基于 N-gram 特征的恶意代码可视化方法

任卓君,陈 光,卢文科
(东华大学信息科学与技术学院,上海 201620)

摘 要: 本文提出了两种基于 N-gram 特征的恶意代码可视化方法. 方法一以空间填充曲线的形式表示,解决了灰度图方法不能定位字符信息进行交互分析的问题;方法二可视化恶意代码的 2-gram 特征,解决了重置代码段或增加冗余信息来改变全局图像特征的问题. 经深度融合网络验证所提方法的识别与分类性能,取得了较优的结果.

关键词: 恶意代码; 可视化分析; 空间填充曲线; 卷积神经网络; 迁移学习
中图分类号: TP309.5 **文献标识码:** A **文章编号:** 0372-2112(2019)10-2108-08
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2019.10.012

Malware Visualization Methods Based on N-gram Features

REN Zhuo-jun, CHEN Guang, LU Wen-ke
(*College of Information Science and Technology, Donghua University, Shanghai* 201620, *China*)

Abstract: We proposed two new methods for visualization analysis based on N-gram features of malware. Method 1 uses space filling curves to solve the problem that the existing grayscale method cannot locate character information for interactive analysis. Method 2 visualizes the bi-gram features of malware to solve the problem that the attackers may relocate code sections or add redundant data to change the global image features of the visualized results. We designed the deep fusion networks to validate the detection and classification performances of the proposed methods, and the experimental results are very promising.

Key words: malware; visualization analysis; space filling curves; convolution neural networks; transfer learning

1 引言

随着网络技术的快速发展,相应的攻击技术也在迅速地演化更新. 常规的恶意代码通过多态或变形等手段进化出更具威胁的变种,不仅能逃脱安全防御扫描,还能利用计算机系统的安全漏洞,窃取、修改或破坏系统上的各类数据,甚至摧毁整个系统,给信息系统和互联网安全带来巨大的威胁. 随着移动终端数量的迅猛增长,攻击目标也由联网主机向更深更广的领域渗透[1,2],这对各种类型的网络及其用户乃至整个国家的信息安全构成了极大的威胁.

为了对恶意代码进行深层分析以确定其功能属性,研究人员探索了多种检测和识别恶意代码的技术和方法[3,4]. 常见方法如提取指纹特征[5],但由于恶意样本数量激增,新增指纹特征如不及时更新就会延误检测. 传统的分析方法还包括静态代码反汇编和动态代码执行,但都存在应用上的局限性[6]. 如静态分析是通过检查程序的控制流来查找恶意模式,只能在恶意代码不使用混淆技术时才能获得较全面的信息;而动态分析是在虚拟环境中运行恶意代码,只能在满足触发条件时才能观测到恶意行为. 为了克服现有分析技术的缺点,提高安全分析人员的工作效率,帮助他们从海量的可疑数据中快速地提取信息特征来分析、识别并分类恶意文件,迫切需要更加智能的数据分析方法. 恶意代码可视化研究就是在上述背景下形成的新兴交叉科学,它通过可视化界面进行科学的推理,以视觉分析派生出可视化对象的新属性,极大地补充了人类的认知.

近年来,这一领域涌现出很多有意义的研究成果. 例如 Zhuo 等人[7]研发的 MalwareVis 系统专注恶意样本的网络活动,并将其可视化为带有时间轴的细胞状图案. Trinius[8]设计的线图分析系统直观地呈现了待测恶意样本所执行的系统调用. Gove 等人[9]实现的视觉分析系统以交互的方式探索和比较恶意样本库中大型的

收稿日期:2018-05-04;修回日期:2019-03-10;责任编辑:孙瑶
基金项目:国家自然科学基金(No.61671006);中央高校基本科研业务费专项资金(No.14D310407)

属性集. 该方法的优点是,分析人员可以直接地比较各种属性;而缺点是很难从视觉上快速了解某个恶意代码的具体工作模式. Han 等人[10]提出了熵图方法,该方法计算恶意代码中各字节块的熵值,从而生成关于局部熵的直方图,之后运用直方图比较算法[11]来检测和分类恶意代码. 由于熵图的长度受原文件大小的影响存在着差异,因此需要采用截取操作才能进行比较,因此 Ren[12]等人对上述方法进行了改进,将局部熵归一化为统一的方图,提高了分类效率. Yoo[13]采用自组织映射的方法来呈现文件中受感染的区域. 还有一些研究能够从许多恶意软件变种中总结出相似的视觉特征,如文献[14,15]. 另外一些工具则使用可视化技术来表达分层聚类[16],如 Anderson 等人[17]绘制热图来反映恶意样本对内核的使用情况. Hashemi 等人[18]提取恶意文件的微观模式来区别不同样本的攻击行为及功能. Zhang 等人[19]生成了一种新的应用程序调用图 PM-CGdroid 来分析样本的恶意行为. 该方法通过标记目标节点、增加隐含边线、修剪无害分支等操作,与原始的应用程序调用图相比,不仅可以获得轻量级调用图,还能保留恶意功能主体. Angelini 等人[20]提出了一套可视化的解决方案 AMICO 来支持分类研究,比如帮助分析人员更好地理解分类决策以及变更分类结果的可能性. Wagner 等人[21]开发了一个基于恶意行为的分析系统 KAMAS. 该系统适用于复杂数据结构的视觉访问,能提供适当的视觉表示以及特定工作流程的交互技术,并将专家知识外化成规则形式以便简化分析流程.

为了有效地检测及分类恶意代码,本文提出了两种可视化分析方法. 这两种方法将恶意代码分析转化为图像间的比较,使用 VGGNet 卷积网络和支持向量机构成的深度融合网络来学习恶意代码生成的图像并分类.

2　相关的研究工作

本文着重研究恶意代码的字节特征. 在传统恶意代码分析方面,Sornil 等人[22]经过实验分析后认为 N-gram 方法能有效地描述恶意代码的字节特征. N-gram 特征是给定恶意样本中每相邻 n 个字节的序列模型,可以理解为一阶马尔科夫链,即字节序列的出现顺序是离散事件的随机过程. Tesauro 等人[23]利用神经网络来分类字节的 Tri-gram 特征,IBM 公司将该方法应用于反病毒扫描器的研究中. Abou-Assaleh 等人[24]针对两类数据集(恶意样本集:I-Worm,正常文件集:Win32),分析了 N-gram 特征及其长度 L 对恶意代码检测的影响,获得了最高 91% 的识别准确率.

在可视化分析方面,Conti 等人[25]提出了直接可视化二进制文件的方法,用独特的视角将字节值与像素的亮度值联系起来,以此来识别文件格式、发现其中异常. 该方法的优点在于,分析人员无需事先进行反汇编操作即可观察到文件内部的情况,相较于十六进制编辑器只用文本形式指明字节值而言,该方法利用像素点的明暗差异变化使分析更为直观. 但在恶意代码检测和分类的应用场景中,该方法还存在以下不足.

(1)用于文件自身或文件间比较时,极大地依赖机器的硬件性能. 例如 1MB 的文件会生成 1TB 的点阵图,尤其在处理大规模数据时,分析效率也受到制约.

(2)就字节层面而言,可打印字符是能直接获取且较重要的分析信息,而在其系统界面中将字符信息与字节可视化结果分开显示,不利于理解待测文件的特征全貌.

之后,Nataraj 等人[26]在 Conti 设计方案的基础上提出了利用恶意代码生成的灰度图纹理进行分类的方法,该方法将二进制文件的每 8 个数据位转换为像素的灰度值,以此将恶意文件转换成灰度图;随后,该方法提取灰度图的 Gist[27,28]纹理特征,运用 K-Nearest Neighbor (KNN)分类算法进行 10 倍交叉验证,取得了较高的分类正确率. Nataraj 的方法虽然在视觉上可以反映出同族恶意代码具有相似的图案纹理,也在分类方面获得了较优的成果,但在恶意样本分析时存在以下几个问题.

(1)该方法生成的灰度图与恶意样本原文件大小成正比,如果原文件数据量较大,则大型的灰度图文件会被系统误判为解压炸弹拒绝服务攻击(Decompression Bomb Dos Attack)[29],从而导致分析程序终止.

(2)该方法同样存在可打印字符的问题. 恶意代码中的可打印字符通过 URL 链接、文件操作位置等信息可以提示该样本的功能. 突出可打印字符可以便于分析人员定位.

为了解决上述问题,本文以同族代码的视觉表征相似、而异族代码的视觉表征差异较大为前提,基于字节的 N-gram 特征提出了两种新的研究恶意代码谱系分类的可视化方法,即基于空间填充曲线的恶意代码可视化方法及基于字节 2-gram 特征的恶意代码可视化方法,目的是从图像分析的角度挖掘恶意代码字节特征的新模式.

3　方法描述

如图 1 所示,方法 1 包括三个部分:字节序列归一化、空间填充曲线映射、图像特征学习与分类;方法 2 包括两个部分:点阵图可视化、图像特征学习与分类.

3.1　基于空间填充曲线的恶意代码可视化方法

3.1.1　字节序列归一化

不同于灰度图与原始恶意样本的数据量一致,方

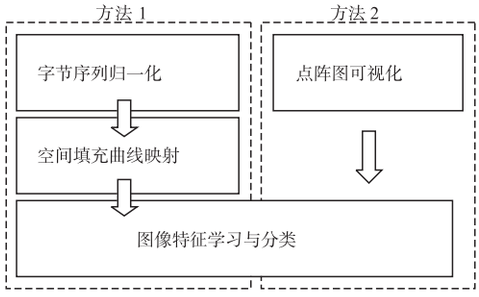


图1 所提的两种方法

法 1 通过等间隔缩放采样算法控制采样步长,使样本的字节序列长度固定为 65536. 为了突出可打印字符(0x20-0x7e)的重要性,将其用绿色通道标记;字节 0 为黑色,字节 255 为白色;其余字节用品红色系明暗值标记. 由此,将固定长度的字节序列转化为带颜色标记的像素点序列.

3.1.2 空间填充曲线映射

观察恶意代码间的相似度,不难将样本之间在形态学或者统计学上的相似性与数学中的分形理论联系起来. 分形描述了复杂的客观存在,不仅具有视觉吸引力,还能反映结构层次的逻辑性. 作为一种分形曲线,空间填充曲线(Space Filling Curve,SFC)[30]将离散的多维空间映射到一维空间,并能保持多维空间中基本单元的线性顺序不变,该性质将空间填充曲线分为递归和非递归两类. 递归填充曲线具有能够被迭代地等分成四个尺寸大小一致、形状相同子曲线的特点. 方法 1 采用以下六种曲线来遍历 256×256 分辨率的方图,其中递归曲线四种(Gray 曲线[31]、Hcurve 曲线[32]、Hilbert 曲线[33]、Zorder 曲线[34]),非递归曲线两种(Sweep 曲线[35]、Zigzag 曲线[36]),如图 2 所示.

3.1.3 图像特征学习与分类

本文使用迁移学习的概念(即运用已有知识对不同但相关领域的问题进行求解的学习方法),将恶意代码转化为图像,运用机器视觉领域中图像分类方法来处理恶意代码研究领域的分析问题. 就图像识别而言,有很多成功的案例,如深度神经网络,尤其是著名的卷积网络[37]. 深度卷积神经网络的优势在于拥有反馈机制可以自动调整滤波器使分类更精确;使用稀疏交互的方式来共享权值,解决了参数过多所导致的过拟合问题. 本文使用 VGGNet[38]卷积神经网络来提取图像特征. 为提升学习性能,该网络重复堆叠小型卷积核(3×3)和最大池化层(2×2)来增加网络深度. 由于本文要解决的恶意代码分类任务不同于 VGGNet 网络最初的应用需求,因此所提方法只迁移基本的网络结构和部分权重,并使用支持向量机分类器代替原先的全连接层和 softmax 分类器以此提高方法的泛化能力.

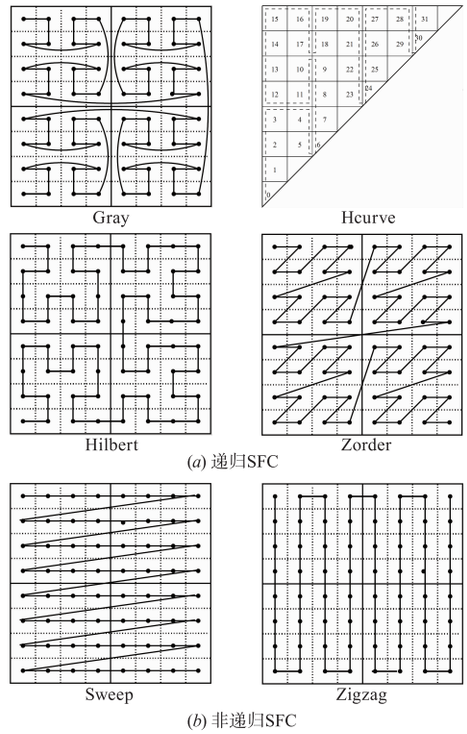


图2 六种空间填充曲线

3.2 基于字节 2-gram 特征的恶意代码可视化方法

“点阵图可视化”主要实现字节转化和亮度显示两方面的功能,即将每相邻两字节转化为像素点的 x,y 坐标,而将相同坐标点 $p(x,y)$ 的数量 N_p 归一化为[0,255]之间的整数 N_p',并用蓝色通道标记为. 方法 2 同样使用深度融合网络实现“图像特征的学习与分类”. 此外,这里给出两套用于比较验证的哈希着色策略,即分别用函数 djb2 和 SimHash 计算 N_p 的 24 位散列值,用以构成相应的 RGB 颜色向量.

4 实验结果分析

本文采用微软公开的恶意样本集[28]来评估和验证所提方法. BIG 2015 | Kaggle[39]的训练集包括了 10868 个来自 9 种恶意代码族的样本,每个恶意样本都有一个唯一识别的 ID(一个 20 字符的哈希值). 从各族样本的数量上可以看出,该数据集的分布很不均匀,见表 1.

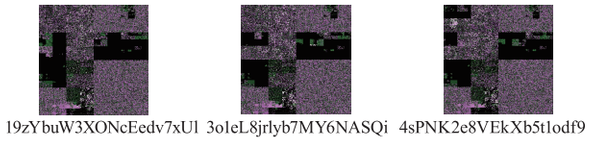
表 1 样本集的数量分布

1	Ramnit(1541)	2	Gatak(2478)	3	Obfuscator. acy(2942)
4	Kelihos_ver1(475)	5	Tracur(42)	6	Simda(751)
7	Vundo(398)	8	Kelihos_ver3(1228)	9	Lollipop(1013)

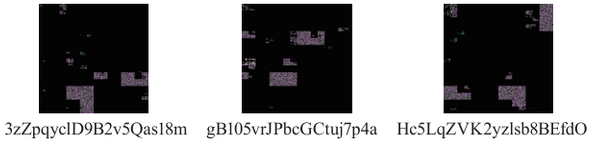
4.1 可视化结果的视觉分析

将方法 1 应用于微软恶意样本集,其可视化结果见图 3 图 4. 图 3 给出了 Gray 曲线映射后的图像示例,从

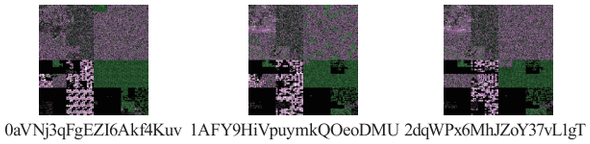
视觉感知上可以反映出,同族样本间的图像很相似而异族样本间的图像差异明显. 由此说明该方法具有较好的视觉区分能力,同时也为实现恶意样本检测与分类提供可能. 为节省篇幅,图 4 展示了图 3 中第 8 类示例样本对应的空间填充曲线映射图.



(*a*) 第1类Ramnit



(*b*) 第5类Tracur



(*c*) 第8类Kelihos_ver3

图3　使用Gray曲线映射后的图像示例

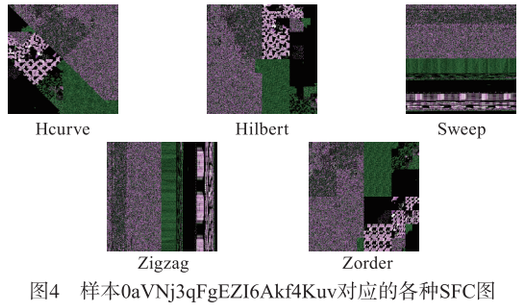


图4　样本0aVNj3qFgEZI6Akf4Kuv对应的各种SFC图

同样地,将方法 2 及哈希对比方案应用于该样本集,其可视化结果如图 5 所示. 在视觉表现上,哈希着色方案增强了图像的对比度;而单色方案可以采用变焦交互的方法提升视觉感知度. 此外,2-gram 特征仅与相邻两字节出现的频率有关,而与字节在原恶意文件中的具体位置无关,很好地解决了代码段重置的问题.

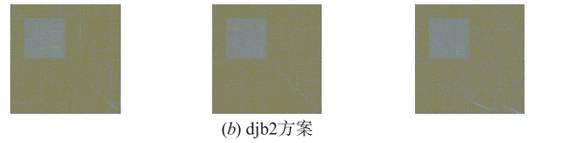
图 6 给出了灰度图方法的可视化结果. 分析图 3 ~ 6 后认为:(1)所提方法均能体现样本族的类内相似性;(2)就方法 1 而言,使用颜色标记可以比灰度图更直观地了解恶意代码内部的结构(如可打印字符),以便通过交互的方式直接定位到感兴趣的区域;(3)方法 2 可以显式地反映高频的字节对,为发现恶意代码的新特征提供依据.

4.2　分类性能比较

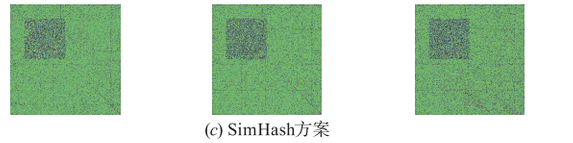
4.2.1　不同的图像特征对分类结果的影响

为了评估不同的图像特征对分类结果的影响,本

(*a*) 方法2: 单色方案



(*b*) djb2方案



(*c*) SimHash方案

图5　方法2与哈希着色方案的可视化结果

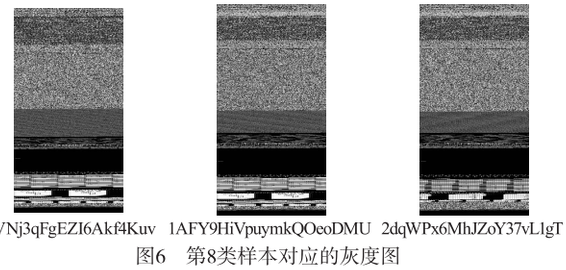


图6　第8类样本对应的灰度图

项实验使用最近邻分类器按欧氏测度对上述各方法生成的可视化结果进行分类. 表 2 分别比较了 Gist 特征和由卷积神经网络(ResNet50[40]、VGG16、VGG19)学习得到的特征. 当使用 Gist 特征时,本文所提的两种方法均优于灰度图方法本身;而使用卷积神经网络学习特征时,VGG16 网络使方法 2 获得了 98.45% 的最高分类正确率.

表 2　各方法在恶意代码分类方面的结果比较

%		Gist	Resnet50	VGG16	VGG19
方法 1	Gray	97.45	96.96	94.56	95.73
	Hcurve	97.07	96.86	93.50	94.82
	Hilbert	97.26	96.75	94.63	95.99
	Sweep	97.05	97.59	97.41	97.34
	Zigzag	97.28	97.68	97.21	97.36
	Zorder	97.18	96.98	94.73	95.88
灰度图		96.87	97.67	97.56	97.73
2-gram	方法 2	98.35	98.19	98.45	98.41
	djb2	95.48	95.82	94.35	94.88
	SimHash	94.09	93.80	91.01	91.14

就方法 1 而言,仅在使用 VGG 系列网络时分类性能才略微低于灰度图方法,分析其原因在于该方法压缩了数据量,保留的图像信息少于灰度图. 在实际分类

应用中,需要权衡正确率、数据量和硬件性能时,方法 1 可以用作备选方案. 就方法 2 而言,其各项分类结果均为最优,由此说明该恶意样本集中存在代码段重置或冗余数据填充的情况,同时也从另一个侧面反映了深度学习可以发现人类感官不易察觉的模式. 就哈希着色策略而言,其分类结果较差,原因在于:由相同坐标数量计算出的散列值是唯一的、绝对的;而方法 2 采用相对概念来归一化相同的坐标数量. 此外,与 Gist 特征相比,使用卷积网络更能提高灰度图的分类性能,尤其是采用 VGG19 网络时的分类正确率 97.73% 高于原方法的 96.87%.

4.2.2 SVC 算法和 LinearSVC 算法对分类的影响比较

本项实验使用深度融合网络来评估 SVC 算法[41]与 LinearSVC 算法[42]对分类结果的影响,结果如表 3 所示. 使用 LinearSVC 算法时,VGG16 网络的分类效果较好;而使用 SVC 算法时,也是 VGG16 网络获得了最优的分类正确率 99.08%(由方法 2 获得,图 7 给出了最优参数条件下 $C=100$、$\gamma=10^{-6}$的学习曲线). 总体来说,SVC 算法稍优于 LinearSVC 算法,这说明图像中存在少量线性不可分的特征.

表 3 使用深度融合网络的分类结果比较

%		Resnet50		VGG16		VGG19	
		Linear SVC	SVC	Linear SVC	SVC	Linear SVC	SVC
方法 1	Gray	96.99	97.69	97.41	97.53	97.34	97.54
	Hcurve	97.32	97.98	97.55	97.63	97.58	97.72
	Hilbert	97.23	97.87	97.40	97.64	97.40	97.59
	Sweep	97.73	98.07	97.98	98.36	98.02	98.22
	Zigzag	97.64	98.08	97.81	97.78	97.71	98.15
	Zorder	97.34	97.87	97.73	97.63	97.63	97.82
灰度图		97.87	98.11	98.03	98.46	98.21	98.45
2-gram	方法 2	98.74	98.81	99.07	99.08	99.02	98.97
	djb2	96.59	97.20	96.26	96.13	96.65	96.62
	SimHash	94.90	95.79	94.91	95.21	95.02	95.37

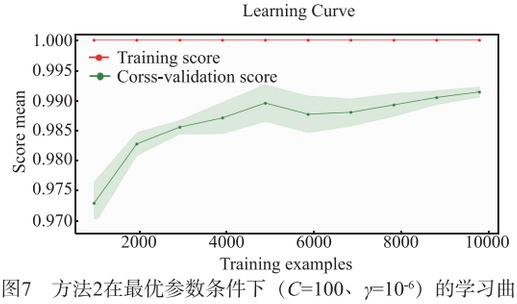


图7 方法2在最优参数条件下(C=100、γ=10^{-6})的学习曲线

4.3 恶意代码识别效果分析

本文从 Windows 系统上搜集了共计 9390 个正常的 PE 文件,包括. exe 可执行文件和. dll 动态链接库. 将这些文件按上述可视化方法生成对应的图像后,再与之前 10868 个恶意样本的可视化结果混合,以 Gist 特征为识别依据,按最近邻原则执行分类决策,获得的恶意代码检测结果见表 4. 从中不难发现,本文所提的两种方法在恶意代码识别检测中均优于灰度图方法,尤其是方法二的识别率达到了 98.31%. 这是因为 Gist 特征通过 5 种感知属性的 4 种比例等级对图像进行小波分解,而彩色图像较灰度图有更丰富的光谱信息,因此在经过离散傅里叶变换得到的特征向量能提供更全面且可靠的估计.

表 4 各方法在恶意代码检测方面的结果比较

	Gray	Hcurve	Hilbert
方法 1	98.24	98.07	98.14
	Sweep	Zigzag	Zorder
	96.85	97.09	98.14
灰度图	96.12		
2-gram	方法 2	djb2	SimHash
	98.31	97.63	97.08

之后,使用深度融合网络来比较所提方法与灰度图方法的检测性能. 如表 5 所示,Zorder 映射图取得了最高 99.21% 的识别率,而灰度图也获得了高于其原始方法的结果(98.99%). 由此说明:(1)深度融合网络的架构优于使用 Gist 特征与最近邻分类器的组合,这是因为卷积神经网络采用的卷积核远小于输入图像的尺寸,而 Gist 方法只是将图像划分为 4×4 个区块,因此卷积神经网络对图像的特征过滤比 Gist 方法更彻底;(2)卷积网络相同的情况下,SVC 算法略微优于最近邻

分类决策,这反映出可视化结果中含有少量线性不可分的特征,而 SVC 算法可以利用有效收敛的凸优化技术来学习非线性模型;(3)方法 1 在恶意代码检测方面的性能优于其他方法.

表 5　所提方法与灰度图方法的检测结果比较

%		Resnet50		VGG16		VGG19	
		NN	SVC	NN	SVC	NN	SVC
方法 1	Gray	97.86	98.45	97.30	98.92	97.55	99.08
	Hcurve	97.67	98.64	96.54	99.01	97.33	99.01
	Hilbert	97.78	98.64	97.19	99.04	97.42	99.07
	Sweep	98.19	98.82	97.81	98.98	98.24	99.06
	Zigzag	98.10	98.67	98.17	98.98	98.09	98.90
	Zorder	97.90	98.75	97.63	99.04	97.64	99.21
灰度图		98.05	98.64	98.25	98.99	98.30	98.97
2-gram	方法 2	97.09	98.45	97.10	98.74	97.01	98.45
	djb2	97.03	98.16	96.71	97.55	96.71	97.87
	SimHash	95.96	97.38	95.28	96.81	95.65	97.46

5　总结

本文提出了两种基于字节 N-gram 特征的恶意代码可视化分析方法,主要贡献体现在以下三个方面.

(1)在视觉表现方面,方法 1 突出了字节属性的差异,便于交互定位,且在视觉区分度上优于灰度图方法;方法 2 解决了恶意样本通过代码重置或增加冗余数据来改变全局图像特征的问题.

(2)在恶意代码检测与分类方面,两种方法均优于现有的灰度图方法. 其中,方法 1 在恶意代码检测方面的优势较明显,而方法 2 则在恶意代码分类时表现优异.

(3)在分析效率方面,两种方法均无须反汇编或沙箱运行,且能以程序自动化的方式操作,从而降低了对分析人员业务水平的要求. 此外,两种方法采用降采样或降维的方式大幅地减少了原始文件的数据量,缓解了硬件压力,统一了用于分析和识别的图像标准,有利于特征的批量提取,提高了工作效率.

参考文献

[1] LI J, WANG Z, WANG T, et al. An android malware detection system based on feature fusion[J]. Chinese Journal of Electronics, 2018, 27(6): 1206 – 1213.

[2] 张焕,武建亮,唐俊杰,等. NeighborWatcher:基于程序家族关系的附加恶意手机应用检测方法研究[J]. 电子学报,2014,42(8):1642 – 1646.
ZHANG Huan, WU Jian-liang, TANG Jun-jie, et al. Neighbor watcher: Detecting piggybacked smartphone applications with their family members[J]. Acta Electronica Sinica, 2014, 42(8): 1642 – 1646. (in Chinese)

[3] YAN H, ZHOU H, ZHANG H. Automatic malware classification via PRICoLBP[J]. Chinese Journal of Electronics, 2018, 27(4): 852 – 859.

[4] 乔延臣,云晓春,张永铮,等. 基于调用习惯的恶意代码自动化同源判定方法[J]. 电子学报,2016,44(10):2410 – 2414.
QIAO Yan-chen, YUN Xiao-chun, ZHANG Yong-zheng, et al. An automatic malware homology identification method based on calling habits[J]. Acta Electronica Sinica, 2016, 44(10): 2410 – 2414. (in Chinese)

[5] RANVEER S, HIRAY S. Comparative analysis of feature extraction methods of malware detection[J]. International Journal of Computer Applications, 2015, 120(9): 1 – 7.

[6] GANDOTRA E, BANSAL D, SOFAT S. Malware analysis and classification: A survey[J]. Journal of Information Security, 2016, 5(2): 56 – 64.

[7] WEI Z, NADJIN Y. Malwarevis: Entity-based visualization of malware network traces[A]. Proceedings of the Ninth International Symposium on Visualization for Cyber Security[C]. USA: ACM, 2012. 41 – 47.

[8] TRINIUS P, HOLZ T, GÖBEL J, et al. Visual analysis of malware behavior using treemaps and thread graphs[A]. International Workshop on Visualization for Cyber Security[C]. USA: IEEE, 2010. 33 – 38.

[9] GOVE R, SAXE J, GOLD S, et al. SEEM: A scalable visualization for comparing multiple large sets of attributes for malware analysis[A]. Eleventh Workshop on Visualization for Cyber Security[C]. USA: ACM, 2014. 72 – 79.

[10] HAN K S, LIM J H, KANG B, et al. Malware analysis using visualized images and entropy graphs[J]. International Journal of Information Security, 2015, 14(1): 1

– 14.
[11] STRELKOV V V. A new similarity measure for histogram comparison and its application in time series analysis[J]. Pattern Recognition Letters, 2008, 29(13): 1768 – 1774.
[12] REN Z, CHEN G. Entropy vis: Malware classification [A]. The 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)[C]. USA: IEEE, 2017. 1 – 6.
[13] YOO I S. Visualizing windows executable viruses using self-organizing maps[A]. Workshop on Visualization and Data Mining for Computer Security[C]. USA: ACM, 2004. 82 – 89.
[14] HAN K S, LIM J H, IM E G. Malware analysis method using visualization of binary files[A]. Research in Adaptive and Convergent Systems[C]. USA: ACM, 2013. 317 – 321.
[15] HAN K, KANG B, IM E G. Malware analysis using visualized image matrices[J/OL]. The Scientific World Journal, 2014. http://dx. doi. org/10. 1155/2014/132713.
[16] PATURI A, CHERUKURI M, DONAHUE J, et al. Mobile malware visual analytics and similarities of attack toolkits (malware gene analysis)[A]. International Conference on Collaboration Technologies and Systems[C]. USA: IEEE, 2013. 149 – 154.
[17] ANDERSON B, STORLIE C, LANE T. Improving malware classification: bridging the static/dynamic gap[A]. Workshop on Security and Artificial Intelligence[C]. USA: ACM, 2012. 3 – 14.
[18] HASHEMI H, HAMZEH A. Visual malware detection using local malicious pattern[J/OL]. Journal of Computer Virology and Hacking Techniques, 2018. https://doi.org/10. 1007/s11416-018-0314-1.
[19] ZHANG Y, et al. Visual analysis of android malware behavior profile based on PMC gdroid: A pruned lightweight APP call graph[A]. Security and Privacy in Communication Networks[C]. Berlin: Springer, 2017. 449 – 468.
[20] ANGELINI M, ANIELLO L, LENTI S, et al. The goods, the bads and the uglies: Supporting decisions in malware detection through visual analytics[A]. Symposium on Visualization for Cyber Security[C]. USA: IEEE, 2017. 1 – 8.
[21] WAGNER M, RIND A, THÜRN, et al. A knowledge-assisted visual malware analysis system: Design, validation, and reflection of KAMAS[J]. Computers & Security, 2017, 67: 1 – 15.
[22] SORNIL O, LIANGBOONPRAKONG C. Malware classification using n-grams sequential pattern features[J]. International Journal of Information Processing & Management, 2013, 4(5): 59 – 67.
[23] TESAURO G J, KEPHART J O, SORKIN G B. Neural networks for computer virus recognition[J]. IEEE Expert, 1996, 11(4): 5 – 6.
[24] ABOU-ASSALEH T, CERCONE N, KESELJ V, et al. Detection of new malicious code using n-grams signatures [A]. Proceedings of the Conference on Privacy, Security and Trust(DBLP)[C]. New Brunswick, Canada: DBLP, 2004. 193 – 196.
[25] CONTI G, DEAN E, SINDA M, et al. Visual reverse engineering of binary and data files[A]. Proceedings of International Workshop on Visualization for Computer Security [C]. USA: DBLP, 2008. 1 – 17.
[26] NATARAJ L, KARTHIKEYAN S, JACOB G, et al. Malware images: visualization and automatic classification [A]. International Symposium on Visualization for Cyber Security[C]. USA: ACM, 2011. 1 – 7.
[27] DOUZE M, SANDHAWALIA H, AMSALEG L, et al. Evaluation of GIST descriptors for web-scale image search [A]. ACM International Conference on Image and Video Retrieval[C]. USA: ACM, 2009. 1 – 8.
[28] OLIVA A, TORRALBA A. Modeling the shape of the scene: A holistic representation of the spatial envelope [J]. International Journal of Computer Vision, 2001, 42 (3): 145 – 175.
[29] AERAsec. Decompression Bomb Vulnerabilities[OL]. http://www. aerasec. de/security/advisories/decompression-bomb-vulnerability. html. 2018.
[30] LIAO S, LOPEZ M A, LEUTENEGGER S T. High dimensional similarity search with space filling curves[A]. International Conference on Data Engineering[C]. USA: IEEE Computer Society, 2001. 615 – 622.
[31] MOKBEL M F, AREF W G. Irregularity in high-dimensional space-filling curves[J]. Distributed & Parallel Databases, 2011, 29(3): 217 – 238.
[32] NIEDERMEIER R, REINHARDT K, SANDERS P. Towards optimal locality in mesh-indexings[A]. International Symposium on Fundamentals of Computation Theory[C]. Berlin: Springer-Verlag, 1997. 364 – 375.
[33] DAI H K, SU H C. Approximation and analytical studies of inter-clustering performances of space-filling curves [A]. Discrete Random Walks (Drw'03)[C]. Paris, France: DBLP, 2003. 53 – 68.
[34] SCHRACK G, STOCCO L. Generation of spatial orders and space-filling curves[J]. IEEE Transactions on Image Processing, 2015, 24(6): 1791 – 800.
[35] MOKBEL M F, AREF W G. Space-filling curves for query processing[A]. Encyclopedia of Database Systems [M]. US: Springer, 2009. 2675 – 2680.
[36] ANOTAIPAIBOON W, MAKHANOV S S. Curvilinear

space-filling curves for five-axis machining[J]. Computer-Aided Design,2008,40(3):350 – 367.

[37] LECUN Y,BOSER B,DENKER J S,et al. Backpropagation applied to handwritten zip code recognition[J]. Neural Computation,2014,1(4):541 – 551.

[38] SIMONYAN K,ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J/OL]. Computer Science,2014,arXiv:1409.1556.

[39] Microsoft. Microsoft Malware Classification Challenge (BIG 2015)[OL]. https://www.kaggle.com/c/malware-classification/data.2018.

[40] HE K,ZHANG X,REN S,et al. Deep residual learning for image recognition[A]. IEEE Conference on Computer Vision and Pattern Recognition[C]. USA:IEEE Computer Society,2016.770 – 778.

[41] GUYON I,BOSER B E,VAPNIK V. Automatic capacity tuning of very large VC-dimension classifiers[J]. Advances in Neural Information Processing Systems,2008,5:147 – 155.

[42] CHIANG W L,LEE M C,LIN C J. Parallel dual coordinate descent method for large-scale linear classification in multi-core environments[A]. The ACM SIGKDD International Conference[C]. USA:ACM,2016.1485 – 1494.

作者简介

任卓君　女,1984 年 2 月出生,浙江湖州人. 东华大学在读博士生,从事网络安全、恶意代码可视化方面的研究.
E-mail:1129110@mail.dhu.edu.cn

陈　光　男,1958 年 5 月出生,广东汕头人. 华东理工大学博士,东华大学通信专业教授,中国通信协会高级会员,上海市电子电器协会理事.　E-mail:gchen@dhu.edu.cn

卢文科　男,1962 年 5 月出生,陕西西安人. 西安交通大学博士,东华大学控制科学与工程学科教授、博导,主要从事小波变换相关技术的研究.　E-mail:luwenke3@163.com