·云计算专题· 文章编号：1000-3428(2014)04-0026-06 文献标识码：A 中图分类号：TP317

基于云计算的恶意程序检测平台设计与实现

韩 奕[1,2]，姜建国[2]，仇新梁[2]，马新建[2]，赵 双[2]

(1. 北京交通大学计算机与信息技术学院，北京 100044；2. 中国科学院信息工程研究所，北京 100093)

摘 要：针对当前恶意程序种类繁多、分析工作量大的问题，利用 VMware vSphere 虚拟化技术，设计并实现云环境下的恶意程序自动检测平台。该平台通过轮询机制获得服务器虚拟机资源的负载情况，将收集的可疑样本分类预处理，调用相应的服务器资源进行检测，可为用户终端节点提供多样化的虚拟环境，实现恶意程序文件、注册表、进程以及网络 4 类主机行为的自动分析，并自动生成分析报告。在真实样本上的实验结果表明，与金山火眼、Threat Expert 平台相比，该平台能够更准确地反映恶意程序的特点及危害性。

关键词：VMware vSphere 技术；恶意代码；自动分析；行为特征；虚拟机；检测

Design and Implementation of Malware Detection Platform Based on Cloud Computing

HAN Yi[1,2], JIANG Jian-guo[2], QIU Xin-liang[2], MA Xin-jian[2], ZHAO Shuang[2]

(1. School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China;
2. Institute of Information Engineering, China Academy of Sciences, Beijing 100093, China)

【**Abstract**】Aiming at the problem of wide range of malware and large analysis workload, in this paper, with the use of VMware vSphere virtualization technology, an automatic malware detection system upon the cloud platform is designed and implemented. This platform adopts polling mechanism to monitor the load of virtual machines in servers, conducts preprocessing of collected suspicious samples according to their type and tests the samples using correspond server resources. It can offer users a variety of virtual environment, automatic analysis malware's four host behavior of files, registry, processes and network, provides online analysis report, and effectively responses to the problem of wide range of malicious programs, eliminates the analyzing workload, improves the efficiency of analysis. Experimental result on real samples shows that this platform can provide more precise character and threat information of analyzed samples compared with Jinshan Fireeye and Threat Expert platform.

【**Key words**】VMware vSphere technology; malicious code; automatic analysis; behavioral characteristics; virtual machine; detection

DOI: 10.3969/j.issn.1000-3428.2014.04.005

1 概述

在现代医学中，化验样本的分析已经从配置试剂、显微镜评估等低效率、高人力的传统手段过渡到将化验样本放入自动分析仪即可打印输出化验报告的时代。和医学科研一样，计算机恶意程序行为自动分析产生的最初动机，也是提高分析效率，模拟专业人员对可疑文件进行分析。

检测可疑文件常用的方法有 3 种：

(1)通过杀毒软件对目标文件执行扫描最常见的方法，但实际扫描结果不能准确判定该可疑文件具有哪些恶意行为。

(2)使用 IDA、OllySafe 等工具，通过脱壳、反汇编手段静态分析代码，这种方法需要分析人员具备较高程序语言基础，完全依赖人工实施，周期较长，且当面临大规模恶意程序爆发时，不能够及时有效地采取措施，另外，通常情况下人们只能得到程序的可执行文件，对于源码未知且多态、变种的可疑文件更显得无能为力。

(3)不具备逆向分析能力的用户，会使用一些安全公司提供的公开 Web 版沙箱系统，沙箱(Sandbox)[1]技术为一些来源不可信、具备破坏力或无法判定恶意程序意图的程序提供实验环境，一些比较常用的 Web 沙箱系统如 Norman Sandbox[2]、Anubis[3]、Joebox[4]等为大众所熟知，但是由于版权、商业利益等原因，各个沙箱系统对公司外部的普通用户只提供 Web 接口，用户必须通过特定网页上传待分析文件，并等待几到几十分钟才能以网页链接或邮件的方式

基金项目：国家自然科学基金资助项目(61372062)。

作者简介：韩 奕(1988－)，女，硕士研究生，主研方向：云计算，恶意代码检测；姜建国，研究员、博士生导师；仇新梁，高级工程师；马新建，博士研究生；赵 双，助理工程师。

收稿日期：2013-10-10 **修回日期：**2013-12-04 **E-mail：**jiaojiao1024@163.com

得到分析结果。这种方式对于偶尔上传几个文件的普通用户而言尚可接受，但如果用它来批量分析大量恶意程序，等待时间和分析报告的格式等限制因素都会给研究工作带来诸多不便。

云计算[5](Cloud Computing)是一种新的计算模式，它通过虚拟化技术将物理资源虚拟化，使之成为可管理的逻辑资源，方便用户通过网络按需使用。其显著优势有：按需服务，高宽带网络接口，共享资源池，快速可伸缩性和服务可测量等。虚拟化技术的成熟也使得云计算成为恶意程序检测技术变革的推动力。如果把虚拟环境整合成一个具有强大计算能力的平台，再将这种强大计算能力分布到互联网终端用户手中，一方面能够惠及更多的用户，另一方面也可以提高恶意程序检测分析效率，对当前信息安全防护具有重要意义。

本文提出一种基于云计算的恶意程序检测平台架构，并给出其关键技术的实现方法。该架构具有用户在线上传的功能，能自动分析可疑文件进(线)程、注册表、文件、网络这 4 类主机行为，并生成综合分析报告。

2　基础平台架构设计

2.1　VMware vShpere 介绍

VMware vSphere[6]基础架构服务是可以将离散的硬件资源转换为大型机式的共享计算平台，并且能够接近本机的性能运行要求最为苛刻的应用程序，VMware vSphere 的组件主要包括：

(1)VMware ESXi

VMware ESXi 提供一个虚拟化层，该层可将物理主机的处理器、内存、存储器及网络资源抽象化为多台虚拟机。

(2)vCenter Server

vCenter Server 运行于 Windows 服务器之上，可集中管理 VMware EXS/ESXi 主机，并提供基本的数据中心服务，如访问控制、性能监控和配置。其实现原理是根据系统管理员设定的策略，管理主机的虚拟机分配，以及给定主机内虚拟机的资源分配。

(3)vSphere Client

vSphere Client 安装在 Windows 平台上，是与 VMware vSphere 进行交互的主要方式，提供创建、管理和监控虚拟机及其资源和主机的主界面。vSphere Client 可用作虚拟机的操作控制台以及 VMware Center Server 系统的管理界面。vSphere Client 包括供管理员和控制台用户使用的文档。

本文所设计的检测平台包括用户端、服务端 2 个部分，如图 1 所示。服务端采用 VMware vSphere5 作为基础架构，ESXi 安装以后，通过 vSphere Client 创建 WinXP、Win7 等虚拟机资源池。资源池在 VMware vShpere 中是一种层次结构相对独立的虚拟机聚合资源，每个资源池可以根据计算容量的大小创建多个虚拟机。

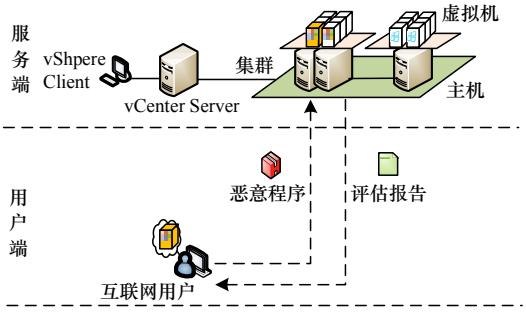

图 1　恶意程序检测平台示意图

用户将可疑文件或者恶意程序样本通过互联网上传至服务端，由服务端根据用户所选检测环境，将待检测文件传送至对应的虚拟机资源池；再将待检测文件投入虚拟机，调用系统控制模块执行检测，分析并记录其在虚拟环境下的行为；最后将恶意行为记录以报告形式反馈回用户。当系统面临多用户上传、批量检测等情况时，vCenter 会先扫描资源池内所有虚拟机的运行状态，判断是否有足够的虚拟机，根据待检测样本与可用虚拟机数量的情况动态分配、调用虚拟机，满足大量和复杂的恶意程序分析需求，服务端虚拟资源结构如图 2 所示。

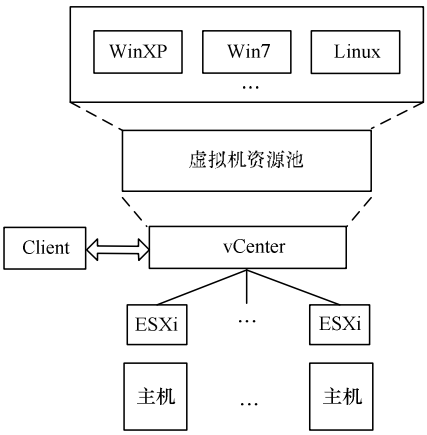

图 2　检测平台虚拟资源结构

2.2　平台检测流程

恶意程序检测平台为了使待检测样本在服务端分配虚拟机的过程得到合理调度，防止多个虚拟机对同一样本重复分析产生冗余信息，就需要在设计程序过程采用状态标记的方式加以识别。

(1)恶意程序上传后，数据库会增加新的样本记录，此时系统会自动填写文件状态标识位，初始为“0”，记为“未检测”状态。

(2)系统对数据库中所有文件循环扫描，若判断某文件标识位为“0”，即将该样本上传至服务端，调用资源池可用虚拟机，运行行为监控模块对其检测分析，同时将文件状态改为“2”，记为“检测中”。

(3)检测过程的实现借助于虚拟机层行为监控、网络流量捕获和行为监控控制 3 个模块，记录目标程序在运行过程中文件、进程、注册表、网络 4 类主机行为。

(4)将程序在虚拟机中产生的行为数据写入数据库，待检测结束后更新文件状态标识位为“1”，并生成综合性分析报告。

恶意程序检测流程如图 3 所示。

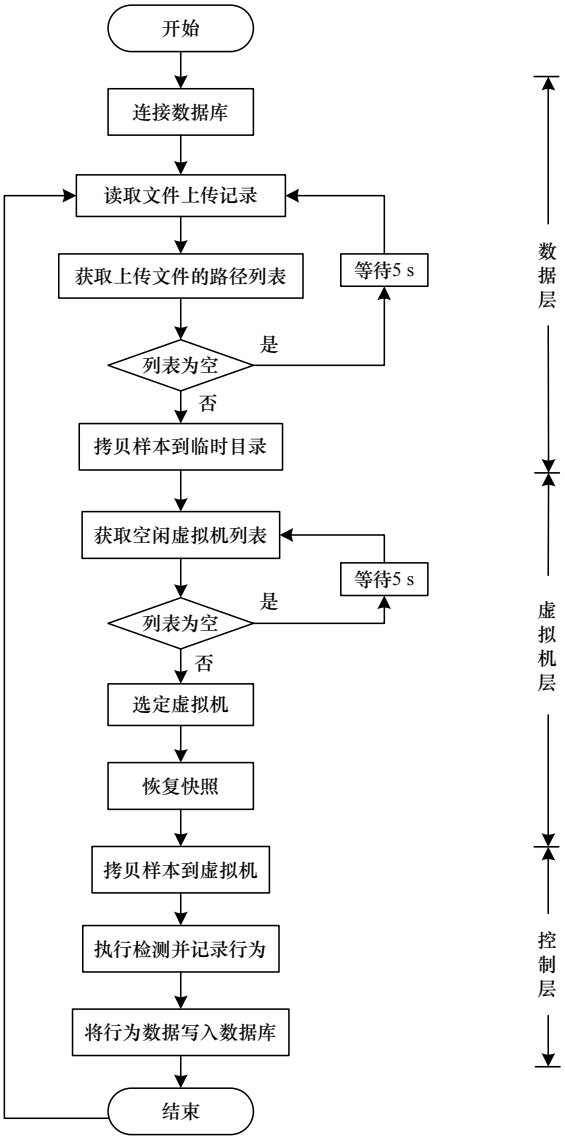

图 3 恶意程序检测流程

3 关键技术与实现

3.1 虚拟机层实现

虚拟机层通过 pysphere[7]实现，pysphere 是一种 python 与 vSphere Web SDK 接口的 API 函数库，对虚拟机的操作主要以脚本或者编程方式实现，方便程序员编写程序自动控制虚拟机。本文利用 pysphere 实现虚拟机电源管理、镜像恢复、传送文件、启动程序等操作，这也是实现恶意程序自动分析的基础，脚本语言对虚拟机主要进行以下控制操作：

(1)建立服务器连接

```
>>>server.connect("http://192.168.100.100:8080/sdk","jdoe","secret")
```

该命令用于寻找服务器地址，发起连接请求，并与服务器建立连接。

(2)获取虚拟机列表

```
>>>vmlist=server.get_registered_vms(resource_pool='Windows XP', status='poweredOn')
```

该命令能够获得数据中心已注册的虚拟机列表，“pool”参数指虚拟机所在的资源池，根据参数“resource_pool”和“status”筛出所需资源池，调用空闲状态的虚拟机。

(3)电源控制

```
>>>vml.power_on()
>>>vml.reset()
>>>vml.suspend() #since pysphere 0.1.5
>>>vml.power_off()
```

上述操作实现虚拟机电源状态的控制，包括虚拟机的开启、重新启动、挂起、关闭等。

(4)恢复镜像

```
>>>vml.revert_to_snapshot()
>>>vml.revert_to_named_snapshot(“base”)
>>>vml.revert_to_path(“/base/updated”)
```

在正常情况下，该命令以快照的形式启动虚拟机。虚拟机快照是 VMware vShpere 提供的一个功能，可以跳过操作系统的启动过程，直接加载预先保存的操作系统状态，缩短整个加载时间。此外，每次检测结束后也无需对环境进行清理，因为每次分析都会重新加载同一个干净的系统状态，这两点无疑大大提高了行为检测效率。

(5)传送文件

```
>>>get_file(guest_path, local_path, overwrite=False)
```

该命令用于从“guest_path”目录下载文件到本地“local_path”中，其中，参数“guest_path”指虚拟机文件的绝对路径。

```
>>>send file(local path, guest path, overwrite=False)
```

该命令用于从物理机中把所需文件传送至虚拟机指定目录，其中“local_path”参数指文件在物理机中的路径，“guest_path”参数指虚拟机中的路径。

(6)虚拟机进程操作，执行检测

```
>>>start_process(program_path, args=None, env=None, cwd=None)
```

该命令用于开启虚拟机中的进程，并返回当前进程的 PID。其中，“cwd”参数指进程所在工作目录的绝对路径。

3.2 数据层实现

数据层以虚拟机传出的监控记录为数据输入，本文构造多个数据表存储恶意程序的主机行为，以实现检测结果按行为操作对象二次分类，表结构如图 4 所示。其中，time_stamp 为行为发生时间；src 为发起行为的进程；dst 为该行为的对象；md5 作为恶意程序名字的唯一标识，通过唯一标识即可汇总该恶意程序相关行为，而后生成检测报告。

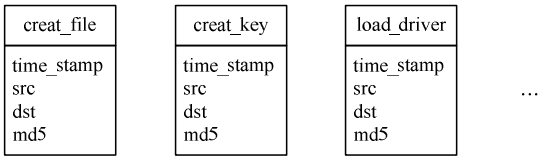

图 4　MySQL 数据表

此外，在分析的过程中，还需要对行为的发起者进行过滤。由于监控日志里的初始数据包括虚拟机客户中所有进程的行为，需要过滤关注的恶意程序及其衍生的行为。

3.3　控制层实现

控制层是整个检测平台的核心部分，控制层各个模块及其关键技术如表 1 所示。

表 1　系统各模块所使用的关键技术

模块	关键技术
监控控制模块	Windows 驱动和驱动调用
网络流量捕获模块	Winpcap
数据分析模块	正则表达式

对表 1 中的关键技术具体描述如下：

(1)Windows 驱动和驱动调用

操作系统的总体设计是基于分层设计思路的，每层由若干组件构成，它依赖于上层组件向下层组件的调用[8]。每层组件有固定的接口，靠近底层的组件有更高的权限，靠近上层的组件将任务转化为对底层组件的调用。

应用程序属于操作系统上层组件，要使驱动程序工作，必须先使用 Windows API 的 CreateService 函数对其进行加载以创建一个服务，然后再使用 StartService 函数来启动该服务。同样，要停止驱动程序工作时，先用 OpenService 函数打开服务，再使用 ControlService 函数停止该服务。

本文在虚拟机内部部署了一个应用层的监控控制程序，用于启动、监控和运行待测文件，它是利用上述这些方法实现监控控制模块的调用。

(2)Winpcap 技术

Winpcap[9]是一个基于 Windows 平台用于捕获网络数据包并进行分析的开源库，包括核心过滤器、底层的动态链接库(packet.dll)和一个高层的不依赖于系统的库(wpcap.dll)，前者提供底层 API，伴随着独立于 Microsoft 操作系统的编程接口，这些 API 可以直接用来访问驱动的函数；后者导出了一组更强大的与 libpcap 一致的高层抓包函数库(capture primitives)。Winpcap 独立于主机协议(如 TCP-IP)发送和接收原始数据包，也就是说，Winpcap 不能阻塞、过滤或控制其他应用程序数据包的发收，它仅仅是监听共享网络上传送的数据包，以一种与网络硬件和操作系统无关的方式进行。

本文在虚拟机内部部署了基于 wpcap.dll 的应用程序，用于捕获待测文件运行时所产生的网络流量，进而为僵尸网络特征提取和下载行为特征提取提供数据源。

(3)正则表达式

正则表达式通常被称为一个模式[10](pattern)，用来描述或者匹配一系列符合某个句法规则的字符串，如利用正则表达式对数据包进行匹配和信息提取。

本文利用正则表达式以及其他字符串处理方法实现了数据分析模块，用于对监控模块生成的监控日志以及网络流量捕获模块生成的 dump 文件进行数据过滤和提取，从而在繁杂的数据中提取相关信息。

4　实验与结果分析

4.1　实验目的

实验目的主要展现恶意程序检测虚拟平台的实用性，检测报告能全面、直观地重现恶意行为，另外，还要体现对多样本检测时系统的分析效率。

4.2　实验环境

实验环境具体如下：

(1)主机平台配置

1)硬件平台：CPU 为 Intel Xeon E5620 2.4 GHz；主存为 12 276.5 MB；

2)宿主机操作系统：ESXi-5.1.0；

3)虚拟机操作系统：Microsoft Windows XP。

(2)网络拓扑

为满足恶意程序可能的网络需求，增设 30 台虚拟机，构建一个内部局域网环境，网内 IP 设置范围为 192.168.1.2~192.168.1.100，测试主机安装恶意代码检测程序，整个网络环境拓扑结构如图 5 所示。

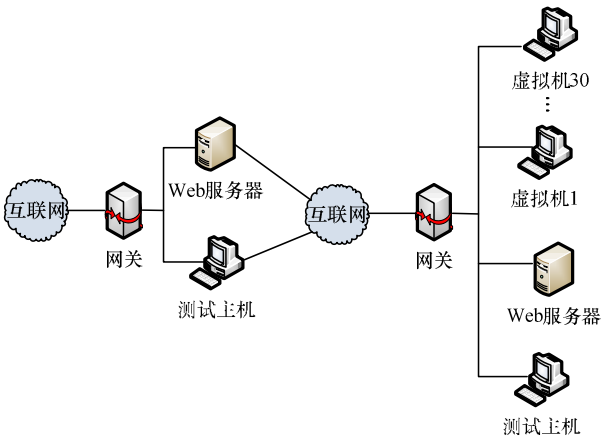

图 5　测试网络环境拓扑

4.3　实验内容

按照恶意程序检测虚拟平台的设计目的，实验采用来自于实验室部署的密网捕获到的恶意程序样本。其中，功能性测试采用的样本 Worm.Viking(威金)[11]，威金属蠕虫类病毒，能够将自身复制到系统文件夹%windows%下，文件名为 rundle32.exe，具有释放病毒文件、修改注册表等恶意行为以达到随机启动、开启端口、终止杀毒软件等恶意行为，且经 360 安全卫士扫描并确认是恶意程序。测试采用

MD5 对恶意程序的命名来标识本次测试的样本。

(1)功能测试

1)总体信息

测试样本：viki0000.exe；

MD5 值：febd2 583e768f8812d1cf0d4e805565f；

文件大小：85 KB。

2)运行检测

3)进程行为

行为描述：创建进程

Logo1_.exe--->C:\WINDOWS\system32\net.exe

cmd.exe--->C:\febd2583e.exe

行为描述：注入线程

Logo1_.exe--->net.exe

cmd.exe--->febd2583e.exe

4)文件行为

行为描述：创建文件

System--->C:\monitorlog.log

Logo1_.exe--->D:_desktop.ini

Logo1_.exe--->C:\WINDOWS\vDll.dll

行为描述：覆盖文件

Logo1_.exe--->C:\WINDOWS\rundl132.exe

行为描述：删除文件

cmd.exe--->C:\febd2583e.exe

5)注册表行为(部分)

行为描述：设置注册表键值

Logo1_.exe--->\REGISTRY\USER

\S-1-5-21-1177238915-1647877149-2147093213-500

\Software\Microsoft\WindowsNT\CurrentVersion

\Windows\load ["C:\WINDOWS\rundl132.exe"]

Logo1_.exe--->\REGISTRY\MACHINE\SOFTWARE

\Soft\DownloadWWW\auto ["1"]

6)网络行为

行为描述：ARP(部分)

Who has 192.168.1.1 ? Tell 192.168.1.17

192.168.1.1 is at 005056893e04

Who has 192.168.1.2 ? Tell 192.168.1.17

Who has 192.168.1.3 ? Tell 192.168.1.17

行为描述：ICMP

Ping(request)192.168.1.17--->192.168.1.1

Ping(reply)192.168.1.1--->192.168.1.17

7)行为分析

威金病毒样本入侵用户电脑后，首先运行 Logo1_.exe 程序，全盘搜索“.exe”执行程序并感染，感染后文件突变全变成了“.exe”执行程序缺省时的图标。其次，将自身复制到 C 盘“WINDOWS”系统文件夹目录下，然后将其命名成“rundl132.exe”，该操作意图将自身伪装成系统程序“rundll32.exe”，但病毒文件与系统文件的区别是文件名中间的数字“1”和字母“l”。再次，释放病毒文件“_desktop.ini”和“vDll.dll”。其中，“_desktop.ini”是病毒配置文件，这个文件是当威金病毒成功感染一个磁盘分区目录下所有“.exe”执行程序后才被创建的，每个驱动目录下各存在一个，它不属于程序，只是配置文件，用于记录病毒是否在今天访问过该文件夹目录；“vDll.dll”是病毒的动态链接库文件，病毒运行后会将“vDll.dll”插入到系统进程中，例如“explorer.exe”或者“iexplore.exe”进程，用于检查网络是否可用，便于下载木马程序。由于威金病毒的不断变种，这个“.dll”文件的名称并不单一，各种不同的变种名称也不同，如“dll.dll”、“vdll.dll”等。

之后修改开机自动启动项，以达到随机启动的目的。同时还会发送 ICMP 探测数据，尝试判断网络状态是否可用，当用户联网后，病毒会打开端口连接黑客服务器，下载木马程序。

感染结束以后，执行自删除操作以便躲过用户追查。

恶意程序行为过程如图 6 所示。

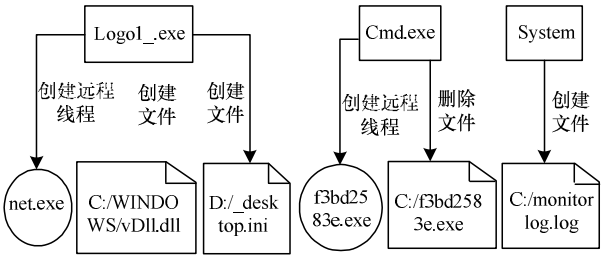

图 6 恶意程序行为过程

综上，本文所搭建的恶意代码检测平台能够较全面地捕获恶意代码在运行过程中所发生的危险行为。

(2)性能测试

实验室所捕获的恶意程序有上千万个，由于时间限制，无法将全部恶意程序样本逐一进行测试，因此本文从中随机抽取了 100 个样本进行分析测试。使用恶意代码自动分析系统对上述恶意样本进行批量分析，测试结果如表 2 所示。可以看出，有 3 个样本不能提供正常的分析报告，经手工分析查看发现，它们因为条件执行需要与操纵者实施交互，或者虚拟机内存耗尽，导致自动分析无法正常工作。

表 2 恶意程序样本分析结果

测试结果	数量	比例/%
分析成功	97	97
分析失败	3	3

将本文平台与金山火眼(https://fireeye.ijinshan.com)和 Threat Expert(http://www.threatexpert.com/submit.aspx)2 款在线沙箱检测平台对比检测粒度。检测样本为 015816 d08fc931c2c5eeaf6e9819d39c82b0125d.exe，检测结果如表 3 所示。

表 3　3 种平台的检测结果比较

检测事件数	金山火眼	Threat Expert	本文平台
文件行为	5	6	21
注册表行为	7	Many	11
进程行为	6	0	9
网络行为	4	6	8

注:表中 Many 表示粒度太细导致检测结果中的事件数过多。

在本文所述系统流程中，程序行为监控时间是可控的，监控时间过短会导致监控信息不全，执行时间过长会降低系统的运行效率。经过实验调试，将监控时间定为 30 s 是对监控信息量和运行效率进行折中的较优方案。

实验结果证明，本文的恶意代码自动分析平台能够较好地检测分析待测程序的主机行为和网络行为。对行为敏感度较高,能够较全面地提取程序的特征行为(Threat Expert 公司几乎枚举了恶意代码所有注册表行为，没有对恶意程序关键行为进行过滤)。

5　结束语

本文构建的恶意程序检测基础平台，可用于互联网用户上传检测可疑文件，同时能满足对大量恶意程序检测分析的需求，提高检测效率。通过动态检测恶意程序的行为，可掌握恶意程序可能进行的破坏活动，为主机恢复、损失评估提供详细信息。但由于恶意代码实际分析的过程较为复杂，往往与运行环境、操纵者交互紧密相关，或者由于样本与僵尸网络插件分离等原因，导致恶意程序不能正常运行；此外，随着恶意代码反检测技术的发展和应用，一些对抗虚拟机的技术、程序休眠、潜伏周期长等诸多原因也会导致无法在检测平台观察到恶意样本的正常行为，因此，对恶意代码的分析还需要进一步改进及研究。

下一步工作将对恶意程序威胁度评估做深入研究，降低系统使用门槛，提高行为分析报告可读性，一方面准确判别被检测程序的恶意性，减少误报率和漏报率；另一方面合理评价恶意程序的危害程度，以便有针对性地做好安全响应，采取有效对抗或清除恶意程序的措施。

参考文献

[1] 维基百科. 沙箱[EB/OL]. [2013-09-10]. http://zh.wikipedia.org/w/index.php?title=%E6%B2%99%E7%9B%92&variant=zh-cn.

[2] Norman Co.. Norman Sandbox Whitepaper[EB/OL]. [2013-09-10]. http://download.norman.no/whitepapers/whitepaper_Norman_SandBox.pdf.

[3] Willems C, Holz T, Freiling F. Toward Automated Dynamic Malware Analysis Using CWSandbox[J]. IEEE Security and Privacy, 2007, 5(2): 32-39.

[4] Joebox Team. Joebox: A Secure Sandbox Application for Windows to Analyse the Behaviour of Malware[EB/OL]. [2013-09-10]. http://www.joebox.org/concept.php.

[5] 陈　康, 郑纬民. 云计算: 系统实例与研究现状[J]. 软件学报, 2009, 20(5): 1337-1348.

[6] VMware Inc.. vSphere 说明[EB/OL]. [2013-09-10]. http://pubs.vmware.com/vsphere-51/index.jsp.

[7] Google Inc.. vSphere SDK for python 说明[EB/OL]. [2013-09-10]. http://code.google.com/p/pysphere/.

[8] 张　帆, 石彩成. Windows 驱动开发技术详解[M]. 北京: 电子工业出版社, 2008.

[9] WinPcap Team. WinPcap: The Windows Packet Capture Library[EB/OL]. [2013-09-10]. http://www.winpcap.org.

[10] 谭玉玲. 基于正则表达式的数据处理应用[J]. 武汉理工大学学报, 2010, 32(2): 249-252.

[11] 百度百科. 威金[EB/OL]. [2013-09-10]. http://baike.baidu.com/view/560093.htm.

编辑　金胡考

(上接第 25 页)

[16] Ostermann S, Iosup A, Yigitbasi N, et al. A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing[J]. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 2010, 34(4): 115-131.

[17] Amazon Inc.. Amazon S3[EB/OL]. [2013-08-26]. http://aws.amazon.com/cn/s3/.

[18] Amazon Inc.. SimpleDB[EB/OL]. [2013-08-26]. http://aws.amazon.com/cn/simpledb/.

[19] Krutz R L, Vines R D. Cloud Security: A Comprehensive Guide to Secure Cloud Computing[M]. Indianapolis, USA: Wiley, 2010.

[20] Sood S K. A Combined Approach to Ensure Data Security in Cloud Computing[J]. Journal of Network and Computer Applications, 2012, 35(6): 1831-1838.

[21] Cai Jianping, Qiao Liping. Research and Application of the Floating License Management Strategy[C]//Proc. of International Conference on Information Science and Engineering. Nanjing, China: [s. n.], 2009: 26-28.

[22] 王全国, 任晨光. 软件狗加密技术透视[M]. 北京: 清华大学出版社, 1996.

编辑　金胡考