)FRS方法的可扩展性不强,无法适应虚拟机计算环境下客户虚拟机的动态扩展和随机创建.

2.1　VFRS 方法

针对 FRS 方法的缺点, 本文基于虚拟机架构[14], 设计了一种面向虚拟计算环境的容侵方法——VFRS 方法, 如图 1 所示. 其基本思想是: 首先通过分析系统行为预测虚拟计算环境下故障的危机区域, 确定需要保护的数据(如: 私有信息、系统配置信息等); 其次将要保护的数据划分成若干片数据, 并对每个片数据作冗余备份; 然后将冗余片数据分散到不同虚拟机上; 最后根据数据被攻击情况作出后续处理.

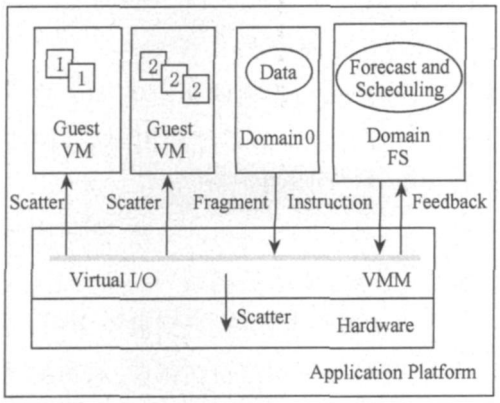

Fig. 1　VFRS intrusion tolerance.

图 1　VFRS 入侵容忍方法

VFRS 工作流程如下:

1) 分片. 将虚拟计算环境中要保护的数据调入特权域 Domain0, 在 Domain0 中将要保护的敏感数据划分成不同的片段. 这样, 即使有入侵发生, 入侵者取得的数据也只是无关联的片数据.

2) 冗余. 利用虚拟计算环境丰富的计算资源, 在客户虚拟机上复制划分好的片段, 以便当入侵或恶意攻击发生时, 有备份可以替换被篡改的片段.

3) 分散. 被保护的数据最终要存储在物理介质上, 而分散机制则决定片段数据具体存储在哪个虚拟机对应的物理域上. 分散的目的是隔离存储划分好的单个片数据, 以防止入侵发生后全部数据被获取. 这样, 可利用虚拟机本身的隔离机制将片段逻辑隔离, 增强单个入侵获取全部数据的防御力.

4) 预测与调度. 和传统分布式系统不同, 虚拟计算环境下虚拟机之间的信息传递与任务分配都是通过 VMM (virtual machine monitor)实现的, 而 VMM 并未提供数据划分、复制与分散的功能, 因此, VFRS 方法需要一个专用虚拟机 Domain FS 实现系统缺陷预测, 以及通过 Virtual I/O 辅助实现划分—冗余—分散的策略调度.

2.2　VFRS 分片

FRS 容侵方法采用了 CBC(cipher block chaining)对页数据加密, 这种方法增加了片内容被解析的难度, 但加密过程增加了 CPU 的负荷, 同时一旦某个工作站被入侵, 其上所有的片数据都有被窃取的危险. 有鉴于此, VFRS 容侵方法采用一种基于成本二元树存储的划分策略, 一方面从降低 CPU 消耗考虑取消了对片数据的加密, 另一方面采用散列和加权成本动态构建二元树的方法增强数据的安全性, 这样即使入侵成功也不会影响数据的完整性. VFRS 分片过程如下:

1) 建立附加信息. 确定数据需要划分的粒度和片数据最大长度, 根据数据完整性需求划分数据, 允许划分片数据的长度存在差异. 同时, 统计每段片数据历史访问的概率, 并建立片数据组织的索引信息.

2) 生成头数据. 将索引、长度、粒度信息以散列方法生成头数据, 作为引导性的片数据, 并将其定义为片数据二元树的根.

3) 构建成本二元树. 将划分的片数据作为节点, 从片数据的访问效率和安全性考虑, 以划分片数据的长度、敏感指数加权作为评价效益, 并采用动态规划算法建立最小成本二元树. 若片数据长度为 L, 敏感指数为 S, 对应权重分别为 w_L 和 w_S, 评价效益记为 p, 则有 $p = \sum(w_L L + w_S S)$.

4) 映射到硬件平台. 对二元树节点按从左到右、自上而下次序顺序编号, 以便将其映射到不同 Guest VM 对应的域(domain)上, 设节点编号为 i, Guest VM 总数为 M, 则节点 i 部署到域($i \bmod M$)上.

2.3　基于 BFT 的 VFRS 冗余

虚拟计算环境下丰富的存储资源为利用冗余技术实现系统的入侵容忍提供了便利, 但与此同时, 虚拟系统的开放性也增加了入侵行为的不可预测性以及系统可能作出反应的不可预料性, 也即增加了拜占庭失效(Byzantine failure)发生的概率. 有鉴于此, 我们在 VFRS 方法中采用了文献[15]提出的基于 BFT(Byzantine fault tolerance)的冗余技术. 基于 BFT 的 VFRS 冗余基本思想是: 整个数据冗余分复制、确认、反馈 3 个阶段; 对于任一片数据 i, 至少复制 3 个备份存储于不同 Guest VM 上, 映射列表保存在监控引擎中, 并有监控引擎负责维护; 随后各冗余备份向原始片数据 i 和其他冗余发送确认消息; 原始数据 i 和其他冗余备份收到确认消息后和自己的备份对比, 并返回反馈信息.

3 FS预测引擎

专用虚拟机 Domain FS 的一个重要功能就是对虚拟计算环境下系统缺陷及可能因入侵而引发系统故障的预测. 从 VMM 的角度看, Guest VM 的服务请求、数据访问(包括入侵)都是应用程序对操作系统的系统调用请求, 这些信息均可被 VMM 获取与监控, 因此, 虚拟计算环境下系统调用序列(sequences of system call)信息就成为最直接、最真实地反应各种行为的数据[16]. 本文设计了一种基于序列分析的预测算法——SCSFA (system call sequence forecasting algorithm), 通过对系统调用行为序列的分析来预测计算环境因入侵而可能引发的故障.

3.1 问题描述

系统调用的行为序列, 如: $\boldsymbol{s}=\langle$open, read, mmap, mmap, open, read, mmap$\rangle$, 可存储为长度为 k 的若干子序列, 设 $k=3$, 则有: $\langle$open, read, mmap$\rangle$ $\langle$read, mmap, mmap$\rangle$ $\langle$mmap, mmap, open$\rangle$ $\langle$mmap, open, read$\rangle$. 因此, 可以将系统调用序列描述为数据点 $s_1, s_2, \cdots, s_n$ 的有序序列, 每个系统调用的参数可以描述成数据点 s_i 上的属性域 $I=\{I_1, I_2, \cdots, I_m\}$. 则序列是数据点的有序表, 记为 $\boldsymbol{s}=\langle s_1, s_2, \cdots, s_n\rangle$, 其中 s_k ($k=1, \cdots, m$) 的取值源于 I. 含有 k 个数据点的序列称为 k 序列($k=\sum |s_i|$), 记为 L_k. 如果一个序列发生的足够频繁, 则称该序列是频繁序列, 若 L_k 频繁, 则记为 F_k. 频繁序列是 VFRS 方法认知和表示系统缺陷和故障的主要手段. 若 L_n 频繁, L_{n+1} 不频繁, 则称 F_n 是最大频繁序列.

3.2 计算模型

从概率论上讲, m 维的序列可表示为一个 m 维随机矢量, 它的分布是一个 m 维的概率分布. SCSFA 算法采用贝叶斯概率来统计行为序列的概率. 令序列 $\boldsymbol{s}$ 是从属性域 I 中随机抽取的概率分布, $p(\boldsymbol{s})$ 为其客观概率, $\boldsymbol{D}$ 为序列的数据值域空间, $p(\boldsymbol{s}|\boldsymbol{D})$ 为后验概率, $p(\boldsymbol{D}|\boldsymbol{s})$ 为先验概率. 设系统调用序列的训练数据集 $\boldsymbol{Ts}$ 中含有 N 个 m 维的序列, 令 N 足够大以保证能获取足够长的系统调用序列, 记为:

$$\boldsymbol{Ts}=\{\boldsymbol{t}_1, \boldsymbol{t}_2, \cdots, \boldsymbol{t}_N\},$$

其中矢量 $\boldsymbol{t}_k=\{t_{k1}, t_{k2}, \cdots, t_{km}\}$.

若 $s_i \subset \boldsymbol{t}_k$ 且 $\boldsymbol{s}_i=\langle s_{i1}, s_{i2}, \cdots, s_{in}\rangle$ 是频繁序列, $l(1\leqslant l\leqslant n)$ 表示当前序列点所在位置, 在单一序列点的情况下 l 也是序列概率树的深度, 则序列概率树可表示成:

$\boldsymbol{Bran}=\{(s_{i1}, l=0)\cdots(s_{i2}, l=1)\cdots(s_{in}, l=n-1)\}$,

$\boldsymbol{Tree}=\{\langle Node, p\rangle \mid Node\in\{s_1$ 相同的 $\boldsymbol{Bran}\}\}$,

其中 p 是序列概率树中节点概率集合, $p(l=k)=p(D_k)$, $p(s_{ik})=p(s_{ik}\mid D_k)$, $p(s_{ik}\mid l=k)=p(s_{ik}\mid D_k)\,p(l=k)$.

3.3 SCSFA 算法

SCSFA 算法步骤如下:

步骤 1. 生成行为序列的初始集 $\{s_1\}$. 在 VMM 中采集虚拟计算环境下的系统调用序列, 构造确定时段内行为序列的初始集 $\{s_1\}$;

步骤 2. 序列概率预测树的构建. 从长度为 1 的序列集合 $\{s_1\}$ 开始, 依次扩展到 $\{s_2\}$, …, $\{s_k\}$, 构造已知行为序列的概率预测树, 直到最长序列 s_k;

步骤 3. 序列概率预测树的裁剪. 从预测树的根节点开始, 按层次查询该层次上的频繁序列, 即若层次为 $l(1\leqslant l\leqslant\log_2 n)$, 则只查询到 F_l. 若从根到该节点所表示的序列频繁, 则在预测树中保留该分支, 否则删除该节点及其下的子树;

步骤 4. 寻找概率最大的 F_{k+1}. 在序列概率预测树中从 F_1 开始, 按 F_2, …, F_k 的演化路径寻找概率最大的 F_{k+1}.

3.4 算法性能

算法的计算时间包括可离线计算的时间、需实时处理的计算时间和 Virtual I/O 之间的通信时间; 算法的计算空间主要集中在序列树的存储开销上. 令 $|I|$ 表示系统调用序列值域空间的平均长度, m 表示序列流的维数, $|L|$ 表示频繁序列的平均长度, 最大频繁序列为 k-序列. 离线计算主要集中在系统调用行为序列初始集的建立上, 其时间开销为 $O(m\times\log|L|)$, 空间开销为 $O(k)$. 实时处理的计算时间主要集中在序列树的构建及历史路径的遍历搜索上, 系统调用行为序列预测树的构造时间为 $O(m^2\times k^2\times|I|\times\log|L|)$, 搜索预测树的时间最坏情况下为 $O(m\times|L|)$, 平均为 $O(m\times\log|L|)$. 算法空间开销主要取决于矢量概率树的存储开销, 为 $O(|L|\times|I|\times m)$.

通信消耗是影响算法效率的重要因素之一, 为降低通信消耗, 算法在传输处理结果时, 仅传输

F_{k+1}和 $P(F_{k+1}|\ l=k+1)$, 若 F_{k+1}的每个数据点的 Virtual I/O 需 1 个单位时间, 则在 F_{k+1}序列点有限的情况下, 最坏情况其需要的时间为 $O(k)$, 平均情况需要时间为 $O(|L|)$, 最好情况需要时间为 $O(1)$. 通信消耗的空间需要主要是存储 F_{k+1}和 $P(F_{k+1}|\ l=k+1)$的空间, 需要 $O(m\times(k+1))$单位空间.

4　实验结果与分析

我们在 Linux 平台上(Fedora8, Xen3.1, 内核 2.6.23)仿真虚拟计算环境, Domain FS 置于 Domain0 中, 用各种攻击入侵仿真虚拟计算系统, 通过对数据库文件的入侵容忍作为原型来验证 VFRS 方法的性能, 实验中采用文献[17-18]中攻击实例, 实验环境包括 1 台双 CPU(每个 CPU 4 核、频率 1.6GHz, cache 4MB, 内存 2GB)的服务器、1 台 P2.0 GHz/512MB/cache128KB 的 PC 和 1 台 MySQL 数据服务器. 我们将从如下 2 个方面验证 VFRS 方法的性能: 1)SCSFA 算法的执行效率和预测精度; 2)VFRS 方法的执行效率.

VFRS 方法的性能很大程度上取决于 SCSFA 算法的执行效率和预测精度, 我们在 Linux 2.6.23 上选取了全部的系统调用(325 个)来检测 SCSFA 的性能. 针对不同的系统调用频繁序列长度 k, 我们在宿主机及客户机上对 login 木马攻击执行 SCSFA 预测算法的效率对比见表 1.

Table 1　Execution Time and Accuracy of SCSFA for "login" Attack

表 1　针对 **login** 攻击的 **SCSFA** 的执行时间与精度

k	Host/s	Guest/s	Feature Database	Forecast Rate/%
3	0.210	0.210	825	74.853
5	0.510	0.555	1222	81.342
6	0.544	0.570	1573	88.970
7	0.567	0.600	1702	89.355
8	0.567	0.601	1875	91.355
10	1.766	1.900	2265	91.684
14	3.826	4.113	3211	91.720
20	7.322	7.679	4396	92.021

实验结果表明: 1)SCSFA 算法在主机和 Guest VM 上的执行时间相差不大, 误差低于 6%, 因此 SCSFA 算法是适用于虚拟计算环境的; 2)k 越大预测的精度越高, 执行时间也越长, k 超过 8 以后算法执行时间大幅增加, 但预测精度变化不大, 因此, 在虚拟计算环境下 SCSFA 算法对短时系统调用的入侵预测更有效.

我们仿真虚拟计算环境模拟攻击对 MySQL 数据库的 134 个 data 文件进行了测试, VFRS 方法对入侵的抗击能力如表 2 所示. 实验结果表明: 除去误报导致的文件错误, 当入侵发生后数据文件的失效率较小, 也即 VFRS 方法对入侵有较好的容忍能力.

Table 2　Intrusion-Tolerant Performance of VFRS

表 2　**VFRS** 方法对入侵的容忍性能

Attack	Invalidation Rate	Invalidation Rate from Misreport
FK 0.4	5/134	1/2
Adore	1/134	1/1
ARK 1.0	2/134	0/4
Knark v.2.4.3	7/134	4/12
hhp-trosniff	5/134	2/2
ulogin.c	2/134	0/7

我们对 MySQL 下 1 个 2GB 的 MYD 文件进行了测试(CPU 时钟周期为 10 ns), VFRS 方法通过 IOZone 对 Linux 主机, Xen－host, Xen－guest 上的读写数据进行了测试, 实验结果对比如图 2 所示. 实验结果表明: VFRS 方法对虚拟计算环境并没有造成太大的负载, 它主要的消耗集中在对数据写操作上, 这是由数据划分和冗余造成的, 就目前虚拟计算环境的计算能力而言是可容忍的.

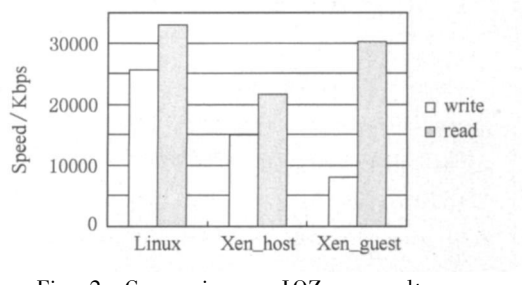

Fig. 2　Comparison on IOZone results.

图 2　IOZone 测试结果对比

VFRS 和 FRS 方法的对比如表 3 所示, 比较结果表明: 1)VFRS 方法不需要额外的硬件设备支持数据容侵策略; 2)VFRS 方法增加了分片的计算时间, 但减少了 CBC 加密的时间; 3)VFRS 方法中的片数据冗余数更低. 虽然 VFRS 方法在执行基于成本二元树存储的划分策略时增加了分片计算的冗余时间, 但其总体时间复杂度仍保持在 $O(n)$; 而 VFRS 方法的空间冗余度为 $O(n)$, 远低于 FRS

方法的$O(n!)$, 因此, 和FRS 方法相比, VFRS 的资源消耗较低, 更适合于动态性的虚拟计算环境.

Table 3 Comparison Between VFRS and FRS

表3 VFRS 和 FRS 对比

Performance	VFRS	FRS
Needing hardware	No	Yes
Fragmentation encrypt	No	CBC
Fragmentation time	$O(2n)$	—
Scattering time	$O(\log n)$	$O(n)$
Frag-data(n) redundancy	$3n+1$	$n!/2$

5 结论与展望

随着计算环境虚拟化的迅速普及, 针对虚拟计算环境的恶意攻击事件日益增多, 潜在安全隐患防不胜防, 因此, 如何容忍入侵就成为虚拟计算环境中一项迫切而富有挑战性的研究课题. 本文提出的VFRS 入侵容忍方法是针对虚拟计算环境的特征对FRS 方法的改进, 该方法能有效预测虚拟计算环境下的入侵企图, 以及较好的容忍复杂性错误, 该方法在多虚拟计算系统中有广阔的应用前景. VFRS 方法在实现过程中需要全面的系统调用特征库作为训练集加以辅助, 特征库越全面, VFRS 的预测能力就越强. 另外, 本文也没有讨论 VFRS 方法数据写操作的优化问题, 我们将对此作进一步的研究.

参 考 文 献

[1] Veríssimo P, Neves N F, Correia M. Intrusion-tolerant architectures: Concepts and design, DI/FCUL TR03-5 [R]. Springfield: University of Lisboa, 2003

[2] Li Qinghua, Zhao Feng. The PBL method: A novel parallel error detection method for intrusion tolerance systems [J]. Journal of Computer Research and Development, 2006, 43(8): 1411—1416 (in Chinese)
(李庆华, 赵峰. 一种面向容侵系统的并行错误检测方法——PBL 方法[J]. 计算机研究与发展, 2006, 43(8): 1411—1416)

[3] Jaeger T, Sailer R, Sreenivasan Y. Managing the risk of covert information flows in virtual machine systems [C] // Proc of the 12th ACM Symp on Access Control Models and Technologies. New York: ACM, 2007: 81—90

[4] Asrigo K, Litty L, Lie D. Using VMM-based sensors to monitor honeypots [C] //Proc of the 2nd Int Conf on Virtual Execution Environments. New York: ACM, 2006: 13—23

[5] Matthews J N, Herne J J, Deshane T M, et al. Data protection and rapid recovery from attack with a virtual private file server and virtual machine appliances [C] //Proc of the 2nd IASTED Int Conf on Communication, Network and Information Security. Phoenix: ACTA, 2005: 170—181

[6] Nagarajan A B, Mueller F, Engelmann C, et al. Proactive fault tolerance for HPC with xen virtualization [C] //Proc of the 21st Annual Int Conf on Supercomputing. New York: ACM, 2007: 23—32

[7] Reiser H P, Kapitza R. VM-FIT: Supporting intrusion tolerance with virtualisation technology [C] //Proc of the 1st Workshop on Recent Advances on Intrusion-Tolerant Systems. New York: ACM, 2007: 18—22

[8] Sun Wenchun, Chen Yiming. VMITN: A novel intrusion tolerance architecture for treating the rapid propagation of malicious programs [C] //Proc of the Int Computer Symp. Piscataway, NJ: IEEE, 2006

[9] Nguyen A, Takefuji Y. A novel approach for a file-system integrity monitor tool of xen virtual machine [C] //Proc of the 2nd ACM Symp on Information, Computer and Communications Security. New York: ACM, 2007: 194—202

[10] Nguyen A, Takefuji Y. A real-time integrity monitor for xen virtual machine [C] //Proc of The IEEE Int Conf on Networking and Services. Piscataway, NJ: IEEE, 2006: 90—98

[11] Deswarte Y, Blain L, Fabre J C. Intrusion tolerance in distributed computing systems [C] //Proc of the IEEE Symp on Security and Privacy. Piscataway, NJ: IEEE, 1991: 110—121

[12] Courtney T, Lyons J, Ramasamy H V, et al. Providing intrusion tolerance with ITUA [C] //Proc of the 2002 Int Conf on Dependable Systems and Networks. Piscataway, NJ: IEEE, 2002: c-5-1—c-5-3

[13] Wang Feiyi, Gong Fengmin, Sargor C, et al. SITAR: A scalable intrusion-tolerant architecture for distributed services [C] //Proc of the 2001 IEEE Workshop on Information Assurance and Security US Military Academy. Piscataway, NJ: IEEE, 2001: 38—45

[14] Barham P, Dragovic B, Fraser K, et al. Xen and the art of virtualization [C] //Proc of the 19th ACM Symp on Operating Systems Principles. New York: ACM, 2003: 164—177

[15] Castro M, Liskov B. Practical Byzantine fault tolerance and proactive recovery [J]. ACM Trans on Computer Systems, 2002, 20(4): 398—461

[16] Rajagopalan M, Hiltunen M A, Jim T, et al. System call monitoring using authenticated system calls [J]. IEEE Trans on Dependable and Secure Computing, 2006, 9(3): 216—229

[17] BackDoor Rootkit Tools [OL]. [2008-05-01]. http://www.antiserver.it/Backdoor-Rootkit

[18] The University of New Mexico. IMMUSE intrusion dataset [DB/OL]. [2008-05-01]. http://www.cs.unm.edu/~immsec/data

Zhao Feng, born in 1976. Postdoctoral fellow and associate professor in the School of Computer Science and Technology, Huazhong University of Science and Technology. Member of China Computer Federation. His current research interests include computer security, data mining, distributed and parallel computing, etc.

赵　峰, 1976 年生, 博士后, 副教授, 中国计算机学会会员, 主要研究方向为信息安全、数据挖掘、分布并行计算等.

Jin Hai, born in 1966. PhD, professor and PhD supervisor in the School of Computer Science and Technology, Huazhong University of Science and Technology. Fellow of China Computer Federation. His current research interests include computer system architecture, grid computing, computing system virtualization, etc.

金　海, 1966 年生, 博士, 教授, 博士生导师, 中国计算机学会常务理事, 主要研究方向为计算机系统结构、网格计算、计算系统虚拟化等.

Jin Li, born in 1978. PhD of in the School of Computer Science and Technology, Huazhong University of Science and Technology. Her current research interests include information security, etc.

金　莉, 1978 年生, 博士, 主要研究方向为信息安全等.

Yuan Pingpeng, born in 1972. Associate professor in the School of Computer Science and Technology, Huazhong University of Science and Technology. Senior member of China Computer Federation. His current research interests include semantic Web, grid computing, etc.

袁平鹏, 1972 年生, 博士, 副教授, 中国计算机学会高级会员, 主要研究方向为语义网、网格计算等.

Research Background

With the emergence of multi-core processor, virtualization technology have aroused attention and developed rapidly in recent years. Virtual computing environment based on virtual machine becomes a hot topic in the field of network computing. Virtual computing environment is open, complex and dynamic computing environment, which has brought new challenges to system security, especially to intrusion tolerance. In this paper, VFRS method is proposed in order to protect sensitive data from intrusion in virtual computing environment. Firstly, a probability computing model is constructed to present system call sequences and SCSFA algorithm is designed to predict the attempt of intrusion and to determine what need to protect, which is based on the analysis of system call sequence in virtual computing systems; secondly, the sensitive data protected are divided into a number of film data, and for the goals of random errors tolerance, each tablet data is redundant backup based on Byzantine fault tolerance; then, the redundant data are distributed to different virtual machines. The VFRS method can predict the anomaly intrusion and well tolerate the complicated errors in virtual computing environment. The VFRS method can help greatly to prevent the intrusion from generating a system failure, especially in virtual computing environment. Our work is supported by the National Natural Science Foundation of China under grant No. 60803114, the National 973 Basic Research Program of China under grant No. 2007CB310900.