doi :10.3969/j.issn.1671-1122.2014.11.007

基于 Minifilter 的多层次文件操作行为提取技术研究

张陈磊，周安民，刘亮，卿粼波

（四川大学电子信息学院，四川成都 610065）

摘　要：文章研究了对于不同层次的文件操作行为的提取及监控，旨在针对目前存在的绕过过滤驱动的检测方法进行改进，更加有效地针对恶意软件行为进行监控，多层次提取其文件操作的技术。文章首先概述了文件过滤驱动技术工作原理及当前应用现状，介绍了目前被广泛应用的微文件过滤驱动（Minifilter）技术的开发原理、步骤和应用领域。随后对文件操作的底层行为全过程进行了分析，并对 Minifilter 在其中的检测原理进行了相关介绍，对其安全性进行分析，提出当前能绕过过滤驱动检测的几种方法原理，包括通过增加过滤驱动以及 Hook 派遣函数等原理绕过过滤驱动，从而造成过滤驱动无法检测。列出了目前存在的从不同层次绕过过滤驱动的几种攻击方法，包括附着新的过滤驱动，直接访问内核，对底层文件结构的派遣函数进行不同的 Hook 等。针对其攻击原理进行分析，提出对应的检测方法。通过在原有 Minifilter 的基础上添加以上几种检测方法，可实现对目前存在的多种攻击手段进行多层次检测，从而添加相应的防护措施。在之后对改进后的过滤驱动进行功能及性能上的针对性测试中，表明改进后的检测驱动能利用更小的时间成完成更深层次的检测。因此，改进后的行为提取技术能绕过普通文件过滤驱动的恶意行为进行拓展检测，更深层次地提取恶意软件的文件操作行为，从而实现对目标程序的可疑文件操作进行更加全面的监控。

关键词：微过滤驱动；多层次；文件监控；行为提取

中图分类号：TP309　　**文献标识码：**A　　**文章编号：**1671-1122（2014）11-0041-05

中文引用格式：张陈磊，周安民，刘亮，等. 基于 Minifilter 的多层文件操作行为提取技术研究 [J]. 信息网络安全，2014，(11)：41-45.

英文引用格式：ZHANG C L，ZHOU A M, LIU L, et al. Multi-level File Operations Recording System Based on Minifilter Driver[J]. Netinfo Security, 2014,(11):41-45.

Multi-level File Operations Recording System Based on Minifilter Driver

ZHANG Chen-lei, ZHOU An-min, LIU Liang, QING Lin-bo

(*College of Electronics and Information Engineering, Sichuan University, Chengdu Sichuan* 610065, *China*)

Abstract: This paper studied for different levels of extraction and monitoring the behavior of file operations, aimed at the existing bypass filter drivers detection method was improved, more effective against malicious software behavior, multi-level technology to extract the file operations. Firstly the paper introduces the file filter driver technology , principle and current application situation,

收稿日期：2014-06-23

基金项目：教育部高等学校博士学科点专项科研基金 [20110181120009]

作者简介：张陈磊（1990-），男，陕西，硕士研究生，主要研究方向：信息系统安全保护技术；周安民（1963-），男，四川，硕士生导师，研究员，主要研究方向：信息系统安全保护技术；刘亮（1982-），男，四川，博士研究生，主要研究方向：网络系统与信息安全；卿粼波（1982-），男，四川，副教授，博士，主要研究方向：通信与信息处理。

通讯作者：张陈磊 15392808@qq.com

then introduces the widely application of micro file filter driver (Minifilter) technology development principle, steps and application field. Subsequent to the underlying behavior of file operations process are analyzed, and the Minifilter detection principle of the related introduction. To analyze its security and puts forward several methods of current can bypass the filter drivers detection principle. Including by adding filter drivers and send Hook function principle to bypass filter drivers, which the filter driver behavior cannot be detected.Lists the existing several attack methods from different levels to bypass the filter driver, including attached new filter drivers, direct access to the kernel, the sending of the underlying file structure function of different hook skills and so on. According to its attack principle is analyzed, puts forward corresponding detection methods.By adding the above on the basis of the original Minifilter several detection methods, which can realize to test the present a variety of means of attack, so as to add multi-layered protective measures. And then the improved filter drivers for targeted on the function and performance test, shows that the improved test drive to be able to use a smaller time cost to complete more deeper detection. Therefore the behavior of the improved extraction technology can bypass the normal file filter driver to expand to detect malicious behavior, the extraction of deeper malicious software file operations, so as to realize the target of suspicious file operations for a more comprehensive monitoring.

Key words: minifilter driver; multi-level; file monitoring; record behavior

0 引言

文件过滤驱动技术作为目前提取记录恶意软件文件行为的技术被广泛应用。该技术可通过附着在目标设备上的驱动设备拦截往来请求，以及创建 IRP、相关例程来实现对文件行为操作的过滤。微文件过滤驱动（即 Minifilter）作为目前比较具有代表性的轻型过滤驱动，较为广泛地应用在安全检测技术中，成为提取文件行为比较有效的方法。然而，当前从安全方面来讲，Minifilter 的文件行为提取技术仍旧面临被新型恶意软件通过附着驱动、对文件系统的直接访问、派遣函数更改等方法从驱动层绕过，无法监控记录其行为。本文将在微文件过滤驱动的基础上，针对目前可绕过一般 Minifilter 的攻击方法进行相应的改进，旨在针对目前出现的绕过文件过滤驱动的方法提出多层次的防御措施，对基于 Minifilter 的文件行为监控系统进行改进，以实现更深度的行为提取。

1 文件过滤驱动技术

过滤驱动 [1,2] 是中间层驱动的一种，可监视、拦截和修改系统发往下层驱动的 I/O 请求包（I/O Request Packet，IRP）流，利用过滤驱动可以给现有的驱动增加新的功能。例如，监控系统的行为，防止某个文件被删除，以及对目标程序的文件操作进行监控、记录，并对文件行为的安全性作出判断。

文件系统微过滤驱动 [3–5] 是 Microsoft 公司提出的一种新的过滤器模型，它位于 I/O 管理器和文件系统之间。文件系统微过滤驱动忽略了对文件系统驱动的内部实现，将关注点转到对文件的操作上，让开发者可以专注于功能上的实现，这样可以加快开发与除错的效率。另外，文件系统微过滤驱动通过微过滤管理器 [6] 提供了一个平台，任何符合规范的微过滤驱动都可以加载，被过滤管理器统一管理，提高了兼容性。本文中提出的改进文件操作提取方法的方案，就是基于微过滤驱动实现的，但微过滤驱动受过滤管理器制约，其灵活性会受到一些限制，且不能做到系统的更底层。

2 绕过过滤驱动的方法及对应检测技术

2.1绕过过滤驱动监控的方法

图 1 表现了当前用户进行文件操作时的行为分析，即应用层访问 FSD（File System Driver），自上而下的正常流程，以及从不同层次绕过过滤驱动监控的几种方法原理 [7,8]。

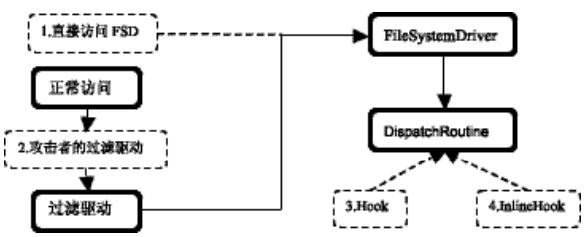

图1 绕过Minifilter监控的几种方法

其中实现框部分代表文件操作的正常行为，用户进行文件行为操作时将访问底层的 FSD，通过调用其派遣函数，

实现对自己应用层进行的相关操作，同时将操作结果体现在应用层。

在添加了 Minifilter 后，在其监控行为下，微文件过滤驱动将绑定在 FSD 之上。用户对文件系统的所有操作行为都将经过过滤驱动访问内核 FSD 区域，因此在其过程中，过滤驱动可以实现记录及监控其行为的作用。

图 1 中虚线框部分则代表了攻击者针对微文件过滤驱动当前检测原理进行的若干攻击方法。主要攻击手段有两种主要思路：绕过过滤驱动直接对文件系统驱动进行访问；对当前存在的过滤驱动进行 HOOK 修改。

2.1.1 直接访问FSD的内核级别文件访问

FSD 层是文件 API 函数经过本地系统服务层（Native API）最后到达的驱动层次。在正常访问中，用户在应用层的操作到达底层后将调用 FSD 的派遣函数，由派遣函数完成底层对应的操作。文件系统过滤驱动通常附着在正常的文件系统之上，所有访问 FSD 的文件行为操作将被过滤驱动所记录，从而实现监视和过滤系统内所有程序的文件访问。文件系统驱动栈就是由这一连串的附着起来的过滤驱动组成，在内核内以链表形式存储。正常的 FSD 信息存储在一个 VPB（Volume Parameter Block）结构中，VPB 指卷参数块，主要作用是把实际存储媒介设备对象和文件系统上的卷设备对象联系起来。这也为攻击者留下了直接绕过过滤驱动访问底层 FSD 的可能，当前存在的主要攻击方法为攻击者使用PDEVICE_OBJECT IoGetBaseFileSystemDeviceObject（IN PFILE_OBJECT FileObject）这个未公开的内核函数得到与文件对象相关的卷设备对象，并以它作为目标发送 IRP 包。该 IRP 包直接以 FSD 为目标发送，绕过过滤驱动，使其无法监控到相关行为。

2.1.2 在FSD上挂载新的文件过滤驱动

文件过滤驱动监控行为原理为附着在 FSD 高层，记录经过其传输的行为。在攻击中可以利用该原理，生成自己的过滤驱动绑定 FSD 以及文件卷设备，新生成的过滤驱动将在检测软件已加载的过滤驱动之上，从而可优先获得应用层发送至下层的 IRP 包进行更改。例如，攻击者可截获用户 IRP_CREATE 标志的创建行为，阻止其发往下层并返回失败，这将会造成应用层端无法访问的结果。由于其全过程并未传至监控过滤驱动及以下层级，造成无法监控提取到任何文件操作行为信息，从而绕过检测软件的过滤驱动检测。

2.1.3 替换DispatchRoutine的FSD Hook

正常的文件操作行为以 IRP 形式发送到 FSD 后，会根据其主功能号进行判断，选择其对应的派遣函数进行调用，从而完成底层操作。因此，由派遣函数端入手完成攻击行为也成为攻击者的一种攻击方法。通过读取内核中原本 FSD 驱动的 .INIT 段或者 .TEXT 段，查找其 DriverEntry 函数，在它的 DriverEntry 函数中通过特征码搜索的方法找到 FSD 驱动对象中各个派遣函数的地址。可在派遣函数中完成内核中其他操作，由于行为在过滤驱动更底层完成，并不经过其监控范围，因此过滤驱动无法监控到其行为，从而攻击者通过修改派遣函数实现绕过过滤驱动的操作。

2.1.4 Inline Hook DispatchRoutine函数本身的FSD Hook

此类方法的原理同样是对 FSD 派遣函数完成 Hook，并且更加隐蔽，使用户更不易察觉。该方法通过在派遣函数开头的几个字节写入汇编指令中的 JMP 等指令来实现跳转，即可以不在原先内核地址段内完成内存修改等操作。此类方法不需改变函数地址，因此通过一般的读取模块比对派遣函数地址的方法无法进行检测。

2.2 针对绕过过滤驱动的多层次检测方法

本检测系统针对 2.1 节中介绍的几种过滤驱动无法捕获到的绕过方法 [9-11] 进行对应的改进。

2.2.1针对直接访问FSD攻击方法的改进

对于直接绕过过滤驱动访问 FSD 的攻击行为，FSD 接受其 IRP 包后会通过调用派遣函数完成下一步操作。因此，可对 FSD 中的各 DispatchRoutine 进行 Hook，在派遣函数中添加监控函数，记录操作，进而交给下层，实现对各操作行为的记录。可通过对比派遣函数端监控到的数据域 FSD 上层过滤驱动监控到的行为，即可识别绕过过滤驱动直接访问 FSD 恶意行为的操作。

2.2.2 针对挂载新的过滤驱动攻击方法的改进

对于绑定在过滤驱动上层的驱动，该驱动将先于检测的过滤驱动获得应用层发送的 IRP 包，并可以阻止其发往下层，造成“拒绝服务”的攻击效果。为了监控其恶意行为，对于在监控过滤驱动 A 之上绑定新驱动 B 的恶意行为可以对其进行反向绑定。可以对绑定 FSD 的驱动进行实时监控，当发现有用户未知的驱动 B 附加在 FSD 之上时，可以在新

的挂载驱动上重新挂载监控驱动 C，形成在其被检测驱动上下层均添加检测驱动的状态。通过上下层截获的数据对比，可以看到时 A 与 C 截获的信息是否有不对称性，从而得出驱动 B 是否进行了“拒绝服务”恶意行为，并对该恶意行为进行监控记录。

2.2.3 针对替换DispatchRoutine的FSD Hook 的改进

该攻击方法可以对原派遣函数进行修改，使其完成其他操作功能。针对该攻击方法，可以通过读取模块基地址并获取地址范围的方法来实现检测。如果派遣函数功能发生改变，则一般需要新申请内存地址来完成函数实现的新功能，通过检测派遣函数地址范围是否越界可判断其是否被更改。在过滤驱动 DriverEntry 主函数中对各个 DispatchRoutine 地址范围进行判断，如果已被 Hook，可将 DispatchRoutine 函数地址修改为原先记录的正确地址，使得正确操作不被更改。同时，在派遣函数接收到 IRP 后发送相同的 IRP 给被 Hook 后的 DispatchRoutine，可监控记录其恶意行为操作，并利用重定向技术 [12-15] 将两次操作结果进行比对，如果是正确的操作行为被禁止，则判断其行为恶意性。

2.2.4 针对DispatchRoutine的InlineHook的改进

此攻击手段在派遣函数中通过 JMP 等跳转指令实现其他功能，由于并未单独申请派遣函数地址，因此无法通过检测其地址范围判断是否存在 Hook 行为。检测基本思路是读取存储在磁盘上的 FSD 文件，将其加载到内存，保存一份干净的备份，然后检测要调用的 DispatchRoutine 开头的几个字节和这个干净备份是否一致。如果不一致，特别是存在 JMP、RET、INT3 之类的汇编指令时，则极有可能存在 Inline Hook。可将干净的函数开头复制过来覆盖被感染的函数头以还原原函数。方法同 2.2.3 节，分别发送 IRP 给原函数及 Hook 后函数，通过重定向技术，将两次操作结果比对，得到记录行为，从而判断其行为恶意性。

2.2.5 集成后的检测模块

将上述的改进检测方法整合，形成集成后的检测模块。基本思路为检测开始首先对 FSD 派遣函数进行检测，判断其是否被 Hook 或 InlineHook。之后在派遣函数中加入检测方的记录函数，防止对于 FSD 的直接内核级别访问。同时，对于绑定 FSD 的驱动进行监控，如发现新的绑定 FSD 的过滤驱动被添加，则生成新的检测驱动进行绑定。可有效针对以上四种攻击手段进行防御，通过在过滤驱动的上下层进行对应的防护，实现多层次的文件操作提取。

3 集成测试环境

测试环境：通过 VmVare 装载纯净虚拟机，在相同硬件条件下安装普通的 Minifilter 以及改进后 Minifilter。

测试过程：测试驱动的编写，针对以上四种绕过过滤驱动的攻击方法，编写对应功能测试驱动。测试驱动分别实现以下四个功能：1）通过在 Minifilter上挂载新的过滤驱动，其过滤驱动实现禁止创建、读取、写入、删除文档功能。2）驱动层直接访问 FSD，实现创建、读取、写入、删除文档功能。3）对 FSD 中派遣函数的 Hook，分别实现禁止创建、读取、写入、删除文档功能。4）对 FSD 中派遣函数的 InlineHook，分别实现禁止创建、读取、写入、删除文档功能。

每次测试前先加载一个测试驱动，之后分别加载 Minifilter 以及改进后的 Minifilter 进行创建、读取、写入、删除功能的分别检测。在每次加载不同驱动时应重置虚拟环境。

4 集成测试及结果分析

测试分为功能测试和性能测试两部分，功能测试主要用于实现测试改进后的过滤驱动功能性是否完整，是否能针对其攻击方法有效检测出其恶意行为。性能测试是测试改进后的过滤驱动与原先 Minifilter 相比的稳定性、运行速度等指标。

4.1 功能测试

为了测试改进后的方案是否有效，根据上述四种绕过过滤驱动的攻击原理分别编写四个测试驱动，分别实现创建文档、读取文档、写入文档和删除文档的功能。对每个测试驱动分别用改进前的 Minifilter 以及改进后的多层次检测模块进行测试，其检测结果如表 1 所示。

表1 部分测试结果

行为功能 / 攻击方法	创建文档		读取文档		写入文档		删除文档	
	改进前	改进后	改进前	改进后	改进前	改进后	改进前	改进后
正常	√	√	√	√	√	√	√	√
挂载过滤驱动	×	√	√	√	√	√	√	√
内核级别文件访问	×	√	×	√	×	√	×	√
派遣函数 Hook	×	√	×	√	×	√	×	√
派遣函数 InlineHook	×	√	×	√	×	√	×	√

表 1 中“√”代表能获取到相关行为信息，“×”代表无

法获得。从表1中的测试结果可以看出，改进后的过滤驱动可以提取到原先Minifilter无法提取到的文件操作。针对文中提到的能绕过Minifilter攻击方法编写的对应的测试驱动，在改进后的过滤驱动中全部被检测记录，从而证明了通过改进后的多层次行为提取技术，可以实现更深度的文件行为监控。

4.2 性能测试

测试改进后的微过滤驱动对四种攻击方法的不同反应，检测其稳定性、兼容性和效率。虚拟环境硬件配置：Intel Pentium(R) Dual-Core E5200 处理器，频率2.5 GHz；1 GB 的DDR2 内存，频率800 MHz。操作系统：32 位Windows XP 和 Windows 7。

针对上述4种攻击方法分别编写对应的检测驱动以及一个集成后的检测驱动。在硬件虚拟化的坏境下，分别在Windows XP下加载原始的Minifilter、Windows XP下加载改进后的过滤驱动、Windows 7下加载改进后的过滤驱动，3种环境中运行相同的100个恶意二进制样本得出时间记录比值，测试结果如图2和表2所示。

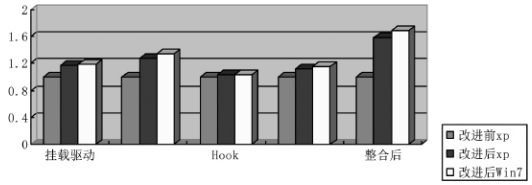

图2 性能测试结果对比图

表2 检测针对安装不同测试驱动的性能对比

改进技术	挂载驱动	内核级别访问	派遣函数Hook	InlineHook	整合后
Windows XP	1.18	1.28	1.03	1.12	1.58
Windows 7	1.19	1.34	1.04	1.15	1.69

从图2和表2中可以看出对于不同模块改进后的检测驱动略微增加了时间成本。其中对于派遣函数Hook的检测相对增加的时间成本较少。同时，在Windows XP操作系统下，相同微过滤驱动的执行效率要略高于Windows 7操作系统，这也是Windows 7操作系统占用系统资源更多的缘故。

总体来说，改进后的过滤驱动检测能实现多层次的文件行为提取，尤其能对当前绕过主流检测技术Minilfiter的几种攻击方法进行对应检测。从功能测试及性能测试结果来看，其功能基本完成，并且带来的时间代价相对较小，比较具有工程价值。

5 结束语

本文对当前微过滤驱动检测文件操作行为方面展开研究，分析了当前过滤驱动无法监控到的几种行为，并针对几种行为在原先检测驱动上添加对应判定，从而实现更深度检测文件操作行为的目的。此方法可运用在虚拟化环境中文件操作行为的提取上，在杀软基础上，做进一步的辅助分析。在后续研究中，将对以上检测做进一步的优化，提高检测速度及稳定性，增强抗攻击性。例如，加入机器学习算法、优化数据结构、深入底层和加强安全性。该检测方法同样可适用于其他方面的恶意行为提取，例如注册表操作、网络操作等，都可以针对驱动层攻击进行专项防御，更有效地提取恶意软件行为。可以将以上技术整合，形成一个更深度的辅助分析系统。（责编 潘静）

参考文献：

[1] SUN Ying-ying, ZHENG Kou-gen,Filemonitoringsystem based on minifilter[J]. Journal of Computer Applications, 2010,30(11):3115-3117.

[2]FileSystemFilterDrivers[EB/OL].http://msdn.microsoft.com/en-us/windows/hardware/gg462968, 2012-02-14.

[3] XUE Sheng-jun , CAO Feng-yan. Security Software Files Protection Based on Minifilter[J]. JOURNAL OF WUHAN UNIVERSITY OF TECHNOLOGY, 2011,33(4):130-133.

[4]FileSystemMinifilterDrivers[EB/OL]. http://msdn.microsoft.com/en-us/library/ff540402.aspx, 2012-02-14.

[5] ZHANG Fan,SHI Cai-cheng. Windows Driver Development Internals[M]. Beijing: Publishing House of Electronics Industry,2008.

[6] TAN Wen,YANG Xiao,SHAO Jian-lei.Windows Kernel security programming[M].Beijing: BEIJING Publishing House of Electronics Industry,2009.

[7]CHEN Jian-xiong,JIE She,ZHANG Xin. Implementation of Virus Prevention Method Based on File System Filter Driver[J].COMPUTER TECHNOLOGY AND DEVELOPMENT,2013,23(3):143-146.

[8] GE Yang-yang，MAO Yu-guang,File Security Protection System Based on Minifilter[J]. Computer & Digital Engineering, 2013,(4):631-634.

[9] YAN Zhen, LIU Ming, File monitoring system based on file filter driver design and implementation[D].Chengdu:University of electronic science and technology of computer system structure. 2012.

[10] HU Ji-ying, ZHAI Jian-song, SONG Yang. A file system based on Minifilter framework design and implementation of security module[D].Ha Er-bin:Harbin industrial university, 2013.

[11]PI Chang-di, LIU Nai-qi.Secure file system based on filter drivers in the research and implementation[D].Chengdu:Department of electronic science and technology of large software engineering,2010.

[12] LIU Sheng,ZHOU An-min, JIANG Lei.File operations detecting system based on minifilter driver[J].Information and electronic engineering,2012,10(6):779-782.

[13]LI Min, FANG Yong, LIU Lin-chao, XIONG Fan. File Driver on FSD and its Applications[J].Information and electronic engineering, 2005, 3(4):290-292.

[14] CHEN Ling , TANG Zhen-min.Redirect implementation and its application in file system filter driver [J].The information technology,The China science and technology information 2007, (22):92-93.

[15] WANG Yang ,WANG Qin,The development of the sandbox security technology research [J].Software Guide,2009,8(8):152-153.