

一种基于系统管理模式的隐藏进程检测模型

赵晓参¹, 任 炬¹, 徐 旸¹, 王国军²

(1 中南大学 信息科学与工程学院, 湖南 长沙 410083; 2 广州大学 计算机科学与教育软件学院, 广东 广州 510006)

摘 要: 近年来, 恶意程序的隐身性越来越强, 针对这个问题, 提出了一种基于系统管理模式(System Management Mode, SMM)的隐藏进程检测模型(SMM-based Hidden process Detection model, SHPD). 该模型能够有效地检测系统中的隐藏恶意进程, 同时保证自身的透明性. 模型包括客户端和监控端两部分. 客户端运行在 BIOS 中, 利用内外语义信息建立操作系统进程的多个视图, 将建立的视图发送到监控端. 监控端通过对比视图间的差异, 识别出隐藏的恶意进程. 在提出的 SHPD 模型理论支持下, 搭建了实验原型系统, 并进行了功能测试和分析, 实验结果证明了该模型的有效性.

关键词: 系统管理模式; 恶意程序; 隐藏性; 视图对比

中图分类号: TP309. 2

文献标识码: A

文章编号: 1000-7180(2017)12-0111-05

A Hidden Process Detection Model Based on System Management Mode

ZHAO Xiao-can¹, REN Ju¹, XU Yang¹, WANG Guo-jun²

(1 College of Information Science and Engineering, Central South University, Changsha 410083, China;

2 College of Computer Science and Educational Software, Guangzhou University, Guangzhou 510006, China)

Abstract: In recent years, the stealth of malware is getting stronger and stronger. In this paper, a SMM-based Hidden process Detection model (SHPD) is proposed. SHPD can effectively detect the stealthy process in system while ensuring its own transparency. SHPD consists of two parts: the client and the monitor. The client, which implemented in BIOS, uses both internal and external semantic information to establish multiple views of processes in OS and sends those process views to the monitor. The monitor identifies the stealthy process by comparing the differences between the views. In the paper, we build a prototype system under the support of the SHPD theory, and conduct functional testing and analysis. The experimental results verify the feasibility of SHPD.

Key words: SMM; malware; stealthy; view comparison

1 引言

近年来, 网络恶意程序迅速增多^[1]. 恶意程序隐藏自身的存在痕迹, 持续感染操作系统, 逃避安全工具的检测. 现阶段恶意程序隐藏手段更底层化, 这就使网络安全面临更严峻的挑战.

关于恶意程序的隐身性问题, 目前有很多研究者提出了相关的解决方案. 其中, 使用最广泛的是基

于操作系统的检测方法^[2], 这类检测系统安装和运行在操作系统内部, 通过对比应用层和内核层视图差异来发现隐藏恶意程序, 但是这类方法很容易被更高等级恶意程序攻击, 导致保护策略失效. 另一类方法是基于虚拟机监控器的检测方法^[3]. 这类方法把检测程序放到操作系统外部, 使其运行在虚拟机监控器中. 该方法通过验证内核完整性或者视图对比来发现隐藏恶意程序, 能够较有效地防止更高级

收稿日期: 2017-03-01; 修回日期: 2017-04-07

基金项目: 国家自然科学基金重点项目(61632009); 国家自然科学基金面上项目(61472451); 湖南省研究生科研创新项目(CX2015B047)

恶意程序的攻击. 但该类方法运行在虚拟机监控器中, 又面临着虚拟机逃逸攻击等威胁^[4-5].

因此, 本文提出一种基于系统管理模式(System Management Mode, SMM)的隐藏进程检测模型(SMM-based Hidden process Detection model, SHPD). 该模型分为客户端和监控端两个部分, 利用视图对比发现隐藏恶意程序, 但客户端检测程序运行在 BIOS 系统中, BIOS 系统能够保证对于物理内存的绝对隔离, 所以能够使更高级恶意程序无法对其攻击, 并且避免了虚拟机逃逸攻击等威胁.

相比与现阶段的安全解决方法, 本文的主要贡献体现在以下几个方面:

- (1) 在系统管理模式的理论基础上, 提出了 SHPD 模型, 将其客户端检测程序放到 BIOS 中, 保证了检测程序的绝对隔离性.
- (2) 利用系统管理模式, 实现了对上层操作系统的语义重建.
- (3) 实验验证了 SHPD 模型理论, 证明其能有效发现隐藏恶意程序, 并且保护自身不受攻击.

2 系统管理模式

系统管理模式^[6-9]是一种为操作系统提供透明机制的运行模式. 该模式相关执行代码在 BIOS 中实现. 在操作系统启动之前, 该模式的相关代码从 BIOS 中加载到系统管理内存中. 系统管理内存是一种隔离内存, 在任何其他模式下都不允许被访问. 系统管理模式的启动依赖于系统管理中断, 当出现系统管理中断时, 开启系统管理模式.

系统管理模型拥有以下优良特性:

- (1) 系统管理模型可以访问系统中的所有物理内存.
- (2) 启动系统管理模型之前, 相关代码加载到系统管理内存, 加载完毕后不能修改. 保证了代码的不可篡改性.
- (3) 系统管理模式可以重建系统语义信息, 不必依赖上层操作系统. 保证信息独立性.

由于系统管理模型拥有以上的优良特性, 根据其相关理论, 构建 SHPD 模型.

3 基于系统管理模式的 SHPD 模型

SHPD 模型包括两部分: 客户端和监控端, 如图 1 所示. 客户端触发系统管理中断, 使其进入系统管理模式. 将系统的可信进程视图和不可信进程视图发送到监控端. 监控端使用视图对比技术, 识别客户

端中的隐藏进程, 发现恶意程序. SHPD 模型主要从以下五个步骤做到隐藏恶意程序的检测:

- (1) 客户端使用内部函数获取操作系统的不可信进程视图, 通过网络发送到监控端, 进而触发模块随机地触发中断, 进入系统管理模式.
- (2) 客户端进入系统管理模式后, 处理语义重建模块.
- (3) 客户端利用重建的系统语义信息, 获取当前系统中的所有进程列表, 得到上层操作系统的可信进程视图.
- (4) 客户端通过网络模块把可信进程视图发送到监控端, 退出系统管理模式.
- (5) 监控端接收到客户端的可信和不可信进程视图后, 进行视图对比, 获取视图差异, 分析客户端中隐藏程序的存在痕迹, 并展示检测结果, 由此发现隐藏恶意程序.

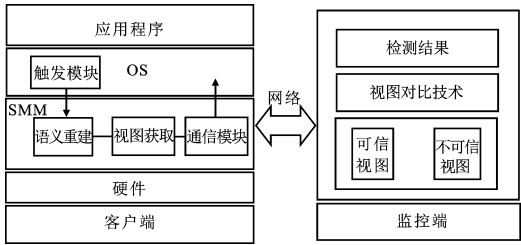


图 1 SHPD 模型架构图

3.1 客户端模块

客户端的主要作用有以下几个方面: 其一是使用操作系统内部函数获取不可信进程视图; 其二是利用系统管理模式获取可信进程视图; 其三通过网络把相关视图信息发送到监控端. 客户端由以下四个模块组成: 位于操作系统中的触发模块, 位于系统管理模式中的语义重建模块、视图获取模块和通信模块.

3.1.1 触发模块

触发模块的主要功能有两点: (1) 触发系统管理中断使系统进入系统管理模式; (2) 获取操作系统不可信进程视图, 并发送到监控端.

触发系统管理中断的方式分为两种: 基于软件和基于硬件的触发中断方式. 本文采用基于软件的触发中断方式, 该方式通过输入输出端口, 直接向高级电源管理控制寄存器写入数据, 由此触发系统管理中断, 进入系统管理模式^[10].

触发模块在触发系统管理中断前, 利用操作系统的内部函数获取当前的不可信进程视图, 不可信进程视图指的是操作系统展示的进程信息. 此类信息将通过网络发送到监控端, 以便进行视图对比

检测.

3.1.2 语义重建模块

经过触发模块触发了系统管理中断,由此进入系统管理模式.客户端中的第二个模块语义重建模块开始进入工作.

语义重建模块的主要工作是将系统管理模式CPU中的上下文信息进行读取,并将上下文中的虚拟地址转换成物理地址.现给出语义重建模块中内核虚拟地址和物理地址转换的伪代码描述.

```
算法 1 内核虚拟地址和物理地址的转换
Virtual_to_Physical( virtual_address)
Input:virtual_address
Output:physical_address
1: Initialize(PAGE_OFFSET)
2:physical_address=(unsigned long)( virtual_address)
- PAGE_OFFSET
3: returnphysical_address
```

3.1.3 视图获取模块

视图获取模块根据语义重建模块提供的上层操作系统的语义信息获取可信进程视图.

该模块的作用是将语义重建之后得到的当前进程物理地址进行分析.所有进程以环形双向链表存储,当获取到当前进程物理地址后,对该环形双向链表进行遍历,得到所有进程的信息,也就是可信进程视图.

```
可信进程视图获取的伪代码描述如下所示.
算法 2 可信进程视图获取
Trusted_Process_View(current_task)
Input:current_task //当前进程的 task_struct 的物理地址
Output: TPV[MAX_PRO_LEN] //可信进程数组
1: TPV[MAX_PRO_LEN] = NULL
2:t_prev = current_task ->prev_task
3:t_next = current_task->next_task
4:i = 0
5: TPV[i] = * current_task;
6: while t_prev != t_next && t_prev->next_task
!= t_next Do: // 遍历链表
7: TPV[++i] = * t_prev
8: TPV[++i] = * t_next
9:t_prev = t_prev->prev_task
10:t_next = t_next->next_task
11: if t_prev == t_next Do:
12: TPV[++i] = * t_prev
13: return TPV[MAX_PRO_LEN]
```

3.1.4 通信模块

通信模块的功能主要有如下两个方面:其一接收模块获取的可信进程视图,将其发送到监控端;其

二退出系统管理模式,返回系统管理中断前的状态.

本文采用在系统管理模式中直接发送数据包的方式进行数据传输,该方式不经过操作系统,可以有效地防止恶意攻击者对发送数据的修改,保证数据的完整性.

当视图获取模块得到可信视图后,可信视图以数据包形式传输给通信模块,通信模块把最终的可信视图数据传输到监控端.传输完毕后,通信模块给出退出系统管理模式指令,结束该模型的整个客户端工作.

3.2 监控端模块

经过上一节客户端模块的所有工作后,监控端模块启动,接收到客户端传送的可信视图和不可信视图.监控模块对视图进行差异对比,由对比结果推测是否有隐藏恶意程序.监控模块还负责监控客户端的通信频率,如果通信频率不正常,说明客户端触发模块被恶意程序攻击.

可信视图和不可信视图差异对比图如图 2 所示.

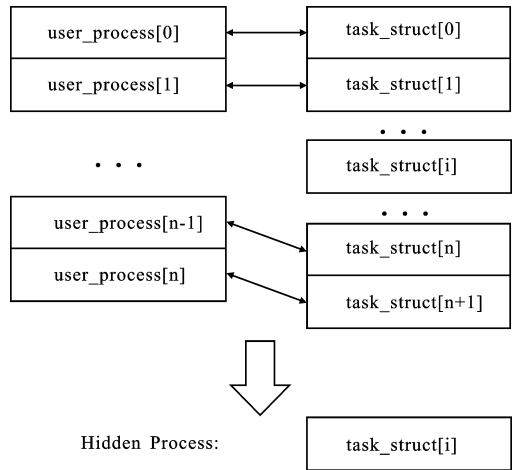


图 2 隐藏进程检测对比图

左边的 user_process 为不可信视图元素,右边的 task_struct 为可信视图元素,由对比差异可知,隐藏恶意程序为 task_struct[i].由此监控模块的任务完成.该模型的总体工作结束,按照预期的理论分析,监控端与客户端的相互配合,能够很好的发现隐藏的恶意程序,提供网络安全保障.

4 实验及结果分析

为了验证 SHPD 模型的有效性,本文实现了 SHPD 原型系统,并根据运行结果分析 SHPD 的检测能力.系统的流程图如图 3 所示.

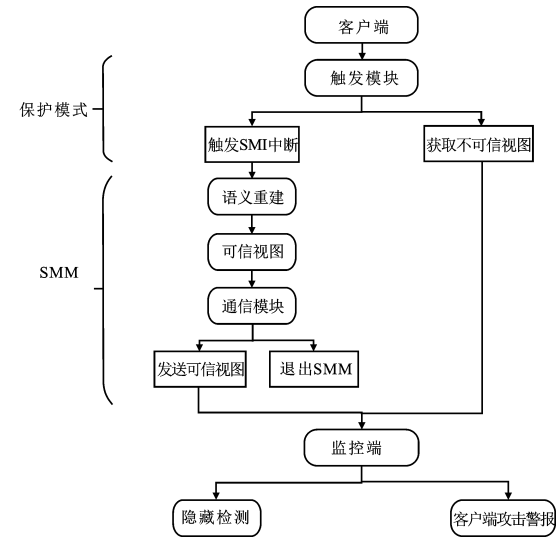


图 3 系统检测流程图

PID	GID	TGID	PPID	UID	NAME
1	0	1	0	0	init
2	0	2	0	0	kthreadd
3	0	3	2	0	migration/0
4	0	4	2	0	ksoftirqd/0
5	0	5	2	0	stopper/0
6	0	6	2	0	watchdog/0
7	0	7	2	0	events/0
27	0	27	2	0	kseriod
28	0	28	2	0	md/0
29	0	29	2	0	md misc/0
30	0	30	2	0	linkwatch
31	0	31	2	0	khungtaskd
32	0	32	2	0	kswapd0
272	0	272	2	0	kdmflush
274	0	274	2	0	kdmflush
292	0	292	2	0	jbd2/dm-0-8
293	0	293	2	0	ext4-dio-unw
963	0	963	2	0	ext4-dio-unw
1058	0	1058	2	0	kauditd
1320	0	1320	1	0	auditd
1405	32	1405	1	32	rpcbind
1425	81	1425	1	81	dbus-daemon
1428	81	1425	1	81	dbus-daemon
1465	29	1465	1	29	rpc.statd
1502	0	1502	1	0	wpa_supplica
1503	0	1503	1	0	cupsd
1535	0	1535	1	0	acpid
1547	68	1547	1	68	hald
1548	0	1548	1547	0	hald-runner
1549	68	1547	1	68	hald
1584	0	1584	1548	0	hald-addon-i
1597	68	1597	1548	68	hald-addon-a
1621	0	1621	1	0	automount
1622	0	1621	1	0	automount
1623	0	1621	1	0	automount
1626	0	1621	1	0	automount
1629	0	1621	1	0	automount
1675	0	1675	1	0	VBoxService
1678	0	1675	1	0	control
1679	0	1675	1	0	timesync
1600	0	1675	1	0	vmtoolsd
1681	0	1675	1	0	cpuphotplug
1682	0	1675	1	0	memballoon

图 4 不可信进程视图

PID	GID	TGID	PPID	UID	NAME
1	0	1	0	0	init
2	0	2	0	0	kthreadd
3	0	3	2	0	migration/0
4	0	4	2	0	ksoftirqd/0
5	0	5	2	0	stopper/0
6	0	6	2	0	watchdog/0
7	0	7	2	0	events/0
27	0	27	2	0	kseriod
28	0	28	2	0	md/0
29	0	29	2	0	md misc/0
30	0	30	2	0	linkwatch
31	0	31	2	0	khungtaskd
32	0	32	2	0	kswapd0
272	0	272	2	0	kdmflush
274	0	274	2	0	kdmflush
292	0	292	2	0	jbd2/dm-0-8
293	0	293	2	0	ext4-dio-unwrit
963	0	963	2	0	ext4-dio-unwrit
1058	0	1058	2	0	kauditd
1320	0	1320	1	0	auditd
1405	32	1405	1	32	rpcbind
1425	81	1425	1	81	dbus-daemon
1428	81	1425	1	81	dbus-daemon
1465	29	1465	1	29	rpc.statd
1502	0	1502	1	0	wpa_supplicant
1503	0	1503	1	0	cupsd
1535	0	1535	1	0	acpid
1540	0	1540	1	0	backdoor
1547	68	1547	1	68	hald
1548	0	1548	1547	0	hald-runner
1549	68	1547	1	68	hald
1584	0	1584	1548	0	hald-addon-inpu
1597	68	1597	1548	68	hald-addon-acpi
1621	0	1621	1	0	automount
1622	0	1621	1	0	automount
1623	0	1621	1	0	automount
1626	0	1621	1	0	automount
1629	0	1621	1	0	automount
1675	0	1675	1	0	VBoxService
1678	0	1675	1	0	control
1679	0	1675	1	0	timesync
1600	0	1675	1	0	vmtoolsd
1681	0	1675	1	0	cpuphotplug

图 5 可信进程视图

4.1 系统配置

客户端: ASUS-M2V_MX_SX 主板、2.2 GHz AMD Sempron LE-1250CPU、2 GB Kingston DDR2 RAM、RS-232 接口;Linux-2. 6. 32.

服务端: Intel Core i5-4590 3.1 GHz CPU; 4 GB DDR3 RAM;Linux-3. 13. 0-32.

4.2 功能测试

本次测试使用具有隐藏进程功能的 Linux Rootkit(wnps)来验证 SHPD 的检测能力. 首先,在客户端使用 insmod 命令加载内核模块 wnps(insmod wnps. ko),其次,在客户端启动后门程序 backdoor,接下来,使用内核模块 wnps 提供的命令隐藏 backdoor 进程,最后,在客户端使用 ps 指令和 proc 文件系统确认 backdoor 进程被隐藏.

通过客户端和监控端的交互,得到了如图 4 所示的不可信进程视图和如图 5 所示的可信进程视图. 对比两个视图的差异,由此发现隐藏恶意程序.

图 4 中的不可信进程视图展示了在操作系统内通过/proc 目录提取的进程列表,图 5 的可信进程视图给出了在系统管理模式中通过提取 task_struct 链表获得的进程列表. 由图 4 和图 5 可知,1540 号进程在可信进程视图中可见,在不可信进程视图中不可见,这说明 backdoor 进程在操作系统内部被隐藏.

为验证 SHPD 模型的适用范围,本文测试了多个典型的隐藏程序,涉及了多种不同的隐藏技术,例如,lrk5 是典型的用户态 Rootkit,通过替换用户态的 ls,ps,netstat 等指令达到隐藏效果,knark,adore_ng 都是典型的内核态 Rootkit,但隐藏技术却不尽

相同,其中,knark 通过修改系统调用表隐藏攻击者,adore_ng 通过修改 VFS 层处理函数实现隐藏目的. 表 1 列出了 SHPD 对多种隐藏程序的检测结果,其中“√”表示成功检测隐藏进程,“×”表示未检测到隐藏进程.

表 1 SHPD 对多种隐藏程序的检测结果

	内核版本	隐藏级别	隐藏进程
lrk5	2. 4. 2. 5	用户态	√
knark_0. 59	2. 2. 2. 4	内核态	√
adore_ng	2. 4. 2. 6	内核态	√
ddrk	2. 6	内核态 用户态	√

由表 1 可知, SHPD 模型能够检测到多种隐藏的恶意程序, 在检测的同时, 还能保证模型自身不被受到攻击, 实验证明了 SHPD 模型的有效性和安全性.

5 结束语

针对新型隐藏恶意程序, 本文提出了基于系统管理模式的隐藏进程检测模型 SHPD. 该模型利用可信视图和不可信视图对比差异, 能够有效的检测出恶意隐藏进程, 并保证自身的安全隔离性. 实验证明了该模型的有效性.

参考文献:

[1] McAfee. Threats report: First quarter 2014[EB/OL]. [2017-01-12]. <http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-ql-2014-summary.pdf>.

[2] Klister. Windows kernel level rootkit detector. 2011 [EB/OL]. [2017-01-12]. <http://www.securiteam.com/tools>.

[3] 李博, 沃天宇, 胡春明, 等. 基于 VMM 的操作系统隐藏对象关联检测技术[J]. 软件学报, 2013, 24(2): 405-420.

[4] Kortchinsky K. CLOUDBURST: A VMware guest to host escape story[C]//Black Hat. USA, 2009.

[5] Branco R R, Barbosa G N, Neto P D. Scientific but not academical overview of malware anti-debugging, anti-disassembly and anti-VM technologies[C]// Black Hat. USA, 2012.

[6] Zhang Fengwei, Kevin Leach, Kun Sun, et al. SPEC-

TRE: A dependable introspection framework via system management mode[C]// Dependable Systems and Networks (DSN), 2013 43rd Annual IEEE/IFIP International Conference on. IEEE, 2013: 1-12.

[7] William A. R, Allan Tomlinson. SMM-based hypervisor integrity measurement[C]// Cyber Security and Cloud Computing (CSCloud), 2015 IEEE 2nd International Conference on. IEEE, 2015: 362-367.

[8] Zhang Fengwei, Kevin Leach, Angelos Stavrou, et al. Using hardware features for increased debugging transparency[C]// Security and Privacy (SP), 2015 IEEE Symposium on. IEEE, 2015: 55-69.

[9] Zhang Fengwei, Zhang Hongwei. SoK: A study of using hardware-assisted isolated execution environments for security[C]// Proceedings of the Hardware and Architectural Support for Security and Privacy 2016. ACM, 2016: 3.

[10] Intel Corporation. Intel 82801DB I/O Controller Hub 4(ICH4)[Z]. Intel Corporation ,2002.

作者简介:

赵晓参 男, (1990-), 硕士研究生. 研究方向为访问控制、信息安全. E-mail: 992536614@qq. com.

任 炬 男, (1987-), 博士, 特聘教授. 研究方向为无限传感器网络.

徐 旻 男, (1988-), 博士研究生. 研究方向为信息安全.

王国军 男, (1970-), 博士, 教授. 研究方向为可信计算、隐私保护.

(上接第 110 页)

超载下 ad-hoc 网络能量均衡算法. 并通过仿真实验证明, 所提算法能有效地延长 ad-hoc 网络的使用寿命.

参考文献:

[1] 李波波, 龙昭华. 基于蚁群优化的 Ad Hoc 网络 QoS 路由[J]. 计算机工程与设计, 2016, 37(1): 12-15.

[2] 孙泽宇, 伍卫国, 曹仰杰, 等. 无线传感器网络中能量均衡参数可控覆盖算法[J]. 西安交通大学学报, 2016, 50(8): 77-83.

[3] 向敏, 唐亮, 王平. 基于 Dijkstra 能量均衡的无线 HART 图路由算法[J]. 仪器仪表学报, 2016, 37(11): 2628-2636.

[4] 徐驰, 邵华钢, 倪圣喆. 均衡网络能量的智能水滴路由算法[J]. 计算机工程与设计, 2016, 37(7): 1725-1728.

[5] 蔡钊, 马林华, 黄绍城, 等. 基于博弈论的能量均衡路由

算法[J]. 计算机工程与科学, 2016, 38(7): 1369-1375.

[6] 刘文进, 周天明, 李新春. 一种能量均衡的 WSN 多级分簇路由算法[J]. 计算机工程与应用, 2016, 52(16): 126-131.

[7] 魏炬熠, 张建军, 魏振春. 一种能量均衡的低开销 WSN 时间同步算法[J]. 计算机工程, 2016, 42(12): 5-10.

[8] 孙毅, 南婧, 武昕, 等. 基于簇的能量均衡无线传感器网络时间同步算法[J]. 计算机应用, 2014, 34(9): 2456-2459.

[9] 鲁登月, 樊建席, 刘文军, 等. 一种能量均衡的最小连通支配集构造算法[J]. 小型微型计算机系统, 2014, 35(3): 443-447.

[10] 赵梦, 李蜀瑜. 云计算环境下基于蚁群优化的任务负载均衡调度算法[J]. 电子设计工程, 2016, 24(8): 30-33.

作者简介:

张 昕 男, (1979-), 硕士, 讲师. 研究方向为计算机网络. E-mail: zhangxin7852@sina. com.