基于行为关系网络的恶意代码检测方法

刘建松，张 磊，方 勇
(四川大学网络空间安全学院，成都 610065)

摘 要：在网络安全领域，恶意代码的威胁是一个不可回避的话题. 如何快速检测出恶意代码、阻止和降低恶意代码产生的危害一直是亟需解决的问题. 本文提出一种基于行为关系网络的恶意代码检测方法. 首先，在沙箱中运行样本获得行为报告，再从报告中提取样本的 API 调用、注册表访问和文件读写操作三种行为记录来构建行为关系网络，所构建的行为关系网络包含“PE”、“API”、“Registry”和“File”4 种类型的节点；然后，使用一种基于元图的方法来计算样本之间的相似度矩阵；最后，使用一种自定义核的支持向量机(Support Vector Machine，SVM)模型来进行训练和预测. 实验结果表明，本文提出的方法可以达到 95.5%的分类准确率，能够有效地对恶意代码进行检测.

关键词：恶意代码；行为关系网络；异质信息网络；元图；支持向量机
中图分类号：TP391.1 **文献标识码**：A DOI：10.19907/j.0490-6756.2022.023001

Malicious code detection method based on behavior relation network

LIU Jian-Song, *ZHANG Lei*, *FANG Yong*
(School of Cyber Science and Engineering, Sichuan University, Chengdu 610065, China)

Abstract: In the field of network security, the threat of malicious code is an unavoidable topic. How to quickly detect malicious code, prevent and reduce the harm caused by malicious code has always been an urgent problem. This paper proposes a malicious code detection method based on the behavior relation network. First, obtain the behavior report by executing the sample in the sandbox, and then construct a behavior relationship network by extracting the three behavior records of the sample's API call, registry access, and file read and write operations from the behavior report . The constructed behavior relationship network includes "PE", "API", "Registry" and "File" 4 types of nodes, we then use a meta-graph-based method to calculate the similarity matrix between samples, and finally the Support Vector Machine (SVM) model, which kernel is custom defined, is used for training and prediction. Experimental results show that the method proposed in this paper can achieve a classification accuracy of 95.5% and can effectively detect malicious code.

Keywords: Malware detection; Behavior ralation network; Heterogeneous information network; Meta-graph; SVM

收稿日期：2021-11-04
基金项目：国家自然科学基金(U20B2045)
作者简介：刘建松(1997—)，男，四川宜宾人，硕士研究生，研究方向为恶意代码检测.
通讯作者：张磊. E-mail：zhanglei2018@scu.edu.cn

1 引　言

恶意代码是一种能够自我激活、可自动执行，并对互联网用户产生巨大威胁的应用程序，常见的有病毒、木马、蠕虫、间谍软件和特洛伊木马等. 近几年来，勒索软件和挖矿木马开始大肆感染用户的设备. 由于利益驱使攻击者会通过各种手段将挖矿程序植入受害者的计算机中，在受害者不知道的情况下利用云计算进行挖矿，从而获取利益. 这些活动都会造成计算机、移动设备及人员财产等受到影响. 根据 2020 年中国网络安全报告[1]，2020 年瑞星“云安全”系统共截获病毒样本总量 1.48 亿个，病毒感染次数 3.52 亿次，病毒总体数量比 2019 年同期上涨 43.71%. 报告期内，木马病毒新增 7728 万个，为第一大种类病毒；排名第二的为蠕虫病毒，数量为 2981 万个. 感染型病毒、灰色软件和后门等分别占到总体数量的 12.19%、9.59%和 3.75%. 除此以外，还包括漏洞攻击和其他类型病毒. 因此，恶意代码的爆发式增长威胁着当今世界的互联网安全.

现有的恶意代码自动化分析技术大多都是基于其样本文件本身的内容，即静态特征或者自身的动态行为特征来做进一步分析的. 比如，Liu 等[2]直接提取 PE 文件的头信息和节信息作为特征，Kolter 等[3]从文件中提取 n-gram 字节码作为特征. 但以字节 n-gram 作为特征的分类器非常容易产生过拟合. 部分安全研究员认为，恶意代码汇编指令序列比单纯的字节流更能表达恶意代码的静态特点. Canfora 等[4]先将可执行程序反汇编，再提取汇编指令序列作为特征进行分析. 随着深度学习的发展，一些深度学习的方法也被应用到恶意代码检测领域. Pascanu 等[5]借鉴自然语言处理中的建模方法，探索使用回声状态网络和递归神经网络讲汇编指令 N-gram 映射到低维向量的方法，再用逻辑回归分类器对向量进行最终分类. 也有将程序的源代码[6]或者 API 序列[7,8]看成语义丰富的文本，再基于现有的 NLP 模型进行改进来训练和分类的. 恶意代码可视化是近年才出现的恶意代码分析新方法. 2011 年 Nataraj 等[9]首次提出了完备的恶意代码样本图像化方法. 他们首先将恶意可执行程序按字节转化为灰度图，然后将灰度图的 GIST 特征作为样本特征，使用支持向量机对样本进行分类. Yuan 等[10]提出一种将恶意代码可执行程序转换为马尔可夫图像，然后使用深度卷积神经网络自动从图像中提取特征进行恶意代码家族分类的方法.

在恶意代码分析领域，存在一些非欧式结构的数据，比如函数依赖图、API 调用图和数据流图，传统的神经网络难以直接处理这种不规则的数据. 为解决这些问题，一些研究人员[11]选择先通过各种手段对图进行标准化处理，转换到统一的规格后再输入神经网络进行训练. 除此之外，图卷积神经网络(Graph Convolutional Network，GCN)[12]的提出给图数据的处理带来新的思路. 但是 GCN 只能处理节点类型单一的同质图，无法处理包含多种节点类型的异质信息网络. 2017 年 Hou 等[13]利用 APP 和不同种类的 API 调用来构建异质信息网络，然后使用基于元路径的方法计算 APP 之间的相似性，次年又将其应用到 Windows 恶意程序检测中[14]. 2019 年 Wang 等[15]利用系统事件数据构建异质信息网络，包含进程、文件和套接字 3 种不同的实体，可以捕获程序的行为，提出 MatchG-Net 模型，能够基于程序行为的不变图模型得到图嵌入向量，再进行未知软件和正常软件的相似度判断，低于一定阈值的都归为恶意软件.

针对同质图无法融合异构特征的问题，受文献[13]的启发，本文利用样本运行期间的行为记录来构建一种系异质信息网络的行为关系网络，将恶意代码的多种敏感行为结合分析，融合样本的多种行为特征. 本文选取恶意代码中常常出现的几种敏感行为作为特征，如关键 API 调用、注册表访问和文件读写操作. 本文提出的方法中，首先在沙箱中执行收集的恶意样本，然后从沙箱报告中提取样本的三种行为记录，利用这些行为信息来构建样本行为关系网络，再使用一种基于元图的方法来计算样本之间的相似度，最后构造 SVM 分类器用于恶意代码检测. 本文的主要贡献如下：(1) 在从沙箱报告中提取的 API 调用记录、注册表访问记录和文件操作记录，构建文件行为关系网络. 所构建的行为关系网络中包含 PE 文件、API、注册表和普通文件 4 种节点类型，3 种行为本身作为边，可将样本期间的多种敏感行为结合起来，并且在合适的方法下能够自然地融合这些动态特征. (2) 不同于文献[13]提出的基于异质信息网络的安卓恶意程序检测方法——基于元路径计算安卓程序之间的相似性，本文使用一种基于元图的方法计算 PE 文件间的相似性，通过计算元图上的交换矩阵来反应 PE 文件间的相似度，最后使用一种自定义核的支持向

量机(Support Vector Machine，SVM)模型来进行训练和预测.(3) 在公开的数据集上对所提模型进行验证，达到 95.5%的分类准确率，优于基于元路径的方法.

2　研究方法

2.1　方法概述

本文提出基于行为关系网络恶意代码检测方法的分类模型框架如图 1 所示，包括沙箱执行、文件行为关系网络构建、相似性矩阵计算和分类与检测模型. 通过在沙箱中执行恶意代码样本得到其行为报告，并从报告中提取 API 调用、注册表访问和文件操作 3 种行为记录，再根据这些行为信息构建文件行为关系网络；然后使用基于元图的方法计算相似性矩阵；而后将各相似性矩阵的线性组合作为最后 SVM 的核矩阵，进行分类与训练；最后将训练好的模型用于恶意代码检测.

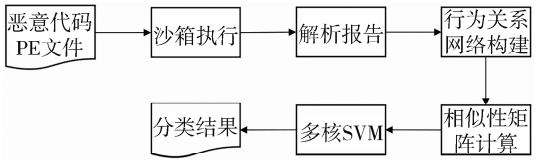

图 1　分类模型框架

Fig. 1　The frame of the classification model

2.2　行为关系网络的构建

定义 1　异质信息网络[16](Heterogeneous Information Network，HIN). 定义为一个具有实体类型映射函数 $\varphi: V \to A$ 和关系类型映射函数 $\psi: E \to R$ 的有向图 $G=(V,E,\varphi,\psi)$. 其中，每个实体 $v \in V$ 属于实体类型集合 A：$\varphi(v) \in A$ 中的一个特定实体类型，每条链接 $e \in E$ 属于关系类型集合 R：$\psi(e) \in R$ 中的一个特定关系类型，并且满足实体类型数 $|A|>1$ 或者关系类型数 $|R|>1$.

本文构建的文件行为关系网络是一种异质信息网络，也叫做异质图. 异质图是相对于同质图而言的，同质图只包含一种类型的节点和边，异质图包含多种类型的节点或者边. 相比于同质图而言，异质图可以更好地反应实体间复杂的关系，可以保留更全面的语义及结构信息. 近年来，异质图迅速成为数据挖掘等领域中的研究热点，常见的有社交网络、科技文献和医疗系统，常被应用于分类、聚类、推荐等. 本文构建的文件行为关系网络包含 4 种节点类型和 3 种边类型，属于异质信息网络.

本文首先在 Cuckoo 沙箱中执行恶意样本得到其行为报告，然后从沙箱报告中的 report.json 中提取 API 调用记录，再解析 logs 目录下的文件得到注册表访问记录和文件读写操作记录，构建文件行为关系网络. 所构建的文件关系网络如图 2 所示，包含了“PE”、“API”、“Registry”和“File”4 种类型的节点和 $PE \xrightarrow{\text{calls}} API$、$PE \xrightarrow{\text{accessed}}$ registry、$PE \xrightarrow{\text{r/w}}$ file 3 种行为作为边. 构建行为关系网络为下一步计算 PE 文件相似度做好准备.

定义 2　网络模式[16](network schema). 记为 $T_G=(A,R)$，是带有实体类型映射 $\varphi: V \to A$ 和关系类型映射 $\psi: E \to R$ 的信息网络 $G=(V,E)$ 的元模式. 具体地，网络模式是定义在实体类型集合 A 上的有向图，并以 R 上的关系为边. 网络模式更直观的反应了行为网络结构，本文的恶意代码网络模式图如图 3.

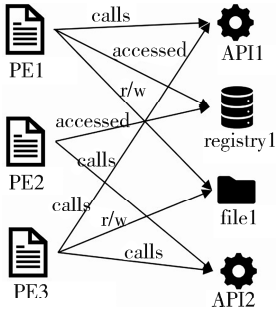

图 2　本文构建的行为关系网络

Fig. 2　Behavior relation network

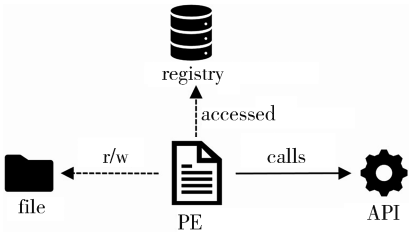

图 3　恶意代码行为网络模式图

Fig. 3　Network schema

定义 3　元路径[16]. 元路径 P 是在网络模式 $T_G=(A,R)$ 上定义的路径，记为 $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$. 同时，定义实体 $A_1, A_2, \cdots, A_{l+1}$ 间的复合关系 $R=R_1 \cdot R_2 \cdot \cdots \cdot R_l$，其中 $\cdot$ 表示关系上的合成运算符.

在异质信息网络中，元路径常常用来定义实体间的关系. 如图 4 所示，元路径 M_1：$PE \xrightarrow{\text{calls}} API \xleftarrow{\text{calls}} PE$，让调用过同一 API 的 PE 文件相关联；元路径

$M_2: PE \xrightarrow{\text{accessed}} \text{registry} \xleftarrow{\text{accessed}} PE$，让访问过同一注册表项的 PE 文件相关联. 文献[13]提出的基于异质信息网络的安卓恶意程序方法，是基于元路径去计算的安卓 app 之间的相似度. 但是元路径不能让同时调用过同一 API 和访问过同一注册表项的 PE 文件相关联. 元图就解决了这一痛点. 元图的定义如下.

定义 4　元图[17]. 元图是一个有向无环图，包含一个源节点(入度为 0)和一个目标点 n_s 和一个目标点 n_t. 因此在异质信息网络 $G=(V,E,\varphi,\psi)$ 和网络模式 $T_G=(A,R)$ 基础上定义元图为 $M=(V_M,E_M,T_M,R_M,n_s,n_t)$，其中 $V_M\subseteq V, E_M\subseteq E$，并约束 $A_M\subseteq A$ 和 $R_M\subseteq R$.

元图是 Zhao 等[17]提出的概念，用于刻画实体间的复杂的信息共享关系. 如图 4 所示，元图 M_4 就能让同时调用过同一 API 和访问过同一注册表项的 PE 文件相关联. 所以本文使用基于元图的方法去计算 PE 文件之间的相似度. 本文构建的行为关系网络中抽取出的元图如图 4 所示，元路径可以看作特殊的元图.

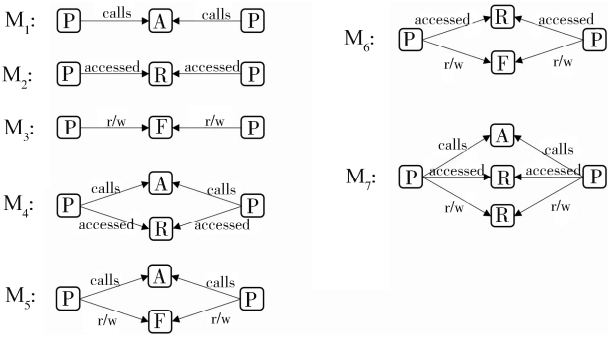

图 4　七种元图

Fig. 4　The seven types of meta-graph

2.3　相似性矩阵计算

定义 5　交换矩阵[16]. 给定一个异质信息网络图 $G=(V,E,\varphi,\psi)$ 和网络模式 $T_G=(A,R)$. 假设有一条元路径 $(A_1,A_2,\cdots,A_l)$，交换矩阵定义为 $C=W_{A_1A_2}W_{A_2A_3}\cdots W_{A_lA_{l+1}}$，其中 $W_{A_iA_j}$ 是实体 A_i 和 A_j 间的邻接矩阵.

在异质信息网络中，交换矩阵通常被用来当作目标节点之间的相似矩阵，由元路径上各实体间的邻接矩阵定义. 例如，图 4 中 M_1 元路径的交换矩阵计算公式为 $C_{M_1}=W_{PA}\cdot W_{PA}^{\top}$，其中 W_{PA} 为 PE 文件与 API 调用间的邻接矩阵，邻接矩阵的行向量表示一个 PE 文件调用过的所有 API. 本文一共涉及 3 个邻接矩阵，见表 1.

表 1　各邻接矩阵的定义

Tab. 1　Description of each matrix

邻接矩阵	元素	定义
W_{PA}，记为 A	a_{ij}	若 PE_i 调用了 API_j，则 $a_{ij}=1$，否则 $a_{ij}=0$
W_{PR}，记为 R	r_{ij}	若 PE_i 访问过 $Registry_j$，则 $r_{ij}=1$，否则 $r_{ij}=0$
W_{PF}，记为 F	f_{ij}	若 PE_i 访问过 $File_j$，则 $f_{ij}=1$，否则 $f_{ij}=0$

而元图的交换矩阵却没有直接的定义. 我们采用一种基于计数的方法来计算元图的交换矩阵. 算法的核心思想为，在分支节点处，遍历所有分支，把每条分支看成新的元路径分别计算其交换矩阵，再将所有分支的交换矩阵的哈达玛积作为分支节点最后的交换矩阵，剩余的计算步骤同计算元路径交换矩阵. 算法的关键就在于采用哈达玛积去合并分支节点的交换矩阵. 例如，图 4 中 M_5 元图的交换矩阵计算伪代码如下.

算法 1　计算 M_5 元图的交换矩阵

输入：PE 与 API 的邻接矩阵，W_{PA}；PE 与文件的邻接矩阵，W_{PF}；

输出：M_5 元图的交换矩阵，C_{M_5}；

1) $C_{P_1}=W_{PA}\cdot W_{PA}^{\top}$；

2) $C_{P_1}=W_{PF}\cdot W_{PF}^{\top}$

3) $C_{M_5}=C_{P_1}\odot C_{P_2}$

4) return C_{M_5}

2.4　分类与检测模型

上文计算得到的相似性矩阵维度跟样本的数量有关，在样本数目较多的情况下，相似性矩阵的维度会比较大，SVM[18]可以在数据维度比较高的情况下保持良好的泛化能力. 而且 SVM 可以自定义设置核函数，通过核函数，SVM 可以将特征向量映射到更高维的空间中，使得原本线性不可分的数据在映射之后的空间中变得线性可分. 本文采用一种自定义的核矩阵[13]来作为 SVM 的输入. $C=\sum_{k}^{K}\beta_k C_{M_k}$，且 $\beta_k\geqslant 0$，$\sum_{k=1}^{K}\beta_k=1$，其中 C_{M_k} 为各元图的交换矩阵，自定义的核矩阵为各元图的交换矩阵的线性组合.

检测阶段要预测未知样本时，将未知样本先与训练集构造异质信息网络，然后求得它与训练数据集的交换矩阵，最后把该交换矩阵输入到已训练好的模型中进行检测.

3　实验结果和分析

3.1　实验数据集

实验数据中的恶意代码样本是从开源恶意代码库 VirusShare[19] 下载，正常样本来源于 Windows 操作系统中的正常软件. 为了证明恶意样本和正常样本的纯净，上传样本到 VirusTotal[20] 进行检测，剔除恶意样本中的正常样本和正常样本中的恶意样本. 在 Cuckoo 沙箱中运行样本获取样本分析报告，通过解析 Cuckoo 沙箱报告文件——report.json 包含的字典中“behavior”子项，获得各个进程下的 API 调用信息；解析 logs 路径下的文件，获得进程访问过的文件和注册表信息. 由于部分样本对运行环境敏感，存在反虚拟机、反沙箱行为，所以并未在分析环境中成功运行，在分析报告中并未捕获到恶意行为，因此剔除这类样本. 80% 的样本用于模型训练，20% 的样本用于测试. 筛选后的数据集样本空间如表 2 所示. 本文的训练样本集和测试样本集中共检测到 183 个 API，1526 多个注册表项和 571 文件.

表 2　样本分布

Tab. 2　Sample distribution

样本类别	训练样本集	测试样本集
恶意样本	1037	259
正常样本	1016	254
合计	2053	513

3.2　实验结果分析

本文使用准确率、误报率、漏报率作为评估指标，这些指标涉及机器学习中的四个检验指标：真阳性（TP）、真阴性（TN）、假阳性（FP）和假阴性（FN），各指标含义如表 3.

表 3　检验指标的含义

Tab. 3　Meaning of the inspection index

检验指标	TP	TN	FP	FN
样本类型	恶意	正常	正常	恶意
检测结果类型	恶意	正常	恶意	正常

准确率（ACC）、误报率（FPR）、漏报率（FNR）定义如下.

$$\mathrm{ACC}=\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$$

$$\mathrm{FPR}=\frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}$$

$$\mathrm{FNR}=\frac{\mathrm{FN}}{\mathrm{TP}+\mathrm{FN}}$$

为了证明本文所提基于文件行为关系网络的恶意代码检测方法的有效性，我们进行了一系列实验. 首先是基于单个元图（包括元路径）和多元图融合后的检测性能评估，实验结果如表 4 所示.

从表 4 可见，首先从基于元路径 M_1、M_2、M_3 的分类结果来看，基于 API 调用行为的元路径 M_1 的表现结果最好，达到了 90.8% 的准确率；基于注册表访问的元路径 M_2 检测效果次之，有 85.2% 的准确率；基于文件操作的元路径 M_3 的表现结果最差，准确率只有 81.5%. 但总体来说，基于单条元路径的分类结果都不是很好，说明以上的单个行为特征都不足以准确刻画恶意代码行为模式.

表 4　检测性能评估

Tab. 4　Detection performance evaluation

Meta-graphs	准确率/%	误报率/%	漏报率/%
M_1	90.8	9.1	9.3
M_2	85.2	13.4	16.2
M_3	81.5	17.3	19.7
M_4	91.8	7.9	8.5
M_5	91.0	8.7	9.3
M_6	84.4	14.6	16.6
M_7	93.4	5.9	7.3
MKL	95.5	4.7	4.2

再把元图 M_4、M_5、M_6、M_7 与元路径 M_1、M_2、M_3 的分类结果作对比，检测效果都得到了有效的提升. 这是因为元图能够融合两个或者多个的行为特征. 比如，元图 M_4、M_5、M_6 分别融合了其中的两个特征. 除了 M_6，融合了两个特征的元图都比对应的单一特征的元路径的分类效果要好. 而元图 M_6 检测效果不如元路径 M_2 的原因可能是在此数据集上，这两个特征存在一些过拟合. 但融合了三个特征的元图 M_7 表现结果非常得好，准确率达到了 93.4%，同时误报率为 5.9%，漏报率为 7.3%. 对比结果表明，基于单元图的检测效果优于基于元路径的.

表 4 的最后一行是所有元图（包含 M_1、M_2、M_3）融合后的分类结果，结果表明基于元图融合的检测效果比所有单元图的分类检测效果都要好，准确率达到了 95.5%，误报率为 4.7%，漏报率为 4.2%，说明了基于元图融合的方法比基于单元图的更有效.

本文还将文献[13]提出的基于元路径融合的方法应用到本文的恶意代码数据集上,并与本文提出的方法做了实验对比,实验结果见表 5.

表 5　检测性能评估

Tab. 5　Detection performance evaluation

方法	准确率/%	误报率/%	漏报率/%	F_1-Score/%
融合元路径[13]	93.8	4.7	7.7	93.7
本文方法	95.5	4.7	4.2	95.6

如表 5 的实验结果所示,基于元路径融合的方法的准确率为 93.8%,误报率为 4.7%,漏报率为 7.7%,F_1-score 分数为 93.7%. 而本文提出的基于元图融合的方法准确率为 95.5%,误报率为 4.7%,漏报率为 4.2%,F_1-score 分数为 95.6%. 除了误报率相同,其余各指标均优于基于元路径融合的方法. 这说明本文所提出的基于元图融合的方法优于基于元路径融合的方法. 这是因为元图可以自然融合多个特征. 本文的方法就是元图自然融合了 API 调用、注册表访问和文件读写三种行为特征,可以更准确全面地捕获恶意代码的行为模式,从而具有更好的区分能力.

4　结　论

针对同质图无法融合多行为特征的问题,本文提出了一种新颖的基于文件行为关系网络的恶意代码检测方法. 首先在 Cuckoo 沙箱中执行恶意代码样本得到其行为报告,并从报告中提取 API 调用、注册表访问和文件读写操作三种行为记录,再根据这些行为信息构建文件行为关系网络. 所构建的行为关系网络中包含 PE"、"API"、"Registry"和"File"4 种类型的节点和 API 调用、注册表访问和文件读写三种类型的边,属异质信息网络. 不同于传统的同质图的处理方法,本文使用一种基于元图的方法——基于元图计算 PE 文件的相似矩阵. 在最后的 SVM 分类模型中,使用一种自定义的核矩阵,结合多个元图. 实验结果表明,本文提出的方法能有效检测恶意代码.

本文提出的方法,选取了 API 调用、注册表访问和文件操作信息三种行为记录构建行为关系网络. 在下一步研究中,可以尝试选取更多的行为特征,比如所调用的 dll,恶意代码所在的主机信息等重要信息,可以更全面准确地刻画恶意代码行为模式. 并且本文构建的行为关系网络中,没有包含 API 序列这种重要的语义信息,恶意代码的恶意行为都是通过底层调用的 API 来实现. 因此通过动态提取的 API 序列特征可以很好地刻画恶意样本的行为. 这也是未来研究中需要重点解决的问题.

参考文献:

[1] 瑞星. 2020 年中国网络安全报告[EB/OL]. (2021-01-14)[2021-03-04]. http://it.rising.com.cn/dongtai/19747.html.

[2] Liu B X, Zhao G, Sun R Y. A deep reinforcement learning malware detection method based on PE feature distribution[C]//Proceedings of the 2019 6th International Conference on Information Science and Control Engineering (ICISCE). Shanghai: IEEE, 2019.

[3] Kolter J Z, Maloof M A. Learning to detect and classify malicious executables in the wild[J]. J Mach Learn Res, 2006, 7: 2721.

[4] Canfora G, Lorenzo A D, Medvet E, *et al*. Effectiveness of opcode ngrams for detection of multi family android malware[C]//Proceedings of the 2015 10th International Conference on Availability, Reliability and Security. Toulouse: IEEE, 2015.

[5] Pascanu R, Stokes J W, Sanossian H, *et al*. Malware classification with recurrent networks[C]//Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). South Brisbane: IEEE, 2015.

[6] Amin M, Tanveer T A, Tehseen M, *et al*. Static malware detection and attribution in android byte-code through an end-to-end deep system[J]. Future Gener Comp Sy, 2020, 102: 112.

[7] 张涛, 王俊峰. 基于文本嵌入特征表示的恶意软件家族分类[J]. 四川大学学报:自然科学版, 2019, 56: 441.

[8] 黄琨茗, 张磊, 赵奎, 等. 基于最长频繁序列挖掘的恶意代码检测[J]. 四川大学学报:自然科学版, 2020, 57: 681.

[9] Nataraj L, Karthikeyan S, Jacob G, *et al*. Malware images: visualization and automatic classification[C]//Proceedings of the 8th international symposium on visualization for cyber security. New York: ACM, 2011.

[10] Yuan B, Wang J, Liu D, *et al*. Byte-level malware classification based on markov images and deep learning[J]. Comput Secur, 2020, 92: 101740.

[11] Huang N, Xu M, Zheng N, *et al*. Deep android malware classification with api-based feature graph[C]//Proceedings of the 2019 18th IEEE Interna-

tional Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). Rotorua: IEEE, 2019.

[12] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[C]. //The 5th International Conference on Learning Representations. Toulon: ICLR, 2017.

[13] Hou S, Ye Y, Song Y, *et al*. Hindroid: an intelligent android malware detection system based on structured heterogeneous information network [C]//Proceedings of the 23rd ACM SIGKDD International conference on knowledge discovery and data mining. New York: ACM, 2017.

[14] Fan Y, Hou S, Zhang Y, *et al*. Gotcha-sly malware! scorpion a metagraph2vec based malware detection system[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM, 2018.

[15] Wang S, Chen Z, Yu X, *et al*. Heterogeneous graph matching networks: application to unknown malware detection [C]//Proceedings of the 2019 IEEE International Conference on Big Data. Los Angeles: IEEE, 2019.

[16] Sun Y, Han J, Yan X, *et al*. Pathsim: meta path-based top-k similarity search in heterogeneous information networks [J]. VLDB J, 2011, 4: 992.

[17] Zhao H, Yao Q, Li J, *et al*. Meta-graph based recommendation fusion over heterogeneous information networks [C]//Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. New York: ACM, 2017.

[18] Cortes C, Vapnik V. Support-vector networks [J]. Mach Learn, 1995, 20: 273.

[19] Melissa. VirusShare. com [EB/OL]. (2016-06-15) [2021-03-25]. https://virusshare. com/torrents. 4n6.

[20] VirusTotal. VirusTotal. com [EB/OL]. (2017-06-08) [2021-03-25]. https://www. virustotal. com/.

引用本文格式：

中　文：刘建松，张磊，方勇. 基于行为关系网络的恶意代码检测方法[J]. 四川大学学报：自然科学版，2022，59：023001.

英　文：Liu J S, Zhang L, Fang Y. Malicious code detection method based on behavior relation network [J]. J Sichuan Univ: Nat Sci Ed, 2022, 59: 023001.