doi 10.3969/j.issn.1671-1122.2015.02.010

基于特征分析和行为监控的未知木马检测系统研究与实现

郝增帅[1]，郭荣华[2]，文伟平[1]，孟正[1]

（1. 北京大学软件与微电子学院，北京 102600；2. 洛阳电子装备试验中心，河南洛阳 471003）

摘 要：木马是以盗取用户个人信息和文件数据，甚至是以远程控制用户计算机为主要目的并尽可能隐藏自身的恶意程序。近年来，随着黑客行为的职业化、利益化和集团化，网络入侵与攻击手段日新月异，木马等恶意代码已成为我国网络安全的重要威胁。现阶段，木马检测通常依赖于病毒软件的检测能力，防病毒软件一般采用特征码比对和行为识别的方式进行木马查杀，这种方式需要防病毒软件拦截木马样本进行分析，提取木马样本，对木马特种库进行升级后对木马进行识别，滞后性很强，无法对新出现的或无已知特征的木马进行查杀。文章对木马反杀毒技术、隐藏技术、突破主动防御技术进行探讨，并以此为基础，提出基于特征分析和行为监控的木马检测技术，完成了未知木马检测系统的设计与实现，能够在一定程度上弥补现有防病毒软件及安全措施只能查杀和监测已知木马而不能识别和查杀未知木马的不足。

关键词：木马检测；木马查杀；特征分析；行为监控

中图分类号：TP309 文献标识码：A 文章编号：1671-1122（2015）02-0057-09

中文引用格式：郝增帅，郭荣华，文伟平，等. 基于特征分析和行为监控的未知木马检测系统研究与实现 [J]. 信息网络安全，2015，(2)：57-65.

英文引用格式：HAO Z S, GUO R H, WEN W P, et al. Research and Implementation on Unknown Trojan Detection System Based on Feature Analysis and Behavior Monitoring [J].Netinfo Security, 2015,(2):57-65.

Research and Implementation on Unknown Trojan Detection System Based on Feature Analysis and Behavior Monitoring

HAO Zeng-shuai[1], GUO Rong-hua[2], WEN Wei-ping[1], MENG Zheng[1]

(1. *School of Software & Microelectronics, Peking University, Beijing* 102600, *China*; 2.*LEETC, Luoyang Henan* 471003, *China*)

Abstract: Trojan is a malicious program that exists mainly to steal user's personal information and file data, and even to control user's computer remotely, while hides itself as far as possible. In recent years, the hacker's behavior has become more professional, interest-oriented, and group-organized, and network intrusion and attacking means have experienced daily changes. Nowadays, Trojan detection depends on the ability of anti-virus software in general, anti-virus software executes Trojan killing usually by using characteristic codes comparison and behavior recognition technology. This way needs anti-virus software to intercept the Trojan samples for analysis, extract the Trojan samples, and identify Trojan after upgrading the Trojan special library. So the hysteresis is very strong, which can't kill the new Trojans and the Trojans without known characteristics. This paper discusses technology against anti-virus, hiding technology and active defense breakthrough technology, puts forward the Trojan detection method based on feature analysis and behavior monitoring, and completes the design and realization of the unknown Trojan detection system. The system covers the shortage that the existing anti-virus software and security measures can only kill and monitor the known Trojans but can't identify and kill the unknown Trojans.

Key words: Trojan detection; Trojan killing; feature analysis; behavior monitoring

收稿日期：2014-12-12

基金项目：国家自然科学基金 [61170282]

作者简介：郝增帅（1976-），男，山东，工程师，硕士，主要研究方向：信息安全测评、软件工程；郭荣华（1972-），男，湖北，副研究员，博士，主要研究方向：信息安全；文伟平（1976-），男，湖南，副教授，博士，主要研究方向：网络攻击与防范、恶意代码研究、信息系统逆向工程和可信计算技术；孟正（1990-），男，河北，硕士研究生，主要研究方向：系统与网络安全、漏洞分析。

通讯作者：文伟平 weipingwen@ss.pku.edu.c

0 引言

互联网在我国政治、经济、文化以及社会生活中发挥着越来越重要的作用，以互联网为核心的网络空间已成为继陆、海、空、天之后的第五大战略空间，而随着互联网的发展，木马、病毒等恶意代码的数量和危害大幅增长。目前国内外针对无已知特征木马（以下简称未知木马）的查杀尚未形成统一标准，由于未知木马往往针对国家重要信息系统专门定制开发，严重威胁涉及国计民生的重要信息系统的安全稳定运行，因此在“震网”病毒之后，越来越多的国家和组织开始重视这一领域，并开展了大量的研究工作[1]。

防病毒软件厂商公开负责未知木马的研究与查杀，主要采取对未知木马的动态检测技术，通过将虚拟机技术运用于如解密、脱壳等对木马样本运行期间的动作给出较为详细的分析，但这些分析往往只是对动态行为的罗列，而没有对行为本身的安全级别进行分类[2]。

目前未知木马检测技术研究成为国内外学术界研究的热点，这些研究不以木马的静态指纹特征码作为判断木马的依据，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。研究木马的隐藏技术、躲避查杀技术及动态检测技术是目前学术界认可的特种木马和未知木马检测技术的研究方向。主流的特种和未知木马检测技术包括对比检测法、系统资源检测法、协议分析法和网络行为分析法[3]。

1 木马免杀技术研究

1.1 文件免杀

文件免杀的最终目标就是在保证原文件功能正常的前提下，通过一定的更改使木马“变”为一个正常的文件，使得原本会被查杀的文件免于被杀。文件免杀基本上就是破坏原有程序的特征，无论是直接修改特征码还是加上一段花指令，或是将其加壳，其最后的目的只有一个，就是打乱或加密可执行文件的内部数据[4]。

1）更改特征码免杀

特征码类似于杀毒软件的黑名单，是从病毒体内不同位置提取出的一系列字节，杀毒软件就是通过对这些字节及位置信息进行比对来检验某个文件是否为病毒或木马。黑客往往使用更改特征码技术进行免杀。更改特征码一般说来包括两种方法。一种是直接修改特征码，这也是黑客做木马免杀最初所采取的方法。第二种是针对校验和查杀技术提出的免杀，校验和是根据病毒文件中特定区域计算出来的，如果一个文件某个特定区域的校验和符合病毒库中的特征，那么反病毒软件就会报警。如果想阻止反病毒软件报警，只要对病毒的特定区域进行一定的更改，就会使这一区域的校验和改变，从而达到欺骗反病毒软件的目的[5]。

2）花指令免杀

花指令其实就是一段毫无意义的指令，也可以称为垃圾指令。花指令的存在对程序的执行结果没有影响，所以它存在的唯一目的就是阻止反汇编程序的运行，或是对反汇编设置障碍。然而这种障碍对于反病毒软件来说同样也是致命的，如果花指令添加得足够高明，就可以使木马逃脱查杀。花指令最根本的思想就是扰乱程序运行顺序，并为破解者（反病毒人员）设下陷阱。

3）加壳免杀

软件加壳也可以称为软件加密（或软件压缩），与通常加密（或压缩）的方式和目的不同，一般的加密是为防止陌生人随意访问数据。加壳可以将一个可执行程序中的各种资源，包括 EXE、DLL 等文件进行压缩，但不会破坏程序，压缩后的可执行文件仍然可以正常运行。加壳之后的文件通常都无法还原。杀毒软件是靠特征码来识别木马，加壳后原有的特征码消失，反病毒软件遇到加壳之后的文件通常难以识别，反病毒软件将会按照一个正常的文件来处置。

4）敏感函数字符串免杀

敏感函数字符串免杀主要采用隐藏导入表的方法进行，无论是简单的异或加密、导入表单项移除，还是稍微复杂的重构导入表、利用 HOOK 方式打乱其调用等，都可以达到隐藏导入表的目的。敏感函数字符串免杀具体有 4 种方法：⑴ 源码动态获取导入表；⑵ 修改 PE 文件；⑶ 构造反汇编代码；⑷ 修改程序入口点[6]。

1.2 内存免杀

内存免杀就是木马对抗杀毒软件内存查杀的技术。其实内存免杀方法与文件查杀相似，都是通过特征码比对进行的。只不过为了强化查杀效果，大多数反病毒公司的内

存扫描与文件扫描采用的不是同一套特征码，这就导致了一个木马同时拥有两套特征码。针对杀毒软件这一特性，黑客们发明了内存、文件特征码修改免杀法，必须要将特征码全部破坏掉才能躲过反病毒软件的查杀，通常先用特征码定位软件定位文件特征码的所在之处，再用编辑器对被定位的特征码进行修改[7]。

1）指令顺序调换法。在程序运行过程中，将内存特定的敏感函数调用字节码、敏感 IP 地址以及字符串等在顺序上做调整，绕过简单的内存扫描查杀。

2）内存可信地址免杀。在程序运行过程中，尽可能不出现或者少出现自动开辟大量执行空间，杀毒软件对可执行的空间开辟非常敏感，这些空间易被直接查出。可以选择零散的可执行空间或者已存在的执行空间作为载体执行恶意代码。

3）可信模块免杀。在程序运行过程中最好将代码放入可信模块中来绕过杀毒软件检测，单独加载独立非可信模块容易被杀毒软件查出。

4）通用跳转法。用跳转的方法把特征码对应的汇编指令跳转走，犹如加入花指令一样。

5）字符串大小写修改法。特征码所对应的内容是字符串时，把字符串的大小写互换。

6）等价替换法。特征码所对应的是汇编指令时，把指令替换成功能类拟的指令[8]。

1.3 逃避主动防御

木马在逃避反病毒软件的主动防御时，一般从两个方向入手，一个是分析反病毒软件的行为，另一个就是逆向反病毒软件的关键代码。已知的突破主动防御技术如下：

1）屏幕截图突破主动防御

通常可执行程序都是有窗口的，但是绝大多数的病毒木马没有与被害用户交互的窗口，因此某些反病毒软件就利用这个特点判断一个程序的合法性。如果某个程序在运行时已经开始执行很多操作，但是却没有窗口，那么这个程序显然就非常可疑。黑客通常先对桌面进行截图，然后将其显示在他们创建的窗口上。由于窗口界面与用户的桌面完全一致，导致用户无法发现窗口的存在。虽然用户可能会感受到键盘和鼠标在几秒钟内无法正常操作，但过后便会很快恢复正常。主动防御也会认为这是正常的，因此无法做出正确判断，从而使得木马程序突破主动防御[9]。

2）利用可信进程突破主动防御

黑客通常会收集一些具有数字签名的程序，如 QQ、阿里旺旺等，然后也会同时加载并运行自己的假 DLL 文件。为了避免假 DLL 文件被发现，大多数黑客会在那个假 DLL 文件执行时将其“ 摘链“ ，这样杀毒软件在遍历进程模块时就难以发现这个假 DLL 文件了。如果假 DLL 文件无法被发现，反病毒软件的各种分析引擎（包括云查杀引擎）也就都无效了。

3）利用系统进程突破主动防御

如果恶意软件启动了一个白名单中的新进程，那么这个进程的父进程显然就是黑客的恶意软件，因此虽然启动的新进程是在白名单内的，但这仍然会被一些防守严密的反病毒软件查杀。黑客为了使自己的恶意软件合法化，他们会模拟用户键盘或鼠标操作，在“ 开始 ” 菜单下用“ 运行 ” 功能去运行那个“ 傀儡 ” 白名单进程，进而达到调用指定恶意 DLL 文件的目的。

4）利用逻辑漏洞突破主动防御

逻辑漏洞一般都要靠实际分析反病毒软件才能得出。逻辑漏洞从根本上来讲与溢出漏洞基本一致，都是由于程序设计人员的大意或偷懒造成的。例如，国内某用户量很大的反病毒产品，其保存进程链表的数据结构是一个指定大小的数组，也就是说如果一个恶意进程试图去递归调用一个可信程序，那么当这种递归调用次数超过反病毒产品所保存的数组长度时，就会将前面的进程信息丢弃掉，这样会导致发起调用的恶意进程信息被淹没掉，反病毒软件也就回溯不到发起调用的恶意进程了。由于这样会导致整个进程链表中的所有进程都是可信的，反病毒软件也就会放行一些高危操作，从而达到免杀的目的[10]。

5）利用变形复制突破主动防御

通常用户在向一些比较敏感的系统目录中写入信息时会被反病毒软件严密监控，因此一些黑客就将一个文件分为很多份，多次在不同的时间段内逐渐写入，甚至一次只写入一个字节，这样就会使大多数反病毒软件失效。

6）利用异同逃逸虚拟机突破主动防御

反病毒软件通常会使用虚拟机技术，即一种可以部分模拟一个真实运行环境的机制。被检查的程序会在虚拟机

中先运行一遍，由于其运行在虚拟机环境中，因此即便这是一个恶意程序，也不会对真实系统产生任何不良影响。运行在虚拟机中的恶意程序的所有行为都会被记录下来，如果一个恶意软件无法逃逸出虚拟机，那么它必将被检测出来。

为逃逸出虚拟机，黑客可以通过执行虚拟机未能模拟的一些操作来判定自己是不是在虚拟机中。如果是，则仅执行一些具有欺骗性的常规操作：如果不是，则执行恶意操作。判断是否在虚拟机中的方法有很多，如打开一个特定的网页查看其返回值，或者读取某个系统目录下的文件等[11]。

7）利用替换文件突破主动防御

每个操作系统总有一些自启动程序不是必需的。因此黑客可以将这些被启动的文件替换掉，以实现自启动的目的。Windows 有一个用于移动文件的 API 函数 MoveFIleEx()，将它的参数 dwFlags 设置为 MOVEFILE_DELAY_UNTIL_REBOOT，就可以实现重启时替换的目的。

8）利用调试接口突破主动防御

部分反病毒软件为了方便自己调试，往往会有一些相关调试接口，因此通过命令行参数可以让这些反病毒软件关闭甚至精确控制其某种行为。这些接口的获得方式往往都是通过分析文档，分析安装包或反病毒软件本身来得到的[12]。

1.4 木马隐藏

木马被种植到计算机后，首先要做的就是躲避各种防病毒软件及恶意代码监控设备，将自已在系统中隐藏起来，以实现其窃密、破坏等功能，因此隐藏是特种木马与未知木马的一个重要特征。木马通常采用文件隐藏、注册表隐藏、进程隐藏、通信隐藏等技术达到隐藏目的[13]。

RootKit 是攻击者用来隐藏自己行踪和保留 root 访问权限的工具。有些木马为了达到隐藏的目的，采用了 RootKit 技术，这类木马通常会利用 Windows 操作系统中各种驱动程序模块和核心模块来隐藏程序文件以及注册表键值，某些特种木马甚至会使用 RootKit 技术来隐藏内存中的进程。

1）用户模式 Rootkit

用户模式的 Rootkit 定义为“能够长时间存在于计算机或者自动化信息系统上的未被发现的处于用户空间（userland）的用户程序和代码集合”。所有用户模式应用程序以用户账户特权级别运行于系统之中，不是操作系统的一部分。反病毒软件对内核的防护越来越严密，直接从用户层躲过反病毒软件的检查并隐藏自己，从而避免碰触内核的做法已经在攻击者中取得越来越多的共识。

2）内核模式 Rootkit

内核模式 Rootkit 就是运行于操作系统中的具有高特权级别（Ring0）的恶意二进制代码。通过用户模式的 Rootkit 可以知道，Rootkit 之所以能够隐藏软件行踪并控制一些系统行为，关键就是可以钩住特定的系统函数，从而达到改变系统函数执行路径的目的，使木马在做关键操作时被我们提供的函数拦截并过滤。因此，Rootkit 技术的本质就是各种钩子的应用。黑客可以使用 Rootkit 技术做出免杀效果非常理想的木马，反病毒工程师也可以利用它建立坚固的安全阵地，以使得所有病毒木马无所遁形。内核中可以使用的钩子类型有十余种之多，由它们衍生出来的各种应用方案更是多达近百种，如 SSDT 钩子、SYSENTER 钩子、内联钩子、IDT 钩子、IRP 钩子等。

1.5 逃避云查杀

黑客为了使得木马能够在系统中生存下来，研究了针对云查杀的逃避或绕过技术。

1）特征码变形

对于已知的木马样本进行特征码变形，破坏它本身的特征值。例如，在编写此类样本时加入编码器和密钥，每次生成的样本使用不同的密钥进行处理，运行时进行解码，这样既躲避了云查杀，也极大增强了样本的对抗性和灵活性。

2）白名单伪造

将样本编写为辅助加载模块写入白名单库的文件中。例如，将恶意样本编写成常见的加载文件如 OCX、DLL 等文件，并绑定在第三方白名单进程上运行，这样云端检测时通常只检测运行的主程序，由于选用的主程序也是合法的白名单，这样也可躲避云查杀[14]。

3）哈希碰撞伪造

通过算法碰撞，使样本文件的 MD5 值和白名单的 MD5 值一致，从而绕过云查杀。

4）阻断云上传

通过制造网络故障，阻断云上传，阻止杀毒软件和服务器的联系和升级，直接绕过云查杀。

2 木马检测系统设计

2.1 检测系统概述

该系统对 Windows 主机中的可疑行为进行监控，关注终端及服务器上所有可疑的程序行为，对系统被攻击后驻留在其中的高级未知木马程序进行深度挖掘、分析和报告，采用静态分析技术检查可疑未知木马程序，并将可疑木马程序提交到未知木马分析系统进行动态分析。

该系统的设计原则主要有以下 4 个方面 ：

1）满足未知木马专项检查要求

当前环境下，国内重要信息系统都强烈要求检查是否已经遭受未知木马攻击，本文未知木马检测系统支持以上需求。可有效满足相关部门或相关安全监管机构未知木马专项检查要求。

2）及时监控未知木马攻击的发生

传统的恶意程序发现系统或者主动防御没有程序分类，只有黑白名单。对于普通的程序，即非白非黑程序而言，将正常程序和未知程序放在同一个角度去分析容易产生误判。当正常的程序发生溢出，这个程序通常是使用面非常广的已知程序，甚至是白名单程序，如 Word 等。由于程序具备数字签名，因此会被传统的主动防御放过。未知木马检测分析系统根据木马的各种特征，从系统中检测出可疑样本，并提交分析系统进行动态分析，看其行为特征是否为已知或未知木马 [15]。

3）有效发现已发生的未知木马攻击

常规边界及网络恶意代码防护设备无法发现已经发生的未知木马攻击，本文系统的特殊架构可以有效发现已发生的未知木马攻击，并可将已发现的未知木马攻击模式转换成检测策略进行更有针对性的查找和清除。入侵者为植入木马并最终获得执行权，可以定期地针对系统中可以获得执行权的点（如启动目录、服务、驱动、系统程序可外接的动态库、打开关联等配置信息和对应的文件）进行检测，以发现客户端被入侵的痕迹。如果发现可疑痕迹，将对应的文件提交分析中心进行进一步的检测与分析。

4）边界恶意代码检测设备的必要补充

边界恶意代码检测设备只能检测通过网络边界进行恶意代码攻击的行为，而实际上存在针对特定目标的各种可能的攻击形式，并且具有长期性，边界恶意代码检测设备对发生的各种未知木马攻击无能为力，并且边界恶意代码检测设备对通过加密通道进来的各种木马程序也无法还原并检测。未知木马检测分析可以对各种形式的木马攻击进行检测，是边界恶意代码检测设备的必要补充。

2.2 网络部署架构

系统支持灵活的部署方式，可以根据需要在机构总部部署一套或多套管理控制中心、分析中心以及客户端检测代理和 U 盘专用检测工具，在下属单位根据需要部署一套或多套管理控制中心、分析中心以及客户端检测代理和 U 盘专用工具。机构总部的管理控制中心和分析中心设置为顶层管理控制中心和顶层分析中心，下属单位的管理控制中心和分析中心设置为下一级管理控制中心和分析中心。系统总体部署示意图如图 1 所示。

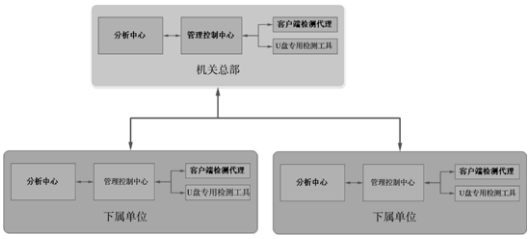

图1 网络部署示意图

2.3 功能模块设计

平台分为 4 个部分 ：客户端检测代理、U 盘专用检测工具、管理控制中心、分析中心。

1）客户端检测代理

客户端检测代理驻留在客户端系统，根据配置信息定期或不定期检测系统中的可疑程序，将可疑度分值大于设定数值的程序样本上传到管理控制中心，管理控制中心将样本提交给分析中心进行动态分析 [16]。

客户端检测代理结构如图 2 所示。

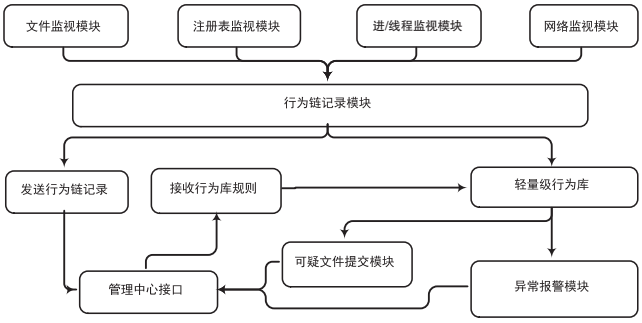

图2 客户端检测代理结构

图 2 中的部分模块及功能描述如表 1 所示。

表1 客户端模块及模块功能

编号	模块名称	模块功能
1	文件监视模块	主要对文件的创建、修改做监视并记录
2	注册表监视模块	主要对注册表的关键位置进行监视并记录
3	进/线程监视模块	主要对进/线程的创建、注入等情况监视并记录
4	网络监视模块	主要对程序请求网络进行监视并记录
5	行为链记录模块	记录从事件发生到结束的整个行为链过程
6	异常报警模块	触发报警
7	可疑文件提交模块	进行初步过滤筛选，决定文件是否需要提交到管理控制中心

2) U 盘专用检测工具

在不能安装代理的客户端的情形下，或者在怀疑系统已经被植入未知木马的情况下，客户端环境已经不可信，需要在干净的系统中进行检测。需要 U 盘启动 WinPE 系统，在干净环境下进行检测，并可以与正常启动情况进行对比分析。使用 U 盘专用检测工具检测后，保存可疑样本，并提交给管理控制中心，管理控制中心将样本提交给分析中心进行动态分析。

U 盘专用检测工具结构如图 3 所示。

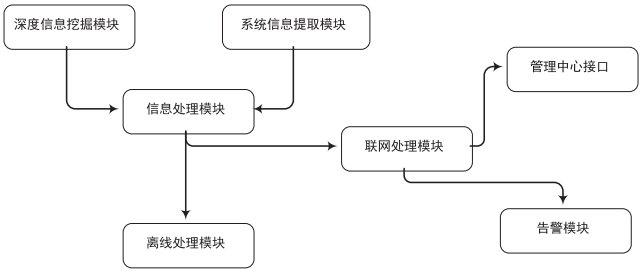

图3 U盘专用检测工具结构

3) 管理控制中心

管理控制中心可以配置、指定相应的分析中心，根据实际环境需要，可以多个管理控制中心对应一个分析中心。管理控制中心支持多级部署，上级管理中心节点可以向下级管理中心节点发布、更新、配置各项管理操作，下级管理中心节点可以向上级管理中心节点提交报警信息。客户端检测代理和 U 盘专用检测工具在管理控制中心的管理下工作。客户端检测代理和 U 盘专用检测工具上传的可疑程序样本在本地存储并做预处理后提交给分析中心进行分析处理 [17]。

管理控制中心模块结构如图 4 所示。

4) 分析中心

分析中心接收管理控制中心提交的样本，并对样本进行动态分析，根据预定义的规则给出分析结论，展示详细

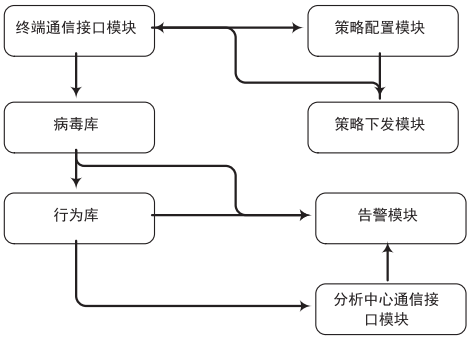

图4 管理控制中心模块结构

分析结果。分析中心分为交互界面与分析引擎，交互界面提供样本接收接口、报告接口与人机交互接口。样本接收接口自动接收管理控制中心提交的样本，并将样本加载到分析引擎进行分析；报告接口根据样本哈希值查询是否已进行过分析，以及根据样本哈希值获取分析结论；人机交互接口提供样本分析结论、样本分析详细结果、样本特征提取、分析中心配置、样本分析结果同步等人机交互界面。

分析中心支持多级部署，下级节点可以向上级节点提交样本及分析结果。上级节点可以向下级节点推送样本信息及分析结论信息，以使所有分析中心节点同步获得样本分析结论信息。同级节点之间根据策略配置为完全同步。分析中心可根据配置策略，提取确定的攻击样本特征，并提交给管理控制中心，管理控制中心根据配置信息将样本特征下发给下级管理控制中心和客户端检测代理，并通过样本特征更新 U 盘专用检测工具策略。

分析中心模块结构如图 5 所示。

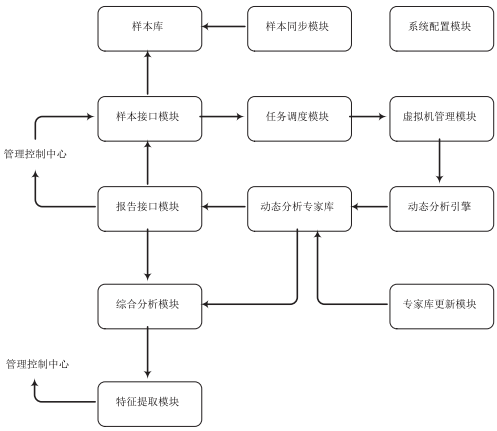

图5 分析中心模块结构

3 木马检测系统实现

3.1 检测模块实现

本系统对主机端和网络中的特种木马和未知木马实施

监控，主要分为两个部分 ：在主机端完成对可疑样本的静态和动态处理分析 ；在网络端通过对网络中数据包的捕获重组分析，定位特种木马和未知木马的通信数据包。主机端的未知木马检测框架如图 6 所示。

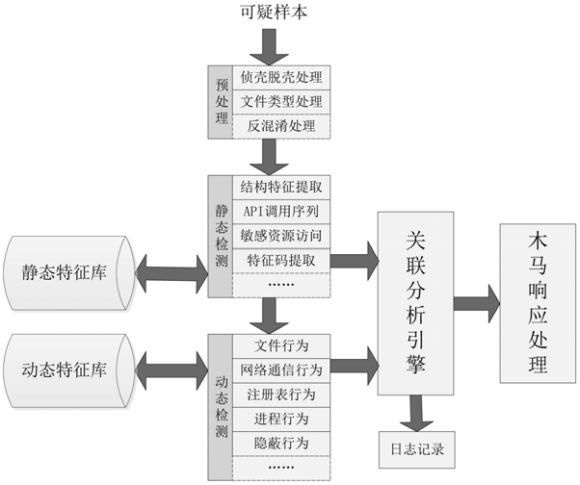

图6 主机端的未知木马检测框架

主机端的未知木马检测框架主要分为预处理、静态检测、动态检测 3 个部分，通过它们之间的协同关联分析，检测可疑样本是否为特种木马或未知木马。

网络端的未知木马检测框架如图 7 所示，主要分为特征分析和行为分析两个部分，特征分析用于对协议、单向数据流、偶发性数据流等进行匹配分析，行为分析能够通过对网络会话 3 个阶段的独立分析以及数据挖掘方法确定未知木马的通信数据包。

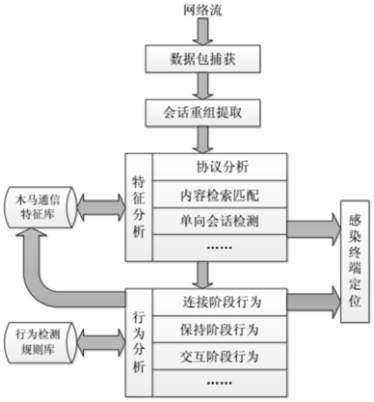

图7 网络端的未知木马检测框架

1) 侦壳脱壳处理模块

软件加壳会影响对文件的正常分析，木马等恶意软件会通过加壳、加花指令的方式保护自己。若要实现对系统中软件的分析，必须先将其脱壳。侦壳脱壳模块首先根据文件架构判断该文件是否被加壳，若已被加壳，则调用脱壳模块将壳脱掉后继续分析。侦壳处理模块流程如图 8 所示 [18]。

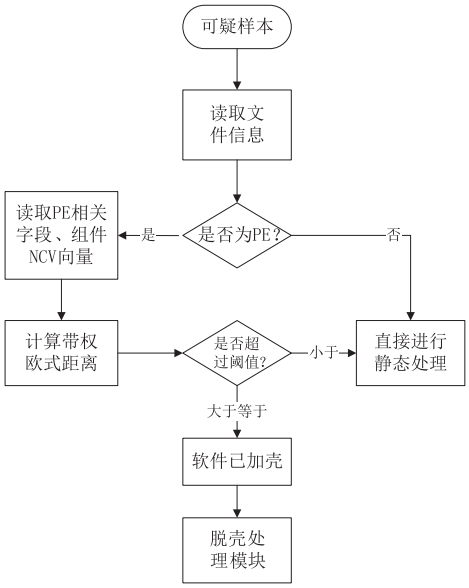

图8 侦壳处理模块

位于系统主机端的预处理模块通过对可疑样本结构特征的提取，构建此样本的特征向量，然后通过带权欧式距离算法判断文件是否带壳。侦壳脱壳处理模块又分为 PE 文件判定模块、向量模块和算法判定模块 3 个小模块。

2) 文件静态特征检测模块

木马等恶意软件的头部结构特征、申请使用的函数和函数的调用序列与正常软件有很大不同，文件静态特征检测模块依据获取的可疑样本的头部特征和反汇编获得的函数调用序列，判断样本是否为木马文件。文件静态特征检测模块流程图如图 9 所示。

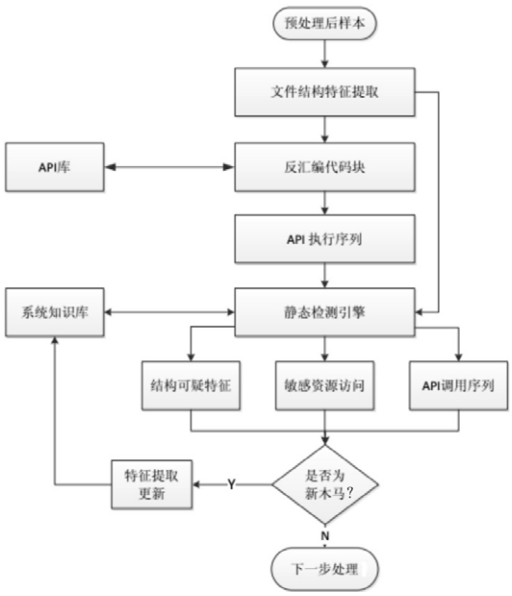

图9 文件静态特征检测模块

文件静态特征检测模块主要分为文件结构特征提取、反汇编获取 API 调用序列、静态检测引擎 3 个主要部分。

3) 动态分析与检测模块

动态分析与检测模块利用仿真技术和沙盒技术实现通过程序模拟交互活动触发木马恶意行为的精确检测，在动态分析特种木马和未知木马的同时，保证终端系统的完整性和无破坏性。动态分析与检测模块主要分为两个部分：用户层模块和驱动模块，如图 10 所示。

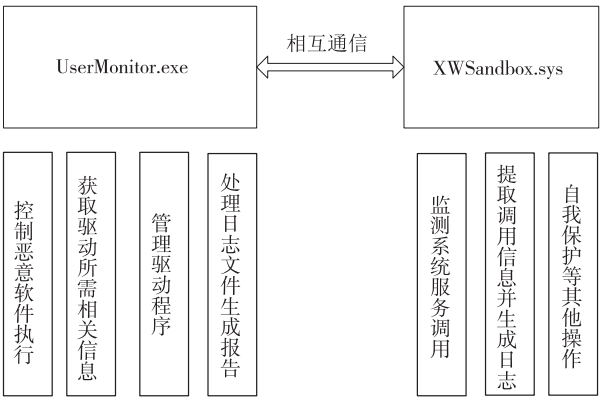

图10 动态分析与检测模块

(1) 用户层模块。用户层模块主要用于完成运行在用户态的功能，如控制恶意软件的运行、管理驱动程序、获取系统服务信息、生成软件运行日志等。用户层模块要监控与未知木马行为相关的所有进程，因此该模块会将获取的软件进程的相关信息传递给驱动模块。为了保证进程的所有活动不被遗漏，系统首先以挂起模式创建木马进程，再获取进程信息且传递给驱动模块，然后重新启动被挂起的进程，同时用户层模块还将获取的 SSDT 索引信息传递给驱动模块，以便驱动模块依据 SSDT 索引信息完成所有的 Hook 操作。

(2) 驱动模块。当驱动模块被用户层模块启动后，驱动模块首先进行 SSDT Hook 操作，然后对软件行为进行检测。驱动模块会对被它检测到的进程做判断，只有当进程为可疑软件进程时才做记录。同时驱动模块还实现了自我保护功能，主要是隐藏自身和监控 SSDT，使得可疑木马软件不会发现自己运行在虚拟隔离的环境中。驱动模块流程图如图 11 所示。

4) 隐藏密码模块

隐藏密码模块也属于木马的动态检测，用于弥补沙盒环境下对采用高级隐藏技术的木马检测率不高的不足。主要采用交叉检测技术、关联检测技术以及系统完整性分析技术来有效检测特种木马与未知木马采用的隐藏技术，如文件隐藏技术、进程隐藏技术、网络连接隐藏技术和内核模块隐藏技术。 隐藏密码模块流程图如图 12 所示。

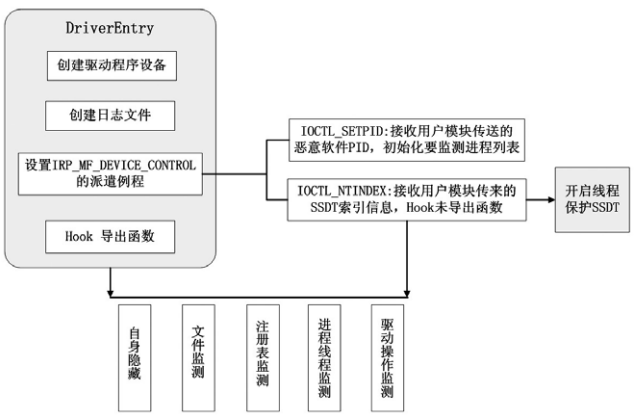

图11 驱动模块

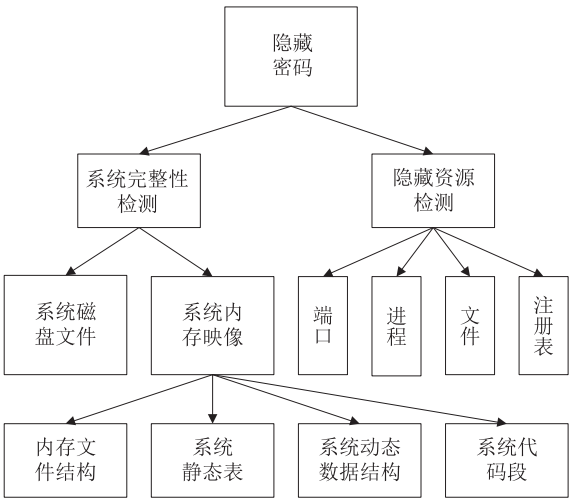

图12 隐藏密码模块

隐藏密码模块主要包括两个核心部分：系统完整性检测和隐藏资源检测，这两个模块协调工作、相互补充，共同构成了核心的隐藏密码模块。

5) 网络流检测模块

网络流检测模块对通过网络出口的数据包进行抓包分析，提取其通信特征，采用特征匹配和行为匹配的方式，检测出未知木马的通信数据包。对于检测出的未知木马或特种木马，网络流检测模块能够自动提取其网络会话特征并更新到网络特征库。网络流检测模块流程图如图 13 所示。

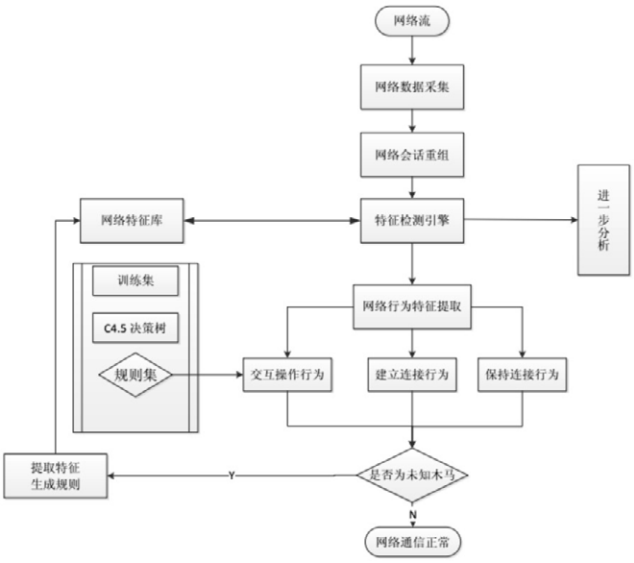

图13 网络流检测模块

3.2 测试结果

本文的木马检测系统与传统防病毒软件的比较如表 2 所示。

表2 测试结果

序号	对比项	本文木马检测系统	传统防病毒软件（瑞星、江民、金山、趋势、卡巴斯基）
1	未知病毒检测	对杀毒软件无法查杀的木马样本运行后进行检测，检测率为 100%	不能检测特征库以外的未知木马
2	病毒库依赖度	低，不依赖特征码升级就可以查杀病毒	高，必须依靠特征码不断升级进行查杀
3	判断、查杀病毒依据	根据程序运行时的行为特征进行判定	根据病毒库内已知的病毒特征码进行判定
4	升级依赖度	无需频繁升级即可全方位保护系统安全	需要频繁升级病毒库，严重消耗用户网络和本机系统资源
5	客户端资源占用	运行时CPU占用 1%~2%，内存占用 10M 左右，杀毒时没有明显变化	未扫描时 CPU 占用 5%，内存占用 15M 左右，扫描时 CPU 占用 60%~100%，内存占用 90M 左右，波动明显
6	客户端关机扫描	客户端数据流量异常时可进行关机扫描	不支持
7	远控地址定位	自带IP地址识别，可报出当前与客户端实际连接IP地址，做到攻击源地址全球定位	不支持
8	未知病毒行为特征分发	对未知病毒的行为特征自动提取和分发，实现未知病毒一点截获，全网分发防护，无须上报等待厂家处理，可做到第一时间保护网络安全	发现病毒库以外未知病毒，必须上报厂家确定，并且频繁下载升级病毒库后才可查杀，失去第一时间保护，过度依赖厂家远程解决方案，网络安全度差
9	软件兼容性	可与传统防病毒软件并存	多种防病毒软件不能并存，冲突严重

4 结束语

本文以特种木马与未知木马的隐藏技术为基础研究了特种木马与未知木马的静态、动态、网络异常等检测技术，研发了一套特种木马与未知木马的主动识别、主动监测及主动查杀系统。该系统能够较为准确地识别未知木马，弥补了目前主流的基于特征码的杀毒软件只能识别已知木马，不能检测未知木马的不足。

本文尚存在以下问题需要研究完善 ：

1）本文提出的静态可疑样本筛查技术基于木马需要在操作系统中实现自启动并采用了隐藏技术、逃避查杀技术，对单次运行（下次开机重启不随系统启动）的木马、正规微软签名程序中集成的木马静态筛查能力较弱，要想对这些木马取得好的检测效果，仍需要进一步的研究。

2）本文提出的未知木马动态分析技术基于木马需要调用操作系统敏感 API，对功能单一的木马检测能力较弱。这需要不断的研究和实验，当发现有新的木马样本时要及时分析获取其行为特征，不断完善木马行为规则库，加强动态检测内容，使得检测系统保持较高的检测准确率[19-22]。（责编 马珂）

参考文献:
[1] Michael A.Davis，Hacking Exposed:Malware & Rootkits Secrets & Solutions[M].NewYork ：McGraw - Hill Osborne Media, 2009.
[2] 王鼎 . 高隐藏性木马的深度检测技术实现研究 [D]. 成都 ：电子科技大学，2010.
[3] 王蕊，冯登国，杨轶 , 等 . 基于语义的恶意代码行为特征提取及检测方法 [J]. 软件学报 ,2012,(2):378 - 393.
[4] 全宇 . 特洛伊木马的表现及防范 [J]. 湛江师范学院学报，2010,31（3）:18 - 26.
[5] 李云亚 . 特种木马的分析与识别 [J]. 江苏科技信息 ,2010,(2):30 - 32.
[6] 王鹏 .Windows PE 文件保护技术研究与实现 [D]. 成都 ：四川师范大学，2009.
[7] 刘 喆 , 张家旺 . Rootkit 木马隐藏技术分析与检测技术综述 [J]. 信息安全与通信保密，2010,(11):61 - 65.
[8] 什么是特洛伊木马及其 6 个特性 [J]. 计算机与网络，2013,39(7):43.
[9] 孙海涛 . 基于通信行为分析的木马检测技术研究 [D]，郑州 ：中国人民解放军信息工程大学，2011.
[10] 李伟 . 基于内核驱动的恶意代码动态检测技术 [J]. 中国科学院研究生院学报，2010. 27(5):695 - 704.
[11] 张新宇，卿斯汉，张恒太，等 . 特洛伊木马隐藏技术研究 [J]. 通信学报，2004，（7）：153 - 159.
[12] 刘牧星 . 木马攻击与隐蔽技术研究 [D]. 天津 : 天津大学，2006.
[13] 张健 . 恶意代码危害性评估标准和检测技术 [D]. 天津 : 南开大学，2009.
[14] 王东 .VMware 虚拟机检测技术研究 [J], 计算机光盘软件与应用，2011，（10）:73.
[15] 康治平，向宏，傅鹂 . 基于 API HOOK 技术的特洛伊木马攻防研究 [J]. 信息安全与通信保密，2007，（2）:145 - 148.
[16] 胡燕京，张冰，王海义，等 . 主流木马技术分析及攻防研究 [J]. 现代电子技术，2007，（13）:96 - 100.
[17] 李阳 . 恶意代码检测及其行为分析 [D]. 西安 ：西安电子科技大学 .2010.
[18] 胡明科 . 未知木马检测技术研究 [D]. 沈阳 : 沈阳航空航天大学，2011.
[19] 赵天福，周丹平，王康，等 . 一种基于网络行为分析的反弹式木马检测方法 [J]. 信息网络安全，2011，（9）:80 - 83.
[20] 彭国军，王泰格，邵玉如，等 . 基于网络流量特征的未知木马检测技术及其实现 [J]. 信息网络安全，2012，（10）:5 - 9.
[21] 杨卫军，张舒，胡光俊 . 基于攻击树模型的木马检测方法 [J]. 信息网络安全，2011，（9）:170 - 172.
[22] 刘昊辰，罗森林 .Android 系统木马隐藏及检测技术 [J]. 信息网络安全，2013，（1）:33 - 37.