后门植入、隐藏与检测技术研究*

孙淑华[1]，马恒太[1]，张 楠[1]，卿斯汉[1]，王晓翠[2]

(1. 中国科学院软件研究所 信息安全技术工程研究中心，北京 100080；2. 吉油集团公司 职业教育中心，吉林 长春 138000)

摘 要：在对现有后门攻击技术进行研究分析的基础上，深入研究了后门的植入方法和隐藏技术并对相应的检测方法进行了讨论。采用相关技术在 Linux 平台上实现了一个内核级后门代理程序，通过实验测试，该后门代理程序达到了良好的隐藏效果。

关键词：后门代理程序；LKM；植入；隐藏；检测

中图法分类号：TP393.08 **文献标识码**：A **文章编号**：1001-3695(2004)07-0078-04

Research on Planting Concealment and Detecting Technology of Backdoors

SUN Shu-hua[1], MA Heng-tai[1], ZHANG Nan[1], QING Si-han[1], WANG Xiao-cui[2]

(1. *Engineering Research Center for Information Security Technology, Institute of Software, Chinese Academy of Sciences, Beijing* 100080, *China*; 2. *Vocational Education Center of Jiyou Group Company, Changchun Jilin* 138000, *China*)

Abstract: In this paper, we make much more research on planting method and concealment technology of backdoors and discuss detecting method of it based on analyzing current backdoor attack technology. By taking use of related technology, we implement a kernel backdoor agent program on Linux. Experimental result shows it has good concealment performance.

Key words: Backdoor Agent Program; LKM; Planting; Concealment; Detecting

1 前言

随着网络的普及，安全问题一直是人们关注的焦点，最近，CERT/CC 连续接到由后门而引起的攻击事件的报告，使得人们不得不对它十分关注。目前对后门的检测没有特别有效的方法，相关的研究工作主要是围绕如何更多地列举已出现后门和如何更好地进行分类及提取有用的特征。

后门也叫做陷阱门，是访问程序、在线服务的一种秘密方式。通过安装后门，攻击者可保持一条秘密的通道，不必每次登录到目标主机都要重新实施攻击才能完成。后门对系统安全的威胁是潜在的、不确定的。它所能实现的功能主要包括：

(1) 方便再次入侵。后门一般是秘密存在的，采用正常的方法一般难以发现，一旦植入成功可以长久保持，即使系统管理员采取了保护措施，截断了植入路径，如改变口令、打补丁、改变系统配置等，入侵者也能利用已植入的后门毫不费力地再次进入系统。

(2) 隐藏操作痕迹。使再次进入系统被发现的可能性降至最低，好的后门会提供一些隐藏手段来躲过日志系统和安全保障系统。入侵者如果能很好地利用后门和相关技术(如隐蔽通道技术等)，可以很好地隐藏其活动。

(3) 避过监控系统。IDS、防火墙和漏洞扫描软件等都是安全辅助系统，它们可以有效地提高系统的安全性，阻止各类恶意代码的攻击。但现有监控系统主要采用模式匹配的检测方法，这是很容易被避过的，好的后门一般都采用一些隐蔽或伪装手段来避过监控系统的检查。

(4) 提供恶意代码植入手段。病毒、木马和逻辑炸弹等恶意代码都对系统的安全造成巨大的威胁，这些恶意代码都可以通过后门来传播和植入。

收稿日期：2003-07-31；修返日期：2003-12-02

基金项目：国家自然科学基金资助项目(60083007)；国家信息网络安全保障持续发展计划资助项目(2002-研 1-B-002)

2 后门的植入

2.1 寻找利用可用作后门的系统漏洞

从实体配置的状态来看，漏洞是指区别于所有非受损状态的容易受攻击的状态特征。软件设计过程中一些过失或一些未检查的边界条件会形成漏洞，而大多数网络服务软件都有许多漏洞能被攻击者所利用。另外，系统管理员在配置系统时常因经验不足或粗心大意，导致系统存在严重的安全漏洞。利用系统漏洞产生的后门非常多，如：

(1) 登录程序后门。其身份验证过程可能存在漏洞，使用这类后门可以方便地登录系统，并且不容易被发现。这类后门的引入可能是由于程序设计漏洞，如存在缓冲区溢出漏洞或验证算法不合理等；也可能是程序开发人员为了调试方便或怀有特殊目的故意加入，如缺省空口令、缺省固定口令、万能口令等；还有就是攻击者替换或修改了登录程序，不影响原有的登录过程，但有捷径可以方便地进入系统。这类后门有 Rootkit, Netbios, SQL Server, PBBSER, Hidepak 和 Hidesource 等。

(2) 网络服务后门。一些系统服务程序或应用服务程序中存在着漏洞，如系统服务 Telnet, Ftp, E-mail, Rlogin 等和应用服务 IRC, OICQ 等。利用这种漏洞产生的后门很多，如常见的

AOL Admin, Attack FTP, Hack'a'Tack 和 GateCrasher 等。

(3) 系统库后门。系统库用于函数的重用而减少代码长度。利用系统库来安装后门可以做到很隐秘, 成功率也很高。如 T0rn Kit。

(4) 内核后门。内核是操作系统工作的核心, 但内核模块的处理过程存在一些漏洞, 可以利用这些漏洞将后门植入系统内核, 达到更好的隐藏目的, 该类后门在第三部分有详细描述。

2.2　植入木马等后门代理程序

远程管理型木马可以提供很好的后门服务。木马的安装主要是利用系统操作者不好的使用习惯和薄弱的安全意识潜入系统, 如操作人员随意上网抓资料或太信任电子邮件送来的文件等。

目前, 大多数的特洛伊木马都可用作后门代理程序。这类后门代理程序很多, 比较常见的有 Netbus, Bo2k, Netspy, Netbuster, BirdSpy, Subseven 等。

2.3　攻击进入目标主机安装后门

攻击进入目标主机安装后门的过程主要分为三个阶段:

(1) 锁定目标, 漏洞扫描。在发动攻击之前, 攻击者一般先确定攻击目标并收集目标系统的相关信息。目前远程漏洞扫描工具很多, 如: Nessus, Shed, IIS Cracker, Dotpot PortReady, Rangscan, Hostscangb, Stormscan, PortScan, Shadow 等, 这些工具一般都很强大, 而且可以不用掌握太多的专业知识就可以使用。如 SFind v1.85 就是一个多线程扫描工具, 它支持端口扫描; CGI 漏洞扫描; Unicode 漏洞扫描; Printer 漏洞扫描; Idq 漏洞扫描; FTP 匿名登录扫描、通过 FTP 扫描 Administrator 密码等。

(2) 目标攻击。攻击者收集到足够的信息后, 实施攻击, 进入系统。攻击者进入系统后, 如权限不够, 需要再次提升权限, 直到获取超级用户权限。权限提升主要有两种方式, 一种是口令破解或截取, 另一种方式是缓冲区溢出攻击。

(3) 后门安装。获得足够的权限后, 后门安装一般是较容易的。只需一条至几条命令即可。

3　隐藏技术

后门隐藏包括应用级隐藏和内核级隐藏。应用级隐藏是常规的隐藏方法, 通过修改、捆绑或替代系统合法的应用程序来实现隐藏。早期的后门一般是在应用级上实现隐藏; 内核级隐藏基本上分三种, 一种是指在支持 LKMs (Loadable Kernel Modules) 的操作系统上利用 LKMs 机制实现隐藏; 另一种是通过系统库来实现隐藏; 比较高级的是利用内存映射来实现隐藏, 它可以在不支持 LKM 技术的情况下实现内核级隐藏。内核级隐藏是比较难于检测的, 能避过目前绝大多数后门扫描工具、查杀病毒软件和入侵检测系统的检测。

3.1　应用级隐藏

3.1.1　后门程序和进程隐藏

(1) 选择一个隐秘的目录存放后门程序文件, 通常这样的目录(在 Linux 平台) 可以选用:

- /dev/.hdd
- /dev/.lib
- /etc/ …
- /etc/rc.d/arch/alpha/lib/.lib
- /etc/rc.d/rsha
- /usr/info/.t0rn
- /usr/lib/.egcs
- /usr/src/.poop
- /usr/src/.puta
- /usr/src/linux/arch/alpha/lib/.lib/.lproc

在 Windows 平台, 可以选用: Autoexec.bat, Config.sys, System.ini 和 Win.ini 注册表等。

(2) 后门程序设置隐藏标记。如, 在 Windows 9x 系统中, 设置隐藏标记, 这可以在任务管理器中实现隐藏。

(3) 定制程序名、修改图标。早期的后门代理程序采用固定的程序名, 使用固定的端口, 因此检测这类后门很容易。现在的后门可以自由定制安装后的程序名, 也可以将自己的服务端程序图标改成 HTML, TXT, ZIP 等各种文件的图标, 因此, 可以进行有效地伪装保护。木马类后门代理程序还可以利用此方法来诱骗操作人员下载植入。

(4) 附着、捆绑或替换合法文件。这种方法不仅可以实现隐藏保护的目的, 还可以通过合法程序的正常运行来执行后门程序, 如果选用的是系统文件, 可以达到更好的隐藏和执行效果。这类后门有 Silk Rope 2K, SaranWrap, Rootkit 和 TCP Wrapper 等。

(5) 修改或替换管理命令。修改或替换用于查看程序文件和进程信息的管理命令, 使后门程序和进程可以很好地隐身。如替换 "ls" "find" "du" 可以实现文件和目录隐藏; 替换 "ps" "top" "pidof" 可以实现进程隐藏; 替换 Netstat, Ifconfig 可实现网络的连接状态隐藏; 替换 Kill, Killall 可防止删除攻击者的进程, 替换 Crontab 能隐藏攻击者的定时启动信息, 替换 Tcpd, Syslogd 隐藏攻击者的连接信息等。

(6) 进程动态注入。将后门程序可执行代码注入正在运行的进程中, 可以极大地提高后门的生存能力。此项技术可以很好地隐藏后门程序, 并且植入后门代理程序后, 不会出现独立进程, 可以达到很好的隐藏效果。该项技术在 Windows 下已比较成熟, 但在 Linux 下还处于讨论试验阶段, 没有公开的后门出现。

(7) 消除痕迹。一个好的后门安装完成后或后门程序任务完成后, 可以主动销毁原后门程序, 这样一来, 服务端管理人员就很难追踪到后门程序的来源。另外在安装过程中留下的一些痕迹也要清除, 如日志、临时文件和所使用过的工具的记录信息等。

3.1.2　通信隐藏

攻击者利用后门可以很容易地再次进入系统, 但在交互过程中易于被发现, 特别是现在入侵检测技术和漏洞扫描技术的发展, 要求后门具有更高的通信隐藏能力。

(1) 通信内容隐藏。通信内容一般采用加密方式进行保护, 但加密方式无法隐藏通信状态, 无法避过基于流量分析的检测方法。现在较好一点的后门代理程序都提供加密通信方式来保护通信内容, 如 Bo2k, Back Orifice 和 Netbus Pro 等。

(2) 通信状态隐藏。

- 通信端口隐藏: ①选用不常用的通信端口, 早期的后门一般都选用确定的端口, 大多数的后门检测工具也是通过判断相应的端口来进行工作的, 因此定制端口可以避过大多数检测软件的检测; ②将后门隐藏在合法端口, 为后门数据包设定标志, 通过标志来区分会话, 同时不会影响原有端口的服务, 这种方法的优点在于无法通过端口来判断后门的类型, 也不容易通

过流量分析来检测,只能通过查找相应的标志来检测,但标志是易变的,因此该方法可以很好地隐藏通信端口。具有端口定制功能的后门代理程序有 Ping Backdoor, WinShell 等。Executor 利用 80 端口传递控制信息和数据,实现其远程控制。而 Code Red II 则利用 80 端口来进行传播。

● 通信连接隐藏。主要通过隐蔽通道技术来实现,在 TCP/IP 协议簇中,有许多冗余信息可用于建立隐蔽通道。攻击者可以利用这些隐蔽通道绕过网络的安全机制秘密地传输数据。如可选择 TCP 包的“Identification”域、“Sequence Number”域和“Acknowledgment Number”域等进行通信隐藏,还可以选用 UDP 和 ICMP 包进行通信隐藏。隐蔽通道技术可很好地隐藏通信内容和通信状态,当前隐蔽通道的检测还是一个难点。目前常见能够提供隐蔽通道方式进行通信的后门有 Bo2k, Code Red II, LOKI, MP3Stego, Nimida 和 Covert TCP 等。

3.2 内核级隐藏

一般商业操作系统的内核是不公开的,对于不公开内核的操作系统,不容易出现具有内核级隐藏能力的后门。下面就以公开内核源码的 Linux 操作系统为例来讨论后门的内核级隐藏。

3.2.1 LKM(Loadable Kernel Modules) 隐藏

LKMs 是指内核加载模块,它主要用于系统扩展功能,不需要重新编译内核就可以被动态地加载。现在许多内核开放的操作系统都支持这种功能,如 Linux, Solaris 和 FreeBSD。Linux 操作系统有两种工作模式,一般用户工作模式和超级用户工作模式。用户进程工作在用户模式,内核进程工作在超级用户工作模式;用户进程访问硬件资源要通过系统调用来进行,如文件操作的系统调用有 Open, Read, Write, Close 等;进程操作的系统调用有 Fork, Exec 等;网络操作的系统调用有 Socket, Connect, Bind, Listen, Accept 等,还有一些其他低级系统调用。在 Linux 下,系统调用的相关信息可从文件/usr/include/sys/syscall.h 中查到。后门的内核级隐藏不是通过替换系统管理命令文件来实现的,而是通过替换或调整系统调用来实现的。如 PS 命令通过/proc(procfs) 来获得进程信息,可以利用 Procfs 隐藏特殊的进程来实现进程隐藏,还可以通过系统调用的重定向来实现隐藏。下面就对如何在内核级利用 LKM 技术实现文件、进程和通信隐藏来进行讨论:

(1) 系统调用重定向: 利用 exec( ) 系统调用

实现重定向,主要当对隐藏目标使用 exec( ) 系统调用时使用替换版本,而对于其他则使用原始版本。

(2) 文件隐藏: 文件和目录信息主要是利用 Ls, Du 等系统命令来显示的,而 Ls, Du 等命令获取目录信息都利用 sys_getdents( ) 系统调用,在内核级上通过改变 sys_getdents( ) 的功能可实现相应的文件、路径隐藏。

(3) 进程隐藏: 进程信息是通过映射到/proc 文件系统中,通过改变 sys_getdents( ) 系统调用使任务结构不可见就可以实现进程隐藏。

(4) 通信连接隐藏: 通信连接信息记录在/proc/net/tcp 和/proc/net/udp 文件系统中,通过改变 sys_read ( ) 系统调用可实现相关信息隐藏。通过打开这两个文件并匹配特定的字符串,则可滤掉相关信息。另外还可以通过修改 TCP/IP 协议栈来实现隐藏。在内核里,每个协议都在* inet_protocol_base 和* inet_protos [MAX_INET_PROTOS] 注册自己。当系统初始化时,所有支持的协议会在 inet_protocol_base 注册并加到 inet_protos 的哈希表里。对于到达的协议包,内核将检查这个哈希表,调用相应的处理函数和系统调用。因此,可以用处理函数来替换原始的协议处理函数。对于到达的 UDP 或 TCP 包,内核将检查这个包是否是指定的包。若是则执行修改了的处理函数,若不是则执行原来的处理函数。所以不用产生新进程等待连接,从而实现通信连接隐藏。

(5) LKM 模块隐藏: Linux 操作系统将 LKM 模块信息存放在一个单链表中,可以通过将该链表中有关后门代理程序的 LKM 模块信息删除,可以避过“lsmod”之类的管理命令的检查。

(6) LKM 符号隐藏: 被加载的内核模块要与其他内核进行通信,所以有必要对符号进行隐藏,通过使用宏“EXPORT_NO_SYSBOLS”可以实现符号隐藏。

(7) 通信隐藏: 当 LKM 被加载后,攻击者要与之进行通信。通信隐藏一般选用加密技术或隐蔽通道技术来实现,加密技术可以隐藏通信内容,隐蔽通道技术不仅隐藏通信内容,也可以隐藏通信状态。内核级后门在利用隐蔽通道通信时需要修改协议栈来处理通信内容。

目前,较好的 LKM 后门有: ① Knark 是 Sekure.net 的成员 Creed 开发的一个基于 Linux 2.2 内核的后门模块,它实现的功能有隐藏文件、隐藏进程、重定向执行程序、隐藏网络连接、并能以 Root 身份运行命令; ② Adore 是由 Teso 小组成员开发的一个 LKM 后门,能够实现的功能有隐藏文件、隐藏进程、以 Root 身份执行程序并提供了卸载功能(当然它的卸载是需要提供密码验证的) ,这个后门模块可以支持 Linux 2.2 及 2.4 的内核; ③ Phide 也是一个 LKM 后门,它只实现了隐藏进程。

3.2.2 系统库后门隐藏

从内核空间(/proc) 到用户空间(如/bin/ps 和 top) 之间传送进程信息时要进行系统库文件调用,这时,使用特殊的系统库文件(Libproc.a) 来替换标准的系统库文件,使之被调用。它没有替换二进制文件,而是使用了自己系统库(Libproc.a) 来过滤掉某些进程信息来实现隐藏。也可以在没有传送到内核前直接调用主系统库(glibc/libc) ,这叫做对应内核的用户空间重定向。利用这些技术可以实现进程与文件的隐藏。T0rn Kit 是这种类型后门的一个典型代表。

3.2.3 内存映射隐藏

这是未来隐藏技术的一个发展方向,主要通过直接更改内存映射(/dev/mem) 而不采用加载内核模块来影响系统调用或内核运行来实现后门隐藏。这方面的技术已由研究阶段逐步转向成品化。这种后门不需要插入任何模块与组件就能实现隐藏。如 SucKit 就是一个例子,其原理由 Silvio Cesare 提出,并由 Sd 实现 ,可以工作在内核 2.2 和 2.4 上。

4 后门的检测

4.1 应用级后门的检测

无论采用什么方式植入后门,并采用什么样的伪装隐藏手

段, 总可以通过一些方法来进行检测, 如通信端口检查、通信特征匹配、植入痕迹查找和完整性检查等。对于应用级隐藏, 最有效的检测方法是完整性检测, 如 MD5 校验和法就能得到很好的效果。现在这方面的检测工具有 Tripwire 和 Aide 等。Tripwire 首先使用特定的特征码函数为需要监视的系统文件和目录建立一个完整性特征数据库, 所谓特征码函数就是使用任意的文件作为输入, 产生一个固定大小的数据(特征码) 的函数。入侵者如果对文件进行了修改, 即使文件大小不变, 也会破坏文件的完整性特征码。利用这个数据库, Tripwire 可以很容易地发现系统的变化。

4.2　内核级后门的检测

内核级后门是在内核级隐藏目录、文件、进程和通信连接等, 它不修改程序二进制文件, 因此 MD5 校验和法也就无效了。按照内核级后门的隐藏特点, 已经出现了一些不同类型的检测方法, 如内核级隐藏的一种方法是通过修改系统调用表来实现系统调用重定向从而隐藏文件、进程和通信连接, 所以通过检查系统调用的内存地址就可能发现是否被植入了后门, 这类检测工具主要有 KSTAT, 利用它可以很好地检测到此类后门, 如 Knark, Adore 无法逃过 KSTAT 的检测; 另外, 内核级后门一般都要进行内核模块的加载, 因此通过监控内核的变化可以很好地防御内核级后门, 如 stMicheal-LKM 就是这样的一种检测防御工具, 它可以很好地防御目前出现的内核级后门。

另外, Vern Paxson 等人针对处于交互过程中的后门, 根据数据包的大小、数据包间隔的时间特性及连接建立的方向提出了后门检测的较通用的算法, 并针对特定传输协议的传输特性是否异常提出特定协议算法, 但算法的有效性并不理想。

5　实现的后门代理程序

5.1　原型实现

我们在 2003 年承担了国家信息网络安全保障持续发展计划的研发项目, 在 Linux 平台上成功地实现了一个内核级后门代理程序, 该后门代理程序具有远程访问功能。它的服务端程序作为 Linux 内核模块, 装载到目标主机的内核空间, 远端的客户端可与之进行通信, 并以 Root 身份执行命令、窃取目标主机的敏感信息。服务端程序利用上面提到的 LKM 隐藏技术来实现文件、进程、通信连接隐藏, 但具体实现方法又有所不同。该后门代理程序没有采用修改系统调用指针实现系统调用重定向来实现隐藏, 而是通过在内核中修改系统调用相关函数的方式来实现隐藏。模块信息隐藏是通过删除模块链表中的对应项实现的, 因此利用 “lsmod ”之类的管理命令无法检测到后门的存在。进程隐藏是通过修改 /proc 文件系统的相应函数来实现的, 因此, 可以成功避过进程管理命令 PS, TOP 等的检查。

该后门代理程序还利用内核级的通信连接隐藏技术, 选用隐蔽通道技术(ICMP 包的数据域) 实现了网络通信状态隐藏和通信内容隐藏。通过修改系统调用 Sys _ read ( ) , 滤过了 /proc/net/raw 中的通信信息记录, 隐藏了通信连接状态, 可以避过 Netstat 的检查。通信内容的处理是通过修改协议栈中的协议处理例程来实现的。

该后门代理程序支持 Linux 2.4 内核。

5.2　与其他著名后门代理程序的比较

目前, 比较著名的 LKM 后门代理程序有 Adore 和 Knark, 它们所采用的隐藏技术主要是修改系统调用指针实现系统调用重定向来实现隐藏。如 Adore 修改了九个系统调用 Getdents, Fork, Clone, Kill, Write, Close, Mkdir, Stat 和 Open 来实现隐藏; Knark 修改了七个系统调用: Fork, Read, Execve, Kill, Ioctl, Settimeofday 和 Clone。

通过修改系统调用指针是不能避过相应检测工具 KSTAT (测试采用 v2.4) 的检测, 因此 KSTAT 可以很容易地检测到 Adore 和 Knark。而我们的后门代理程序是通过直接修改系统调用相关函数来进行隐藏, 没有修改系统调用表, 所以能够成功避过 KSTAT 的检测。Chkrootkit(测试采用 v0.38) 是通过一些恶意代码签名来检测的, 后门代理程序能成功避过其检测。stMicheal-LKM (测试采用 v0.11) 是检测内核变动的工具, 我们的后门代理程序对抗它的方法是, 加载过程中在内核中找到 stMicheal 的代码, 并将其删除, 使其丧失监视内核改动的能力。

由于我们的后门代理程序在通信中采用隐蔽通道技术, 所以可以成功避过目前常见的 IDS(Snort, Realsecure 等) 。

6　结束语

本文详细地研究了后门的植入方法、隐藏技术, 并在基础上实现内核级后门代理程序, 通过测试它具有很好的隐藏性能, 能避过许多工具的检测。本文对相应的检测方法进行了讨论。在目前的计算机体系结构和技术条件下, 还不可能给出一种通用的算法来检测和防御这类攻击。在未来的检测道路上, 更多趋向于混合利用几种检测方法, 这样可以达到互相补充, 互相弥补, 以期达到提高检测精度的目的, 从而制止各种攻击。

参考文献:

[1] Zhang, V Paxson. Detecting Backdoors[C]. 9th USENIX Security Symposium, 2000.

[2] ackdoors, Trojan Horses. By the Internet Security Systems ’X-Forse [M]. Information Security Technical Report, 2001, 6(4).

[3] ragmatic /THC, Complete Linux Loadable Kernel Modules(v1.0) [EB/OL]. http://packetstormsecurity.nl/docs/hack/LKM_HACKING.html, 1999-03.

[4] oby Miller. Detecting Loadable Kernel Modules (LKM) [EB/OL]. http://www.incident-response.org/LKM.htm, 2000.

[5] SS X-Force White Paper: Back Orifice 2000 BackdoorProgram: In TECS[EB/OL]. http://www.itsecurity.com/papers/bo2k.htm, 1999-07-15.

[6] oseph Lo, et al. NetBus Backdoor Attack[EB/OL]. http://yasarozg.webhostme.com/eng/trojans/netbus.htm, 2002.

[7] rabhaker Mateti. Viruses, Worms and Trojan. http://www.cs.wright.edu/~pmateti/Courses/499/Viruses, 2002.

[8] tealth[EB/OL]. http://www.team-teso.net/releases/adore-0.34.tgz, 2003.

作者简介:

孙淑华(1968-) , 女, 工程师, 硕士生, 主要研究方向为信息安全对抗; 马恒太(1970-) , 男, 助研, 博士, 主要研究方向为网络信息安全和分布式计算; 张楠(1979-) , 男, 主要研究方向为信息安全对抗; 卿斯汉(1939-) , 男, 研究员, 博士生导师, 主要研究方向为信息安全理论与技术; 王晓翠(1971-) , 女, 讲师, 主要研究方向为多媒体教学。