第 38 卷 第 2 期
Vol.38 No.2

计 算 机 工 程
Computer Engineering

2012 年 1 月
January 2012

• 安全技术 •

文章编号：1000—3428(2012)02—0153—03 文献标识码：A 中图分类号：TP309.2

基于动静特征加权的木马检测系统

钟明全，李焕洲，唐彰国，张 健

(四川师范大学网络与通信技术研究所，成都 610066)

摘 要：传统木马检测方法的漏报率较高。为此，结合木马的动态特征与静态特征，设计并实现一个基于动静特征加权的木马检测系统。研究木马工作机制，建立自定义的木马特征库，介绍木马检测思路和系统工作逻辑，分析木马特征的提取过程，并给出权值分配方法。实验结果表明，该系统的检测准确率较高。

关键词：木马特征；动态检测；静态检测；加权算法

Trojan Detection System Based on Weighting of Dynamic and Static Characteristics

ZHONG Ming-quan, LI Huan-zhou, TANG Zhang-guo, ZHANG Jian

(Institute of Network and Communication Technology, Sichuan Normal University, Chengdu 610066, China)

【Abstract】In allusion to the shortage of high unreported rate of current detection method for Trojan, using dynamic and static characteristics of Trojan, Trojan detection system based on weighting of dynamic and static characteristics is designed and realized. By in-depth research of work mechanism of Trojan, custom characteristic library for Trojan is built. Detection idea for Trojan and work logic of detection system is introduced, pick-up procedure of Trojan characteristic is analyzed, and distribution method of weight for Trojan characteristic is given. Experimental result proves that the Trojan detection system has high accurate rate.

【Key words】Trojan characteristic; dynamic detection; static detection; weighting algorithm

DOI: 10.3969/j.issn.1000-3428.2012.02.050

1 概述

特洛伊木马(简称木马)是以盗取用户个人信息、甚至远程控制用户计算机为主要目的的恶意代码，具备破坏或删除文件、发送密码、记录键盘和远程攻击等功能。据《中国互联网网络安全报告(2010 年上半年)》报道，2010 年上半年监测到境内被木马控制的主机 IP 数量近 124 万个，境外有 12.8 万个主机 IP 参与控制上述境内主机，与 2009 年相比，木马境内受控主机数量增加了近 4 倍。木马程序在数量上逐年递增、数值巨大、造成的影响和危害很大，仍然是计算机网络安全的最大威胁，因此，反木马将是一项长期而艰巨的任务。目前木马的检测方法归纳起来可以分为 2 类：基于文件静态特征的检测方法和基于文件行为特征的检测方法。两者单独使用都存在各自的不足，为此，本文提出一种将动态特征与静态特征相结合的检测系统，利用加权算法实现对各种木马程序的检测。

2 木马检测的基本原理

2.1 木马特性

木马的生存周期包括传播阶段、植入阶段、运行阶段和网络通信阶段。在其生存周期中，尽管木马程序的编程语言不同，运行的环境不同，实现的功能也不同，但它们在宏观上表现出以下一些共同的动态或静态特性[1-2]：

(1)隐蔽性。隐蔽性是木马的首要特征，木马程序为了不被发现，通常会采用各种隐藏自己的技术，如文件隐藏、进程隐藏、通信隐藏等。

(2)可执行性。每个木马程序都为一定的目的而设计，木马程序只有运行起来才能实现其目的，因此，木马程序都是可执行的，并且部分木马程序还被设置为开机自启动运行。木马程序运行后，还必须通过网络通信与控制端取得联系，才能实现其最终目的。

(3)功能特殊性。木马程序的功能与正常程序差异很大，呈现出特殊性，木马程序常见的一些功能包括搜索计算机口令、记录键盘操作、扫描 IP 地址、远程注册表操作、远程抓屏、锁定键盘和鼠标、删除系统文件、破坏系统等。

(4)反检测性。现有的木马程序多采用模块化设计，每个模块都有自己的功能分工，有的模块实现安装，有的模块负责监视木马的运行，一旦发现木马程序被强制退出或被强制删除，立即从备份文件处进行恢复。有的木马甚至运行起来之前，先关闭各种杀毒软件或安全工具。

2.2 木马检测方法

基于木马程序与正常程序的差异性，目前有多种方法可以检测木马。其中，杀毒软件通常采用基于特征码的检测方法，特征码是一个特殊的二进制标志字，相当于木马程序的指纹，只要木马程序稍作改动，特征码就会改变。因此，这种方法的缺点是无法检测变形木马和未知木马。文献[3]指出，基于 PE 文件静态信息，并运用决策树、神经网络等智能信息处理技术，构造基于文件静态信息的木马检测模型，可具有较高的检测效率。但该方法要求样本类别齐全、信息

基金项目：四川省教育厅基金资助项目(08ZA043)

作者简介：钟明全(1975－)，男，讲师、硕士，主研方向：网络与信息安全，网络监控；李焕洲，副教授、博士；唐彰国，讲师、硕士；张 健，讲师、博士研究生

收稿日期：2011-07-04 **E-mail：**mqzhong@sina.com

采集量足够高。文献[4]根据木马在运行时所表现的行为特征，提出了一种基于行为监控的木马检测机制，对已知木马和未知木马，检测结果都较为准确，但该方法存在不能检测潜伏木马的缺点。文献[5]指出，基于对大量病毒程序的功能、行为分析，从中寻找病毒程序的特有行为特征并收集起来形成行为特征集，利用熵值的原理对各个特征进行处理，通过计算其信息增量来判断其对区分病毒的贡献程度。该方法的特征主要来自于程序调用的 API 函数序列，对于一些大型程序来说工作量大、效率低。

针对上述方法的不足，本文提出一种将动态特征与静态特征相结合的检测系统，利用加权算法实现对各种木马程序的检测。

3 基于动静加权的木马检测系统

3.1 系统架构

本文建立一个基于动静加权的木马检测系统检测模型，该系统借鉴了动态检测法与静态检测法的优点，通过采集被检测程序的行为信息与静态信息，提取出文件的动态特征与静态特征，将 2 种特征信息按加权的思想相结合来判定被检测对象的危险等级。系统架构如图 1 所示。

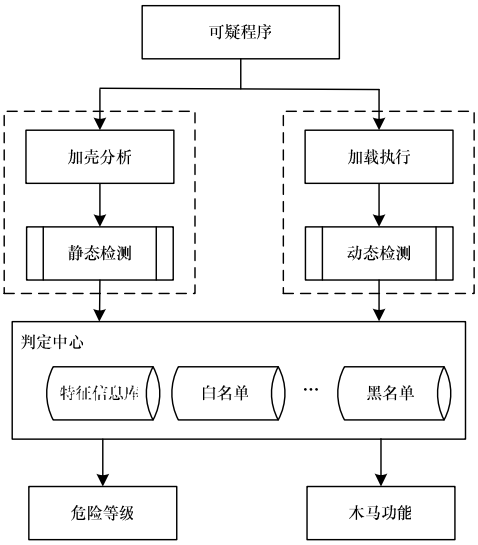

图 1 基于动静特征加权的木马检测系统架构

检测系统的工作流程总体上分为以下 2 个方面：

(1)从静态角度实施检测，首先判断被检测程序是否加壳，然后对文件静态信息进行采集，并将采集到的静态数据送往判定中心。

(2)从动态角度实施检测，先将被检测程序加载到内存运行起来，然后从动态行为的多个方面进行实时数据采集，将采集到的动态数据也送往判定中心。判定中心是信息处理中心，包括特征信息库、黑名单、白名单等。在判定中心，基于特征信息库，利用自定义的木马判定算法对可疑程序进行判定，得出被检测对象的危险等级，同木马功能分析等信息一起输出到检测报告中。

3.2 系统逻辑

系统采用模块化思想进行设计，将检测分为动态检测与静态检测两大模块，并将动态检测细分成 4 个子模块，从而形成以控制中心为核心、动静检测相结合、子模块与控制中心通过共享内存实现进程间通信的工作模式。系统逻辑如图 2 所示。其中，空心箭头表示 Windows 消息。

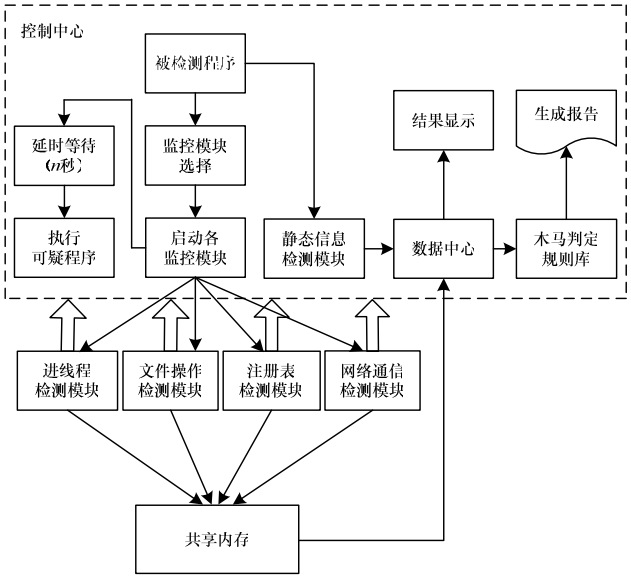

图 2 基于动静特征加权的木马检测系统逻辑

静态检测对执行之前的被检测程序实施检测。由于木马程序具有特定的设计目的，执行之前在文件信息方面已存在与正常程序的差异，这是静态检测的基础。静态检测从数字签名、加壳信息、PE 结构、文件属性等方面进行检测，因此，该方法能够检测潜伏木马、反虚拟机木马等多种木马程序。

动态检测首先运行被检测程序，然后对正在执行的被检测程序产生的行为进行实时采集。由于木马程序具有一定的针对性和目的性，通过采集其运行过程中的行为数据，并与正常程序的行为进行对比，即可判断是否为木马。动态检测采集的行为信息包括进程增量信息、文件增量信息、注册表增量信息和网络通信增量信息，该检测方法能够检测变形木马、未知木马等。

3.3 特征信息提取方法

木马的特征信息是指它与正常程序的差异信息，能够用来区分木马程序和正常程序，分为静态特征信息和动态特征信息。在检测模型中，为达到良好的检测效果，既考虑了常见木马的特性，又考虑了针对某硬件的各种木马(如 BIOS、扇区、硬盘、网卡等木马)的特性，以此获得足够多的特征信息。在这种条件下，将静态特征信息与动态特征信息结合起来判定木马，是一种较好的检测方案。

本文实验用到的静态和动态特征信息共有 40 条左右，全部来源于对木马样本的分析，并进行验证。限于篇幅，木马的一些主要特征信息可参见文献[4,6-7]，本文重点阐述特征信息的提取方法。

根据特征信息出现的概率及危险性，其等级分为高、中 2 个级别，这 2 个级别的特征信息都能用于判定可疑程序。本文将以木马样本的部分静态数据为例，讨论其静态特征信息的提取方法，木马的动态特征信息提取方法与此类似。

静态特征信息提取方法如下：

(1)首先准备 2 批样本，一批为木马程序样本，另一批为正常程序样本；然后分别获取样本的静态数据，对样本的静态数据进行记录与统计。

(2)对比 2 类样本静态数据取值的概率分布，只要木马样本相对概率大于预设值的信息就可以作为特征信息。

(3)设定一个相对概率分界点，将特征信息分为高、中 2 个等级。

3.4　特征信息提取记录

本文实验中的所有预设值最初都是假定的，在实验的过程中要根据实验结果做相应的动态修正。样本程序的静态数据较多，包括数字签名、加壳信息、PE 结构、文件属性等，其中，PE 结构是一个庞大的结构体，其成员也是结构体，因此，静态特征提取的工作量比动态特征大。

为说明静态特征的提取过程，以数字签名、加壳信息、文件版本、公司名称 4 个静态数据为例，分析它们能否作为判定木马的特征信息，具体记录见表 1 和表 2。

表 1　木马样本的部分静态数据记录

数据	取值	个数
数字签名	有	0
	无	50
加壳信息	有	21
	无	29
文件版本	0.0.0.0	33
	非 0.0.0.0	17
公司名称	有	12
	无	38

表 2　正常样本的部分静态数据记录

数据	取值	个数
数字签名	有	28
	无	22
加壳信息	有	6
	无	44
文件版本	0.0.0.0	9
	非 0.0.0.0	41
公司名称	有	26
	无	24

对比分析表 1、表 2 中的数据可发现：

(1)木马样本有数字签名的概率为 0%，正常样本有数字签名的概率为 56%，相对概率大小为 0，不能将“有数字签名”作为木马的特征信息，但可以将数字签名作为正常程序的特征信息，用于间接区分木马。

(2)木马样本加壳的概率为 42%，正常样本加壳的概率为 12%，相对概率大小为 3.5。如果预先定义相对概率大小在 3~8 之间的信息作为中危险等级的特征信息，相对概率小于 3 的信息不能作为特征信息，相对概率大于 8 的信息作为高危险等级的特征信息，则该信息可作为木马中危险等级的特征信息。

(3)“文件版本为 0.0.0.0”可以作为木马中危险等级的特征信息，公司名称不能作为木马的特征信息。

3.5　权值分配与判定

木马判定算法选用基于木马特征信息的加权算法，需要确定每条特征信息的权值系数。权值系数赋值原则如下：

(1)权值系数分为 2 个等级，高危险等级特征信息的权值系数大于中危险等级特征信息的权值系数。

(2)相同危险等级的特征信息其相对概率越大，权值系数也越大。

对于上文提取到的 40 条特征信息，每条都要赋予其权值系数。预先定义中危险等级特征信息的权值系数取值范围在 20~60 之间，高危险等级特征信息的权值系数取值范围在 61~100 之间。部分静态特征信息和动态特征信息的权值系数预设值如表 3 所示，其他特征信息的权值系数可参考表 3 进行赋值。

表 3　特征信息的权值系数

特征信息	类别	等级	权值系数
文件加壳	静态	中	45
文件版本为 0	静态	中	35
进程隐藏	动态	高	80
文件自删除	动态	高	90
设置自启动	动态	中	40
对外通信	动态	中	55

假定特征信息库包含 n 条特征信息，第 i 条特征信息的权值系数为 a_i，则被检测程序的木马疑似度等级(综合等级)定量描述为：

$$P_{\text{Trojan}}=\frac{\sum_{i=1}^{n}m_i a_i}{n}$$

其中，m_i 为检测过程中第 i 条特征信息出现的次数，没有出现时 m_i 为 0，出现多次时表示权值累加。

实验使用正常样本和木马样本对该计算式进行测试训练，以找出正常程序与木马程序 P_{Trojan} 的分界点。如果对 P_{Trojan} 选取 2 个分界点，则综合等级为高、中、低 3 个级别。

4　实验结果与分析

对于木马程序的检测环境首先要求未受木马病毒的感染，其次是能够还原到检测之前的干净环境，因此，常用的检测环境有虚拟机、沙箱、影子系统等。其中，虚拟机由于具有良好的稳定性、快速的还原性而作为典型的测试环境使用。本文系统选用 VMware Workstation 6.0 作为测试环境。

在 Windows XP 平台下基于动静加权的木马检测模型实现了一个木马检测功能的原型系统，该原型系统既可以检测动态特征，又可以同时检测动态特征与静态特征。通过设定原型系统在不同的工作模式下，经反复测试，平均一次检测所需的时间为 2 min 左右。2 种工作模式下检测系统的误报率对比如表 4 所示。

表 4　2 种模式下的误报率比较

工作模式	检测结果个数			误报率/(%)
	高	中	低	
动态检测	3	4	43	14
动静特征加权	2	2	46	8

在前面的误报率对比测试中，基于动静加权的检测方案比基于动态特征的检测方案误报率更低，原因在于前者采用了数字签名等静态检测技术，使具有部分木马动态特征的正常程序危险等级降低，从而使误报率减小。2 种工作模式下检测系统的漏报率对比如表 5 所示。

表 5　2 种模式下的漏报率比较

工作模式	检测结果个数			漏报率/(%)
	高	中	低	
动态检测	31	10	9	18
动静特征加权	38	8	4	8

在表 5 中，基于动静加权的检测方案与基于动态特征的检测方案相比较，前者的漏报率大幅下降，原因是基于动静加权的检测方案中，一方面充分检测了木马运行过程中的动态行为信息，另一方面对木马程序运行之前的文件进行了较深入的静态分析，即使潜伏木马也很难逃避检测。

5　结束语

对于木马的检测方法，早期采用较多的是文件静态分析

(下转第 162 页)