·微机网络与通信 ·

Windows下内核级木马隐藏技术研究

刘 德,甘早斌

(华中科技大学计算机科学与技术学院,武汉 430074)

摘 要:木马技术是网络安全的重要方面,也是网络攻击中获取信息的重要途径。隐藏技术是木马的关键技术之一,其直接决定木马的生存能力。从 Rootkit的原理分析出发,深入的研究 Windows下内核级木马的隐藏技术,并在此基础上实现一个内核级木马原型,最后介绍内核级木马的检测和应对策略。

关键词:特洛伊木马;内核 Rootkit;隐藏技术

中图分类号: TP309 **文献标识码**: A **文章编号**: 1002 - 2279(2009)01 - 0041 - 04

Research on Concealment Technology of Kernel- based Trojan Horse Under Windows

LIU De, GAN Zao - Bin

(Dept of Computer Science, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: Trojan horse technology is an important part in the research of network security, and it's also an important way to get information from the network. This paper mainly analyzes the concealment and detecting technology of windows Rootkit - based trojan horse. A kernel - level Trojan horse system is implemented as well. Finally, some ways to detect Trojan horse are described in detail.

Key words: Trojan horse; Rootkit; Concealment

1 引 言

特洛伊木马是网络攻击的主要手段之一,它可以伪装欺骗进入目标系统,并在进入目标系统后继续保持对它的控制。木马的首要特征是它的隐蔽性。为了提高自身的生存能力,木马会采用各种手段来伪装隐藏以使被感染的系统表现正常。木马的隐藏包括启动隐藏、文件隐藏、进程隐藏、通信隐藏等方面的内容。随着安全技术的进步,木马的开发者也在不断寻找新的木马隐藏和自我保护技术。

Rootkit是一种特洛伊后门工具,通过修改现有的操作系统软件,使攻击者获得访问权并隐藏在计算机中。Rootkit最初是为了攻击 UNIX系统而开发的。自从 1999年 Greg Hoglund开发了第一个针对 Windows的 Rootkit(即 NTRootkit)以来,Windows Rootkit受到了人们的广泛关注并得到了很大的发展。目前流行的 Windows Rootkit有 NTRootkit, Hacker Defender, AFX Rootkit, PWS - Progent, FU Rootkit等等。

针对当前 Windows下内核级木马的隐藏技术进行深入的研究,分析和总结内核级木马的隐藏和检测技术,并利用相关技术开发了一个内核级木马原型 BOES,通过实验测试该木马达到良好的隐藏效果。

2 Windows Rootkit技术分析

Windows系统服务是由操作系统提供的一组函数,应用程序接口使得开发者能够直接或者间接的调用系统服务,操作系统以动态连接库的形式提供应用程序接口。用户程序所调用的 API一些是直接来自相应的系统服务,另一些则依赖其他多个系统服务调用。系统服务调用是操作系统向用户提供硬件设备服务和请求内核服务的接口。在用户态执行时为了与 Windows系统交互,要执行各种对 Win32子系统 DLL文件的 API函数调用(Win32系统服务调用机制如图 1所示)。Microsoft提供了大量 API函数,它们被组织到不同的 DLL文件中,包括 User32. dll, kernel32. dll等。当用户程序执行,向 Win32 DLL发出函数调用请求时,这些请求并非被 DLL直接发送给内核,而是首先通过 Ntdll. dll文件。Ntdll. dll利用详细记录的函数调用把这些函数

作者简介:刘德(1982 -),男,湖南攸县人,硕士研究生,主研方向:网络安全和电子商务。

收稿日期: 2006 - 12 - 19

调用转换为内核能理解的基本函数调用,同时把内核基本函数的系统服务号载入 EAX寄存器,把用户内存空间中参数地址载入 EDX寄存器中。Ntdll dll完成上述操作后发出 INT 2E或 SYSENTER指令,使函数调用从用户态转入内核态。在内核态完成函数调用处理后返回用户态,调用结果通过调用函数返回应用程序。

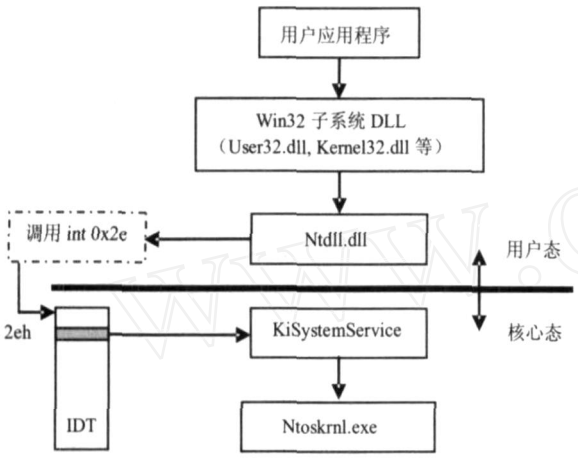

图 1 Win32系统服务调用机制

2.1 改变程序执行路径

(1)替换程序和 DLL文件

Windows操作系统中包含着大量的可执行文件和 DLL文件。当需要函数调用时系统会通过 DLL文件名和文件位置来加载 DLL文件。如果把系统真正的 DLL文件改名,而用攻击者编写的 DLL文件替换系统真正的 DLL文件,那么当系统调用时加载的就是攻击者的 DLL文件,其中的 API函数是攻击者实现的,攻击者就可以实现其所想要到达的隐藏目的(如:通过替换 gina dll木马可以得到用户的登录口令)。替换用户可执行文件也是一样。

这种直接在硬盘上的用户程序和 DLL文件的替换可能会受到 Windows 2000及以后版本的 Windows文件保护 WFP(Windows File Protection)的阻碍。WFP相当于一个内置的文件完整性检测,能够保护重要系统文件不被改变。但是这种保护也可以通过修改 Sfc dll文件或更改注册表项 SFCDisable项的值被突破。

(2)钩子(Hook)技术

钩子(Hook)是 Windows消息处理机制的一个平台,钩子机制允许应用程序截获处理 window消息或特定事件。钩子实际上是一个处理消息的程序段,通过系统调用,把它挂入系统。每当特定的消息发出,在没有到达目的窗口前,钩子程序就先捕获该消息,得到控制权,然后可以改变该消息,也可以不作处理而继续传递该消息,还可以强制结束消息的传递。

钩子有以下几种类型:①函数钩子,可以向被注入线程插入木马代码;②系统 API钩子,通过给系统的 API附加上一段小程序,达到监视甚至控制应用程序对 API函数的调用;③输入地址表(IAT)钩子,用另一个函数的地址替换 DLL输入函数的地址;④中断分配表(IDT)钩子,替换中断分配表中中断服务程序(ISR)的入口地址;⑤IO请求包函数表(IRP)钩子,替换 IRP函数表的一些表项来达到隐藏目的,如文件、网络端口、注册表项等。⑥系统服务描述表(SSDT)钩子,通过修改 SSDT表中某些函数地址为自己的函数地址,可以达到控制函数处理过程和过滤返回结果的目的。

(3)回调

Windows API函数集中大量采用回调函数。上层的应用程序把回调函数的名称转化为地址作为一个参数传给系统底层,底层在符合设定的条件时自动调用上层回调函数,回调函数可以完成一定的操作。Windows平台的消息机制则是另一种更广泛的回调应用,通过系统提供的接口注册消息处理函数(即回调函数),实现接收和处理消息的目的。

很多间谍软件通过 Windows键盘消息钩子,由系统回调一个用来记录用户各种输入的处理函数从而达到窃取用户密码和用户兴趣的目的。CIH病毒通过调用 INT20来完成调用一个 IFSMgr_InstallFileSystemApiHook的子程序,在 Windows内核的文件系统处理函数中挂接钩子,以截取文件调用的操作,这样一旦系统出现要求开启文件的调用,CIH病毒的传染部分程序就会被回调。另外,通过在网卡混合模式下 NDIS过滤驱动程序的回调,可以过滤特定的 IP包从而实现无端口的后门。

(4)特殊寄存器

通过对一些系统特定的寄存器(如调试寄存器 DrX、模型特定的寄存器 MSR)的控制和修改,可以实现提升权限从而获得系统控制权的目的。

(5)分层过滤驱动

设备驱动执行了很多操作系统的重要任务,如网络连接和文件存储。Windows允许程序员在现有的驱动上进行层叠而不用重写整个驱动。许多病毒扫描引擎通过文件过滤驱动在文件打开时得到通知,操作系统将文件操作信息通过文件系统驱动传给病毒扫描引擎。Rootkit通过使用分层过滤驱动修改文件和 socket通信的访问,从而实现文件和通

信端口的隐藏。

2.2 修改内核对象

(1)修改进程和驱动链表

Windows操作系统将进程和线程的信息保存在执行体对象(executive objects)中,Taskmgr.exe等进程监控工具通过ZwQuerySystemInformation函数获得各种进程信息。Windows为某个进程创建了一个EPROCESS数据结构,所有的EPROCESS数据结构被连成一个双向链表。用户请求列出当前所有进程是系统通过遍历EPROCESS双向链表来实现的。当把某个进程的EPROCESS从双向链表中删除时(如图2),系统就不能用此方法列出所有的进程,从而达到隐藏进程的目的。把EPROCESS从双向链表中删除并不会影响进程运行,因为Windows任务调度是基于线程的。FU Rootkit就是通过这种修改内核对象的方法实现进程的隐藏的。

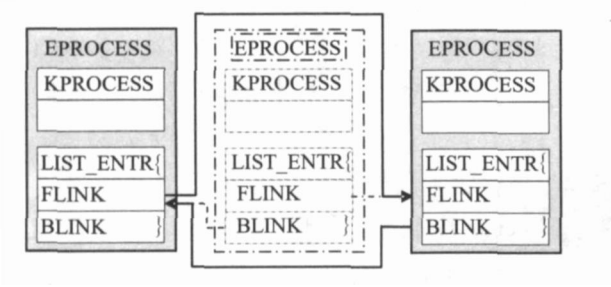

图2　被修改后的进程链表

很多Rootkit都以驱动的形式存在,因此实现驱动的隐藏是Rootkit的一个重要组成部分。驱动的隐藏与进程的隐藏类似。Windows为每个设备驱动创建了一个MODULE_ENTRY数据结构,这些MODULE_ENTRY数据结构被连成一个双向链表。通过把某个设备驱动的记录从双向链表中删除,系统列出的驱动中就不会包含Rootkit的驱动,从而完成驱动的隐藏。

(2)修改进程和线程令牌

进程令牌标记进程权限,决定进程能做什么和不能做什么。进程令牌根据它的属主用户生成。每个进程内的线程也有自己的令牌,大多数时候这个令牌继承自它所属的进程。进程令牌的内存地址在进程的EPROCESS结构中找到。进程令牌中包括特权的本地唯一标识LUID和用户或组的安全标识SID。攻击者要获得系统的控制权,就必须提升Rootkit的权限,通过向令牌中添加不该有的权限,可以实现进程特权的提升。如果同时修改令牌中的AUTH_ID和第一个SID(即用户SID),那么将欺骗系统日志,使之错误记录事件发生进程的属主用户名。

(3)修改句柄表

Windows系统通过句柄管理系统对象。使用SystemHandleInformation类调用NtQuerySystemInformation可以得到所有被打开的句柄的数组,这个数组保存在_SYSTEM_HANDLE_INFORMATION_EX结构中。为了更深层次的隐藏自己,Rootkit将所有与自己和被隐藏进程有关的句柄从句柄数组中删除。

2.3 其他方法

除了修改程序执行路径和更改内核对象外,还有些Rootkit通过更改BIOS程序(如eEye BootRootkit)甚至改变CPU微指令来实现。一些更先进的Rootkit(如SubVirt和Blue Pill)甚至使用虚拟机技术,修改引导区记录使之通过虚拟机引导,然后在这个虚拟机之上运行原来的操作系统。VM Rootkit还可以在原始操作系统之外运行另一个为入侵者提供服务的操作系统。这种VM Rootkit更难检测和移除,因为运行在目标系统中的软件无法访问和操作这个虚拟机。

3 木马原型BOES所采用的隐藏技术

内核级木马原型BOES(Back Orifice for Electronic Scout)提供常见木马的远程控制功能,如远程文件操作、注册表操作、截屏、远程命令行解释器,使用Rootkit实现文件、进程、通信端口和注册表的隐藏。

Rootkit可通过两种层次来实现,即"用户模式Rootkit"和"内核模式Rootkit"。内核模式Rootkit对系统所有用户和内核的内存空间拥有完全访问权限,能够修改任何程序代码和系统数据结构。但是内核模式Rootkit需要以驱动程序调入内核,涉及大量系统底层API函数的使用,复杂性高容易导致系统不稳定,因此BOES使用一种综合用户模式和内核模式的混合型Rootkit(其结构如图3所示)。

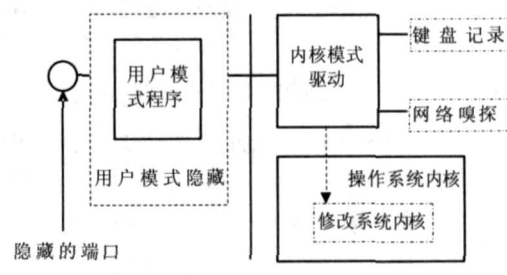

图3　用户模式和内核模式混合型木马

用户模式Rootkit通过植入或修改用户程序实现大部分的功能,如提供网络连接和远程控制后门,记录系统操作记录和获取私隐信息等,而内核模式Rootkit通过改变系统的内核以实现对文件、目录、

进程、注册表项、网络端口、设备驱动器,网卡模式等的隐藏。

用户模式程序可以通过很多种方法与内核驱动进行通信,常用的有共享内存,共享事件,IOCTL宏,或者直接用 ReadFile()或 WriteFile()进行读写,BOES采用最常用的方法即调用 I/O控制命令(IOCTL)。

在自身的隐藏技术上,BOES主要通过修改内核对象实现BOES及其相关文件和文件夹、进程、设备驱动、注册表健值和网络端口的隐藏。

在通信的隐藏方面,BOES提供 ICMP、Rootkit隐藏下的 TCP和邮件发送等多种渠道(如图4所示)。BOES通过定期检查指定网站上的木马配置文件和客户端的控制命令以及防火墙的情况来决定具体采用哪种策略。ICMP消息能有效的穿透防火墙从而躲过网络设备的拦截,所以BOES在大部分时候使用 ICMP来接受远程控制命令。当木马启动时或者被控机器的 IP地址等重要信息发生改变时,BOES通过邮件将这些信息发到攻击者的邮箱从而使攻击者时刻保持对木马所在计算机的跟踪。由于ICMP使用小数据包,不适于大量信息和文件的传输,这时 BOES可以通过 TCP协议传输信息,使用Rootkit隐藏所打开的通信端口。

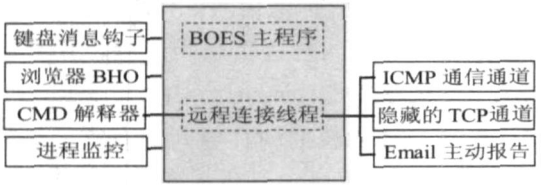

图4 BOES功能结构图

BOES还通过 PsSetCreateProcessNotifyRoutine API钩子实现对系统进程创建事件的监控,并能在收到特定进程(某些新型 Rootkit检测工具)的启动消息时狡猾地终止自身进程从而逃避新型 Rootkit检测工具的检测。

通过实验,木马原型 BOES能躲避主流反病毒软件(诺顿、McAfee、卡巴斯基、金山毒霸)和部分Rootkit检测工具(knlsc13、VICE 2.0、klister 0.4、pm 1.2)的检测,达到良好的隐藏效果。

4 内核级木马的检测和防御

4.1 Rootkit的检测

Rootkit隐藏技术层出不穷,每一种隐藏技术出现时都会出现相应的检测技术,但很快又有更新的隐藏技术绕过检测。现行的 Rootkit检测技术主要有完整性检查、基于特征码的检测技术、一致性检查、执行路径分析技术等。

反病毒软件依赖特征码检查 Rootkit,这种方法可以有效地对付流行的 Rootkit,但是对于新型和多态的 Rootkit无能为力。

Komoku和 SVV检查系统组件和内存数据的完整性,通过对系统文件的 Hash签名检测出替换和新增系统组件的 Rootkit。

许多 Rootkit应用 Hook技术修改 IAT/EAT表、IDT表、SSDT表等,VICE、SDT restore等工具通过检查被挂钩的地方是否被修改来检测 Rootkit,不过许多系统安全软件也使用挂钩技术。

Pathfinder是利用执行路径分析技术(EPA)通过检查某些关键系统函数运行时所执行的指令个数,并与事先记录的数据比较,如果执行了多余的代码,就可认为被安装了 Rootkit。

Klister通过内核线程列表列出所有线程,根据线程与进程的关系得到所有进程的列表,然后比较该列表与通过 API函数得到的进程列表检查系统的一致性,判断是否存在隐藏进程。

4.2 内核级木马的防御

Rootkit技术不仅在恶意软件中广泛应用,在很多第三方安全软件中也大量使用,从而导致内核级木马的检测会遇到很大困难。有些 Rootkit(如 VM Rootkit)甚至超出操作系统的控制之外,根本无法检测。但是木马要实现它对系统的控制总会留下蛛丝马迹,而且木马总有一个植入的过程,这个过程中它是难以实现自身隐藏的,因此对付内核级木马重在防御。通过加强系统的安全配置、及时下载安装补丁、禁用不必要的服务、安装反病毒软件和入侵检测软件进行实时监控、选用安全评估和漏洞扫描工具对系统进行安全性审计然后有针对性的加强系统安全、细心安全规范的操作和随时保持对相关技术的关注,将 Rootkit木马拒之门外。还可以选择最新的Anti-Rootkit软件定期对系统进行彻底的扫描以及早发现内核级木马。

如果已经检测到 Windows系统被攻击者安装了内核级木马,这是一件相当严重的安全事件,处理办法依赖于公司或组织预定的信息安全策略。但无论信息安全策略如何,该 Windows操作系统将不可信。恢复系统之前必须查明系统被攻破的原因,这有利于以后有针对性地加强系统和网络安全。恢复系统最好的方式是重新安装操作系统,重新加固操作系统和安装防护软件。 (下转第48页)

IETF协议,其推广应用尚有待完善。

4.4 优势

MPLS的主要优势在于其中引入了 TT(Traffic Trunk)的概念。TT是同一类业务流的集合,它是对具有相同特征的业务量的抽象,是流量工程实施的基本单位。在MPLS中,可以对不同等级的 TT进行分类服务,保证各自要求的业务质量,这主要是通过分配不同等级的 LSP实现的。其主要好处是将网络设施的开销和网络的大小及业务量分离开来,当网络业务规模增加时,Trunk内的业务量增加,而Trunk的数量不增加。MPLS不仅支持流的聚合,而且还支持流的分解。此外,MPLS比其他机制更容易集成 QoS路由。

MPLS流量工程还有很多其他优势,主要包括:①通过手工的网管配置或是下层协议的自动配置,可以很容易的建立起不受传统逐跳路由协议限制的显式 LSP;②LSP可以被高效的维护;③流量主干可以被使用并被映射到 LSP上;④可以给流量主干规定一套属性来调整流量主干的行为;⑤可以给各种网络资源规定一套属性,以便对以上建立的 LSP通过的流量主干加以限制;⑥既可以对业务进行组合,也可以对业务进行分割,而基于传统的路由协议的IP转发只支持对业务的组合;⑦可以较容易的实现"约束路由";⑧MPLS流量工程的开销要比其他的流量工程技术小得多[4]。

对于流量工程来说,MPLS是当前最好的解决方案。MPLS流量工程的特性允许 MPLS骨干网继承并扩展了二层 ATM和帧中继网络的流量工程能力,是当前性价比最好的最具竞争力的宽带网络技术。同时,MPLS具有提供自动化流量功能的可能。

5 结 束 语

MPLS和流量工程的结合,发挥了 MPLS技术固有的优势:在加速数据包传递的同时,能够更加合理地利用网络资源,同时减少了管理成本,使自动流量工程成为可能,彻底克服了 IP over ATM等重叠模式的种种局限,为 IP网的发展指明了方向。

MPLS中的流量工程技术随着运营商和设备商的应用在不断完善,同时也出现了许多需要解决的新课题。另外,网络技术本身的发展也对流量工程技术提出了新的要求。

参考文献:

[1] 凌永发,王杰. 多协议标签交换技术的应用[J]. 云南民族大学学报, 2005, 14(3): 255 - 258.
[2] 李娟娟. MPLS网络中基于流量工程的边缘准入控制研究[D]. 天津大学硕士论文, 2005.
[3] E Rosen, A Viswanathan, R Callon. Multiprotocol Label Switching Architecture[S]. RFC3031, IETF, Jan 2001.
[4] D Awduche, J Maicolm, J Agogbua, et al. Requirements for Traffic Engineering over MPLS[S]. RFC2702, IEFE, Sep 1999.
[5] Swallow C. MPLS Advantages for Traffic Engineering[J]. IEEE Communications Magazine, 1999, 37(12): 54 - 57.
[6] 彭晖. 新型的骨干网路由平台——MPLS[M]. 北京:人民邮电出版社, 2002.
[7] Eric Osborne. 基于 MPLS的流量工程[M]. 北京:人民邮电出版社, 2003.

(上接第 44页)

如果攻击者修改了操作系统内的一部分,那么用户无法知道是否修改了其它组件。在处理发现已被攻击系统的同时,严格检查附近机器系统,发现问题也作同样处理。原有系统中的秘密信息(如帐号/密码)也需要改变,因为可能已被攻击者盗取或网络嗅取。恢复系统后,要加强对网络和系统的防护和监控,攻击者可能会再次返回犯罪现场,试图再次进入系统。

5 结 束 语

隐藏技术是木马的关键技术,研究隐藏技术对特洛伊木马来说至关重要。特洛伊木马不仅可以作为网络攻击的主要手段,也可以成为我们反击黑客的重要工具。把木马技术的思想用在虚拟网络环境下的动态电子取证也有很重要的应用价值,这是我们下一步的研究方向。

参考文献:

[1] Greg Hoglund, James Butler. Rootkits: Subverting the Windows Kernel[M]. Addison Wesley Professional 2005. 7.
[2] [美]Ed Skoudis, Lenny Zelter. 决战恶意代码[M]. 北京:电子工业出版社, 2005.
[3] Jamie Butler, Bill Arbaugh, Nick Petroni R_: The Exponential Growth of Rootkit Techniques[EB/OL]. http://www.blackhat.com/presentations/bh-usa-06/BH-US-06-Butler.pdf. 2006.
[4] James Butler, Sherri Sparks. Windows rootkits of 2005, part one[EB/OL]. http://www.securityfocus.com/print/infocus/1850. 2005-11-04.
[5] Holy Father. How to become unseen on Windows NT[EB/OL]. http://www.rootkit.com/newsread.php?newsid=36. 2003. 12.
[6] Greg Hoglund, et al. ROOTKIT home[EB/OL]. http://www.rootkit.com.
[7] Holy Father. Hacker Defender Home[EB/OL]. http://hxdef.org.
[8] DiiDog, et al. Back Orifice 2000[EB/OL]. http://www.bo2k.com.