基于行为分析的主动防御技术及其脆弱性研究

罗晓波 王开建 徐良华
(江南计算技术研究所 江苏 无锡 214083)

摘 要 主动防御技术的出现弥补了传统的病毒扫描技术和防火墙技术的不足,给计算机系统提供了更加严密的保护。首先阐述了基于行为分析的主动防御技术的原理和实现,然后从几个方面对其脆弱性进行了分析,并研究了突破这种主动防御系统的可行性,最后提出了一些方法来增强主动防御。

关键词 主动防御 病毒扫描 防火墙 挂钩 SSDT

ON ACTIVE DEFENCE TECHNOLOGY BASED ON BEHAVIOUR ANALYZING AND ITS VULNERABILITIES

Luo Xiaobo Wang Kaijian Xu Lianghua
(Institute of Jiangnan Computer Technology, Wuxi 214083, Jiangsu, China)

Abstract The emergence of active defence technology makes up the pitfalls of tradition virus scan and firewall technologies, and gives an all-around protection on computer system. First, the principle and implementation of active defence technology based on behaviour analyzing are discussed, and then the vulnerabilities of this technology are analyzed from several aspects, the possibility of breaking through the system of active defence technology are studied as well, and at last several methods are presented for strengthening the active defence technology.

Keywords Active defence Virus scan Firewall Hook System service descriptor table

0 引 言

计算机的普及和 Internet的飞速发展极大地推动了社会信息化进程。人们在享受计算机网络所带来的便利的同时,却不得不面对网络攻击所导致的安全问题。破坏文件的病毒可能摧毁计算机上存储的所有文件,给用户造成无法挽回的损失;窃取信息的间谍程序可能导致账号密码的泄漏或丢失、商业敏感信息被非法窃取、国家重要机关的涉密文件被盗等;木马程序[1]除了有信息窃取的功能外,还可能导致计算机完全被黑客控制;而蠕虫的大规模爆发则可能使得计算机系统和计算机网络趋于瘫痪,并失去应有的功能。近年来,网络攻击行为又有了新变化,从炫耀型攻击发展到有目的的攻击,攻击手段从以往的单凭技术发展到技术和社会工程学的综合运用,攻击的对象从系统应用层发展到系统内核[2]。这些网络攻击行为的显著变化使得传统的杀毒软件和防火墙面临很大的挑战,主动防御技术就是在这种背景下出现的。

1 主动防御的原理和实现

主动防御是最近两年才开始的新技术,主要采用了基于行为分析的行为杀毒技术。这里的杀毒包括查杀病毒、蠕虫、间谍软件和木马等,为叙述简便,以下统称病毒。

1.1 主动防御的原理介绍

传统的防病毒技术主要是通过扫描文件,分析文件的特征码,并与病毒库中的特征码进行比对,如果匹配,则认为该文件是病毒。这种病毒扫描的缺点是明显的,由于病毒变化多样,其特征码也是不断变化的,因此杀毒软件的病毒库更新总是落后于病毒的更新。一个新生的病毒从产生到大规模爆发这一阶段,在计算机系统上畅行无阻,等到杀毒软件收集到特征码并更新病毒库时,该病毒已经造成了巨大的损失。

基于行为分析的行为杀毒技术(简称行为杀毒技术)主要是通过监视程序的行为来判断程序的危害性。程序运行时会调用各种应用编程接口(API),行为杀毒技术通过监视这些 API的调用,即可了解程序的运行状态,从而可以判断出程序的危害性。这种杀毒技术的优点是原理简单、操作简便,能够准确地掌握程序的行为,缺点是不能合理判断程序的一系列行为的逻辑关系,容易出现很多的误报,使得用户被频繁的报警所困扰。但是由于其简便易行,目前主流的安全软件大都采用了这种方式来保护计算机系统,如卡巴斯基、ZoneAlarm Pro、趋势科技、瑞星等。

本文关于主动防御技术的研究对象是 Windows操作系统下的基于行为分析的行为杀毒技术。

1.2 主动防御的实现

要实现行为杀毒技术,必须要能够监视程序的一举一动,而这种监视是靠挂钩 API函数来实现的。程序能够调用的 API数量众多,而一个关键的必经之路就是 Native API的调用,因此安全软件一般选择挂钩 Native API来达到有效监视程序行为的目的。其实现过程大致如下:首先挑选出有利于监控程序行为的

收稿日期:2007 - 11 - 08。国家 863高技术研究发展计划基金项目(2006AA01Z431)。罗晓波,硕士生,主研领域:信息安全。

若干 API,将这些 API的名字记录下来,然后获取到 ntoskrnl exe (或其他内核,视计算机系统而定[3])所导出的 KeSeviceDescriptorTable变量[4],得到这些 API在操作系统内核中的入口地址,修改这些入口地址到自己构造的一些 API(这些 API的参数和原 API相同),在进行一定的处理后再次调用原 API,这样就达到了监控程序行为的目的了。图 1描述了在挂钩一个函数 fun()前后同一个程序调用该函数的流程。

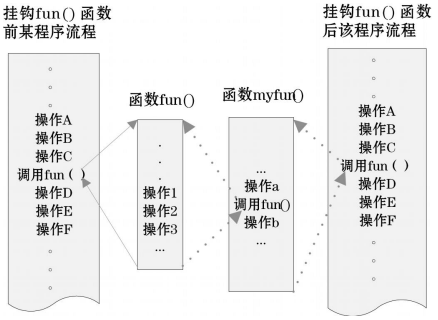

图 1 HOOK函数示意图

2 主动防御在主流安全软件中的实现

主动防御系统的实现一般要经过以下三个阶段:确定监控行为、确定 Native API、确定监控逻辑。确定监控行为表明了一个主动防御的总体部署,挂钩 API是一个主要手段,而确定监控逻辑则是最复杂、最能体现主动防御优劣的一个步骤。

2.1 确定监控行为

当前主流的安全软件监控的程序行为包括:注册表修改、服务注册、进程读写、驱动程序加载、内存映射、动态库注入、系统文件修改、进程创建、程序输入输出重定向等,这些行为是病毒所惯用的手段,对其进行监控将能够非常有效地保护主机的安全。例如,彩虹木马 (Bifrost exe)的服务端在得到执行后,将会创建一个傀儡 IE进程,然后将木马体注入到该 IE进程,同时修改注册表启动项,使得木马随系统开机运行,这一系列动作就包含了进程创建、进程读写、动态库注入、注册表修改等。而具备了主动防御系统后,木马的这一系列行为将会引起报警或直接被阻止运行,计算机系统得到了最大程度的保护。目前市面上的安全软件种类繁多,监视的行为稍有不同,因而在保护计算机系统安全的效果上也存在差异。造成这种差异是因为安全软件对于威胁定位的侧重点不同,比如一款安全软件的主动防御将一个添加服务的行为定义为正常行为,而另一款可能将其视为危险动作而发出警报。

2.2 确定 Native API

在确定了需要监控的行为之后,主动防御系统下一步的工作是需要决定挂钩哪些 API,并对这些 API的调用采用何种处理来达到监控的目的。这是一个复杂的过程,而且不同的安全软件采用的方法可能有很大的区别。比如对动态库注入的监控,这种注入调用的 API顺序为 OpenProcess、WriteProcessMemory、CreateRemoteThread,反映到 Native API层上,分别是 NtOpenProcess、NtWriteVirtuaMemory、NtCreateThread,在 ZoneAlarm Pro 7.0的主动防御系统中,它挂钩了 NtOpenProcess并进行相应的处理,而在卡巴斯基 6.0的主动防御中,则挂钩的是 NtWriteVirtuaMemory和 NtCreateThread。由于选择挂钩的 API有所不同,不同的主动防御系统的监控效果也有所侧重。

2.3 确定监控逻辑

监控逻辑是指根据怎样的条件来确定被监控对象是否构成威胁。在选定了需要挂钩的 Native API之后,接下来的工作是如何操作针对这些 API的调用。一个 API通常会带有若干参数,其中一些参数对于监控程序行为具有重要的作用,主动防御系统通常会对这些参数进行审计,并根据一定的逻辑来判断本次 API调用是否可能构成危害,如果构成危害,则发出警报或直接阻止其运行,否则将调用原 API使得程序继续运行。这种处理过程在不同的安全软件实现上也存在较大的分歧,有的安全软件的主动防御系统可能仅仅根据单个 API的调用动作而确定该动作的危害,而另一些主动防御系统可能经过若干 API调用的组合来综合判断程序行为的危害性。由于这种差异,主动防御系统的监控能力和效果也有区别。

经过这三个阶段之后,程序的主要行为就被监控起来,主动防御系统就基本实现了。由于每一阶段的实现都存在着可变因素,因此目前的主动防御系统之间存在一些差异。

3 安全软件主动防御系统的脆弱性分析

由于实现上的差异,不同的主动防御系统的监控范围和效果存在一定的差异,这些差异可能会成为一个主动防御系统的薄弱环节,下面分别讨论实现主动防御的三个阶段中可能存在的脆弱性。

3.1 监控的行为不全

程序的行为具有多样性,同一个行为作用在不同的对象上可能产生完全不同的效果,因此,选择哪些行为、哪些对象进行监控,在不同的主动防御系统上也存在区别。正是由于这种区别的存在,主动防御系统可能由此而产生一定的脆弱性。比如,感染系统文件的行为是常见的病毒行为之一,但一些安全软件的主动防御系统并没有对这种行为进行严密的监控,往往在若干时间之后通过病毒扫描才发现系统文件被感染,甚至根本没有察觉到系统文件被感染。这就给系统安全留下了很大的隐患,因为系统文件常常是系统启动就加载的,而且加载后一般具有管理员以上的权限,系统文件被感染后,所产生的危害也是巨大的。

虽然 Windows系统自 2000系列开始就引入了文件保护机制,但目前黑客技术的发展使得关闭 Windows系统的文件保护变得相当容易,互联网上已经广泛流传了相关的源代码[5],攻击者只需要根据需求稍加修改,便可以将这一功能嵌入到病毒程序中了。关闭了 Windows系统的文件保护之后,病毒程序就可以肆意修改系统文件 (也称为感染系统文件,主要是通过修改 PE[6]文件体)了。修改后的系统文件在计算机重启之后将发挥作用,它可以在系统初始化阶段便执行攻击代码,而这一动作在安全软件启动之前,因此安全软件根本无法检测到,从而给系统带来很大的威胁。

3.2 监控的 Native API不完备且 API层次偏高

一个程序行为一般要以一定的顺序调用若干 API,而主动防御系统一般选择某个或某几个 API来进行监控,而且,即使达到同一个目的,程序可能选择调用不同类型的 Native API来实现,因而存在漏报的可能。比如,常用的修改注册表方法是调用 NtOpenKey、NtQueryValueKey、NtSetValueKey等 API,但也存在一个另类的方法同样可实现对注册表的修改[7],就是编辑 hive文

件 (即注册表备份文件)的方法,这种方法通过 NtSaveKey备份注册表项、编辑备份的 hive文件、NtRestoreKey将修改后的 hive文件还原到注册表的方法来实现。

Native API是由系统内核 ntoskrnl exe最终执行的系统服务函数,它向上层 Windows API提供调用接口,属于相对层次较低的 API。但是随着网络技术的发展,病毒技术也得到了飞速的发展,一些病毒程序采用了比 Native API更加底层的技术来实现。近年来被广泛关注的 RootKit技术甚至实现了另一套接口来操作硬盘数据,在驱动层执行任务,并操作 NDIS接口来收发网络数据[8],完全不受一些主动防御系统的监控,给系统安全带来了极大的威胁。

3.3 监控逻辑不够严密

监控逻辑决定了一个主动防御系统的监控效果的优劣。Native API函数、函数的参数、调用者都是监控逻辑需要考虑的对象,一个好的监控逻辑往往还需要根据一系列 API调用来进行判断。因此,监控逻辑不够严密的情况或多或少存在于各个主动防御系统中,攻击者对一个主动防御系统进行详细的测试分析后,就可能发现其监控逻辑上的缺陷,从而可能绕过该主动防御系统的监控。例如,系统进程 svchost exe发起的网络连接事件,主动防御系统通常是允许的,但是这个进程是由谁创建的,往往会被主动防御系统忽略,攻击者完全可以创建一个傀儡 svchost exe进程,而后通过其向外发送数据,这就是在调用者上存在的逻辑缺陷。再比如,主动防御系统一般都会监控注册表的 Run键,但是攻击者在修改注册表时,可能会将注册表操作函数的一个参数设置为 Run键的上一级目录,另一个参数拼装到 Run键下,从而绕过一些监控逻辑过于简单的主动防御系统。

由此可见,对一个主动防御系统作全面仔细的测试分析后,就可能发现该主动防御系统的薄弱环节,如果对这一薄弱环节进一步研究分析,就可能破坏或绕过这种主动防御系统的监控。表 1列出了部分安全软件的主动防御系统在默认配置下可能存在的薄弱环节。

表 1　部分安全软件的主动防御系统存在的薄弱环节

项目 / 主动防御	监控行为不全	监控 API不全	监控逻辑不严密
卡巴斯基 6.0 安全套装	病毒可以感染部分系统文件、编辑注册表 hive文件	病毒可以用 ZwSetSystemInformation加载驱动	对挂钩函数的参数校验不严密,可能导致蓝屏
ZoneAlarm 7.0	病毒可以感染部分系统文件、编辑注册表 hive文件	病毒可以用正常方法加载驱动	对挂钩函数的参数校验不严密,可能导致蓝屏
趋势科技 2007 安全套装	病毒可以感染部分系统文件	病毒可以用 ZwSetSystemInformation加载驱动	病毒可能伪造受信任的进程

4 增强主动防御系统的安全性

根据前面对主动防御系统脆弱性的分析,可以采取以下措施增强主动防御系统的安全性。

4.1 完善监控行为

病毒程序要发挥作用,势必修改系统中的敏感资源,这些敏感资源包括注册表文件、系统文件、系统运行时的内核数据,对这些行为进行严密的监控,将能够非常有效地阻止病毒的入侵。

修改这些敏感资源的方法比较多,安全软件的开发人员必须掌握丰富的知识,及时了解网络攻击的新技术、新思路,做到有的放矢。

4.2 丰富挂钩的 API,增加监控的层次

程序达到同一个目的可以选择调用不同的 API来实现,因此安全软件的主动防御系统必须能够监控所有可能造成危害的 API,从入口上封堵病毒的入侵。另外,主动防御系统应该增加监控的层次,在不同的层次上防范病毒的攻击。

4.3 优化监控逻辑

监控逻辑的优化需要主动防御系统考虑更多的因素,包括 API的组合、参数之间组合、调用者身份审核等等,将这些因素综合起来,形成一份有意义的知识库,将能够有效地阻止针对监控逻辑的攻击。

5 结束语

主动防御技术的出现弥补了传统的病毒查杀技术和防火墙技术的不足[9]。在病毒查杀方面,主动防御技术的应用使得杀毒引擎摆脱了仅仅依靠病毒特征码进行扫描的尴尬局面,它通过监控程序行为来及时发现病毒程序并阻止程序的运行。有了主动防御系统的支持,绝大多数的病毒即使产生变种也无法逃过主动防御的监控。在防火墙方面,主动防御的出现使得防火墙不再局限于依据通信协议和端口来进行数据包的拦截,结合主动防御技术,防火墙将能够根据防御策略建立一个立体的防护网,在程序企图发起网络连接之前,主动防御系统就能够发现并提示程序的这种企图。

然而由于主动防御技术是一个新兴的技术,在系统实现上还存在着脆弱性,一些不足可能会被攻击者利用,给计算机系统安全带来威胁。只有充分理解主动防御的原理和实现,才能够充分发挥主动防御的作用,扬长避短,最大限度地保护计算机系统的安全。

参 考 文 献

[1] 邓吉,柳靖. 黑客攻防实战详解 [M]. 北京:电子工业出版社, 2006: 325-373.

[2] Greg Hoglund, James Butler. Rootkits: Subverting the Windows Kernel [M]. Pearson Education, 2006.

[3] Mark E Russinovich, David A Solomon. Microsoft Windows Internals, Fourth Edition[M]. Microsoft Press, 2005.

[4] Prasad Dabak, Milind Borate, Sandeep Phadke. Undocumented Windows NT[M]. M&T Books, 1999.

[5] Ntoskrnl Windows File Protection: How To Disable It On The Fly [CP]. https://www.rootkit.com/newsread.php?newsid=212, 2004(11).

[6] 段钢. 加密与解密 (第 2版) [M]. 北京:电子工业出版社, 2003.

[7] xyzreg 突破主动防御之注册表监控篇 [EB/OL]. (2007-02-26) http://www.xyzreg.net/.

[8] Addylee. 基于 PassThru的 NDIS中间层驱动程序扩展 [EB/OL]. (2006-05-05) http://www.xfocus.net/articles/200605/865.html

[9] William R Cheswick, Steven M Bellovin. 防火墙与英特网安全 [M]. 戴宗坤,罗万伯,等,译. 北京:机械工业出版社, 2000.