

反病毒软件防护 EANSS 模型的设计与实现

刘鑫¹, 宋显华², 吴世鹏², 李增光², 张欢², 王凯²

(1. 哈尔滨理工大学 现代教育技术中心, 黑龙江 哈尔滨 150080; 2. 哈尔滨理工大学 应用科学学院, 黑龙江 哈尔滨 150080)

摘要: 针对 ANSS(ANTivirus software shield)模型的不足, 扩展了 ANSS 模型提出了 EANSS(enhanced ANTivirus software shield)模型, 将新的攻击规则融入到新的模型中, 并且在 EANSS 模型中加入了实时监控机制、自学习机制来提高保护的能力。实验结果表明: EANSS 模型不影响系统的正常运行, 可以对反病毒软件进行更好的保护, 同时占用较低的系统内存和 CPU 占用率。

关键词: 杀毒软件; ANTivirus Software Shield 模型; 恶意软件; Application Programming Interface Hook

DOI: 10.15938/j.jhust.2017.06.014

中图分类号: TP393.08 文献标志码: A 文章编号: 1007-2683(2017)06-0072-04

The Design and Implementation of EANSS Model for Antivirus Software Protection

LIU Xin¹, SONG Xian-hua², WU Shi-peng², LI Zeng-guang², ZHANG Huan², WANG Kai²

(1. Modern Educational Technology Center, Harbin University of Science and Technology, Harbin 150080, China;

2. School of Applied Sciences, Harbin University of Science and Technology, Harbin 150022, China)

Abstract: Aiming at ANSS(ANTivirus Software Shield) model disadvantages, this paper extends ANSS model and proposes a new model named EANSS(Enhanced ANTivirus Software Shield). There are some new rules of the attack are added into the new model, and the real-time monitoring mechanism and the self learning mechanism also be joined in the EANSS model to improve the ability of protection. Experiment results show that the EANSS model not only don't influences the operations of normal applications but also can protect antivirus software from being terminated by them with low CPU performance overhead and memory write performance overhead.

Keywords: antivirus software; antivirus software shield model; malware; application programming interface hook

0 引言

恶意软件为了躲避杀毒软件的查杀, 经常采用

加密、编码、压缩、加壳等多种技术手段对抗基于特征码的杀毒软件的查杀^[1]。病毒如果检测到有调试器等进程存在就不会执行自身; 同时恶意代码中会参杂很多无意义的代码来躲避静态分析; 最具有

收稿日期: 2016-05-31

基金项目: 国家自然科学基金(61501148)。

作者简介: 刘鑫(1984—), 男, 硕士, 助理研究员, E-mail: liuxin@hrbust.edu.cn;

宋显华(1981—), 女, 博士, 讲师;

吴世鹏(1992—), 男, 学士。

攻击性也是危害最大的恶意软件会直接结束安全软件进程^[2-3]。据不完全统计,大约有 11.7% 的病毒在执行后会破坏安全软件的功能,结束杀毒软件技术成为了恶意软件的重要自我保护技术^[6-8]。因此,如何保护杀毒软件成为一个重要研究课题。

下面介绍国内外的相关研究。Hsu 等人在国际顶级期刊上发表了杀毒软件防护的学术论文,首次提出了杀毒软件防护的整体方案^[9]。Stolfo^[10-12]等人提出了 PAD(Probabilistic Anomaly Detection) 模型检测入侵。PAD 建立了一个注册表正常行为的模型,不符合正常行为的注册表访问都被认为是异常的,PAD 模型的错误率较低。Registry Prot^[13-14]是一个实时注册表监控保护的工具体,任何程序试图修改、写入注册表键值,程序都会报警,Registry Prot 对使用电脑人的知识要求较高,不具有通用性。Registry Protector^[15-16]也是保护注册表的工具,用户可以通过设置个人偏好,选择需要保护的注册表键值。Active Registry Monitor^[17-18]通过建立注册表的快照来实现监控,在系统纯净的时候建立注册表的快照,通过比较不同时刻的快照来判断修改和增加的注册表键值。Registry Cleaner^[19-20]可以在任何时刻备份和还原注册表,用户可以删除无用的键值或者还原丢失的键值。

目前为止,国内外的研究主要集中在单一层面的防护,如注册表防护、进程防护等,完整的、只针对杀毒软件防护的相关研究很少,本文基于文[9]的研究成果,进一步扩展了 Hsu 的 ANSS(ANtivirus Software Shield)模型提出了 EANSS(Enhanced ANtivirus Software Shield)模型来保护杀毒软件,扩展的模型相比于 Hus 的模型更加智能、更具有实际意义。

1 攻击规则扩展

攻击规则是模型中判断攻击行为最为重要的依据,需要进一步扩展,本文扩展了攻击规则。在相关工作中文[9]给出了七种结束杀毒软件进程的方法,基本概括如下:

1) 空调试器。恶意软件采用 API DebugActiveProcess 来把自身作为一个调试器附加到杀毒软件的进程上,当调试器退出后,杀毒软件的进程被终止。

2) DLL 卸载。恶意软件可以采用 API ZwUnmapViewOfSection 卸载杀毒软件所必须的核心 dll,

如 ntdll.dll,一旦 ntdll.dll 在杀毒软件的进程中被卸载,杀毒软件进程也直接退出。

3) 进程终结。恶意软件可直接通过 API NtTerminateProcess 或者 ZwTerminateProcess 在内核层直接结束杀毒软件进程。

4) 发送关闭消息。恶意软件通过 FindWindow 函数寻找操作系统中符合杀毒软件名称的窗口,之后发送进程退出的消息如 WM_CLOSE 或者 WM_QUIT,可以通过 PostMessage 或者 SendMessage 函数不断发送关闭消息给杀毒软件,直到杀毒软件进程退出。

5) 鼠标模拟方法。恶意软件可以获取一系列的鼠标事件,再通过计算屏幕宽度高度等方式,找到杀毒软件的托盘图标,采用模拟正常客户退出的方法结束杀毒软件。

6) 线程终结。恶意软件可以通过 API TerminateThread 来结束杀毒软件进程中的线程,直到杀毒软件进程完全退出。

7) 注册表修改。恶意软件可以通过 ZwOpenKey 和 ZwSetValueKey 来修改注册表键值,一种方法可以修改注册表来映像劫持,杀毒软件会无法正常运行;另外一种方法,删除杀毒软件的自启动键值。

在此基础上,经过深入研究,本文提出了以下几种结束杀毒软件方法,加入攻击规则。

1) 通过 OpenProcess API 以 PROCESS_CREATE_THREAD | VM_OPERATION 方式打开进程,然后创建远程线程,使线程开始地址是 ExitProcess 参数为 0,程序会自行退出。

2) 发送键盘 ESC 单击的消息。这点和上边的第四点发送关闭消息有些类似,不过这里发送的是键盘 ESC 消息,因为一般的软件发送 ESC 消息,程序都会退出。

3) 卸载驱动。杀毒软件为了获得底层权限、实现 Hook 等功能,一般会加载自身的驱动,卸载驱动的 API 是 ZwUnloadDriver,恶意软件可以通过卸载杀毒软件的驱动来结束杀毒软件进程,或者使杀毒软件的核心功能失效。

4) 对驱动加载的检测。加载驱动操作本身不会结束杀毒软件,不过要调用原始 API 需要进入 Ring0 层,任何企图加载驱动进入 Ring0 层的操作都是危险的,在本文中将 ZwLoadDriver、ZwSetSystemInformation、SystemLoadAndCallImage 都认为是敏感 API,因为它们会加载驱动获得底层权限。

2 EANSS 模型

通过对 ANSS 模型的仔细分析,发现 ANSS 模型具有很大的局限性,主要有以下几方面不足:

1) ANSS 模型基于已知的杀毒软件结束方法,对于未知的杀毒软件结束方法没有防护能力,例如本文提出的方法,ANSS 模型就没有检测能力,本文将新的结束杀毒软件的方法制定成规则集成到 ANSS 模型中。

2) 对反 Hook 型结束方法缺乏防护能力,即恶意软件先恢复 SSDT 表之后再结束杀毒软件,ANSS 模型失去防护能力。

3) 缺乏学习机制。ANSS 模型不能够自学习,这样随着时间的推移,会逐渐失去作用。

针对 ANSS 模型的不足,本文在多个方面进行改进,本文改进后的 ANSS 模型称为扩展的 ANSS 模型 EANSS,其基本结构如图 2 所示。

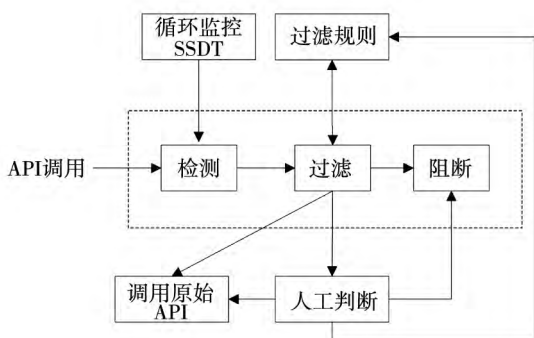


图1 EANSS 模型结构图

EANSS 模型主要由检测模块、过滤模块和阻断模块三部分组成。当一个程序试图调用敏感 API 时,检测模块在敏感 API 被执行之前拦截程序流的执行并将执行流转向过滤模块;过滤模块是最重要的模块,对于每个敏感 API 都有一个相对应的 EANSS API,同时对应一条规则,通过这些内嵌的规则检查 API 调用的参数,过滤出可能结束或者停止杀毒软件的 API 调用;阻断模块通知给用户阻断了一次对杀毒软件的攻击。对于正常的 API 调用,直接调用系统原始的 API 代码。

和 ANSS 模型采用相同的方法,EANSS 通过 SSDT(System Service Descriptor Table) Hook 技术来检测 API 调用,SSDT 就是系统服务描述表,里面存取

了操作系统中每个 API 的地址,EANSS 模型对相关的 API 进行了 Hook,这样监测调用 API 就可以及时发现攻击。EANSS 模型加入了 SSDT 循环监控机制,防止恶意软件再调用 API 之前先进行 SSDT UnHook 来躲避检测,EANSS 模型每隔固定时间就会检查 SSDT Hook 状态,发现被修改立即进行再次 Hook;

最重要的模块依然过滤模块,过滤模块检测敏感 API 的参数来判断是否是恶意行为,EANSS 在以下几个方面进行了改进:

1) 丰富了过滤模块的规则库,将本文提出的 4 种方法加入到了规则库,发送 ESC 消息的行为、API 函数 ZwUnloadDriver、ZwLoadDriver、ZwSetSystemInformation、SystemLoadAndCallImage 都加入到了规则库,卸载驱动和加载驱动都是敏感行为,卸载驱动会导致杀毒软件退出,而通过加载驱动恶意软件可以获得内核权限,这是调用原始 API 的直接途径,不调用原始 API,恶意软件的很多功能无法实现。

2) 引入人工判断机制。即使调用了相关 API 也不能完全判定是恶意行为,通过参数可以判断出大多数的恶意行为,但是如果存在不在规则之内的安全进程软件存在,则会出现误报率,为了解决这个问题,EANSS 模型引入了人工判断机制,交由人工来判断是否为恶意行为。

3) 引入学习机制。ANSS 模型完全依靠内嵌规则,规则不能动态的学习更新,EANSS 模型引入学习机制,人工判断的结果会直接被规则学习到,为以后的判断制定了规则。

为了和 ANSS 模型进行明显的对比,EANSS 模型同样在 Microsoft Windows XP Professional Service Pack 3、Intel 2.60 GHz Core 2 CPU G1610 和 4096 MB RAM 环境下实现,首先来比较 EANSS 模型是否与系统功能产生冲突,如表 1 所示:

可以发现,ANSS 模型和 EANSS 模型的运行不会对系统常见操作和杀毒软件的使用产生任何影响。

表1 ANSS 模型与 EANSS 模型是否与系统冲突对照表

操作	ANSS 模型	EANSS 模型
系统更新	正常	正常
系统扫描	正常	正常
人工关闭杀毒软件	正常	正常
人工开启杀毒软件	正常	正常
关闭计算机	正常	正常

为了对比 ANSS 模型 EANSS 模型的实际效果,我们做了实验效果对比,实验结果对比结果如表 2 所示,表中符号 X 代表模型不能对此种技术进行保护,符号√表示模型可以对这种技术进行保护,表 2 显示 ANSS 模型可以对进程终结、鼠标模拟、注册表修改、关闭消息、空调器、DLL 卸载、线程终结方法进行保护,不过对于新的攻击方法 ESC 消息、驱动卸载和驱动加载方法不能提高很好的保护,EANSS 模型可以对全部的方法进行保护。表 3 显示了 ANSS 模型和 EANSS 模型占用系统性能,从表中可以看出 ANSS 模型和 EANSS 模型对 CPU 和内存的占用率都很低。

表 2 ANSS 模型与 EANSS 模型保护功能对照表

方法	ANSS 模型	EANSS 模型
进程终结	√	√
鼠标模拟	√	√
注册表修改	√	√
关闭消息	√	√
空调器	√	√
DLL 卸载	√	√
线程终结	√	√
ESC 消息	X	√
驱动卸载	X	√
驱动加载	X	√
规则学习	X	√

表 3 ANSS 模型与 EANSS 模型占用系统性能对照表

性能	ANSS 模型	EANSS 模型
CPU 占用率	0.42%	0.43%
内存占用率	1.77%	1.75%

3 结 论

针对 ANSS 模型的一些不足,本文提出了 EANSS 模型,扩展了内嵌的攻击规则,将 ESC 关闭消息、卸载驱动 API、加载驱动 API 等新的方法加入到内嵌攻击规则,提升了模型检测的准确率;引入学习机制来不断丰富攻击规则库,对于新的攻击同样可以保证检测的准确性;同时增加的循环检测机制,可以更有效的检测恶意软件。实验结果表明 EANSS 模型并不与操作系统发生冲突,同时占用较

少的系统资源,EANSS 模型具有更好的效果和优点。

参 考 文 献:

- [1] 诸葛建伟,唐勇,韩心慧,等. 蜜罐技术研究与应用进展 [J]. 软件学报,2013,24(4):825-842.
- [2] 李海峰,王栓奇,刘畅,等. 考虑测试工作量与覆盖率的软件可靠性模型 [J]. 软件学报,2013,24(4):749-760.
- [3] 赵会群,孙晶,张爆,等. 嵌入式 API 测试套生成方法和技术 [J]. 软件学报,2014,25(2):373-385.
- [4] 陈锦富,李青,毛澄映,等. 基于安全规则变异的 Web Services 安全性测试方法 [J]. 计算机学报,2013,36(10):1967-1981.
- [5] 杨欢,张玉清,胡予濮,等. 基于多类特征的 Android 应用恶意行为检测系统 [J]. 计算机学报,2014,37(1):15-26.
- [6] 席荣荣,云晓春,张永铮,等. 一种改进的网络安全态势量化评估方法 [J]. 计算机学报,2015,38(4):749-758.
- [7] 翟继强,乔佩利. Windows 平台下 Snort 和 IPSec 联动实现 [J]. 哈尔滨理工大学学报,2012,17(3):96-99.
- [8] 鲁颖欣,王健,齐宏卓. 模糊判断在网络安全风险评估中的应用研究 [J]. 哈尔滨理工大学学报,2014,19(1):36-39.
- [9] HSU Fu-Hau, WU Min-Hao, TSO Chang-Kuo, et al. Antivirus Software Shield Against Antivirus Terminators [J]. IEEE Transactions On Information Forensics And Security, 2012, 7(5): 1439-1447.
- [10] STOLFO S J, APAP F, ESKIN E, et al. "A Comparative Evaluation of two Algorithms for Windows Registry Anomaly Detection," J. Comput. Security, 2005, 13(4): 659-693.
- [11] 朱建明,宋彪,黄后发. 基于系统动力学的网络安全攻防演化博弈模型 [J]. 通信学报,2014(1):54-61.
- [12] 陈珂,柯文德,王爱国,等. 基于沙盒技术的行为分析系统研究 [J]. 计算机技术与发展,2015(8):166-169.
- [13] 陈进,夏虞斌,臧斌宇. 基于新型 VMI 技术的内核 Rootkit 检测方案 [J]. 计算机应用与软件,2015,32(7):266-271.
- [14] 郝增帅,郭荣华,文伟平,等. 基于特征分析和行为监控的未知木马检测系统研究与实现 [J]. 信息安全,2015(2):57-65.
- [15] 赵巍,秦杰,郭佳,等. 基于不确定性推理的木马检测技术 [J]. 信息工程大学学报,2014(5):617-621.
- [16] 夏爱民,张宏志,杨伟锋. 基于综合行为特征的木马检测技术研究 [J]. 信息安全与通信保密,2014(6):109-113.
- [17] 侯仁平,李孝忠. 基于 Petri 网的病毒攻击建模 [J]. 天津科技大学学报,2014(1):69-74.
- [18] 李明. 网络型病毒分析与计算机网络安全技术的实践探究 [J]. 网络安全技术与应用,2015(5):20-21.
- [19] 罗文华. 基于内存注册表信息解析用户行为方法研究 [J]. 信息安全,2013(6):72-74.
- [20] 张永刚. 一种新型系统安全检测软件的设计与实现 [J]. 电气自动化,2015(5):27-29.

(编辑:王 萍)