基于 PE 文件结构异常的未知病毒检测

樊 震, 杨秋翔
(中北大学 电子与计算机科学技术学院, 山西 太原 030051)

摘 要: 目前基于行为分析的未知病毒检测方法, 需要可执行文件运行后才能检测到, 无法检测出以静态形式存在计算机中的病毒文件。文中提出了一种基于静态文件的未知病毒检测新技术, 通过分析 PE 文件结构中的异常值, 运用贝叶斯方法和支持向量机来识别静态和非静态的未知病毒。相比基于行为分析的未知病毒检测方法, 在不需要运行可执行文件的情况下即可检测出是否可能为未知病毒文件。本方法相比基于函数调用 API 序列的数据挖掘方法的病毒检测方法, 不需要对文件进行脱壳等复杂计算处理, 明显提高了检测速度。试验结果表明, 该方法对未知病毒有较快的检测速度、较高的识别率和较低的误判率。

关键词: 静态文件; PE 文件; 未知病毒检测
中图分类号: TP309.5 文献标识码: A 文章编号: 1673-629X(2009)10-0160-04

Unknown Virus Detection Based on Exceptional PE File Structure

FAN Zhen, YANG Qiu-xiang
(School of Electronic and Computer Science and Technology,
North University of China, Taiyuan 030051, China)

Abstract: Behavior-based analysis of currently unknown virus detection methods, necessary to run an executable file can be detected after, can not be detected in static form of computer virus file. In this paper, a document based on the static unknown virus detection of new technologies, by analyzing the PE file structure of the abnormal value, the use of Bayesian methods and support vector machine to identify the static and non-static unknown virus. Compared to behavior-based analysis of the unknown virus detection methods, do not need to run the executable file in the case of the possibility to detect the virus file is unknown. The method is compared to API function call sequence based on the data mining method of virus detection methods, the documents do not need to shell deal with such complex calculations clearly improve the detection speed. Test results indicate that the method of the unknown virus detecter has faster detection speed, higher recognition rate and lower rate of misjudgment.

Key words: static state file; PE file; unknown virus detection

0 引 言

对未知病毒的检测技术可以在很大程度上弥补反病毒软件总是滞后于病毒的发展[1]。当前的未知病毒检测技术都是基于行为判定方法[2], 需要病毒文件实时运行。实践证明, 因为种种原因(如杀毒软件的漏杀, 病毒库未及时更新, 实时监控失效等), 有相当数量的病毒文件是以静态形式存在用户计算机上的。

目前的未知病毒检测技术的主要方法是对当前运行文件的相应行为进行分析, 进而给出一定的可疑概率, 对未知病毒的检测取得了较好的识别率。但其前提需要病毒文件实时运行起来, 经实践证明, 处于运行状态的病毒文件只占总病毒文件的 40% 左右甚至更少(一个病毒文件会在几个不同的文件路径下保存自己, 但是同时运行的只有一个副本)。基于文献[3]行为判定的未知病毒检测方法, 对于由病毒释放[4]到其他本地文件路径下的病毒文件, 或通过其他方法存在于计算机磁盘中的静态病毒文件无法检测到, 导致病毒清除不干净, 随时会给计算机带来破坏。

基于虚拟机技术[5]的未知病毒检测方法, 虽然有较高的识别率, 但因为其需要模拟处理器的指令虚拟运行, 所以运行速度很慢。基于函数调用 API 序列的数据挖掘[6]方法的未知病毒检测方法虽然不需要病毒文件的运行, 但其缺点是需要对文件进行脱壳等复杂计算处理, 计算量较大, 实用性较差。

文中主要基于以上问题提出解决方法。不需要病

收稿日期: 2008-12-06; 修回日期: 2009-04-07
基金项目: 山西省自然科学基金(20011040)
作者简介: 樊 震(1982-), 男, 山西运城人, 硕士研究生, 主要研究方向为计算机网络和信息安全; 杨秋翔, 硕士研究生导师, 副教授, 研究方向为计算机网络。

毒文件的运行, 不必脱壳, 直接通过分析 PE 文件结构中的异常作为判定项, 进而进行相应分析计算, 判断该文件是否为病毒文件。

1　PE 文件异常

1.1　PE 文件的结构

PE (Portable Executable) 是微软 Windows 操作系统环境中通用的可执行文件格式[7], Windows 操作系统上能够正常运行的应用程序必须是可移植可执行格式的 32 位可执行文件[8]。其主要结构[8]如图 1 所示。PE 文件由一个 MS- DOS 的可执行体开始, 在 DOS 头后部的是 PE 头部, 它包含了主要的头部信息和一些可选头部信息。紧跟在 PE 头部后的是节表, 它包含了所有节的名称、位置、长度和属性。节表还包含了数据节或代码节的种类信息。代码只有一种类型, 而数据有多种类型, 除了程序读写的数据外, 在节中还有 API 引入表和输出表, 资源和重定位信息等数据。

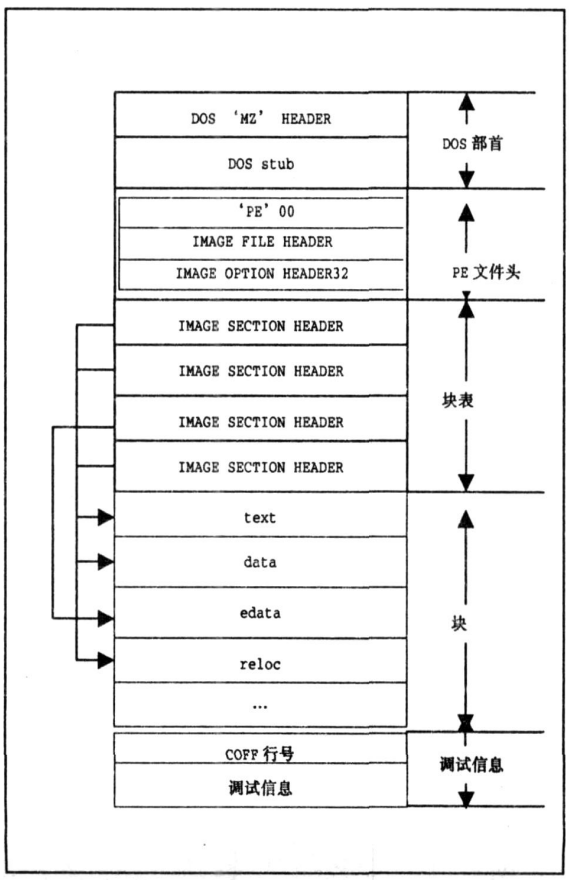

图 1　PE 文件的框架结构

1.2　病毒文件在 PE 文件格式上的异常特征

以下是一些病毒文件格式项中的异常特征[7]:

1) IMAGE_ OPTIONAL_ HEADER32 中 Check2 Sum 项:

病毒及一些非法程序, 在修改 PE 文件格式时, 常将此项置 0, 而正常编译器编译的文件有正确的数值。

2) Section Table 段表项:

Section Table 段表一般包括: Name、Pointer2 ToRawData、SizeOfRawData、Characteristics 等表项。

(1) Name 块名, 这是一个 8 位 ASCII, 多以/ .0开头, 这个点实际是不需要的。常见的块名: . text 存放指令代码, . data 存放初始化的数据, . idata 包含其他外来 DLL 的函数及数据信息, 即输入表, . rsrc 存放模块的全部资源数据, . reloc 存放基地址重定位表, . edata 存放文件的输出表。

病毒常使用一些无规则的命名, 与正常文件的块名有较大区别, 如图 2 所示。

Name	VOffset	VSize	ROffset	RSize	Flags
PS 素腌	00001000	00006000	00000010	000001F0	E0000060
	00007000	00009000	00000200	00001ED4	E0000060
诺@	00010000	00001000	00000010	000001F0	E0000060

图 2　Trojan- PSW. Win32. OnlineGames. rc

(2) PointerToRawData[7]。该块在磁盘文件中的偏移。一些病毒文件常会将此项置零。对应图 2 中 ROffset 项。

(3) SizeOfRawData[7]。该块在磁盘文件中所占的大小。如果块表中某段此项值为零, 说明该文件结构异常, 比较可疑。见图 2 中 RSize 项。

(4) Characteristics[7] 块属性。是判断是否可疑的重要标志。该字段是一组指出块属性的标志。多个标志求或即为 Characteristics 值, 如下是一些常见的标志 (对应图 2 中的 Flags 项):

常见的有可写标志的段名: . data 、DATA、BSS、. tls、. idata、. adata。如果某段有可写标志, 而又不是常见的这些段名, 就比较可疑了。

常见的块属性见表 1。

表 1　常见文件的块属性

字段值	用途
IMAG_ SCN_ CNT_ CODE 00000020b	包含可执行代码, 常与 10000000h 一起设置
IMAGE_ SCN_ CNT_ INITIALIZED_ DATA 00000040h	包含已初始化的数据
IMAGE_ SCN_ CNT_ UNINITIALIZED_ DATA 00000080h	该块包含未初始化的数据
IMAGE_ SCN_ MEM_ DISCARDABLE 02000000h	该块可被丢弃, 因为它一旦被装入后, 进程就不再需要它了。常见的可丢弃块. reloc(重定位块)
IMAGE_ SCN_ MEM_ SHARED 10000000h	共享块
IMAGE_ SCN_ MEM_ READ 40000000h	可执行块, 通常当 00000020h 标志被设置时, 该标志也被设置
IMAGE_ SCN_ MEM_ WRITE 80000000h	该块可写, 如果可执行文件没有设置该标志, 装载程序就会将内存映像页标记为可读和可执行

3)资源段中含有 PE 文件:

普通文件的资源段一般存放文件的图标、文件相关的位图、版本信息等内容。如果发现在资源段中出现了以 MZ 为标志的 PE 文件, 则很大程度上可能是病毒了。在运行时病毒会把此段中的 PE 文件释放出来。

4)代码段, 代码段的段名如果不是. text . rsrc, 也比较可疑。数据段, 段名不是. data。

5) BaseOfData 指向的地址不在段表中。BaseOf2Code 指向的地址不在段表中。

6)段错位: PE 头中正常的 BaseOfCode 和 BaseOf2Data 分别指向对应段表中的 code 和 data 段。

7)文件版本信息: 正常的文件一般都会有版本信息。而病毒文件大多没有。

1.3 特征的提取

将所有文件分为病毒样本集和正常样本集, 分别统计 1.2 节中所述各种异常项的在两个样本集中的次数即字频。根据贝叶斯公式[9]计算其后验概率:

$$p(w_i|X)=\frac{p(X|w_i)p(w_i)}{\sum_{j=1}^{c}p(X|w_j)p(w_j)}$$

$p(X|w_i)$ 是模式向量 X 在 w_i 状态下的条件概率密度, $p(w_i)$ 为 w_i 类的先验概率。

然后根据熵函数计算其可分离性。设 $\sum_{i=1}^{c}p(w_i|x)=1$。

则熵为: $H_c(p)=-\sum_{i=1}^{c}p(w_i|x)\log p(w_i|x)$

取熵的期望[9]: $J_H=E_x[-\sum_{i=1}^{c}p(w_i|x)\log p(w_i|x)]$

作为可分性判断函数。

2 支持向量机的训练和分类

假设有 m 个训练样本, 每个训练样本由一组向量 X 组成:

$x_i \in R^n$, $i=(1,2,,,m)$ 并且每个 x_i 都有一个 y_i 与之对应。即: x_i y $y_i \in \{+1,-1\}$。+1 表示正常样本, -1 表示病毒样本。超平面方程为: $w^* x+b=0$。

满足 Mercer 条件后, 优化函数:

$$\max Q(\alpha)=\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j K(x_i,x_j)$$

其中 $K(x_i,x_j)$ 为核函数, α_i 为拉格朗日乘子。文中选择径向基函数为核函数[9]: $K(x_i,x_j)=\exp(-\frac{+x_i-x_j+^2}{R^2})$

对应的最优分类函数[9]: $f(x)=\mathrm{sgn}(\sum_{i=1}^{m}\alpha_i y_i K(x_i,x)+b)$

3 实 验

基于以上分析设计本试验的过程, 分为: 特征选取, 向量机训练[10]和分类两部分。

3.1 特征选取

试验选取不同种类的 168 个病毒样本和 1500 个正常文件样本。按图 3 所示流程选取特征。

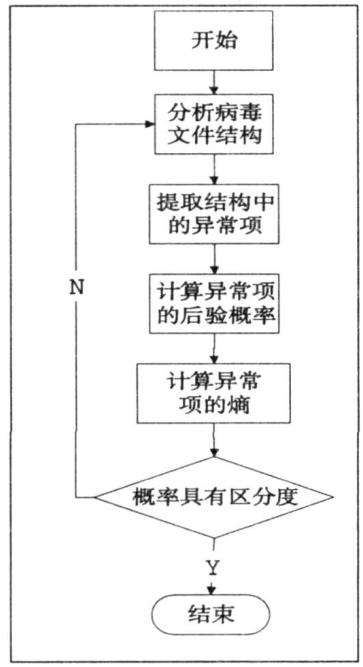

图 3 特征选取流程

最终选取熵值最小的前 12 个特征项:

1. IMAGE_DOS_HEADER 中 e_lfanew 项;

2. IMAGE_OPTIONAL_HEADER32 中 NumberOfSections 项;

3. IMAGE_OPTIONAL_HEADER32 中的 BaseOfCode 项;

4. IMAGE_OPTIONAL_HEADER32 中的 BaseOfData 项;

5. IMAGE_OPTIONAL_HEADER32 中 Image2Base 项;

6. IMAGE_OPTIONAL_HEADER32 中 FileAli2gnment 项;

7. IMAGE_OPTIONAL_HEADER32 中 Check2Sum 项;

8. IMAGE_OPTIONAL_HEADER32- > DataDi2rectory[1]. VirtualAddress 项;

9. IMAGE_OPTIONAL_HEADER32- > DataDi2rectory[1]. Size 项;

10. IMAGE_OPTIONAL_HEADER32 - > DataDirectory[5]. VirtualAddress 项;

11. IMAGE_OPTIONAL_HEADER32 - >

DataDirectory[5]. Size 项;

12. IMAGE_ SECTION_ HEADER 中 Name 项。

3.2　向量机训练和分类

试验随机选取 168 个病毒样本和 1500 个正常文件样本。文件检测流程如图 4 所示。

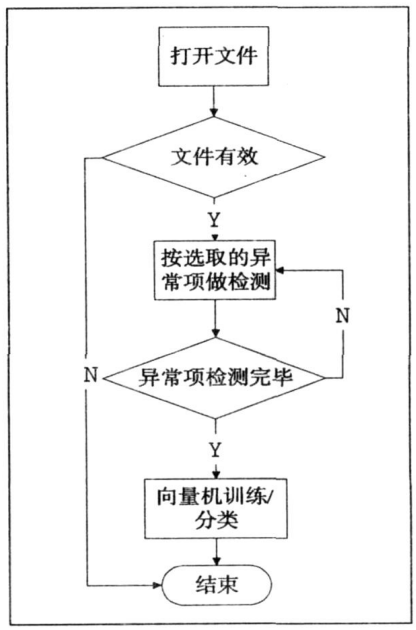

图 4　文件检测流程

经过反复试验选择核函数的参数: $R^2 = \frac{1}{8}$

然后选取 368 个病毒样本(其中木马类 89, 后门类 88, 蠕虫类 191), 和 Windows 系统目录下 15001 个正常样本做分类试验。

表 2 为病毒文件和正常文件的检测概率。

表 2　分类检测概率

类别名称	数量	漏报率	误报率	正确率
木马	89	6%	8%	94%
后门	88	3.5%	14%	96.5%
蠕虫	191	17%	5%	83%
正常文件	15001	0	0.26%	99.74%

文件平均检测速度: 15000 个 / 475s = 31.58 个/ s。

4　结论和进一步研究

(1) 试验证明该方法具有较高的检测率(大于 90%)和较低的误检率(小于 1%)。

(2) 由于部分应用软件为缩小文件体积和防止破解, 使用加壳软件[11] 对文件进行特殊处理, 会导致检测出较高的可疑概率。另外部分破解软件及黑客工具软件[12] 等也会给出较高的可疑概率。为提高检测方法的识别率, 可以针对病毒常见的出现位置进行扫描:

系统目录: \ Documents and Settings \ Administrator \ Local Settings 下的 Temp 和 Temporary Internet Files 文件夹

系统目录: \ WINDOWS

系统目录: \ WINDOWS \ system32

系统目录: \ WINDOWS \ Temp

系统目录: \ Program Files \ Internet Explorer

系统目录: \ 根目录

(3) 在检测文件块名项异常项时, 可以使用白名单做文件块名的检测, 提高区分度。

5　结束语

提出了一种基于静态文件和 PE 文件结构的未知病毒检测方法。弥补了目前业界所采用的基于行为分析的未知病毒检测技术的不足, 具有较高的检测率和较低的误检率。相比基于 API 序列等的检测方法, 不必进行脱壳、解压等繁重计算。此技术可以做为单独的检测工具, 也可以和杀毒软件相结合, 作为启发式扫描[13] 的一部分。

参考文献:

[1] 刘　涛, 邓璐娟, 丁孟宝. 计算机反病毒技术及预防新对策[J]. 计算机技术与发展, 2007, 17(5): 104- 106.

[2] 张仁斌, 李　钢, 侯整风1 计算机病毒与反病毒技术[M]. 北京: 清华大学出版社, 2006.

[3] 王海峰, 段友祥, 刘仁宁. 基于行为分析的病毒检测引擎的改良研究[J]. 计算机应用, 2004, 24(z2): 109- 110.

[4] Gregory P. Computer Viruses For Dummies[M]. [s. l.]: Wi2ley Publishing, Inc, 2004.

[5] Szor P. The Art of Computer Virus Research and Defense[M]. [s. l.]: Addison Wesley Professional, 2006.

[6] Witten I H, Frank E. Data Mining: Practical Machine Learn2ing and Teachniques[M]. Second Edition. Singapore: Elsevier Pte Ltd, 2006.

[7] 段　钢. 加密与解密[M]. 第 2 版. 北京: 电子工业出版社, 2003.

[8] 韩筱卿1 计算机病毒分析与防范大全[M]. 北京: 电子工业出版社, 2006.

[9] 钟　珞, 潘　昊, 封　筠, 等. 模式识别[M]1 武汉: 武汉大学出版社, 2006.

[10] 张　苗, 张德贤. 多类支持向量机文本分类方法[J]. 计算机技术与发展, 2008, 18(3): 139- 141.

[11] Cerven P. Crackproof Your Software- The Best Ways to Pro2tect Your Software Against Crackers[M]. [s. l.]: No Starch Press, Inc. 2002.

[12] Erbschloe M. Trojans、Worms and Spyware[M]. [s. l.]: Else2vier Butterworth- Heinemann, 2005.

[13] 秦志光, 张凤荔1 计算机病毒原理与防范[M]. 北京: 人民邮电出版社, 2007.