私有云平台上的虚拟机进程安全检测*

曹立铭，赵逢禹
（上海理工大学 光电信息与计算机工程学院，上海 200090）

摘　要：针对网络防火墙在私有云平台安全防护上的单调与缺陷，提出了一种基于进程资源监控的安全监测方法（PAMon）。首先利用虚拟机监控器获取平台上虚拟机的物理资源信息；然后通过映射表重构进程资源信息；再对重构的进程信息从关键进程、进程隐藏和进程占用资源异常三方面分析恶意进程；最后对分析出的恶意进程进行了适当的处理。实验结果表明，PAMon 不仅可以有效地检测出恶意程序，而且反馈给防火墙的信息可以进一步增强网络防火墙的防御能力。

关键词：私有云安全；安全监测；进程资源；虚拟机
中图分类号：TP309　　**文献标志码**：A　　**文章编号**：1001-3695(2013)05-1495-05
doi:10.3969/j.issn.1001-3695.2013.05.055

Security detection of virtual machine process in private cloud platform

CAO Li-ming, ZHAO Feng-yu
(*School of Optical-Electrical & Computer Engineering*, *University of Shanghai for Science & Technology*, *Shanghai* 200090, *China*)

Abstract: In terms of the network firewall monotone and defects on the security of the private cloud platform, this paper proposed a method (process analysis monitor, PAMon) based on process resource monitoring. Firstly, the virtual machine monitored the physical resources of the virtual machine platform information. Secondly, it reconstructed process resource information through the mapping table. Thirdly, it identified malicious processes by analysis of the key executing processes, hidden ones and those abnormally occupied with resources. Finally, it disposed the identified malicious processes properly. Experiment shows that PAMon can not only detect malicious programs effectively, but also enhance the defense capacity of network firewall by submitting malicious process information to it.

Key words: private cloud security; security monitoring; process resource; virtual machine

0　引言

私有云由于其能够整合存储、网络、服务器等企业的资源，提供集中统一管理与高质量服务从而降低成本，吸引了公司和企业的青睐。IBM、微软、惠普、Sun 等公司都拥有自己的私有云平台，并且还为其他各类公司提供安装、配置和运营基础设施，以及支持公司企业数据中心防火墙内的专用云。在国内，许多公司和企业也选择了部署自己的私有云平台来增强自身的竞争能力。

在私有云上，多个操作系统或应用程序以虚拟机的形式同时部署在物理服务器上，这些虚拟机同时共享该服务器的硬件资源，虚拟机间的网络流量可以不被外部网络感知。图 1 是服务器虚拟化架构，平台的安全防范措施建立在与交换机相连的防火墙和 IDS 上，disk 为虚拟主机的硬盘，store 为平台的海量存储器，VM1 和 VM2 通过虚拟管理器 host 进行通信。从图 1 中可见，虚拟服务器 VM1 和 VM2 之间的通信以及虚拟机和磁盘等关键区域的通信没有任何的安全防范措施。这样一旦某个病毒采用欺骗技术通过防火墙并在虚拟机上执行时，所有兄弟虚拟机都会很快被感染；当某个特权虚拟机被感染后，病毒程序就会绕过特权虚拟机修改整个系统的核心代码以及关键组件的核心驱动程序，从而导致数据丢失、资源盗用，甚至整个系统的崩溃。由此可见，需要通过监控内部虚拟机的通信、执行状态、异常操作等来加强虚拟机的安全。

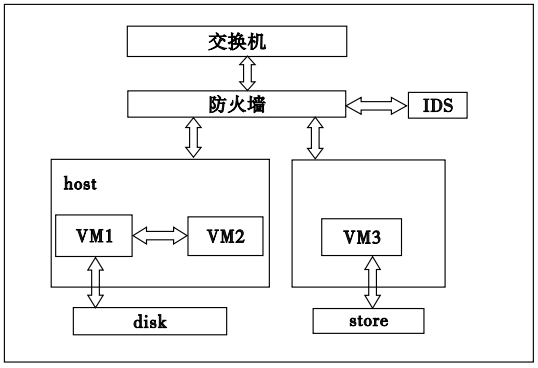

图1　私有云应用架构

瑞星 2007 年病毒统计分析报告[1]指出，病毒在入侵后会终止任务管理器、softice 等用来手工清除病毒的专业软件；大部分的病毒会启动多个耗费资源的进程、抢占系统运行资源，影响其他合法程序的正常运行；还有部分病毒通过设置 CPU 的优先级占用大量的 CPU 运行时间，使得 CPU 一直在运行恶意程序。微软公司发布的恶意代码清除工具（malicious software removal tool）统计[2]显示，超过 90% 的恶意软件使用了进程隐藏技术。

收稿日期：2012-08-17；**修回日期**：2012-09-29　　**基金项目**：国家自然科学基金委员会与中国民用航空局联合资助项目(60979011)

作者简介：曹立铭(1988-)，男，硕士研究生，主要研究方向为云计算与虚拟机安全(clm238@126.com)；赵逢禹(1963-)，男，教授，主要研究方向为 Web 服务与安全、软件工程与软件质量控制.

本文基于上面提出的问题和病毒的这些特点,提出了一个针对私有云平台上虚拟主机的基于进程的安全检测方法 PAMon。通过检测隐藏进程、分析抢占资源的进程、验证关键程序的运行状态,可以有效地检测出运行在虚拟机上的恶意程序。利用资源监控和多视图对比验证的方法,检测出被监控虚拟机上的隐藏进程与占用大量资源的进程、验证关键进程是否在虚拟机上正常地运行,然后通过综合分析得出是否有恶意程序运行在虚拟机上;并对虚拟机或虚拟机上的进程进行处理,然后把该进程的病毒文件特征提交给防火墙 IDS,帮助加强防火墙的防御能力。整个的安全防范模块安装在虚拟机管理层 VMM,PAMon 的分析数据通过 VMM 的监控功能获取,对虚拟机的操作处理也都由 VMM 来执行,减少了对虚拟系统的依赖,有助于提高检测结果的可靠性和检测机制的安全性。

1 相关工作

由于应用的范围广泛和使用者的复杂性,云的安全已经吸引了社会和学术界的强烈关注。在文献[3]中,Pearson 对云计算的隐私问题作了深入的描述和讨论。2009 年 Cachin 等人[4]中对云存储服务环境下的安全进行了完整的调查。ENISA[5]提出了一套详尽的云安全风险的评估方案。2010 年 Lombardi 等人[6]提出了一种加强云计算安全的方法 ACPS,其使用虚拟化技术加强云计算的安全,通过对关键数据和关键组件驱动的监控和完整性验证,主要实现了保护虚拟机和云基础设备组件的安全。2011 年 Mondol[7]提出了使用 FGPA 的云计算安全解决方案,并提出了四种类型的云安全解决方案,即可信任的云平台、安全组内用户数据协同、数据安全、可信认证,通过把四种方案应用到硬件上,实现了用户的验证和数据的安全。

随着虚拟化技术的不断成熟,私有云上面的虚拟机安全检测已成为人们研究的热点问题。传统的安全检测系统[8]运行在被监控主机内部,可以轻松地获取能被操作系统识别的高级语义信息,通过比较真实进程队列和可疑进程队列,能够容易地分析出系统中的隐藏进程;但直接操作内核对象(direct kernel object manipulation,DKOM)[9]类攻击可以将待隐藏进程控制块从进程队列中摘链,因而该方法可能会出现漏检现象;而且它也没有提供相应的隐藏进程的处理机制,需要配合其他病毒查杀工具一起工作。为了提高检测结果的准确性,虚拟机监控器为增强系统的安全提供了新的研究方向,通过它提供的隔离机制,可以有效地保证检测系统的完整性和检测结果的可靠性。但是这种隔离机制却给研究带来了语义断层问题;传统的虚拟机内部检测系统可以轻易地获取进程、磁盘上文件等操作系统级别的高级语义信息,而云平台上虚拟机监控器(virtual machine monitor,VMM)位于被监控系统外部,检测系统仅仅能获取寄存器值、内存数据、磁盘块数据或当前指令流等低级信息,严重影响了检测功能的有效实施。基于这个问题,AntFarm[10]、XenAccess[11]、VMwatcher[12]从不同角度去重构虚拟机内部的高级语义信息。AntFarm 跟踪虚拟机内部的生命周期(包括进程的创建、切换以及销毁等);XenAccess 和 VMwatcher 从信息的刻画出发,重构进程控制块信息,进而获取进程名、进程号以及内存地址空间等。实际上,虚拟机的安全可以从资源利用、异常操作、数据通信、关键区域数据和文件的完整性验证几个方面进行研究。王丽娜等人[13]提出了一种虚拟隐藏进程的检测方法,利用虚拟机自省机制获取被监控主机的底层状态信息,借助语义视图重构技术重构其进程队列,然后通过交叉视图的方式比较各进程队列间的差异,从而确定隐藏进程,最后还把隐藏进程详细汇报给相应的处理机制,并提供终止和挂起隐藏进程的功能。但其也只是对进程的隐藏进行了分析,对进程的其他方面没有进行分析,所以只能分析出带有隐藏进程功能的可疑进程;而且恶意程序一般会设置自身随着系统的启动自动运行,简单地挂起或者终止进程也不能保证系统在下次重启后不再继续运行。因此,该方法对恶意进程的处理还不够彻底,还需要进行更深入的研究。本文也是通过 AntForm、XenAccess、VMwatcher 的技术来获取分析所需要的真实进程队列和进程资源等信息,然后通过分析这些信息来确认虚拟机的安全状况。

2 基于进程资源监控的检测模型设计与实现

在云平台上,虚拟机监视器(VMM)具有高特权、并对底层硬件资源的高可控性,存储并管理虚拟机运行时所有对软硬件资源的使用情况。本文假定 VMM 是安全可信的,对 VMM 受到攻击的问题暂时不进行讨论。本文把讨论的重点放到对云平台虚拟机进程信息的监测分析上。PAMon 的处理过程如下:

a)检测系统在 VMM 层通过 hook 函数获取寄存器值、内存数据、磁盘块数据或当前指令执行流等信息,然后通过语义信息重构得到完整的操作系统能识别的高级系统语义信息。这里主要构造两个队列:真实进程队列 true_List 和进程使用 CPU 记录队列 per_cpu_list。

b)通过对被监控虚拟机提供的进程队列、语义重构模块重构的真实进程队列、每个 CPU 正在执行的进程队列进行多视图对比的方法,检测出隐藏进程队列。

c)通过检测虚拟机提供的运行进程队列查看错误报告、系统调试工具程序(softice)、任务管理器等关键进程是否正常运行。

d)通过统计分析重构出来的真实进程队列中进程使用 CPU、内存等资源的信息,检测出大量占用资源的进程队列。

e)再综合分析隐藏进程、大量占用资源进程队列以及关键进程验证结果,得出运行在虚拟机上的恶意进程和可疑进程。

f)对分析结果进行处理,把恶意、可疑进程信息反馈给防火墙,同时终止、挂起进程或迁移虚拟机数据并重建虚拟机。

PAMon 的系统架构如图 2 所示,它部署于特权管理区域中,主要由六个模块组成。语义重构模块负责重构能被操作系统等识别的文件和进程等上层语义信息;隐藏进程检测模块负责检测被监控虚拟机中是否运行有隐藏进程;关键进程验证模块负责验证虚拟机上的关键程序是否在正常运行;资源使用率分析模块负责检测是否存在大量占用服务器 CPU、内存的进程;综合分析模块接收关键进程检测模块、隐藏进程检测模块、资源使用率分析模块得到的结构数据,然后审核是否存在恶意进程或软件;操作响应模块根据管理员的命令作出响应(终止进程、挂起进程、虚拟机资源迁移、关闭虚拟机等)。

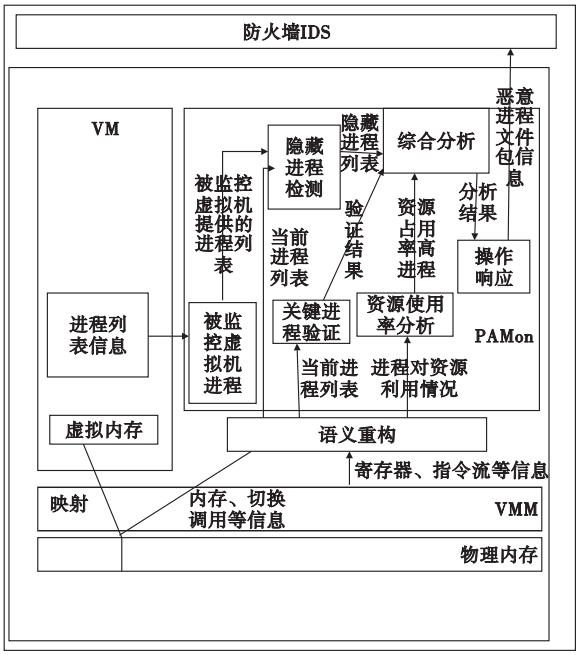

图2　PAMon系统架构

2.1　语义重构

PAMon 首先利用 hook 函数获取寄存器值、物理内存数据、磁盘块数据或当前指令执行流等低等级信息,并通过 VMM 来监控和记录 CR3 寄存器信息的更换时间;然后 VMM 把获取到的信息发送给语义重构模块。语义重构模块利用 Xen 提供的物理地址和虚拟地址的相互转换记录表以及底层对进程对建立和销毁指令信息流,分析出进程的建立和注销时间以及进程对内存的使用信息,从而建立真实进程队列 true_list;根据执行指令的 CPU 和 Xen 的地址转换表建立每个 CPU 的可运行队列 per_cpu_list;根据进程 CPU 状态与 CR3 寄存器信息的对应关系,通过分析 CR3 寄存器信息的更换来提取进程的切换时间,建立进程占用 CPU 的记录队列 per_cpu_list。重构出的真实进程队列、进程占用 CPU 的记录队列和 CPU 可运行队列,提供给关键进程验证、隐藏进程检测、资源使用率分析模块进行下一步的分析。由于寄存器值、物理内存数据等信息的获取没有依赖被监控的虚拟机,因此对被监控虚拟机是透明的。真实进程信息 true_process 和进程对 CPU 使用记录节点 CPU_user_record 的定义如下:

```
struct True_Process {
    char ProcessName[30];//进程名称
    long BasePriority;//进程优先级
    unsigned long ProcessId; //进程号
    unsigned long gpdaddr;//进程描述地址
    unsigned long fileaddr;//程序文件地址
    __64 Start_Time;//第一次执行时间
    __64 Last_Time;//最后一次执行时间
    unsigned long MemorySize//内存占用大小
};
struct CPU_User_Record{
    unsigned long ProcessId;//进程号
    __64 Start_Time;//开始执行时间
    __64 End_Time;//结束执行时间
    long BasePriority;//执行时的优先级
}
```

2.2　隐藏进程检测

目前隐藏进程的方法主要有三种:a)劫持系统调用(system call hijacking,SCH);b)内核对象劫持(kernel object hooking,KOH);c)直接操纵内核对象(direct kernel object manipulation,DKOM)。前两种属于劫持系统内核的函数指针,后一种属于直接修改系统内核的数据结构。然而,在隐藏进程通过系统调用获取 CPU 执行权时,都能被 VMM 监测到。

在虚拟机系统中维护着一个双向进程队列 task_list,以系统启动的第一个进程为首元素。同时 VMM 维持着一个系统运行的真实队列 true_list 和为每个 CPU 维持着一个可运行队列 per_cpu_list,任何一个可运行的进程都只属于一个可运行列表中。真实进程队列 true_list 和 CPU 的执行队列 per_cpu_list 由语义信息重构模块,通过分析底层寄存器等资源访问信息得到。

PAMon 通过进程队列的多视图对比得出隐藏进程,确定隐藏进程的方法为:

a)对所有的进程控制块 P,如果 P 出现在 true_list 队列中,未出现在虚拟机自己报告的进程队列 task_list 中,则 P 属于隐藏进程 hideprocess,即

$$\forall P(P \in \text{true\_list} \wedge P \notin \text{task\_list}) \Rightarrow P \in \{\text{hideprocess}\}$$

b)对所有的进程控制块 P,如果 P 出现在 per_cpu_list 中,未出现在 true_list 中,则 P 属于隐藏进程 hideprocess,即

$$\forall P(P \in \text{per\_cup\_list} \wedge P \notin \text{true\_list}) \Rightarrow P \in \{\text{hideprocess}\}$$

2.3　关键进程验证

每一台虚拟机系统上都运行 softice、错误提交、任务管理器、信息反馈、自动更新等维护系统安全运行的关键程序;这一个模块主要是根据语义重构模块重构出来的真实进程队列 True_list,检测这些关键程序是否出现在被监控的虚拟机的真实进程列表中,并把结果保存到变量 IsIntegrated 中。一旦发现某些关键进程不存在于真实进程队列中,说明被监控虚拟机已经被病毒感染,并且病毒已经终止了这些安全进程的运行;那么就把没有找到的关键进程保存到 KeyFile 中,并把 IsIntegrated 设置为 false,否则 IsIntegrated 设置为 true。然后再通过和其他两模块进行综合分析,确定恶意进程或者程序。

2.4　资源使用率分析

语义重构模块通过对 VMM 获取的寄存器、内存等信息,重构系统执行的高级语义信息,并记录下进程每一次对资源的占用时间和占用空间的大小。然后由资源使用分析率模块进行统计分析,总结出除了系统主进程之外对 CPU 占用率很高的或者内存占用很大的进程。统计分析按照一定时间段内(本文设置为 1 000 个 CPU 周期)的每一个进程占用百分比来确定,CPU 的判定由每一个进程的执行时间比例来决定。

进程占用 CPU 的比例为

$$\text{ProExec} = \frac{\text{ExecTime}}{(\text{LastTime} - \text{FirstTime})}$$

其中:ExecTime 为进程的累计执行时间,LastTime 为进程最后一次执行的时间,FirstTime 为进程第一次执行的时间。

大资源进程(LargeSourceProcess):

判定规则 1　对于进程 P 使得 ProExec > m,其中,m 为

CPU 使用比例的阈值,由实验 1 统计分析获得。

判定规则 2 多个同名的进程 P 同时属于一个程序 Program;那么进程 P 属于大资源进程。

所有的大资源进程组成一个大资源占用队列 lager_source_list,大资源进程列表最终提交给综合分析模块。

2.5 综合分析模块

综合分析模块接收隐藏进程检测模块检测出来的隐藏进程队列 HideProcessList、关键进程验证结果 IsIntegrated、资源占用分析的大资源进程队列 lager_source_list;然后结合这三方面的结果进行分析判断,并把分析结果报告给响应机制。恶意进程最终分析为可疑进程和恶意进程两种,分析的原理如下:

可疑分析规则:

a)关键进程验证证明部分关键进程已经终止或者破坏(即 IsIntegrated = false),那么确定该虚拟机上已经存在了威胁进程或软件;

b)如果进程 P 属于隐藏进程或者是占用大量资源的进程,那么,进程 P 就被确认为可疑进程,即

$$\forall P \in (\text{HideProcessList} \vee \text{lager\_sourrce\_list}) \Rightarrow P \in \{\text{SuspiciousProcess}\}$$

恶意进程确定规则:

a)同时满足可疑规则 a)和 b)时,则确定进程为恶意进程,即

$$((\forall P \in (\text{HideProcessList} \vee \text{lager\_source\_list}) \wedge !\,\text{IsIntegrated})) \Rightarrow P \in \{\text{MaliciousProcess}\}$$

b)存在进程控制块 P 同属于隐藏进程和大资源占用进程,那么这一进程确定为恶意进程,即

$$\forall P(P \in \text{HideProcessList} \wedge P \in \text{lager\_source\_list}) \Rightarrow P \in \{\text{MaliciousProcess}\}$$

2.6 响应机制

接收综合分析模块的分析结果,并对恶意进程和可疑进程进行处理。PAMon 提供了两种处理模式,即报告模式和强制处理模式。

当检测到的是可疑进程时,采用报告模式,把进程的详细信息(包括进程号、进程名、内存、CPU 占用率、网络端口、源文件等)提交给防火墙重新作进一步分析检测。

当检测到的是恶意进程时采用报告和强制处理结合的方法进行处理,同样把进程的详细信息反馈给防火墙,更新防火墙病毒特征库,下次再有携带类似恶意程序的数据包经过时,进行直接的过滤;同时对关键进程完好的虚拟机上的恶意进程实行终止操作,结束恶意进程;操作人员在查看到结果后对关键进程已经不完整的虚拟机进程数据进行迁移,终止并重建新的虚拟机来执行任务。

3 实验与结果

为了验证系统的可行性,本文借助于 Xen 利用 eucalyptus 技术搭建小型私有云,并在其上实现了 PAMon 系统。云平台使用 Ubuntu10.04 自带的云构建工具建立,包含一个云控制器 cloud control、一个簇控制器 cluster control 和两个云节点 node control 组成。在 eucalyptus 平台上创建了两个虚拟服务器(一个提供 Linux 服务,另一个提供 Windows 服务)和一个装有 PAMon 系统的虚拟机,每一个虚拟服务器分配 1 GB 的虚拟内存。总的拓扑结构如图 3 所示。

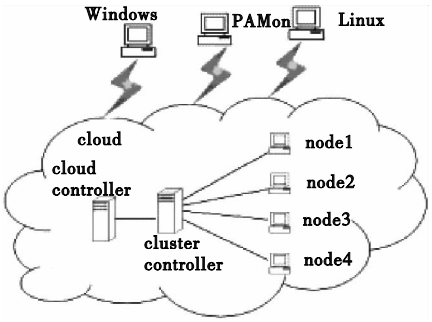

图3 系统拓扑结构

实验 1 为了获取到 CPU 使用率的阈值,本文在检测之前对部分比较占用主机资源的病毒进行了分析,分别在三台电脑上依次运行了 Backdoor. Agobot. bcl、Trojan. PSW. Maple. a、Backdoor. IRCBot. bay、Worm. Viking. bo 四个病毒和比较占用资源的安全程序 QQ,它们的平均占用 CPU 比例的统计如表 1 所示。表 1 显示四个病毒中 CPU 占用时间比最低的是 Worm. Viking. bo 病毒,最小占用比例为 69%,而安全程序 QQ 的最大占用比例为 23%。因此,为了减少漏检现象和把安全程序误检成可疑程序的情况,本文设定 2.4 节中大资源评定的阈值 m 为 60%。

表 1 病毒占用 CPU 时间比例 /%

病毒	在电脑 A 占用比例	在电脑 B 占用比例	在电脑 C 占用比例	平均占用比例
Backdoor. Agobot. bcl	100	96	97	97.7
Trojan. PSW. Maple. a	81	76	73	76.7
Backdoor. IRCBot. bay	85	90	78	94.3
Worm. Viking. bo	69	76	70	71.7
QQ	23	8	9	13.3

实验 2 先运行 PAMon 系统,检测程序自动运行,并且把检测的结果保存到系统根目录下;然后创建 Windows 服务,本文在 Windows 服务虚拟机上下载并运行了“麦托变种 XP”病毒 MyTopXPS. exe;结果 Windows 服务器虚拟机的运行变得十分缓慢而且出现了部分操作无法执行。在 PAMon 系统上运行监控结果显示程序,出现的信息如图 4 所示。

PAMon 检测后,在图 4(c)大资源进程检测信息模块可以看到程序 MyTopXP. exe 同时出现了多个进程;在图 4(b)关键进程检测信息模块中检测到关键进程 wuauclt. exe/lsass. exe 没有出现在被监控的虚拟机上;在图 4(d)分析处理结果模块中 PAMon 判定 MyTopXP. exe 为病毒程序并终止了所有的 MyTopXP. exe 进程并提交病毒的特征给防火墙。这个实验说明,PAMon 可以有效地检测出 Windows 上破坏关键进程和占用大量资源的程序,并且能对程序作出相应的处理。

实验 3 在实验 1 的基础上,再创建一个 Linux 平台系统,并在系统上安装一个 backdoor 程序,backdoor 采用 adore-ng rootKit 技术[14]隐藏自己的进程信息,使得操作系统检测不到 backdoor 进程信息;利用 ELF 文件感染器(ELF infector)[15]技术对 ELF 文件进行感染,运行 backdoor 程序后,backdoor 程序自动查找到虚拟机上的 ELF 文件,并把自身的代码插入到符合条件的 ELF 文本段尾,ELF 文件在执行时会先执行被插入

的病毒程序,然后反复循环。PAMon 的检测结果如图 5 所示。

(a) 隐藏进程检测信息　　(b) 关键进程检测信息

(c) 占用大量资源的进程检测信息　　(d) 分析和处理结果

图4　Windows系统监控

(a) 隐藏进程检测信息　　(b) 关键进程检测信息

(c) 占用大量资源的进程检测信息　　(d) 分析和处理结果

图5　Linux系统监控

从图 5 的结果可以看到,在图 5(a)隐藏进程信息模块的虚拟机中出现了多个隐藏进程 backdoor;在图 5(c)占用大资源队列信息模块中检测到这些进程在不断地运行抢占资源;在图 5(d)分析和处理结果模块,通过 PAMon 的综合分析,系统挂起了所有的 backdoor 进程,并把进程信息提交给了与外部网络交互的防火墙。因此,这个实验证实了 PAMon 系统可以检测到 Linux 上的隐藏进程和占用大量资源的恶意程序。

通过以上实验可以验证,PAMon 检测系统可以有效地从关键进程、进程隐藏、进程占用大量资源三个方面综合分析出运行在被监控的虚拟机上的恶意执行软件,比文献[13]等提出的通过检测隐藏进程来检测运行在被监控器上的恶意软件的方法对进程的研究更加全面,分析得到的结果在误判率方面要低。在对分析结果的处理上,对于恶意程序的处理要比文献[13]等更加彻底,防止了恶意程序在下一次系统启动或者其他时间再一次启动而对系统再一次产生威胁;而且把恶意和可疑进程的信息特征发送给网络防火墙,加强了防火墙的防御能力,阻止了相同的恶意程序再一次入侵,减少了内部检测系统的工作量。PAMon 检测系统在对真实队列建立时,依赖的重构信息是 VMM 利用 hook 函数直接获取的真实物理操作信息,不依赖于被监控虚拟机提供的任何信息,因此比文献[8]等利用被监控虚拟机提供的信息进行分析的结果更加可信。

4　结束语

基于云计算的服务现在已经被广泛地应用到了各个领域,私有云也被广大的企业所应用;而对于云上面的安全问题,大部分的企业和学者都把重点放到了公共云的安全上面,而忽略了私有云上面的安全问题。私有云的安全问题包括了私有云文件、数据、资源、通信等的安全。本文通过分析虚拟机上程序进程的状态,提出了基于进程资源监控的检测模型 PAMon;通过分析进程对资源占用、隐藏状态、破坏关键进程等方式,成功地实现了私有云平台上的有效安全检测,并且反馈信息到外部防火墙,进一步帮助防火墙提高了对整个系统的安全防护。

参考文献:

[1] 瑞星电脑病毒统计[EB/OL]. [2008-03-05]. http://www.rising.com.cn/2007/annual/index.htm.

[2] Microsoft. Windows malicious software removal tool[EB/OL]. [2007-12-20]. http://www.microsoft.corn/seeurity/malwareremoge/.

[3] PEARSON S. Taking account of privacy when designing cloud computing services[C]//Proc of ICSE Workshop on Software Engineering Challenges of Cloud Computing. Washington DC:IEEE Computer Society,2009:44-52.

[4] CACHIN C, KEIDAR I, SHRAER A. Trusting the cloud[J]. SIGACT News,2009,40(2):81-86.

[5] ENISA. Cloud computing risk assessment[EB/OL]. (2009). http://www.enisa.europa.eu/act/rm/files/ deliverables.

[6] LOMBARDI F, Di PIETRO R. Secure virtualization for cloud computing[J]. Journal of Network and Computer Applications,2010,34(4):1113-1122.

[7] MONDOL J A M. Cloud security solutions using FPGA[C]//Proc of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing. 2011:747-752.

[8] BAI Guang-dong,GUO Yao,CHEN Xiong-qun. Windows rootkit detection method based on cross-view[J]. Computer Science,2009,36(8):133-137.

[9] FUTo uninformed [EB/OL]. [2010-12-10]. http://uniformed.Org/? v=3&a=7&t=sumry.

[10] JONES S T, ARPACI-DUSSEAU A C, ARPACI-DUSSEAU R H. Antfarm:tracking processes in a virtual machine environment[C]// Proc of Annual USENIX Technical Conference. Berkeley, CA: USENIX,2008:1-14.

[11] PAYNE B D, De CARBONE M D P, LEE W. Secure and flexible monitoring of virtual machines[C]//Proc of the 23rd Annual Computer Security Applications Conference. Piscataway, NJ: IEEE Press, 2007:385-397.

[12] JIANG Xu-xian, WANG Xin-yuan, XU Dong-yan. Stealthy malware detection through VMM-based "out-of-the-box" semantic view reconstruction[C]//Proc of the 14th ACM Conference on Computer and Communication Security. New York:ACM Press,2007:128-138.

[13] 王丽娜,高汉军,刘炜,等. 利用虚拟机监视器检测及管理隐藏进程[J]. 计算机研究与发展,2011,48(8):1534-1541.

[14] The adore-ng rootkit[EB/OL]. [2012-02-23]. http://stealth.openwall.net/rootkits/.

[15] ELF infector[EB/OL]. [2004-12-13]. http://www.linuxforum.net/forum/showflat.php? Cat=&Board=security &Number=531289&page=0&view=collapsed&sb=5&fpart=.