Computer Science and Application 计算机科学与应用, 2015, 5(12), 429-435
Published Online December 2015 in Hans. http://www.hanspub.org/journal/csa
http://dx.doi.org/10.12677/csa.2015.512054

Hans汉斯

Study of Trojans Detection and Prevention Technology

Shaohua Wu, Yong Hu

College of Electronics and Information Engineering, Sichuan University, Chengdu Sichuan
Email: huyong@scu.edu.cn

Received: Dec. 5th, 2015; accepted: Dec. 25th, 2015; published: Dec. 28th, 2015

Abstract

Based on reverse analysis of many current popular Windows' Trojans behavior, the new technologies used by Trojans were summarized, including program hidden, process hidden, communication pattern and means to avoid killing. Combined with the current mainstream security software to detect the Trojan, some new technologies and opinions against the Trojan threat were present.

Keywords

Trojan, Masquerading Technology, Process Hidden, Communication Protocol, Avoid Killing, Trojan Detection

木马检测与防护技术的发展

吴少华，胡　勇

四川大学电子信息学院，四川　成都
Email: huyong@scu.edu.cn

收稿日期：2015年12月5日；录用日期：2015年12月25日；发布日期：2015年12月28日

摘　要

通过对大量当前流行的windows木马程序进行逆向，分析木马在伪装技术、程序隐藏方式、进程隐藏方

文章引用: 吴少华, 胡勇. 木马检测与防护技术的发展[J]. 计算机科学与应用, 2015, 5(12): 429-435.
http://dx.doi.org/10.12677/csa.2015.512054

式、通信方式和免杀手段上所使用的各种技术，并结合当前主流的安全软件对木马的检测效果和检测方式，提出对抗木马新技术的方法。

关键词

木马，伪装技术，进程隐藏，通信协议，免杀，木马检测

1. 引言

特洛伊木马具有隐蔽性、迷惑性和针对性等特点，且破坏性大，成为网络安全的最大威胁之一。

由于网络安全技术的发展，在攻防斗法中，木马的相关技术也得到了快速发展，先后涌现出很多技术，主要体现在木马的伪装迷惑、程序隐藏、进程隐藏、网络通信模式、免杀技术等方面。

2. 木马伪装技术

源于古希腊神话的特洛伊木马通过伪装和隐藏，进行用户不知道或不期望的行为是其天性[1]。伪装技术体现在很多方面。

2.1. 图标伪装

对用户来说，图标是识别文件是否是可执行文件的显著特征，最早的隐藏是将木马可执行文件的扩展名隐藏，并将图标设置成图片文件、Word 文件、Pdf 文件、文件夹等类型的图标，以此欺骗用户打开木马文件。而此类木马一般会在运行后释放并打开一个与图标相应的文件，具有较好的伪装性。

2.2. 文件名伪装

随着安全事件的披露和人们安全意识的提高，很多人将 windows 的默认设置修改为显示已知文件的扩展名，此时仅仅伪装文件图标显然很容易被发现，文件名伪装出现，主要有两种方式：

1) 采用 windows 下直接运行的其它文件类型，如.scr、.pif、.com 等格式。

2) 在文件名中插入 Unicode 控制字符使文件名反转，“隐藏”真实扩展名，以欺骗用户运行。如 sexe.jpg 的真实文件名为 sgpj.exe。

2.3. 应用程序漏洞利用

图标伪装和文件名伪装只是改变木马程序的“外表”来诱骗用户运行木马，但其仍是 PE 文件。

利用常用应用程序如 Microsoft Office、Adobe Reader、Winhelp 等的漏洞，构建相应格式的特定文件以缓冲区溢出等方式获取应用程序的控制权，从而执行木马程序。此类木马的文件头和文件结构与正常应用程序的文档文件一致，因此迷惑性较 PE 文件大。

3. 木马隐藏技术

在用户运行了木马伪装文件后，此时伪装程序会释放出真正的木马文件，并通过修改注册表等来实现开机自启动，以长期隐蔽地潜伏在用户计算机中。因此，木马的程序隐藏技术主要分为四方面：木马文件的隐藏、木马启动方式的隐藏、木马进程的隐藏、木马通信的隐藏。

3.1. 木马文件隐藏技术

木马的伪装文件被运行后，为了长期潜伏在系统中，会向硬盘释放木马文件。由于系统文件夹的文

件众多，许多木马将自身释放到系统文件夹，达到“鱼龙混杂”隐藏自己的目的。而安全软件对系统文件的监视越来越严格，很多木马又将自己隐藏到其他不易被发现的目录中，如 Program Files。

还有一些木马开始使用 Rootkit 技术，通过系统内核拦截系统遍历文件的 API 函数，实现应用层文件的不可见。也有木马修改磁盘引导区记录(MBR)来执行木马程序，隐蔽性也较强。

3.2. 开机启动方式隐藏技术

开机启动是木马运行的基本手段，其方法主要有：注册表、服务、DLL 劫持、系统文件替换等。

3.2.1. 注册表自启动

注册表是 Windows 系统中一个重要的数据库[2]，用于存储系统和应用程序的配置信息。通过修改注册表能够实现应用程序开机启动，如常用的 HKCU\Software\Microsoft\Windows\CurrentVersion\Run。由于该启动项容易被用户和安全程序发现，更隐蔽的启动路径被利用，如注册表的 ActiveSetup, WinLogon 等。

3.2.2. 服务启动

木马将自身注册为系统服务，并设置服务类型为自动启动，实现木马随计算机启动而启动。木马会替换系统服务对应的动态链接库(DLL)文件或者自己新建服务来实现自启动，如远控工具 ZXShell 等。

3.2.3. DLL 劫持

Windows 程序加载 DLL 文件时，首先会查找应用程序目录下是否存在指定文件名的文件，如果不存在，才会搜索系统环境变量指定的目录下是否存在该文件，因此将木马文件改成与系统的动态链接库文件同名，放在想要劫持的应用程序目录中，便可以实现动态链接库劫持。木马伪造的动态链接库一般会先启动木马程序，然后加载正常的系统动态链接库，从而实现隐蔽执行。常见的 DLL 劫持有对 explorer 的劫持，以及对 iexplorer、firefox 等常用应用程序的劫持。

3.2.4. 系统文件替换

许多系统文件是应用程序和操作系统正常运行所必须的，如 ws2_32.DLL 是许多网络应用程序必须加载的系统文件，通过修改这些系统文件，在应用程序使用此文件时就能启动木马程序。

3.3. 木马进程隐藏技术

木马常用的隐藏进程的方式有：

3.3.1. DLL 注入

由于防火墙对每个应用程序的连网行为做了严格限制，为了突破防火墙，木马常用的方式是 DLL 注入，将一个包含有恶意代码的 DLL 放在另一个进程的地址空间里，由这个进程加载并运行。如果将恶意的 DLL 注入到防火墙信赖的应用程序中，一旦应用程序开始运行，这个 DLL 就可以发送和接收网络数据。比较常见的是将恶意的 DLL 注入到 iexplorer 进程中，因为 windows 系统都给予 IE 足够的权限访问互联网。DLL 注入的方式主要有 LoadLibrary 和设置全局钩子两种方式。

3.3.2. 进程注入

进程注入和 DLL 注入类似，是在一个远程进程中创建并执行一个恶意的线程。只不过进程注入使用的是直接插入代码并创建远程线程的方式来运行自己，这样便能做到在远程进程中无模块，更加不易被检测到。木马以被信赖的应用程序的一个线程方式运行，任务管理器中不会出现新的进程。查看进程模块也无法看到任何木马相关的模块。这种方法受到木马程序编写者的青睐。

3.3.3. 注册为服务

木马将自身注册为 windows 服务便可以长时间地在系统后台偷偷运行，这种服务很多是由系统的 svchost.exe 进程加载的，用户通过任务管理器无法查看到服务(木马)的进程，但是通过查看进程模块的话，可以在相应的进程中看到木马对应的 DLL 文件。有些新木马甚至会将系统服务对应的 DLL 替换成自己，从而实现自启动和隐藏进程。

3.3.4. Rootkit 隐藏进程

随着黑客技术的发展，木马的隐藏技术日趋成熟，隐藏手段也由 Ring3 级发展到了 Ring0 级，常见的 rootkit 隐藏进程的方法有：删除进程双向链表上的进程对象，SSDT 内核调用挂钩等。

3.4. 木马网络隐藏技术

木马与控制端通过网络进行通讯，传统的木马通常会绑定一个端口，控制端则与指定的 IP 和端口建立连接和通信。但随着防火墙技术的发展，从外部连接内部计算机的行为受到了严格的控制，因此出现反弹连接的木马和使用网络隧道的木马。

3.4.1. 反弹连接

目前，用户和 ADSL 设备之间往往有 NAT 设备，使得木马程序为建立连接而在目标系统中打开监听端口变得无意义。反弹连接的木马主动连接控制端，这是对防火墙的严重挑战[3]。如果黑客将控制端的端口设置成 80，防火墙就很难判断一个程序是正常访问网站，还是木马的连接。一个简单的建立反弹连接的方法是从服务器中的动态 DNS 解析出域名，然后连接该域名指向的 IP 地址。目前，几乎所有的木马都采用反弹连接了。

3.4.2. 网络隧道

无论使用正向连接还是反弹连接，其通信特征如协议和 IP 都是能分析出的。而防火墙等对网络数据包的检测越来越严格。木马为躲避检测，采用网络隧道技术，利用一种网络协议来封装传输另一种网络协议的数据。例如，VPN 中网络隧道被用来在两个独立的网络之间建立一个可靠的连接。

网络隧道中很多协议可以使用，只要该协议被防火墙信任并允许通过，如 SMTP, DNS, ICMP 和 HTTP 协议等。

进一步，为了隐藏自己和方便通信，攻击者往往会利用互联网上的网络空间做中转，如 FTP 服务器或者知名网站，通过特定的文件或评论向木马发送指令和传输数据等。即使能够检测并跟踪木马到攻击者的免费空间，但没有一个受害者的计算机与攻击者直接连接，很难追查到攻击者。

4. 木马免杀技术

木马免杀技术主要有结束杀软进程，修改配置，多态代码，插件化，大文件等。

4.1. 结束安全软件进程

木马试图结束安全软件进程，使系统失去任何检测和防御能力。通过发送“窗口关闭消息”关闭目标进程。安全应用程序可以设置忽略此消息。但还有其它方法可以结束进程，如挂起目标程序的所有子线程或者修改内存中的代码，就会使程序崩溃或退出。许多安全软件并非对这些攻击免疫[4]。

4.2. 修改安全软件配置

安全工具启动时，将加载自定义的配置文件信息。此配置文件一般包含了用户的配置，如白名单，信任列表等。木马程序通过伪造、修改安全程序的配置文件，来实现躲避安全软件的检测和查杀。

4.3. 大文件免杀

对于可疑的文件，许多安全软件会选择将文件上传到服务器，由人工分析样本是否是木马程序。由于木马程序一般较小，许多安全软件为了避免占用带宽资源等，对上传样本的大小都做了限制。一些木马正是利用了这一特点，在释放到指定路径时，在木马文件的“尾巴”加上了很多垃圾数据，使得木马文件变得很大，以避免杀毒软件将自己上传，从而达到较长时间免杀的目的。

4.4. 随机文件名和随机哈希值

随着互联网的发展，很多安全软件企业推出了“云查杀”，即在互联网上一旦发现木马，立即提取其哈希值，那么所有与服务器连接的安全软件便能立即通过哈希值检测和查杀该木马。很多通过文件名、路径名进行的快速查杀，也能轻易地杀掉木马。为了避免此类查杀，一些木马使用了随机的文件名和随机的哈希值，从而实现延长免杀期的目的。

4.5. 多态和变形代码

一些木马使用了多态代码技术，以改变代码的特征但维持函数的功能。

多态技术可用来产生大量不同的木马。它可以添加到木马服务端的配置代码中，所以没有大小的限制也不用防止逆向。

4.6. 插件和传输代码

目前，有一些远程控制木马，使用了插件技术，即将木马的各项功能分别写在每个插件里面，方便木马功能扩充的同时，还增加了查杀的难度。因为每个插件都有不同的特征码，而根据不同功能的配置，木马每次“携带”的插件文件也会不一样。

还有一些改进型的插件木马，会在成功连接网络后，根据控制端的命令从控制端下载相应的插件文件到木马目录并进行加载，以实现相应功能的扩充。更有甚者，直接通过网络传输 shellcode 到自身内存，并加以运行，大大增加了查杀的难度。

5. 硬件木马

出于各种目的，攻击者在不断寻找新的攻击方法。对固件和硬件的攻击已大量出现。今年，不仅出现了针对显卡 GPU 的攻击[5]，还发现了对数十种常见品牌的硬盘固件重新编程的木马[6]。

同时，集成电路技术的发展使得片上系统功能越来越复杂，采用的 IP 核可能含有恶意代码，EDA 工具可能被篡改，而硬件的设计和制造往往是分离的，硬件外包制造中也可能被嵌入木马电路，这些都可能导致硬件被植入木马[7] [8]。

硬件木马的检测可以是破坏式的芯片反向分析、侵入式的嵌入特征电路检测、非侵入式的逻辑测试和旁路分析。非侵入式检测不用修改原始电路设计，目前研究较多[9]。

6. 手机木马

移动互联网的迅猛发展和移动终端智能化，各种 APP 争相推出，而这些 APP 或从官方网站下载或来源于第三方网站，有些并没有进行严格的审查，易被利用捆绑上传木马软件。这些木马软件的很多技术与计算机木马相通，但也有特殊之处。以手机木马为例，可由被感染的安装包植入，由无线网络 ARP 欺骗植入，二维码扫描植入[10]，短信链接植入，网页挂马，开发工具挂马(如 XcodeGhost)等方式植入。手机木马可通过短信接收命令，执行发送短信、收集 GPS 位置信息、ROOT 攻击、窃取各种帐号和口令及

文件信息、通话录音以及伪关机[11]等功能。同时，还可能通过修改 boot 分区和启动配置脚本等方式取得高权限，自动安装各种流氓和广告软件等[12]。

手机木马为了阻止被卸载，可能将自己伪装成系统文件，使其迷惑性更强，更易进入用户手机中，而用户看到此类名称的文件，怕影响手机使用也不敢随意删除。也可能以屏蔽卸载应用的界面、使得卸载按钮无法点击、监听系统日志、抢占屏幕焦点、注册设备管理器等[13]方式阻止卸载。

因此，用户不要随意开放手机 Root 权限；日常使用中谨慎点击软件推送广告、来源不明的手机软件、安装包、文件包等勿随意下载；手机上网时，不明链接和二维码不随意点击、扫码等。安装手机安全软件定期查杀手机木马[14] [15]。

7. 木马程序的检测技术

针对木马的伪装性和隐藏性，木马程序的检测技术也随之发展，针对各种木马新技术，有相应的检测方法[16]。

7.1. 针对木马伪装技术的检测方法

此类检测针对木马文件，即捆绑了木马的文件，由于尚未运行，只要判断出是木马程序，直接删除即可。木马程序可以通过特征码扫描和人工识别等方式加以判断。

对于使用图标伪装的木马，显示扩展名就能判断其类型。

对于文件名伪装的木马，windows 能够直接运行的文件扩展名除了.exe，还有.src、.pif、.com、.cmd 等格式。通过重命名/显示 Unicode 字符等方式能够识别出是否使用了文件名反转的伪装方式。

对于利用应用程序漏洞的木马，通过人工是无法判断的，应当使用专业的软件进行静态扫描。并及时为应用程序打上补丁。

7.2. 针对木马隐藏技术的检测方法

7.2.1. 针对文件和启动项隐藏的检测

检测注册表自启动项目、注册表指定路径键值或利用基于特征码扫描的静态扫描技术扫描硬盘文件，能够扫描出可疑的木马，对于使用 rootkit 技术的木马，可以使用专门的 rootkit 扫描工具。

7.2.2. 进程隐藏技术的检测

针对木马进程隐藏的方式，发展出内存查杀的检测方法。内存查杀既是扫描每个进程的内存，将内存数据与数据库中的内存特征码进行匹配，以此检测是否有木马运行。

不管木马使用的是 DLL 注入还是进程注入，或者是以服务的方式运行，其代码必然存在于内存中，所以内存检测能够及时有效地扫描出当前所有正在运行的应用程序中是否有可疑的代码。

针对 Rootkit 木马，目前也有很多种检测方法，如：通过读取 Windows 的线程分派器(dispatcher，scheduler)所用的链表来列出进程、通过 SSDT 内核调用挂钩的检测与脱钩等方式来检测出可疑的木马程序。

7.2.3. 针对通信隐藏技术的检测

对于木马的通信检测主要有通信数据包检测、通信域名/IP 检测、通信 URL 检测等。通过这些检测能够检测出已知的木马。使用防火墙和入侵检测系统也能较好地在通信数据包层面上检测木马的存在。

7.3. 木马检测技术的发展方向

针对木马技术的日益发展和木马数量的快速增加，传统的基于特征码扫描的检测技术越来越力不从心。木马特征库的日益庞大，导致了查杀速度等性能的下降。目前，主要的检测新技术有云查杀、启发

式查杀和主动防御技术、网络数据包检测技术等。

7.3.1. 云查杀

云查杀技术主要解决本地特征码数据库庞大的问题。其将特征码数据放在服务器上，通过网络传输数据进行特征码匹配，从而实现快速查杀。该技术将木马特征码存储在服务器上，更新速度快，特征码扫描引擎也在服务器上，不耗费本地 CPU 等资源。

7.3.2. 启发式查杀

启发式查杀通过分析判断正常程序和木马程序指令行为的区别，提取木马行为的特征和规律，是一种基于代码行为的人工智能检测方式[17]。这是一种通用的，不依赖于特征库的检测技术，还在不断发展中。

7.3.3. 主动防御

主动防御不以特征码作为木马的判断依据，而是从最原始的木马定义出发，直接将程序的行为作为判断木马的依据，从系统底层监视木马的敏感行为(如向系统目录释放文件、修改启动项、钩子、注入等)，解决了传统安全软件无法防御未知恶意程序的弊端，从技术上实现了主动防御。

8. 结论

“道高一尺、魔高一丈”，攻击与防御是一对矛盾，攻击手段的进步最终必然导致防御技术的提高，而为了突破防御能力增强的系统，攻击者又会找到新的攻击方式。因此木马相关技术也在不断地扩充和发展。信息安全一直在迅速发展变化中。

参考文献 (References)

[1] 刘澜, 高悦翔. 木马隐藏技术的研究与分析[J]. 通信技术, 2010(4): 78-80.

[2] 谢宗仁. 木马原理分析与实现[D]. 山东大学, 2009.

[3] 梅登华, 林耀通. 基于 Multi-Agent 的木马模型设计[J]. 电子技术应用, 2008, 34(5): 138-140.

[4] 许国顺. 木马攻击与防范技术研究[D]. 成都: 四川大学, 2006.

[5] 新型木马入侵方式显卡也会感染病毒[EB/OL]. http://vga.zol.com.cn/521/5216200.html

[6] 修改硬盘固件的木马探索方程式(EQUATION)组织的攻击组件[EB/OL]. http://www.antiy.com/response/EQUATION_ANTIY_REPORT.html

[7] 谢海. 基于 FPGA 的硬件木马检测[D]. 广州: 广东工业大学, 2013.

[8] Ngo, X.T., Bhasin, S. and Danger, J.-L. (2015) Linear Complementary Dual Code Improvement to Strengthen Encoded Circuit against Hardware Trojan Horses. 2015 *IEEE International Symposium on Hardware-Oriented Security and Trust* (*HOST*), Washington DC, 82-87.

[9] Khaleghi, B., Ahari, A. and Asadi, H. (2015) FPGA-Based Protection Scheme against Hardware Trojan Horse Insertion Using Dummy Logic. *IEEE Embedded Systems Letters*, **7**, 46-50. http://dx.doi.org/10.1109/LES.2015.2406791

[10] Android 平台下二维码漏洞攻击杂谈[EB/OL]. http://www.wtoutiao.com/p/17472Js.html

[11] Android 间谍软件可劫持设备并伪装成已关机以便继续监听[EB/OL]. http://www.cnbeta.com/articles/372017.htm

[12] 警惕“不死”木马逆袭安卓手机[EB/OL]. http://article.pchome.net/content-1699805.html

[13] 张建国. “顽固木马”对抗的技术剖析[J]. 计算机与网络, 2015(Z1): 72.

[14] 2015 年末最流行手机木马分析报告[EB/OL]. http://news.163.com/15/1209/15/BADGDM2Q00014JB6.html

[15] 360 公布十大手机恶意程序木马伪装微信消息传播[EB/OL]. http://mobile.people.com.cn/n/2014/0212/c183175-24338339.html

[16] Gudipati, V.K., Vetwal, A. and Kumar, V. (2015) Detection of Trojan Horses by the Analysis of System Behavior and Data Packets. 2015 *IEEE Long Island Systems, Applications and Technology Conference* (*LISAT*), Farmingdale, 4.

[17] 赵玉明. 木马技术揭秘与防御[M]. 北京: 电子工业出版社, 2012.