基于虚拟机自省的隐藏文件检测方法①

乌 云[1], 李 平[2,3], 李勇钢[2,3]

[1](中国科学院合肥物质科学研究院 应用技术研究所, 合肥 230088)
[2](中国科学院 合肥智能机械研究所, 合肥 230031)
[3](中国科学技术大学 自动化系, 合肥 230027)

摘 要: 通过检测虚拟机内部的隐藏文件, 检测工具可以及时判断虚拟机是否受到攻击. 传统的文件检测工具驻留在被监视虚拟机中, 容易遭到恶意软件的攻击. 基于虚拟机自省原理, 设计并实现一种模块化的虚拟机文件检测方法 FDM. FDM 借助操作系统内核知识, 解析虚拟机所依存的物理硬件, 构建虚拟机文件语义视图, 并通过与内部文件列表比较来发现隐藏文件. FDM 将硬件状态解析和操作系统语义信息获取以不同模块实现, 不仅具备虚拟机自省技术的抗干扰性, 还具备模块化架构的可移植性与高效性. 实验结果表明, FDM 能够准确快速地检测出虚拟机内部的隐藏文件.

关键词: 虚拟机自省; 文件检测; 隐藏文件

Method of Hidden File Detection Based on Virtual Machine Introspection

WU Yun[1], LI Ping[2,3], LI Yong-Gang[2,3]

[1](Institute of Applied Technology, Chinese Academy of Sciences, Hefei 230088, China)
[2](Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, China)
[3](Department of Automation, University of Science and Technology of China, Hefei 230027, China)

Abstract: The detection tools can judge whether the virtual machine is under attack or not through detecting the hidden files. The traditional file detection tools reside in the monitored virtual machine, which are vulnerable to attack by the malicious software. According to the virtual machine introspection, a modularized virtual machine file detection method (FDM) is designed and implemented. With the operating system kernel knowledge, FDM can parse the physical hardware and build the semantic view of the files. Then FDM can identify the hidden files by comparing with the internal file list. Meanwhile, parsing hardware status and obtaining semantic information are implemented in different modules. FDM has not only the tamper resistance of the virtual machine introspection, also has a modular architecture, portability and efficiency The experimental results show that the FDM can quickly and accurately detect the hidden filesinside virtual machine.

Key words: virtual machine introspection; file detection; hidden file

随着互联网技术的发展, 网络安全备受人们的关注. 中国互联网络信息中心 2013 年发布的《2013 年中国网民信息安全状况研究报告》[1]显示: 有 74.1%的网民在过去半年内遇到过信息安全问题, 总人数达 4.38 亿. 其中网购、浏览网页和即时通信时遇到病毒或木马的比例分别为 22.6%, 70.9%和 40.6%. 由此可见, 计算机病毒、木马等恶意代码已经成为威胁互联网安全的重大隐患.

云计算对计算机资源的利用主要通过虚拟机的形式实现. 虚拟机作为具备完整操作系统和应用服务的个体, 同样面临着来自恶意软件的威胁. 例如, Rootkit 是攻击者向计算机系统植入的, 能够隐藏自身踪迹并

① 基金项目:中国科学院合肥物质科学研究院院长基金(YZJJ201329)
收稿时间:2015-05-03;收到修改稿时间:2015-06-15

且保留超级用户访问权限的恶意代码[2]. Rootkit 能够隐藏文件, 使其不被操作系统甚至杀毒软件所发现. 因此, 快速准确地检测出被恶意代码隐藏的文件对于提高计算机安全有着重要的意义.

传统的入侵检测工具可以用在虚拟机安全检测上. Tripwire[3]是一个典型的使用完整性检测方法来检测文件系统安全性的例子. 首先, Tripwire 为需要监视的文件建立一个完整性特征数据库. 然后, 在不同的时间点比较文件的特征数据库. 最后, 判断文件是否被篡改. Tripwire 是一个基于主机的入侵检测系统. 当恶意软件控制主机后, Tripwire 很容易被攻击者篡改. 刘传[4]等人提出了基于 VFS(Virtual Filesystem)后门技术的文件隐藏方法, 通过对 VFS 层进行攻击, 重定向 readdir、filldir 等操作函数来实现文件隐藏. 文中虽然提到利用一些工具直接去读取磁盘上的内容, 然后与经过 VFS 接口去读的内容相对比, 就可能发现文件是被隐藏的. 由于 VFS 后门与操作系统内核源代码紧密相关, 而操作系统内核版本很多, 因此可移植性很差.

虚拟机自省机制[5]是一项用于监视和获取 VM 内部状态的技术, 广泛应用于入侵检测、隐藏文件检测和恶意软件分析等领域. 主要用于解决虚拟机监视器和内部操作系统之间的语义鸿沟问题, 利用操作系统的内核数据结构特征来为虚拟机监视器提供语义信息.

Livewire[6]首次提出了虚拟机自省技术, 它本身并不主动截获客户操作系统内的相关操作, 而是周期性检查操作系统内的变化, 对于文件系统而言, 就是检查系统文件是否被恶意更改. 但是, Livewire 并没有对隐藏文件进行相关研究. VMWatcher[7]采用虚拟机自省技术, 在虚拟机外部对虚拟机进行监测. VMWatcher 从内核源码来获取信息, 解决了语义鸿沟问题, 构建出虚拟机的视图信息, 通过信息对比来发现隐藏的病毒. 但是, VMWatcher 需要在线解析各种类型 OS 和各种版本内核信息, 通用性不强.

因此, 本文在兼顾安全性、通用性的前提下, 提出了一种内核语义解析和语义视图重构的任务分离式的文件检测方法 FDM. FDM 能够为虚拟机提供主动的、透明的和实时的文件列表检测. FDM 采用虚拟机自省技术在虚拟机外部监视虚拟机物理内存而不在虚拟机内部安装任何监控代码. 最后, 在 Xen 虚拟化平台上实现了 FDM 的系统原型, 并通过实验验证了 FDM 的有效性和可行性.

1 背景

1.1 Linux 文件管理

计算机中所存储的大量程序和数据都是通过文件系统来组织和管理的. 从用户的观点来看, 文件被组织在一个树结构的命名空间中[8]. 在 Linux 中, 每个目录被看作一个文件, 可以包含若干文件和其他的子目录. 一旦目录项被读入内存, 虚拟文件系统就把它转换成基于 dentry 结构体的一个目录项对象, dentry 结构体包含了许多和文件描述有关的成员变量, 如 d_parent、d_iname 和 d_subdirs. 图 1 是 Linux 目录树对应的数据结构示意图. 父目录通过 d_subdirs 指向子目录链表, 然后子目录通过 d_child 连接起来. 子目录通过 d_parent 指针指向其父目录. 整个目录依此层次结构构成 Linux 的文件系统.

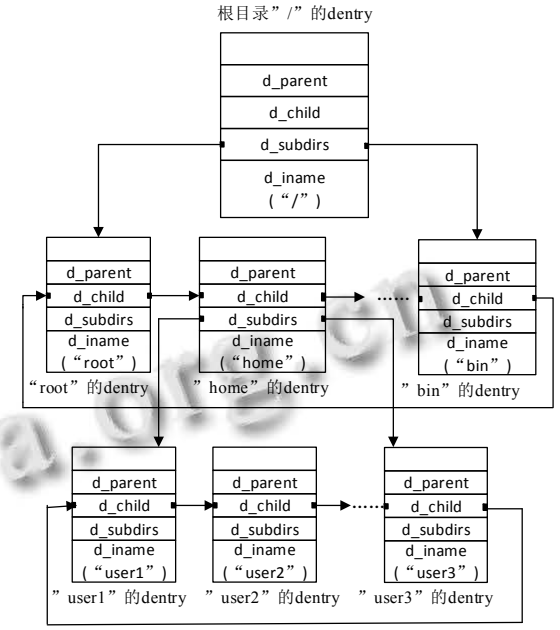

图 1　目录树的数据结构示意图

1.2 Xen 环境下内存管理

Xen[9]是剑桥大学开发的一个运行在 X86 架构上的虚拟机监视器. 典型的 Xen 虚拟化环境主要由机器硬件、Xen 虚拟机监视器(Xen Hy-pervisor)、Dom0(特权域)和 DomU(guest OS)组成[10]. 虚拟机监视器位于硬件和操作系统之间, 向虚拟机提供虚拟硬件资源, 同时分配和管理这些资源, 并保证虚拟机之间的相互隔离. 在 Xen 中使用影子页表[11]技术直接实现客户机

虚拟地址到宿主机机器地址的转换, 提高了机器内存的利用率.

2 系统实现

FDM 采用虚拟机自省技术监视虚拟机的物理内存. 它能够在虚拟机外部重构虚拟机内核数据结构的高级语义视图. 因此, 它不需要在客户虚拟机内部安装任何钩子函数, 与客户虚拟机具有很好的隔离性, 能够很好的避免恶意软件的攻击, 提高了检测系统的安全性. FDM 由前端模块和后端模块两个部分组成. 后端模块负责解析操作系统内核数据结构的语义信息, 前端模块负责完成“init”进程定位和文件语义视图构建. FDM 系统架构如图 2 所示.

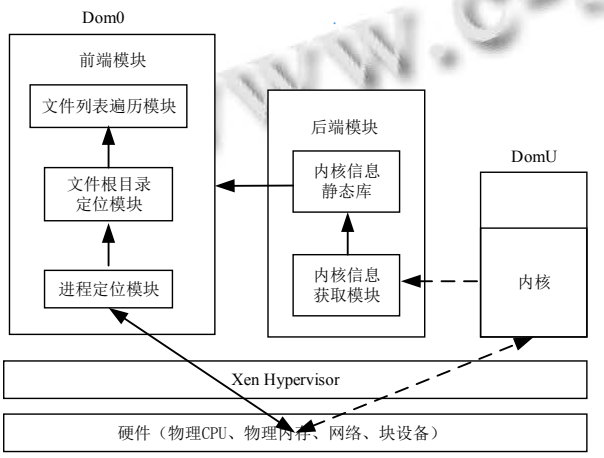

图 2 FDM 系统架构

2.1 FDM 前端模块

FDM 的前端模块由进程定位模块, 文件根目录定位模块和文件列表遍历模块三个部分组成. 前端模块通过调用后端模块提供的内核信息静态库来构建高级语义视图.

进程定位模块主要用来定位 init 进程的地址. 首先, 通过虚拟机监视器读取 ESP 寄存器的信息, 并进行相关运算得到虚拟机中当前进程的 thread_info 结构体的地址. 然后调用后端的内核信息静态库通过 thread_info 的地址来获取进程描述符 task_struct 的地址. 由于所有的进程描述符 task_struct 都是通过双向循环链表连接在一起的, 因此我们通过遍历双向循环链表来定位到 init 进程的地址, 即进程描述符中 comm 项为“init”的 task_struct 地址.

文件根目录定位模块用于定位 init 进程的可执行文件的根目录的位置, 即“/”的地址. 在前面获取到 init 进程的 task_struct 结构体的地址之后, 调用后端的内核信息静态库并使用影子页表技术通过 task_struct->mm_struct->vm_area_struct->file->path->dentry->d_iname 来获取文件名. 获取 init 进程文件名的过程就是获取 dentry 结构体地址的过程, 然后再次调用后端的内核信息静态库通过 dentry 结构体的 d_parent 成员变量来得到当前文件的父目录(dentry 结构体), 依次向上追溯, 直到找到根目录“/”的地址 root_addr 为止. 从进程定位到根目录定位的整个过程如图 3 所示.

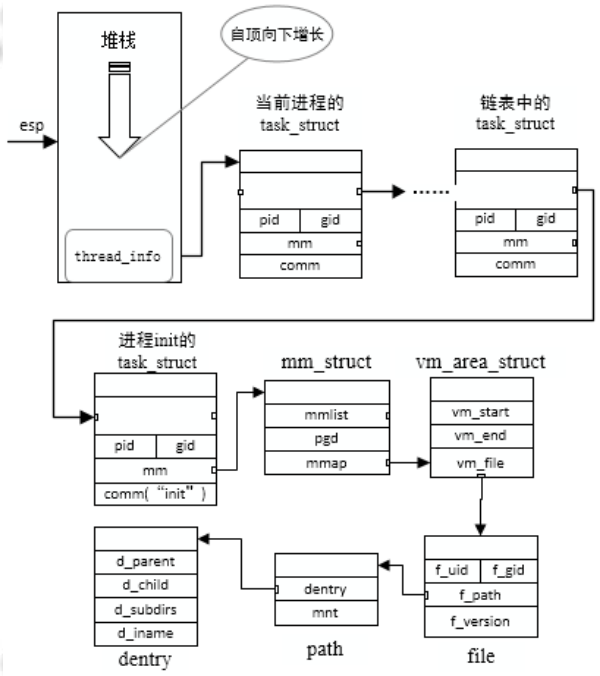

图 3 根目录定位过程

文件遍历模块用于从“/”目录开始查找到所需遍历的目录(n 级目录, 如“/usr/tmp”为三级目录, 依次为“/”, “usr”和“tmp”)并进行遍历. 首先, 从根目录开始采用广度优先搜索方法来定位到需要遍历的目录位置. 然后通过 dentry 结构体的 d_u 成员变量来遍历此目录下的所有文件. 具体步骤如下:

(1) 调用后端的 getd_subdirsoffset 函数和 getd_uoffset 函数获得 dentry 结构体中 d_subdirs 和 d_u 的偏移量 o_d_subdirs 和 o_d_u. 利用影子页表 SPT(root_addr, o_d_subdirs)得到 d_subdirs 的宿主机机器地址. 解析该地址中的内容, 并根据 o_d_u 的值得到子目录结构体 dentry 的地址 dentry_addr.

(2)调用后端的 getd_inameoffset 函数获得 dentry 结构体中 d_iname 的偏移量 o_d_iname. 利用影子页表 SPT(dentry_addr, o_d_iname)得到 d_iname 的宿主机机器地址, 解析该地址里的内容得到文件名称. 若此名称和二级目录名称一致, 则执行步骤(3), 否则调用后端的 getd_uoffset 函数获得 dentry 结构体中 d_u 的偏移量 o_d_u. 利用影子页表 SPT(dentry_addr, o_d_u)获得 d_u 的宿主机机器地址, 解析该地址里的内容并根据 o_d_u 值得到父目录中下一个文件的地址 dentry_addr, 重复执行步骤(2).

(3)再以二级目录作为根目录, 从步骤(1)开始依次向下执行, 直到各级目录项全部匹配为止.

(4)输出最后一级目录下的所有子目录名及文件名.

2.2 FDM 后端模块

FDM 的后端模块主要由内核信息获取模块和内核信息静态库构成. 内核信息获取模块需要解析各种客户虚拟机操作系统的内核数据结构, 获取 task_struct、dentry 和 file 等内核信息, 生成一系列内核信息反馈接口函数, 构建内核信息静态库. 由于整个过程是在离线状态下完成的, 所以后端模块完成时间对前端模块在线执行时间几乎没有影响.

内核信息静态库主要是为前端模块提供语义重构时所需的语义信息. 从进程定位到根目录定位, 再到文件信息获取的整个过程都需要后端模块提供相应内核数据结构的语义信息解析. 当需要获取某一文件名称时, dom0 首先从 domU 的内存中读取 dentry 结构体的数据信息. 由于不同虚拟机之间的内存隔离, dom0 无法直接获取 domU 的内存信息, 需要通过影子页表技术将 domU 的内存映射到 dom0 中. 这时 dom0 中获取到的内存信息只是底层视图中的“0”和“1”. 然后根据后端提供的 dentry 结构体的信息并按照相应格式将这些“0”和“1”还原成文件的名称. 虚拟机自省的过程就是利用后端模块提供的内核语义信息解析这些“0”和“1”的过程.

3 实验与结果分析

3.1 实验环境

对于本文提出的客户虚拟机文件检测方法 FDM 的效果进行了实验验证. 实验环境参数如表 1 所示.

表 1　实验环境

类型	名称	配置
宿主机	操作系统版本	Ubuntu 12.04
	内核版本	3.2.16
	CPU	Intel Core i7 870 2.93GHz
	内存	4GB
Xen	版本	4.1.2
虚拟机	操作系统版本	CentOS 6.4
	内核版本	2.6.32
	VCPU	1 核
	内存	1GB

3.2 实验结果

下面以 CentOS 6.4 为例, 针对客户虚拟机内的初始文件、新增文件和隐藏文件进行检测, 并与虚拟机内部得到的文件列表进行对比来验证 FDM 的检测效果.

图 4 显示了虚拟机的/usr 目录下文件的内外列表对比结果. 其中, 上部窗口是通过 FDM 在虚拟机外部(Dom0 中), 所获得的文件列表, 下部窗口是在虚拟机内部(guest OS 中)执行“ls -1 /usr”命令获得的文件列表. 通过对比可以发现, 在虚拟机外部所获得的文件列表与虚拟机所包含的文件列表完全一致. 因此, 图 4 的实验结果证实了 FDM 能够准确地检测出虚拟机的/usr 目录下的文件列表.

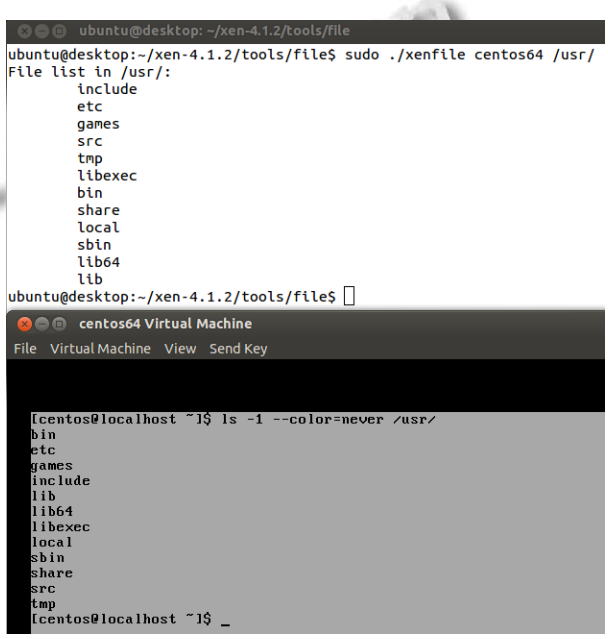

图 4　虚拟机的/usr 目录下文件的内外列表对比

FDM 不仅能够在虚拟机外部获得虚拟机内部文

件列表, 还能够检测出虚拟机内隐藏的文件或目录.

adore-ng[12]是一个功能强大的 Rootkit, 可以隐藏文件目录、文件、进程和网络连接等信息. 在 CentOS 上编译安装 adore-ng, 通过执行"./ava h /usr/tmp"命令将 tmp 文件隐藏起来. 图 5 是 adore-ng 在虚拟机内隐藏文件的结果. 通过与"ls /usr/"的结果对比可以发现, tmp 被隐藏起来了.

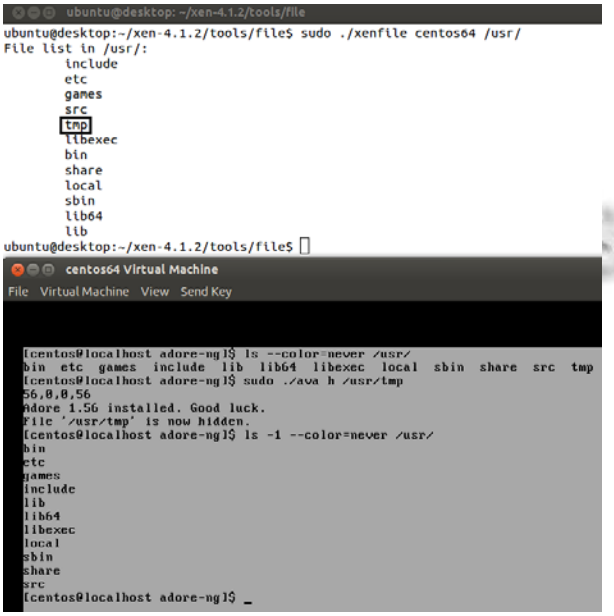

图 5 虚拟机内隐藏文件的检测结果

接下来, 我们在 Dom0 中运行 FDM 获取 DomU 中的文件列表. 通过观察可以发现, 被 adore-ng 隐藏起来的 tmp 被检测了出来. 因此, FDM 也可以有效地发现隐藏文件.

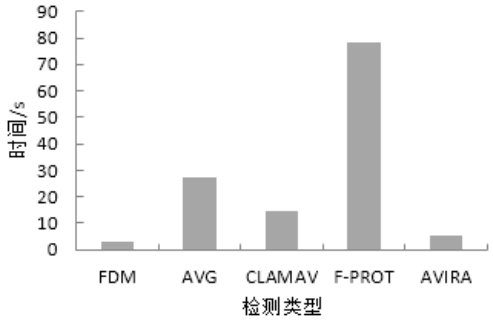

图 6 文件扫描时间柱形图

FDM 扫描文件的效率与当下几款流行的 Linux 杀毒软件相比也有较大的优势. 图 6 是 FDM 与 AVG、CALMAV、F-PROT、AVIRA 四款杀毒软件扫描 10000 个文件的时间对比柱形图, 从图中可以看出, FDM 的文件扫描速率远高于其余几款杀毒软件.

4 结论

本文提出了一种模块化的客户虚拟机内的文件检测方法 FDM. FDM 由前端模块和后端模块两个部分组成, 具有很好的可移植性和扩展性. 同时, FDM 驻留在宿主机里面, 能够很好的避免恶意软件的攻击, 提高了系统的安全性. 最后, 通过实验证实了 FDM 能够准确地获取虚拟机内的文件列表, 并能够检测到被 Rootkit 隐藏的文件, 且验证了 FDM 的文件扫描效率要高于当下许多杀毒软件. 在以后的工作中将会继续增强该方法的使用范围和效率, 并将对基于签名的文件扫描进行研究, 然后与 FDM 技术进行结合, 实现隐藏检测与感染监测的统一.

参考文献

1 2013 年中国网民信息安全状况研究报告. http://www.cnnic.cn/hlwfzyj/hlwxzbg/mtbg/201312/P020131219359905417826.pdf.

2 Mahapatra C, Selvakumar S. An online cross view difference and behavior based kernel rootkit detector. ACM SIGSOFT Software Engineering Notes, 2011, 36(4): 1–9.

3 Kim GH, Spafford EH. The Design and implementation of tripwire: a file system integrity checker. Proc. of the 2nd ACM Conference on Computer and Communications Security. New York. 1994.

4 刘传,薛质.基于 LINUX VFS 的后门机制分析与实现.信息安全与通信保密,2006:137–139.

5 Fu Y, Lin Z. Space traveling across VM: automatically bridging the semantic gap in virtual machine introspection via online kernel data redirection. Proc. of the 2012 IEEE Symposium on Security and Privacy. San Francisco, CA. 2012. 586–600.

6 Garfinkel T, Rosenblum M. A virtual machine introspection based architecture for intrusion detection. Proc. of the 10th Annual Network and Distributed System Security Symposium. San Diego, California. 2003.

7 Jiang X, Wang X, Xu D. Stealthy malware detection and monitoring through VMM-based "out-of-the-box" semantic view reconstruction. ACM Trans. on Information and System

Security, 2010, 13(2): 12.

8 Bovet DP, Cesati M. Understanding the Linux Kerne. 3rd ed. California: O'Reilly Media Inc, 2005.

9 Barham P, Dragovic B, Fraser K, Hand S, Harris T. Xen and the Art of virtualization. Proc. of the 19th ACM Symposium on Operating Systems Principles. New York. ACM. 2003. 164–177.

10 余洋.基于 Xen 的安全虚拟化平台及应用研究.南京:南京邮电大学,2013.

11 Wang Z, Jiang XX, Cui WD, et al. Countering kernel rootkits with lightweight hook protection. Proc. of the 16th ACM Conference on Computer and Communications Security. New York. 2009.

12 Riley R, Jiang X, Xu D. Multi-aspect profiling of kernel rootkit behavior. Proc. of the 4th ACM European Conference on Computer Systems. New York. 2009.