·安全技术· 文章编号:1000-3428(2017)09-0167-05 文献标志码:A 中图分类号:TP309

熵可视化方法在恶意代码分类中的应用

任卓君,陈 光

(东华大学 信息科学与技术学院,上海 201620)

摘 要: 恶意代码激增极大地威胁着信息系统安全。为提高辨识效率,加快应急响应速度,结合信息熵的定义,利用 Jaccard 度量和 K 最近邻分类算法,提出一种新的用于研究恶意代码分类的可视化方法。将二进制文件经局部熵计算转换成熵像素图,从视觉角度直观呈现恶意代码内部特征,通过降维显示机制提高相似度比对和分类的效率。实验结果表明,该方法使用 66 个族的 664 个由卡巴斯基命名规则命名的样本进行评估,平均分类准确率为 93.67%,能有效地分类恶意代码样本。

关键词: 恶意代码;可视化;谱系分类;信息熵;Jaccard 指数;K 最近邻分类算法

中文引用格式: 任卓君,陈 光. 熵可视化方法在恶意代码分类中的应用[J]. 计算机工程,2017,43(9):167-171.

英文引用格式: REN Zhuojun,CHEN Guang. Application of Entropy Visualization Method in Malware Classification[J]. Computer Engineering,2017,43(9):167-171.

Application of Entropy Visualization Method in Malware Classification

REN Zhuojun,CHEN Guang

(College of Information Science and Technology,Donghua University,Shanghai 201620,China)

【Abstract】 Soaring malwares threat the security of information systems. For increasing identification efficiency and improving response speed,this paper presents a new malware visualization method for classification based on Shannon entropy,Jaccard index and K-Nearest Neighbor(KNN) algorithm. This method transforms binary files into entropy pixel images by computing the local entropy values of samples to show the inner features of malwares directly in the visual mode,and uses dimension reduction for display to accelerate the process of similarity and classification analysis. Experimental results show that the method is quite promising with 93.67% classification accuracy on 664 samples named by Kaspersky of 66 different families,it can classify malware families effectively.

【Key words】 malware; visualization; pedigree classification; information entropy; Jaccard index; K-Nearest Neighbor (KNN) classification algorithm

DOI: 10.3969/j.issn.1000-3428.2017.09.030

0 概述

恶意代码利用计算机系统漏洞,能窃取、修改或者破坏系统上的数据,甚至摧毁整个系统,是当前信息系统安全的最大威胁。例如,2010 年 6 月首次发现的震网病毒 Stuxnet[1],是专门以摧毁真实世界中工业系统(能源基础设施)为目的的蠕虫病毒,感染了全球超过 45 000 个网络,尤以伊朗最为严重,使得其位于纳坦兹的铀浓缩离心机失控,其能掩盖故障发生,造成管理部门决策误判。更为严重的是,恶意代码的数量增殖惊人,仅 2014 年的第三季度,迈克菲实验室[2]每分钟检测到的新威胁数量超过 307 个,即每秒钟就超过 5 个;第四季度的恶意软件更是同比激增了 76%。赛门铁克[3]最新的年度安全报告就指出 2015 年新增了 4.3 亿个恶意软件。

因此,为了提高辨识效率、加快应急响应速度,研究人员必须解决如何在较短的时间内完成新恶意代码属性分类的任务。为此,科研人员尝试了多种自动化的分类方法,包括动态分析和静态分析。在动态方面,例如文献[4]基于恶意代码行为将分析所得特征用来分类研究样本。具体方法是使用商业杀毒软件标记过的数据集,在沙箱环境中监视这些样本的行为,由沙箱的虚拟环境自动导出行为报告。他们依据报告中特定字符串出现的频率为每个恶意

基金项目: 国家自然科学基金(61671006);中央高校基本科研业务费专项资金(14D310407)。

作者简介: 任卓君(1984—),女,博士研究生,主研方向为网络与信息安全;陈 光,教授、博士。

收稿日期: 2016-10-14 **修回日期:** 2016-12-09 **E-mail:** 1129110@mail.dhu.edu.cn

代码生成特征向量。随后运用支持向量机来训练和测试这些样本,分类的平均准确率达到 88% 。与文献[4]相比,文献[5]采用的是非常简单的特征,即运用程序长度来分类 7 种类型的特洛伊木马,分类的准确率也为 88% 。在之后的研究中采用可打印的字串信息来评估 13 类恶意代码族的样本,得到 98% 的分类正确率[6-7]。文献[8]则是通过查找恶意代码行为图中的最大公约子图来实现分类。虽然这些方法的分类准确率较高,但都涉及内存跟踪(污点分析)、符号执行、程序切片等一系列时间密集、资源消耗型的操作,不利于这些方法在今后的可扩展性。此外,这些方法需要执行于虚拟环境中,行为报告是基于执行轨迹而生成的,如果沙箱环境不能满足特定的触发条件,一些恶意行为就无法观测到。在静态方面,最知名的就是反汇编技术[9-11],该技术无需执行就能分析受感染的文件。该技术通过查找代码中的控制流来发现恶意模式,能提供分析的完整维度,但是如果攻击者使用混淆技术就能逃避静态检查。

而运用可视化方法在恶意代码分析时则无需反汇编操作,因此,对混淆手段具备一定的弹性。恶意代码的可视分析方法有基于自组织映射(Self-Organizing Maps,SOM)的方法[12]、基于主机行为的方法[13-15]和基于网络特征的方法[16]等。具体案例如文献[17]研发了一个从字节层面上分析二进制文件的可视化系统,用灰度图来展示文件的内部结构。文献[18]在文献[17]的基础上,运用纹理提取的图像处理方法来获得用于分类的代码特征。两者的优点在于:利用人类视觉处理抽象数据,可以获得被分析代码更全面的信息感知,提高研究人员的辨识效率,快速发现文件间的差异及其中隐藏的模式。不足之处在于:前者直接对 2 个文件进行逐字节比对,时间复杂度为 $O(n^2)$,因此,受设备硬件条件制约,分析效率低,且灰度图的方法对确定加密或压缩与否的视觉表现力不够;而后者将全局图定义为特征,如果攻击者重排二进制节的位置、或者增加冗余数据,则会影响分类的判定。

为弥补以上研究的不足,提高分析效率,本文结合信息熵、Jaccard 指数的概念,运用 KNN 分类算法,提出一种新的可视化方法来分类恶意代码。该方法具备自动推断所观测样本属类的能力,操作过程既无需逆向分析,又不用执行代码,且对混淆技术有一定的适应性。

1 熵可视化方法

熵可视化方法的分析流程包括 3 个步骤,如图 1 所示。

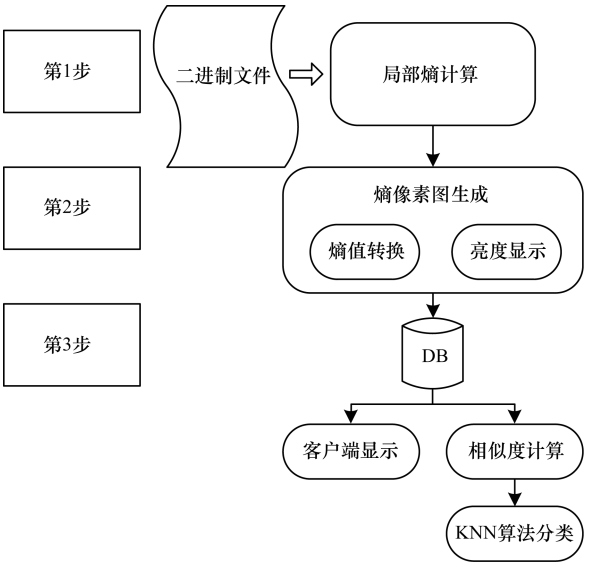


图 1 熵可视化方法的分析流程

熵可视化方法分析流程如下:

1)局部熵计算

信息熵是用来描述信源不确定度的概念,即数据集中元素的无序程度。如果数据集中某个元素出现频率很高,则熵趋于 0;如果集合中每个元素的出现率相同或相近,则熵值最大。逆向工程师关注熵的原因在于:有的恶意代码中存在信息压缩或加密的情况,往往熵值较高。因此,为识别潜在的加密常量或密钥、甚至加密内容本身,本文方法对待测样本进行局部熵计算。

二进制文件中字节的取值范围是 0x00 ~ 0xFF,故预处理时将待测样本划分成 256 Byte 的数据块,末尾不足 256 Byte 的用 0 Byte 填充。对每个数据块使用式(1)计算信息熵值,其中,i 代表数据块中出现的特定字节,即 0x00-0xFF;p_i 表示字节 i 出现的概率。

$$entropy = -\sum_{i=0}^{255} p_i \times \mathrm{lb}\ p_i \tag{1}$$

2)熵像素图生成

通过局部熵计算可以大大加快显示速度,只是这一计算结果与原文件大小成正比,而各样本的文件尺寸大小不一,因此,还不能直接用于分类比较。为获得评判相似程度的统一标准,继续将局部熵计算结果通过降维映射的方法处理得到“熵像素图”,即恶意代码的可视化特征,其结果均规定为 256 ×256 点阵的方图,由此可为之后的谱系分类提供依据,并进一步提高分析人员观测样本文件的效率。

降维映射的方法主要提供两方面的功能:熵值转换和亮度显示,即对局部熵的计算结果按顺序逐一扫描,将熵值转化为像素图中的坐标,并统计同一坐标点的叠加数量,将其映射成明暗差异的亮度感知。熵值转换后的坐标与原文件字节所处位置无关,因此,能应对攻击者重排二进制节位置;而亮度

显示采用占比的相对概念,可以缓解冗余数据带来的影响。实现该方法的 python 代码如下:

```
for j in range (begin,end-1):
c_j = (entropy[j],entropy[j+1])
num[c_j] = num.get(c_j,0) + 1
```

其中,begin 和 end 分别对应局部熵值 *entropy*[ ]的起讫位置。"熵像素图"中各点的位置坐标 c_j 按扫描顺序逐一地由 *entropy*[ ]中相邻的 2 个熵值组成;同一坐标点出现的次数 $num[c_j]$ 与其亮度显示值 *brightness*(取值范围为 0 ~255)的关系如下:

$$brightness = \left\lceil \frac{255 num[c_j]}{\sum_{j=0}^{end-1} num[c_j]} \right\rceil \tag{2}$$

通常,熵值 *entropy* 的计算结果在[0,8]之间,如果只是线性地放大熵值的布局,则不能很好地体现高熵值的重要性。因此,本文特别将 *entropy* 按函数关系(式 3)以指数形式放大,则可令具有高熵值的像素点位于图像较中心的区域,熵值与像素点在图中的映射关系见图 2。处理后的熵值区间变更为[0,255],与亮度值区间一致,便于以后新方法的功能扩展。

$$f(entropy) = \lfloor 2^{entropy} \rfloor \tag{3}$$

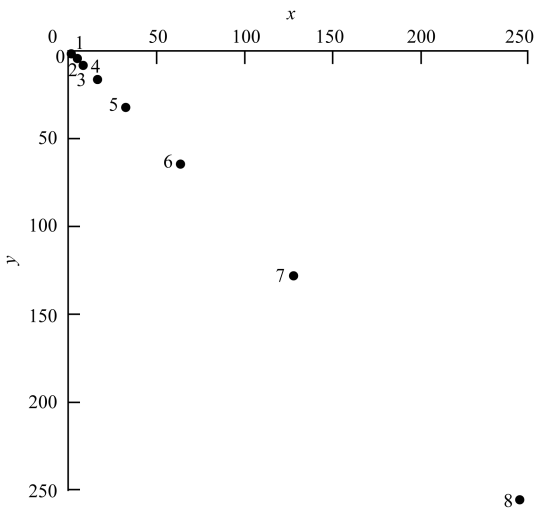


图 2　熵值与像素点在图中的映射关系

3)相似度比较及分类

将熵像素图的结果存入数据库后,即显示于客户端并进行相似性分析及样本分类。

本文提出的方法利用 Jaccard 度量比较"熵像素图"的相似性,其优势在于将图形图像间的比较转化为数据集合间的比较可以加速计算、提高分析性能。Jaccard 指数是用于比较集合间相似性或差异性的统计概念,其中 Jaccard 相似系数定义为样本集交集与样本集并集的比值。如式(4)所示,其中,A,B 是待比较的 2 个集合,相似性取值在[0,1]之间。本文中使用的集合概念为恶意样本呈现于熵像素图中的点集,点集中的元素为坐标参数与亮度值构成的信息三元组(x,y,*brightness*)。

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}, 0 \leqslant J(A,B) \leqslant 1 \tag{4}$$

依据相似度的计算结果,本文利用 K-Nearest Neighbor 算法进行样本分类。这里给出算法的定义:设已知类别的样本数量为 N,其中有 N_i 个样本来自 ω_i 类,k_i 表示与未知样本 x 在距离测度上最接近的 k 个近邻中属于 ω_i 类的样本数。使用判别函数式(5),在已知样本中计算出 k 个近邻样本,之后通过决策函数式(6)确定最大的 k_i 值,则决策 x 属于 k_i 所代表的类。

$$g_i(x) = k_i, i = 1,2,\cdots,c \tag{5}$$

$$g_j(x) = \max_i k_i \tag{6}$$

2　实验与结果分析

本文提出的熵可视化方法采用 python 语言编程,以自动化的方式执行。程序在 Ubuntu 系统上调试,硬件配置选用 Intel Xeon X3450 处理器 4 核 8 线程,时钟核心速度 2.66 GHz,DDR3 内存 8 GB。鉴于 VX Heavens 官方网站的恶意代码均由卡巴斯基命名规则预分类,这能为评估本文方法是否正确分类提供量化的指标,故本文随机采集了 66 个族的 664 个恶意样本,涉及后门(Backdoor)、木马(Trojan)、蠕虫(Worm)等。

图 3 给出了部分样本的熵像素图,分别来自 Rootkit. Agent 族、Trojan. Regrun 族和 Worm. Delf 族。如图 3 所示,同族样本的熵像素图相似,而异族样本的熵像素图差异明显,由此说明本文方法在视觉效果上能较好地区别各类样本,表 1 是这些样本的相似性计算结果,该数据较好地证明了上述观点。

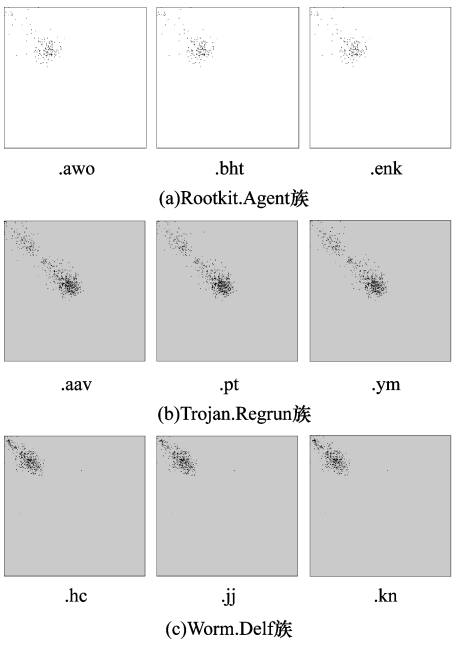
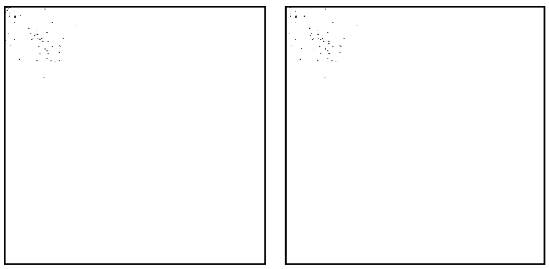


图 3　部分样本的熵像素图比较

表 1 所列样本平均相似度结果 %

恶意代码族	R. A 族	T. R 族	W. D 族
Rootkit. Agent 族	98.18	0.74	1.25
Trojan. Rugum 族	-	93.90	3.16
Worm. Delf 族	-	-	95.99

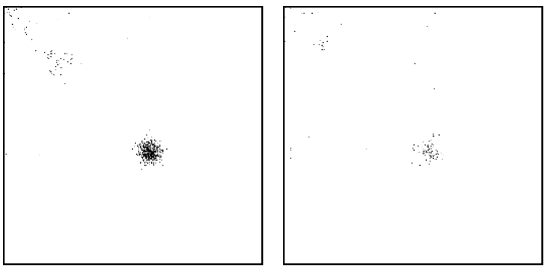
图 4 给出了 Backdoor. Delf 族中部分样本的熵像素图,其中样本. jp 经第三方软件(PEiD)检测未加壳,而样本. lb 则采用 Upack0. 39beta 加壳;计算两样本间的 Jaccard 相似度为 0. 793 5。结合图 4 的视觉效果说明本文方法在一定程度上不受加壳混淆的影响。



(a).jp样本 (b).lb样本

图 4 Backdoor. Delf 族的部分样本熵像素图

图 5 分别给出了样本 Backdoor. Nuclear. uv 和样本 Email-Worm. Joleee. ac 的熵像素图,可发现高熵区域都有像素点聚集。经第三方插件(KANAL)检测,发现样本Backdoor. Nuclear. uv使用了数据压缩函式库 ZLIB 的 deflate 算法,而样本 Email-Worm. Joleee. ac 由第三方软件 IDA 分析确定使用了 TEA/N 加密算法的 0X9E3779B9h 常量。虽然这些样本的具体行为还需进一步分析,但是本文方法从视觉表现上,可以通过高熵区域确定样本是否含有加密或压缩的成分。



(b)Backdoor.Nuclear.uv样本 (b)Email-Worm.Joleee.ac样本

图 5 样本采用加密或压缩算法的熵像素图

本文按恶意代码采集时标注的样本名做预分类测试。在已知 66 族分类的情况下,根据 Jaccard 相似系数的定义计算出这 664 个恶意样本的同族平均相似性和异族平均相似性,结果如图 6 所示,同族平均相似度大大高于异族平均相似度,平均区分度为 92. 94%,其中,最大值为 99. 25%;最小值为 80. 23%。由此,说明本文方法可以用于恶意代码的分类研究。

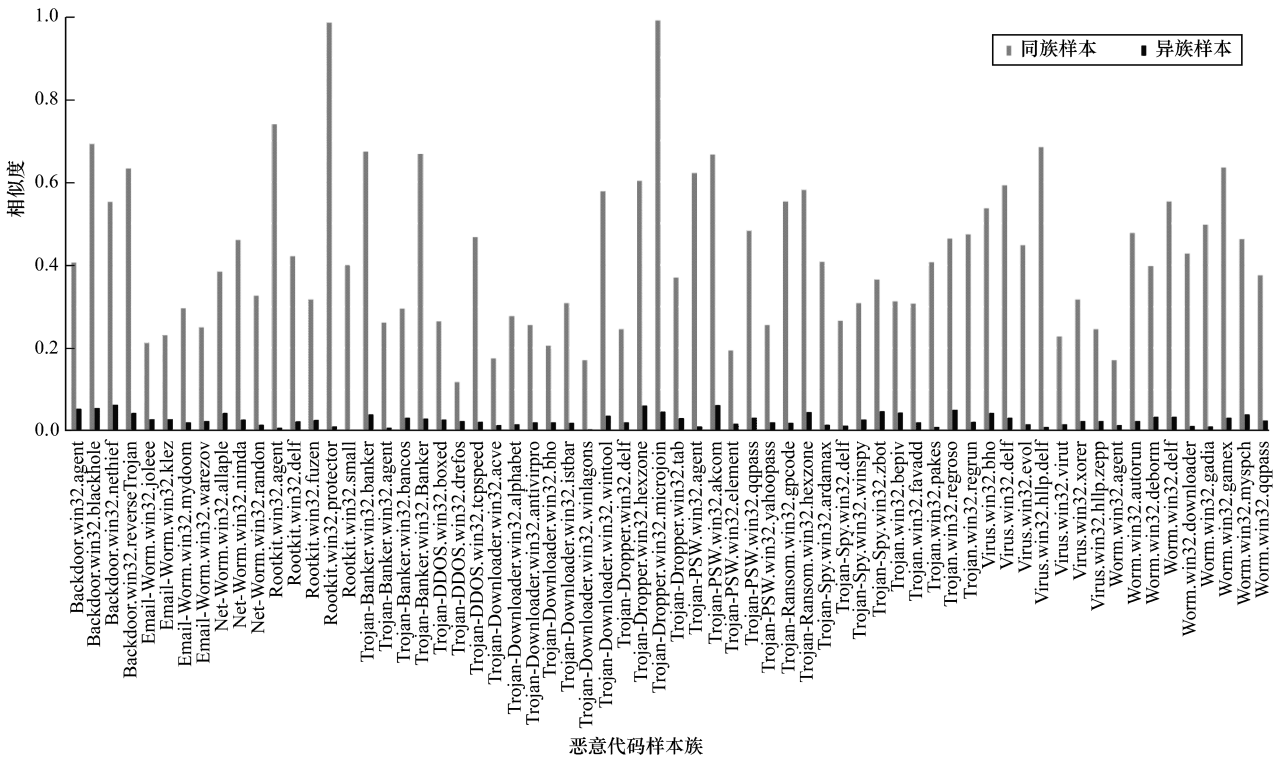


图 6 同族相似性与异族相似性比较

按 KNN 算法的定义计算某一样本的 k 个近邻,距离测度使用之前计算获得的 Jaccard 相似系数。如果这 k 个近邻经比较得出多数属于 ω_i 类,则将该样本归为 ω_i 类。在实验中,k 取[0,11]间的奇数,以

避免 $k_1 = k_2$ 的情况。经验证,本文方法可以正确区分 66 个族。KNN 算法结果如图 7 所示,分类准确率较高,均在 90% 以上。尤其是当 $k = 5$ 时,分类效果最佳,准确率为 93.67% 。分析其原因认为:本次验证所采集的样本中,有小部分族的代码个数为 6 或 7,因此,近邻信息不足($k < 5$)或干扰信息增加($k > 5$)在一定程度上会影响分类效果,但差别并不大。即使采用最近邻决策(即 $k = 1$),分类准确率也在92.32% ,由此说明本文方法能提供稳定的分类判定。

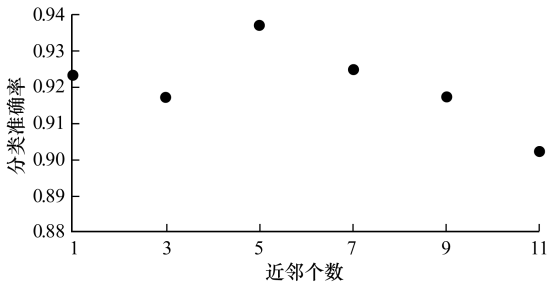


图 7　近邻个数取值与分类准确率的关系

此外,熵像素图(即恶意样本特征)的平均生成时间为 0.91 ms,图像间相似度比较平均用时为 0.56 ms,这些时间数据是进行 100 次实验后计算的平均值。与文献[18]中提取特征平均用时 54 ms 和分类比较平均用时 1.4 s 的结果相比,说明用本文方法时间效率高,同时说明本文方法采用的 Python 编程结构设计合理。

3　结束语

本文提出一种运用熵、Jaccard 相似系数、KNN 算法等实现恶意代码分类的可视化方法。该方法将二进制文件转化为熵像素图的恶意代码特征可视化,并采用局部熵值降维映射机制,使得特征(图像)生成时间开销小、同族相似度较高,另外以自动化方式实现二进制文件,操作简单。实验结果表明,该方法能有效分类各族代码。但是,本文方法较高的分类准确率是建立在已知类别标签的基础上,因此,为扩大该方法的应用范围以发现新变种,下一步将采集更多族的样本,以非监督学习的方式实现间接聚类。

参考文献

[1] 瑞星安全资讯. Stuxnet 病毒全球肆虐将影响我国众多企业[EB/OL]. [2016-10-07]. http:// www. rising. com. cn/about/news/rising/2010-09-25/8226. html.

[2] COCHIN C, CRUZ B, DENNEDY M, et al. McAfee Labs Threat Report [Z]. [S. 1.]: McAfee Corporation, 2014.

[3] WOOD P, NAHORNEY B, CHANDRASEKAR K, et al. Internet Security Threat Report[Z]. [S. 1.]: Symantec Corporation, 2016.

[4] RIECK K, HOLZ T, WILLEMS C, et al. Learning and Classification of Malware Behavior[C]//Proceedings of the 5th Conference on Detection of Intrusions and Malware & Vulnerability Assessment. Paris, France: [s. n.], 2008: 215-223.

[5] TIAN R, BATTEN L, VERSTEEG S, et al. Function Length as a Tool for Malware Classification [C]// Proceedings of the 3rd International Conference on Malicious and Unwanted Software. Los Alamitos, USA: IEEE Press, 2008: 369-378.

[6] TIAN R, BATTEN L, ISLAM R, et al. An Automated Classification System Based on the Strings of Trojan and Virus Families[C]//Proceedings of the 4rd International Conference on Malicious and Unwanted Software. New York, USA: ACM Press, 2009: 459-468.

[7] ISLAM R, TIAN R, BATTEN L, et al. Classification of Malware Based on String and Function Feature Selection[C]//Proceedings of the 2nd Cybercrime and Trustworthy Computing Workshop. Ballarat, Australia: IEEE Press, 2010: 159-167.

[8] PARK Y, REEVES D, MULUKUTLA V, et al. Fast Malware Classification by Automated Behavioral Graph Matching [C]//Proceedings of the 6th Annual Workshop on Cyber Security and Information Intelligent Research. Oak Ridge, USA: ACM Press, 2010: 156-165.

[9] 岳　峰,庞建民,赵荣彩,等. 反汇编过程中 call 指令后混淆数据的识别[J]. 计算机工程, 2010, 36(7): 144-146.

[10] 王新志,孙乐昌,张　旻,等. 基于序列模式发现的恶意行为检测方法[J]. 计算机工程, 2011, 37(24): 1-3.

[11] 张一弛,庞建民,范学斌,等. 基于模型检测的程序恶意行为识别方法[J]. 计算机工程, 2012, 38(18): 107-110.

[12] INSEON Y. Visualizing Windows Executable Viruses Using Self-organizing Maps[C]//Proceedings of ACM Workshop on Visualization and Data Mining for Computer Security. New York, USA: ACM Press, 2004: 154-166.

[13] THOMAS P. Signature Visualization of Software Binaries[C]//Proceedings of the 4th ACM Symposium on Software Visualization. New York, USA: ACM Press, 2008: 246-257.

[14] DANIEL Q, LORIE L. Visualizing Compiled Executables for Malware Analysis[C]//Proceedings of IEEE Workshop on Visualization for Cyber Security. New Jersey, USA: IEEE Press, 2009: 258-267.

[15] PHILIPP T, THORSTEN H, JAN G, et al. Visual Analysis of Malware Behavior Using Treemaps and Thread Graphs [C]//Proceedings of 2009 IEEE Workshop on Visualization for Cyber Security. New Jersey, USA: IEEE Press, 2009: 547-558.

[16] WEI Z, YACIN N. MalwareVis: Entry-based Visualization of Malware Network Traces[C]//Proceedings of VizSec'12. Seattle, USA: ACM Press, 2012: 354-363.

[17] CONTI G, DEAN E, SINDA M, et al. Visual Reverse Engineering of Binary and Data Files[C]//Proceedings of VizSec'08. Cambridge, USA: IEEE Press, 2008: 265-278.

[18] NATARJ L, KARTHIKEYAN S, Jacob G, et al. Malware Images: Visualization and Automatic Classification [C]//Proceedings of VizSec'11. Pittsburgh, USA: IEEE Press, 2011: 152-168.

编辑　索书志