一种未知病毒智能检测系统的研究与实现

张波云[1,2]， 殷建平[1]， 唐文胜[1]
(1. 国防科技大学 计算机学院，湖南 长沙 410073；2. 湖南公安高等专科学校 计算机系，湖南 长沙 410138)

摘 要：设计了一种用于检测未知计算机病毒的查毒系统，其检测引擎基于模糊模式识别的算法实现，检测过程中选用的特征向量是被测试程序所引用的 API 函数调用序列。该系统既可以实现对已知病毒的查杀，又可以对可疑程序行为进行分析评判，最终实现对未知病毒的识别。最后，收集了 423 个 Windows PE 格式的正常程序和 209 个病毒程序组成样本空间进行实验以测试系统的性能。

关键词：病毒检测； API 函数； 模式识别； 模糊集
中图法分类号：TP309.05 **文献标识码**：A **文章编号**：1000-7024 (2006) 11-1936-03

Study and implementation intelligent detection system to recognize unknown computer virus

ZHANG Bo-yun[1,2]， YIN Jian-ping[1]， TANG Wen-sheng[1]
(1. School of Computer Science, National University of Defense Technology, Changsha 410073, China;
2. Department of Computer Science, College of Hunan Public Security, Changsha 410138, China)

Abstract: An intelligent detection system to recognize unknown computer virus is presented. Using the method based on fuzzy pattern recognition algorithm, an unknown computer virus detection model is designed. Characteristic vectors used during detecting are call sequence of API functions. Known and unknown computer virus are deteded by analyzing their behavior. 423 benign programs and 209 malicious programs are gathered as dataset for experiment.

Key words: virus detection; API function; pattern recognition; fuzzy set

0 引 言

当前的计算机病毒检测技术主要基于特征检测法，其基本方法是提取已知病毒样本的特征，并将此特征数据添加到病毒特征库中，在病毒检测时通过搜索病毒特征库查找是否存在相匹配的病毒特征来发现病毒。这种检测办法只能用于检测已知的病毒，对于新出现的病毒的检测无能为力[1]。为了解决这一问题，各反病毒研究机构都在努力探讨病毒检测的智能方法。在 IBM 病毒研究中心，曾经成功地将神经网络用于判断引导型病毒[2]，他们声称在不久的将来把人工免疫方法应用于病毒检测产品中去。M. Schultz 等人[3]曾提出用数据挖掘的算法如朴素贝叶斯算法等检测未知恶意代码，其算法建立在对程序静态分析的基础上，他们直接选用程序的机器码、程序中的 ASCII 字符串以及对 PE 程序的静态分析而取得的 API 引用序列做为样本程序的特征向量，各算法以此为基础进行学习与分类。显然，该法对目前日渐盛行的变形病毒的检测无能为力，因为变形病毒的机器代码序列变化十分迅速[1]。

基于上述思想的启发，本文提出一种基于模糊模式识别分类算法[4]的检测方法来实现对计算机病毒的近似判别。采用该算法对提取的可疑文件行为特征进行分析，并利用病毒程序与正常程序的行为特征的差异性进行分类，从而达到检测未知病毒的目的。本文的分类算法主要针对程序的执行过程中使用的系统 API 函数的调用，它监视程序的行为并进行分析，能有效地检测未知病毒和各种多态与变形病毒。

1 模型结构

系统结构框图如图 1 所示。系统由病毒防火墙、应用服务器和病毒检测服务器组成，进入系统的可疑文件首先经过系统的第一道防线病毒防火墙的检测，若没有检测出病毒 则将其复制两份，一份拷贝直接存放于应用服务器中；一份拷贝则进入基于模糊模式识别分类算法的检测服务器中，对其行为特征进行检测，如果发现其可疑行为，判断为感染了未知病毒，检测服务器则将信息反馈给应用服务器，然后在应用服务器中对该文件进行隔离、监管或删除，也可以将其发送给相关专家进行进一步精确分析。

由于基于程序行为的杀毒系统可能引发对系统不可预料

收稿日期：2005-04-28。
基金项目：国家自然科学基金项目 (60373023)；湖南省自然科学基金项目 (04JJ6032)。
作者简介：张波云 (1972－)，男，湖南永州人，博士研究生，讲师，研究方向为网络信息安全； 殷建平，教授，博士生导师； 唐文胜，博士研究生，副教授。

的破坏，故在该系统中配置了专门进行未知病毒检测的检测服务器，它对正常服务器的性能影响极微，十分适用于实时在线系统中的未知病毒检测。检测服务器中的病毒检测引擎采用模糊模式识别分类算法实现。其分类算法主要针对程序的执行过程中使用的系统 API 函数调用，它监视程序的行为并进行分析，能有效地检测未知病毒和各种多态与变形病毒。

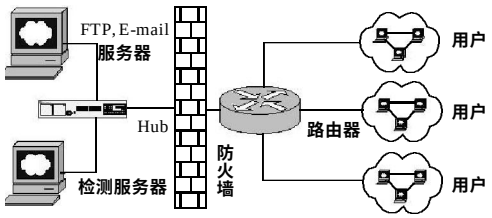

图 1　系统结构

2　算法数学模型

设有 n 个样本组成样本集合：$X=\{x_1,x_2,\cdots,x_n\}$，每个样本用 m 个指标特征向量表示：$x_j=(x_{1j},x_{2j},\cdots,x_{mj})^T$，则样本集可用 m×n 阶指标特征值矩阵表示：

$$x_{m\times n}=\begin{pmatrix}x_{11}&x_{12}&\cdots&x_{1n}\\x_{21}&x_{22}&\cdots&x_{2n}\\\cdots&\cdots&\cdots&\cdots\\x_{m1}&x_{m2}&\cdots&x_{mn}\end{pmatrix}=(x_{ij}) \tag{1}$$

式中：x_{ij}——样本 j 指标 i 的特征值，$i=1,2,\cdots,m$，$j=1,2,\cdots,n$。

设将 n 个样本依据样本的 m 个指标特征按 c 个级别(或类别)加以识别，其模糊识别矩阵为

$$U=\begin{pmatrix}u_{11}&u_{12}&\cdots&u_{1n}\\u_{21}&u_{22}&\cdots&u_{2n}\\\cdots&\cdots&\cdots&\cdots\\u_{c1}&u_{c2}&\cdots&u_{cn}\end{pmatrix}=(u_{hj}) \tag{2}$$

式中：u_{hj}——样本 j 从属于级别 h 的相对隶属度，$h=1,2,\cdots,c$。满足条件：$\sum_{k=1}^{c}u_{kj}=1,\forall j 0\le u_{hj}\le 1$，$\sum_{k=1}^{c}u_{kj}>0,\forall h$。

设每个级别 h 有 m 个指标特征值(称为标准指标特征值)，则 c 个级别的指标特征可用 m×c 阶标准指标特征值矩阵表示：

$$Y=\begin{pmatrix}y_{11}&y_{12}&\cdots&y_{1c}\\y_{21}&y_{22}&\cdots&y_{2c}\\\cdots&\cdots&\cdots&\cdots\\y_{m1}&y_{m2}&\cdots&y_{mc}\end{pmatrix}=(y_{ih}) \tag{3}$$

式中：y_{ih}——级别 h 指标 i 的标准指标特征值。

从训练样本集中我们可获得每一类别的相应模糊集描述 $\tilde{T}_c$。而对于一个要分类样本，获得其 m 个特征的隶属度 $\mu_1,\mu_2,\cdots,\mu_m$ 后进而得到该文件的模糊集描述

$$\tilde{M}_j=\{\mu_1/x_{1j},\mu_2/x_{2j},\cdots,\mu_m/x_{mj}\} \tag{4}$$

然后，计算 $\tilde{M}_j$ 与每一个 $\tilde{T}_c$ 间的贴近度 $\psi(\tilde{M}_j,\tilde{T}_c)$，在此我们采用的是 *Euclid* 贴近度，其计算公式为

$$\psi(\tilde{A},\tilde{B})=1-\frac{1}{\sqrt{t}}\left(\sum_{i=1}^{t}(\mu_i^A-\mu_i^B)^2\right)^{1/2} \tag{5}$$

最后根据“择近原则”分类，即对于模糊集 $\tilde{A}_i,\tilde{B}(i=1,2,\cdots,n)$，若存在 i 使得 $\psi(\tilde{A}_i,\tilde{B})=\vee_{1\le j\le n}\psi(\tilde{A}_j,\tilde{B})$，则认为 $\tilde{A}_i$ 与 $\tilde{B}$ 最贴近，将 $\tilde{A}_i$ 与 $\tilde{B}$ 归于一类。

3　未知病毒检测

3.1　病毒特征提取

病毒与一般程序的区别是要在于执行了一些特殊的动作来破坏系统。它们都是一种程序，需要调用操作系统提供的各种功能函数才能达到传播自身和破坏系统的目的。因此我们将程序在所使用的 DLL 中调用的 API 函数[5]作为待检测程序的行为特征。

对样本库中的程序进行 *API* 调用跟踪处理后，可以得到大量的系统调用序列，这些不同的 *API* 对于病毒识别所起的作用是不一样的。可以想象，当一个 *API* 调用在病毒文件中出现的频率非常高，而在正常程序文件中出现频率较低时，该 *API* 对识别病毒所作的贡献就比较大。因此尽量提取此类 *API* 构成“模糊特征集”。在此，我们采用概率统计学中的均方差法来进行实验。类间频率均方差能较好地体现各不同的 *API* 调用的“贡献程度”，一个被调用 *API* 的分类作用与它的类间频率均方差成正比。

均方差的计算过程如下：①对所有训练库的样本进行系统调用跟踪，获得调用的 *API* 序列 $A=\{A_1,A_2,\cdots,A_p\}$，$(1\le i\le p)$ 统计各 *API*(即 A_i) 在每一病毒程序 V_j 中出现的频率 A_{ij}^V 及在每一正常程序 N_j 中出现的频率 A_{ij}^N；②计算每个被调用的 *API* 在病毒程序与正常程序文件中出现的频率均值 $E(A_i^V)=\frac{1}{S}\sum_{j=1}^{s}A_{ij}^V$，$s$ 为病毒样本数量，$E(A_i^N)=\frac{1}{n}\sum_{j=1}^{n}A_{ij}^V$，$n$ 为正常程序数量；③计算每个 *API* 的总出现频率均值 $E(A_i)=\frac{E(A_i^V)+E(A_i^N)}{2}$；④计算每个 *API* 的类间频率均方差：$D(A_i)=\sqrt{(E(A_i)-E(A_i^V))^2+(E(A_i)-E(A_i^N))^2}$。

将所有的 *API* 按均方差大小排序，选出前 t 个组成“模糊特征集”。通过实验后选取了 88 个主要 *API*，一个示例如表 1 所示。并对不同的程序的行为特征划分为 3 个危险级别：即一般、较严重和严重，在表中分别用☆、☆☆和☆☆☆表示。

表 1　特征列表

序号	程序行为	相关 API 调用	危险度	动态链接库
1	文件搜索	FindClose；FindFirstFileA；FindNextFileA；FindResourceA	☆☆☆	Kernel 32.dll
2	目录搜索	GetWindowsDirectoryA；GetSystem DirectoryA；SetCurrentDirectoryA	☆☆☆	

3.2　病毒检测引擎

计算机病毒检测是一个二值分类问题，即病毒与非病毒两类。对于样本空间中的每一个样本程序我们均可以从中提取出感兴趣的一组特征集，定义分类集为{正常，病毒}，我们的目标是在获得给定样本程序文件中的特征集 F 后，判别出该样本是正常程序或是病毒程序。

因为一个程序具有相应的特征，故可以对某对象或对象类通过它所具有的特征来描述，进而可用一个定义在特征类上的模糊集来描述它们。对于程序文件而言，设 $Q=\{q_1,q_2,\cdots,q_n\}$ 为由 n 个特征量组成的论域，则一个可执行程序或程序类 P 可用定义在论域 Q 上的一个模糊集来描述，即 $\tilde{P}=\{\mu_1/q_1,\mu_2/q_2,\cdots,\mu_n/q_n\}$。它表示文件 P 具有特征 q_i 的隶属度是 μ_i，其中 μ_i 是 [0，1] 间的一个实数。同理，病毒程序类与正常程序类也可用 Q 上的

模糊集来描述。

设某*API*函数在病毒文件中出现的频率均值为$E(A_i^V)$，在正常文件中出现的频率均值为$E(A_i^N)$，则用*F*分布中的正态偏大型模糊分布来构造病毒程序集$\tilde{V}$的隶属函数

$$\mu_V(E(A_i^V))=\begin{cases}0 & ,E(A_i^V)<0\\ 1-e^{-(E(A_i^V))^2/\sigma^2} & ,E(A_i^V)\geqslant 0\end{cases} \tag{6}$$

式中：$\sigma=\max\{E(A_1^V),E(A_2^V),\cdots,E(A_t^V)\}/3$；$t$——特征量的个数。

同样构造正常程序集$\tilde{N}$的隶属函数为

$$\mu_N(E(A_i^N))=\begin{cases}0 & ,E(A_i^N)<0\\ 1-e^{-(E(A_i^N))^2/\sigma^2} & ,E(A_i^N)\geqslant 0\end{cases} \tag{7}$$

对于待检测程序文件，其隶属函数为

$$\mu_M(A_i)=\begin{cases}0 & ,A_i<0\\ 1-e^{-(A_i)^2/\sigma^2} & ,A_i\geqslant 0\end{cases} \tag{8}$$

在检测引擎自学习阶段，对如表 1 中所列特征量在病毒程序测试集中出现的频率进行统计，并据隶属函数$\mu_V(E(A_i^V))$计算得到模糊集

$$\tilde{V}=\{\mu_1^V/A_1,\mu_2^V/A_2,\cdots,\mu_t^V/A\} \tag{9}$$

同样，对它们在正常程序测试集中做同样处理，可以得到模糊集

$$\tilde{N}=\{\mu_1^N/A_1,\mu_2^N/A_2,\cdots,\mu_t^N/A\} \tag{10}$$

对于进入系统的待检测文件*M*，首先对它进行*API*调用跟踪，获得其调用序列。并对如表 1 中所列特征量在其中出现的频率进行统计，进而得到：$\{A_1,A_2,\cdots,A\}$，式中t=88，由隶属函数$\mu_M(A_i)$计算可得到该文件的模糊集描述：

$$\tilde{M}=\{\mu_1/A_1,\mu_2/A_2,\cdots,\mu_t/A\} \tag{11}$$

计算$\tilde{M}$与$\tilde{V}$之间的贴近度$\psi(\tilde{M},\tilde{V})$以及$\tilde{M}$与$\tilde{N}$之间的贴近度$\psi(\tilde{M},\tilde{N})$。在此我们采用的是*Euclid*帖近度，其计算方法如 (5) 式。最后根据“择近原则”将待检测文件归于病毒类或正常类。

4 实验分析

用于实验的样本数据如表 2 所示。样本空间中样本总数为 632，分为正常程序与染毒程序。正常程序从操作系统平台中选取，我们选用的是 Windows 2000 Server 首次安装后，机器中的全部 PE 文件共 423 个。通过各种途径收集的病毒程序 209 个。为获得样本空间中程序的特征向量，我们编写了 API 调用跟踪器，可以实现 Windows 2000 Server 环境下的全部 API 函数调用的拦截。

表 2 实验样本数据

	样本空间	训练集	测试集
正常程序	423	373	50
染毒程序	209	159	50
合计	632	532	100

实验目标主要是对查毒引擎的错误率进行测试。我们将错误类型分为两种：①将正常程序判断为病毒，称为 False Negative；②将病毒程序判断为正常程序，称为 False Positive。用测试集（共 532 个文件）作为训练数据先对分类器进行训练，然后对测试集中的文件（共 100 个）进行分类测试，并进行自学习。然后用全体数据集作为训练数据，测试集从该样本空间中随机选出不等数目的文件而获得，实验结果如图 2 所示。

从实验结果知分类器用较少训练集中的样本测试与用较多训练样本进行测试获得的精度几乎相等，这对于计算机病毒样本较难获得的情况下检测未知病毒非常有用。我们曾在文献[6]中利用基于实例学习的K-最近邻算法对本组样本空间中的数据进行过测试，其检测错误率最小值约为 4.8 %。使用模糊模式识别分类算法的效果稍好。但 KNN 算法开销很小，而本文算法的开销更多，在实际应用中这二方面都要兼顾。

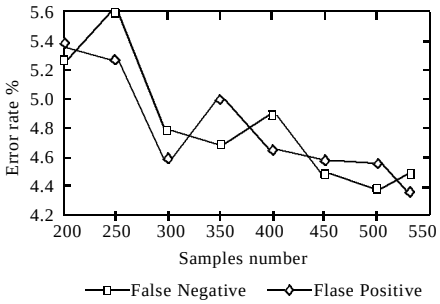

图 2 病毒检测系统性能

5 结束语

本文尝试用基于模糊模式识别的算法来实现对计算机病毒的检测，其中结合了特征码检测技术与模糊智能学习技术，检测成功率较高。选择的程序特征向量是 API 调用序列，它能有效地对付变形与多态病毒。系统开销主要集中于程序特征的提取和隶属度的获得，为简便起见我们采用的是统计方法来计算后者。

该系统是一个病毒检测器，只能隔离病毒而不能实现对病毒的清除，与其它反病毒工具的集成是一个可行的解决方案。因为基于病毒行为的检测会对系统造成不可预料的破坏性，故如何寻找高效与安全的特征提取工具是我们以后研究的要点。在我们的测试床中，将计划进行其它的基于机器学习算法的恶意代码检测实验，以期对其它的恶意代码如木马、蠕虫和间谍程序等进行测试，并对各种方法的性能与开销进行对比选择和优化以期用于实际的产品中。

参考文献:

[1] Diomidis Spinellis. Reliable identification of bounded-lengthviruses is NP-complete[J]. *IEEE Transactions on information Theory*, 2003,49(1):280-284.

[2] Gerald J Tesauro,Jeffrey O Kephart. Neural networks for computer virus recognition[J].*IEEE Expert*, 1996,8:5-6.

[3] Schultz M, Eskin E, Zadok E,et al. Data mining methods for detection of new malicious executables[A]. Roger Needham Proceedings of the 2001 IEEE Symposium on Security and Privacy [C].*Washington: IEEE press*, 2001.38-49.

[4] Bargiela, Pedrycz A, Hirota W. Granular prototyping in fuzzy clustering[J]. *IEEE Transactions on Fuzzy Systems*, 2004,12(5): 697-709.

[5] Kruglinski David J,Scot Wingo,George Shepherd. Programming Visual C++[M]. *Washington: Micosoft Press*, 1998.

[6] 张波云,殷建平,张鼎兴,等.基于 K-最近邻算法的未知病毒检测[J].计算机工程与应用, 2005,41(6): 7-10.