基于注册表Hive文件的恶意程序隐藏检测方法

任云韬，李毅超，曹 跃

(电子科技大学计算机科学与工程学院 成都 610054)

【摘要】研究当今恶意程序的发展趋势，系统比较了在注册表隐藏和检测方面的诸多技术和方法，综合分析了它们存在的不足，提出了一种基于注册表Hive文件来进行恶意程序隐藏检测的方法，使得针对恶意程序的检测更加完整和可靠。实验表明，该方法可以检测出当前所有进行了注册表隐藏的恶意程序。

关 键 词 Hive文件； 恶意程序； 注册表隐藏和检测； RootKit程序
中图分类号 TP393.08 文献标识码 A

A Methodology to Detect Malware Based on Registry Hive Files

REN Yun-tao，LI Yi-chao，CAO Yue

(School of Computer Science and Engineering，University of Electronic Science and Technology of China Chengdu 610054)

Abstract Based on the research on the current developing trends of malicious programs, comparing systematically the various technologies and methodologies with respect to the hiding and detection of registry. analyzing comprehensively their deficiencies existing, we provide a brand-new hiding and detection method based on hive files of registry, which makes the detection especially on malicious programs more integrated and reliable. The experiment indicates that this method can detect all the current malicious programs which hide registry.

Key words Hive files; malware; registry hiding and detection; Rootkit

恶意程序[1]，尤其是间谍软件和Rootkit[2]，已经成为了计算机安全领域重点防范的对象。从恶意程序的发展来看，具备一定的信息隐藏功能已经成了一种趋势。隐藏的信息中包括了注册表、进程和端口等。这些信息的隐藏在延长恶意程序的生存周期，加大检测它们难度的时候，也暴露了恶意程序的踪迹。与此同时，针对恶意程序的检测技术也在不断的提高中并形成了理论体系。常见的恶意程序的检测方法包括了启发式分析法和特征码比较法等[3]。但随着恶意程序隐藏技术的提高，使得对它们的检测越发困难，传统的方法和技术已经不能适应发展的需要。

本文的研究表明，就现有的恶意程序实现而言，不管其如何隐藏，其最终实现依然是在操作系统框架内并且依赖于系统提供的某些功能。在Windows系统下，恶意程序的存在和运行大都离不开系统注册表[4]中的相关信息。恶意程序也常常隐藏于注册表，因此不可能真正删除它们。但可以通过获取系统注册表的可靠原始信息检查隐藏项，进而获知恶意程序的存在，并最终清除恶意程序。

1 注册表隐藏及检测

注册表是Windows系统存储关于计算机配置信息的数据库，包括了系统运行时需要调用的运行方式的设置，是系统的核心。操作系统是用特殊的文件(Hive文件)[5]来存储注册表的内容，并提供给用户相关的编程接口(APIs)来进行注册表的操作，例如Advapi32.dll中的Registry Functions(如：RegEnumKey)。同时，在Windows操作系统中，函数的调用有着极其规范的层次结构，图1所示是系统实现RegEnumKey函数的调用体系图。

正常情况下，函数调用返回的结果应该是操作系统提供的实际信息，但是随着间谍软件和Rootkit等恶意代码技术的发展，越来越多的恶意程序都具备了隐藏自身存在信息的功能，这其中最重要就是对注册表项信息的隐藏[6-7]。注册表隐藏就是将系统呈现给用户的注册表信息进行修改，使得用户或检测工具无法直观地发现事实上存在的注册表内容。目前的注册表隐藏，大都采用的是Hook技术[8]。根据它们不同的运行环境和模式，把常见的注册表隐

收稿日期：2006－04－27

作者简介：任云韬(1981－)，男，硕士，主要从事网络攻防、网络安全方面的研究；李毅超(1969－)，男，硕士，副教授，主要从事网络攻防、网络安全领域方面的研究；曹 跃(1981－)，男，硕士生，主要从事网络攻防、网络安全方面的研究.

藏技术称为用户态下注册表隐藏技术和核心态下注册表隐藏技术。它们通常是利用IAT Hook, Inline Hook、远程线程注入、SSDT Hook、中断调度表挂钩、IRP函数表挂钩、原始内核代码修改和特殊指针修改等方式来实现，文献[8]中进行了详细的解释。

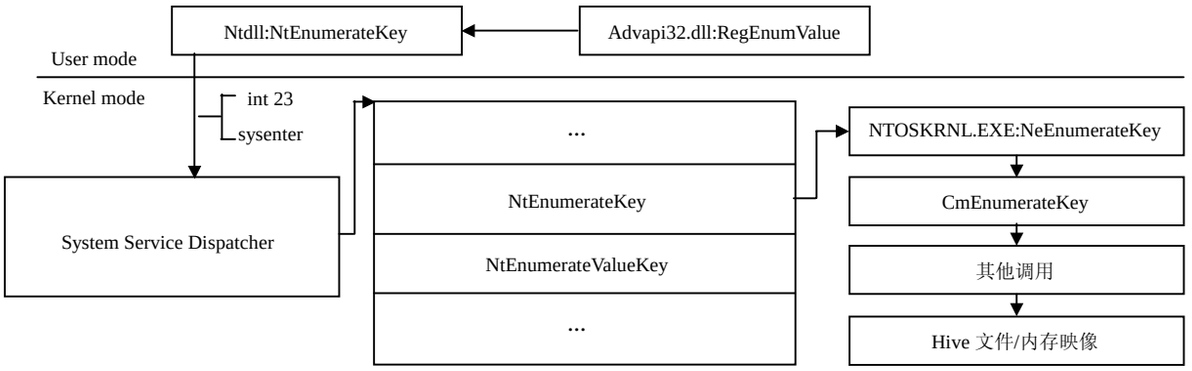

图1 RegEnumKey函数系统调用图

在反恶意程序的发展过程中也产生了一些检测理论和方法。当前基于注册表的恶意程序隐藏检测，基本上采用了如下步骤：(1) 在函数调用体系中，检查并恢复被Hook的部分或利用更底层的函数直接获取指定的注册表项内容，把它们视为原始数据；(2) 利用操作系统最上层的函数调用(通常是Advapi32.dll中的Registry Functions)获得注册表信息，视为系统数据；(3) 比较两种数据是否相同，不同者即视为被隐藏的表项，再结合其他判断，基本可以确定隐藏项的作用，进而找出主机中可能存在的恶意程序。

基于注册表的恶意程序隐藏检测关键在获取可靠的原始数据。针对隐藏技术，现在主要是通过函数调用及执行代码修复的方式来解决。IceSword[9]就是这样一款较为优秀的软件，通过它可以发现一般情况下的注册表隐藏。它在内核模式下直接调用NtEnumerateKey函数，但在调用前会用从Ntoskrnl.exe文件中读取正确的函数起始地址及其函数体前若干代码指令来覆盖内存中的NtEnumerateKey函数，所以一般的隐藏方式是没用的。但是，随着Rootkit技术的不断发展，注册表的隐藏也越来越深入系统内部。目前通常的检测工具和技术已可以被突破或绕过，就连IceSword这样公认的检测工具，也可以被挑战——通过硬编码搜索到NtEnumerateKey中调用CmEnumerateKey函数的指令，通过改变该条指令，使之执行流程转向到Hook函数中去，在Hook函数中再调用CmEnumerateKey，并修改返回结果，从而绕过IceSword的检测。

本文的研究发现，当前的所有注册表隐藏和检测都是在操作系统框架之内，它们之间的博弈还将不断继续下去。同时，注意到由于恶意程序需要操作系统依靠注册表的信息来执行某些操作，所以它们并不能从根本上改变其中的内容，也就是改变系统注册表Hive文件，从而使得它们里面记录的信息是最原始和最完整的。如果不是通过函数调用，而是直接获取到它们的内容来作为上面讲的原始数据，就可以跳出了现有的检测的模式，并且这样检测出来的结果显而易见应该是最可靠的。

2 基于注册表Hive文件的恶意程序隐藏检测

2.1 Hive文件格式

关于Windows系统注册表文件的组织格式，即Hive文件格式，并没有对开发者公开，但是针对它们的研究可通过逆向工程和其他方法进行。图2所示是本文给出的Windows 2000中Hive文件的组织形式，各部分的意义可详见文献[10]。

2.2 Hive文件获取

通常情况下，所能看到的或者可以获取注册表的信息不是以Hive文件格式的形式存在，要通过Hive文件来进行注册表的检测和分析，首先要获取注册表信息对应的Hive文件。

在Windows系统中，可以通过使用API RegSaveKey/RegSaveKeyEx 函数来将指定键下的注册表信息转储为Hive格式的文件。但是，由于这些文件的获取是通过API调用来的，这样就给了恶意程序进行信息篡改的可能。虽然到目前为止还没有挂钩这类函数的程序，但是理论上这样做是可以的，从而存在安全上的不足，因而需要寻找另外的获取方法。

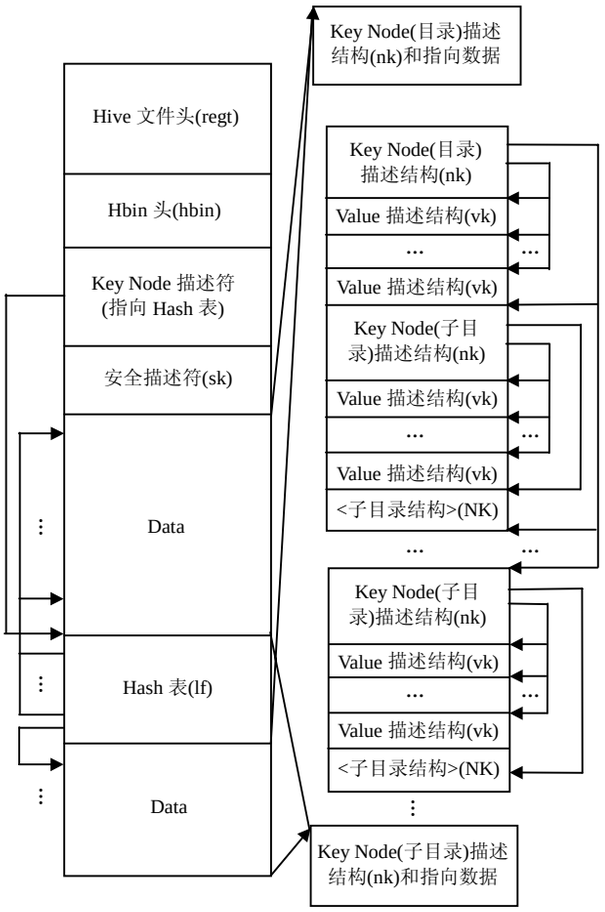

图2　Hive文件组织形式

注册表中HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\hivelist记录着系统注册表Hive文件的保存路径，而系统许多功能的实现是依赖于里面记录的这些文件提供信息的。正是由于这一性质，决定了系统Hive文件应该是安全的，内容也是最完整的。同时，由于这些文件对于系统的重要作用，一般情况下操作系统是不允许其他程序在系统范围内去访问这些文件的，这就需要通过编写文件驱动来绕过系统对这些文件的保护，从而实现文件的读取。

2.3　基于注册表Hive文件的恶意程序隐藏检测

在获取了注册表Hive文件的基础上，就可以实现对它们进行完整的检测。借鉴了Petter Nordahl-Hagen所编写的NT Registry Hive access library(该访问库是文献[10]中的一部分)，通过2.1节中对Hive文件的分析，另外设计了一个实用的Hive文件访问接口，利用该接口可以用来分析Hive文件存储内容，获取其中信息。从而，基于注册表Hive文件的恶意程序隐藏检测步骤如下：(1) 使用访问接口对Hive文件进行Dump分析，来获取其中记录的原始信息；(2) 利用Windows提供的注册表访问API进行相应的注册表查询和遍历，作为系统数据；(3) 对比上述不同来源的两种数据信息，检测隐藏的注册表项，标识出恶意程序隐藏的行为及其所在；(4) 通过分析注册表中的启动项，提供恶意程序自启动信息。按照上述步骤，本文提出了相应的实现流程图，如图3所示。

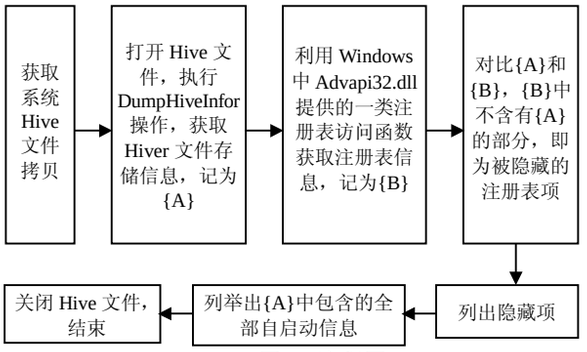

图3　流程图

3　实现与结果

针对本文提出的基于注册表Hive文件的恶意程序隐藏检测方法，可设计能检测出目前所有恶意程序隐藏注册表行为的工具软件。基于本文思想，实现了一个检测软件——T-Hive，并且选择Windows自带的注册表查看程序Regedit.exe和流行的系统检测工具IceSword.exe来做对比实验，实验对象是下面两种技术和方法都很有代表性的Rootkit：(1) T-SSDTHook，它是利用SSDT Hook技术来实现隐藏；(2) T-CmHook，它通过修改NtEnumerateKey中调用CmEnumerateKey函数的指令来实现隐藏。

表1给出了对比实验的结果。从表1中可看出T-Hive可以列举出各种注册表的隐藏项，其内容的完整性和可靠性都是较高的。

表1　实验结果

隐藏项/Rootkit	检测工具	检测结果
HKLM\SYSTEM\ControlSet001\Services\SSDTHook Rootkit: T-SSDTHook	Regedit	无法检测
	IceSword	可以检测
	T-Hive	可以检测
HKLM\SYSTEM\ControlSet001\Services\CMHook Rootkit: T-CmHook	Regedit	无法检测
	IceSword	无法检测
	T-Hive	可以检测

4　结 束 语

基于注册表Hive文件的恶意程序隐藏检测方法是针对现有的恶意程序中注册表隐藏最行之有效的检测方法，并可以作为检测恶意程序的一种有效的思路。同时，针对部分恶意程序并没有注册表隐藏功能，除了列举出所有的系统自启动信息外，如何

在大量的注册表数据中发现可疑内容，将是下一步研究的重点。此外，如何与其他检测方式，如进程检测相结合来进行全面的恶意程序检测和全面分析也是下一步研究的方向。

参 考 文 献

[1] CHRISTODORESCU M, JHA S. Testing malware detectors[C]//In Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis 2004(ISSTA'04). Boston USA: ACM SIGSOFT, ACM Press, 2004: 34-44.

[2] JAMES R, BUTELER I I. Detecting compromises of core subsystems and kernel functions in windows NT/2000/XP [D]. Baltimore USA: University of Maryland, 2002.

[3] CONOVER M. 恶意代码剖析和Rootkit检测[EB/OL]. www.xfocus.net/projects/Xcon/2005/Xcon2005_Shok.pdf, 2005-06-15.

[4] Microsoft Knowledge Base. Description of the Microsoft Windows registry[DB/OL]. http://support.microsoft.com/kb/256986/en-us, 2006-02-21.

[5] Clark. Security accounts manager[DB/OL]. http://www.beginningtoseethelight.org/ntsecurity, 2006-03-02.

[6] HOGLUND G. Nt rootkit - the original and first public NT ROOTKIT[DB/OL]. https://www.rootkit.com/vault/hoglund/ rk_044.zip, 2006-03-02.

[7] Fireworker. Kernel-mode backdoors for windows NT [DB/OL]. http://www.phrack.org/archives/62/p62-0x06_Kernel_Mode_Backdoors_for_Windows_NT.txt, 2006-03-02.

[8] HOGLUND G, Butler J. Rootkits: subverting the mindows kernel[M]. Boston, USA: Addison Wesley Professional, 2005.

[9] PJF. The homepage of iceSword[DB/OL]. http://pjf.blogone.net, 2006-03-02.

[10] HAGEN P N, OFFLINE N. TPassword & registry editor [DB/OL]. http://home.eunet.no/-pnordahl/ntpasswd/, 2006-01-13.

编 辑 孙晓丹

(上接第610页)

实验证明，BoIP的隧道技术可实现IPv4孤岛的连接，并没有出现延迟增大、连接次数增多的现象。

3.2 分析与讨论

目前，IPv4 over IPv6隧道技术还没有统一的标准，国际IETF组织正在为该技术建立专门标准工作组，制定相关的国际系列标准。对比已经成熟的IPv6 over IPv4隧道技术，BoIP改进了报文的封装方式，简化了报文的封装操作。基于IPv6网络高带宽、安全和QoS等功能，BoIP采用UDP协议构造隧道，具有简单、高效的特点。

BoIP能使IPv4孤岛透明地通过IPv6主干网互连，并不失端到端的连接性。对于带宽需求量大、跨域传输数据的IPv4网络应用，它还可以作为分流IPv4网络的数据至负载相对较轻的IPv6主干网以利用IPv6高带宽的一种方法，实现IPv4端到端的高性能连接。

4 结 束 语

IPv4 over IPv6隧道技术是一种重要的IPv4网络向IPv6网络过渡的技术。随着大规模IPv6主干网的建设，会出现IPv4网络孤岛，需要通过IPv4 over IPv6隧道互连，从而实现最后的过渡。本文提出了基于UDP协议的IPv4 over IPv6隧道方案BoIP，描述了该方案的工作原理和实现过程，并以通过IPv6网络连接两个IPv4以太网为例做了实验。实验结果表明，BoIP具有透明(Transparent)、轻量(Lightweight)、支持单播组播和IPv4地址的动态分配等优点。该方案可进一步扩展成为支持多个IPv4孤岛连接的情况，并可用作无线网络接入有线网络的解决方案，具有较大的应用前景和实用价值。

参 考 文 献

[1] 吴建平. CNGI核心网CERNET2的设计[J]. 中兴通讯技术, 2005, 11(3): 17-20.

[2] 张云勇. 基于IPv6的下一代互联网[M]. 北京: 电子工业出社, 2004.

[3] DURAND A, FASANO P, GUARDINI I, et al. IPv6 Tunnel Broker, RFC3053[S]. 2001.

[4] TATIPAMULA M, GROSSETETE P, ESAKI H. IPv6 Integration and coexistence strategies for next-generation networks[J]. IEEE Communications Magazine, 2004, (1): 88-96.

[5] STEVENS W R, Fenner B, Andrew M. UNIX network programming[M]. Beijing: China Machine Press, 2003.

编 辑 孙晓丹