计算机科学
Computer Science
ISSN 1002-137X,CN 50-1075/TP

《计算机科学》网络首发论文

题目：基于异构溯源图学习的 APT 攻击检测方法
作者：董程昱，吕明琪，陈铁明，朱添田
收稿日期：2022-03-04
网络首发日期：2022-12-29
引用格式：董程昱，吕明琪，陈铁明，朱添田．基于异构溯源图学习的 APT 攻击检测方法[J/OL]．计算机科学.
https://kns.cnki.net/kcms/detail//50.1075.TP.20221228.1320.016.html

基于异构溯源图学习的 APT 攻击检测方法

董程昱　吕明琪　陈铁明　朱添田

浙江工业大学计算机科学与技术学院　杭州 310023
（dcyzjut@foxmail.com）

摘　要　APT 攻击（Advanced Persistent Threat），指黑客组织对目标信息系统进行的高级持续性的网络攻击。APT 攻击的主要特点是持续时间长和综合运用多种攻击技术，这使得传统的入侵检测方法难以有效地对其进行检测。现有大多数 APT 攻击检测系统都是在整理各类领域知识（如 ATT&CK 网络攻防知识库）的基础上通过手动设计检测规则来实现的。然而，这种方式智能化水平低，扩展性差，且难以检测未知 APT 攻击。为此，通过操作系统内核日志来监测系统行为，在此基础上提出了一种基于图神经网络技术的智能 APT 攻击检测方法。首先，为捕捉 APT 攻击多样化攻击技术中的上下文关联，将操作系统内核日志中包含的系统实体（如进程、文件、套接字）及其关系建模成一个溯源图（Provenance Graph），并采用异构图学习算法将每个系统实体表征成一个语义向量。然后，为克服 APT 攻击长期行为造成的图规模爆炸问题，提出了一种从大规模异构图中进行子图采样的方法，在此基础上基于图卷积算法对其中的关键系统实体进行分类。最后，基于两个真实的 APT 攻击数据集进行了一系列的实验。实验结果表明，提出的 APT 攻击检测方法的综合性能优于其他基于学习的检测模型以及最先进的基于规则的 APT 攻击检测系统。

关键词：　**APT** 攻击检测；图神经网络；溯源图；主机安全；数据驱动安全

中图法分类号　TP393

Heterogeneous Provenance Graph Learning Model Based APT Detection

DONG Chengyu　LYU Mingqi　CHEN Tieming　ZHU Tiantian

College of Computer Science & Technology，Zhejiang University of Technology，Hangzhou 310023，China

Abstract　APT (advanced persistent threat) are advanced persistent cyber-attack by hacker organizations to breach the target information system. Usually, the APTs are characterized by long duration and multiple attack techniques, making the traditional intrusion detection methods ineffective. Most existing APT detection systems are implemented based on manually designed rules by referring to domain knowledge (e.g., ATT&CK). However, this way lacks of intelligence, generalization ability, and is difficult to detect unknown APT attacks. Aiming at this limitation, this paper proposes an intelligent APT detection method based on provenance data and graph neural networks. To capture the rich context information in the diversified attack techniques of APTs, it firstly models the system entities (e.g., process, file, socket) in the provenance data into a provenance graph, and learns a semantic vector representation for each system entity by heterogeneous graph learning model. Then, to overcome the problem of graph scale explosion caused by the long-term behaviors of APTs， APT detection is performed by sampling a local graph from the large scale heterogeneous graph, and classifying the key system entities as malicious or benign by graph convolution networks. A series of experiments are conducted on two datasets with real APT attacks. Experiment results show that the comprehensive performance of the proposed method outperforms other learning based detection models, as well as the state-of-the-art rule based APT detection systems.

Keywords　APT detection, Graph neural network, Provenance graph, Hosted-based security, Data-driven security

到稿日期：2022-03-04　　返修日期：2022-09-28
基金项目：国家自然科学基金联合重点项目（U1936215）；浙江省重点研发项目（2021C01117）；国家自然科学基金青年项目（62002324）；浙江省自然科学基金重大项目（LD22F020002）；浙江省自然科学基金探索项目（LQ21F020016）；浙江省“万人计划”科技创新领军人才项目（2020R52011）
This work was supported by the Joint Funds of the National Natural Science Foundation of China (U1936215), Zhejiang Key R&D Projects (2021C01117), National Natural Science Foundation of China (62002324), Major Program of Natural Science Foundation of Zhejiang Province (LD22F020002), Zhejiang Provincial Natural Science Foundation of China (LQ21F020016) and "Ten Thousand People Program" Technology Innovation Leading Talent Project in Zhejiang Province (2020R52011).
通信作者：吕明琪（E-mail：lvmingqi@zjut.edu.cn）

1 引言

APT 攻击指利用先进的攻击手段对特定目标进行长期、持续性网络攻击的攻击形式，是威胁国家网络空间安全的最大隐患之一。APT 攻击最主要的特点就是采用多样的攻击技术，通过多个步骤的攻防对抗，逐步达到网络攻击的目的[1]。而传统的网络入侵检测技术（如恶意流量检测、恶意代码检测、日志异常检测）[2-3]往往只能检测“单点”的网络攻击行为，无法感知 APT 攻击过程中的上下文关联，因此难以有效地检测 APT 攻击。

针对这个问题，现有的 APT 攻击检测研究工作发现将系统日志数据建模成有向无环图（称为溯源图）可有效捕捉系统事件的上下文关联，是实现 APT 攻击检测的最佳手段[4-6]。其中，溯源图用于描述系统实体（如进程、文件、套接字）和系统实体之间的交互（交互通常表示为系统事件，如读取、写入、派生）。图 1 给出了一个溯源图的示例 。

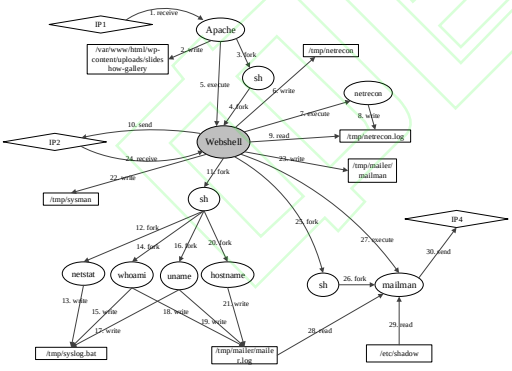


图 1 溯源图示例

Fig 1. Example of provenance graph

在基于溯源图的 APT 攻击检测方面，大多现有研究（如 SLEUTH[7]，HOLMES [8]和 CONAN[9]）尝试利用网络攻防知识（如杀伤链模型[10]、ATT&CK 知识库[11]）人工设计各种规则来检测 APT 攻击。基于规则的检测策略具有准确性高、可解释性强、便于部署等一系列优势，是产业界的主流方法。然而，基于规则的检测策略仍存在以下局限性。第一，规则的设计门槛极高，因为它需要网络攻防、操作系统、计算机网络等跨领域的综合专业知识。第二，规则缺乏泛化能力，因为它们是在特定的环境（如网络攻击策略、网络环境）下设计的，难以适应环境的变化。第三，规则依赖人类经验，无法处理人类难以理解的复杂和潜在的模式。

另一方面，基于学习的技术长期以来一直应用于网络安全任务，例如入侵检测[2-3]、恶意软件检测[12-13]和欺诈检测[14-15]。与基于规则的技术相比，基于学习的技术具有以下优点。首先，基于学习的技术（尤其是深度学习技术）可以从训练数据集中自动提取潜在模式并构建检测模型。其次，检测模型可以通过自动重新训练或微调来适应新的环境。为此，本文研究基于深度学习的 APT 攻击检测方法。针对溯源图的结构，利用图神经网络（Graph Neural Network， GNN）[16]来构建 APT 攻击检测模型。然而，由于 APT 攻击和溯源图的复杂性，基于 GNN 构建 APT 攻击检测模型仍然是一项具有挑战性的任务。

第一，溯源图复杂度较高，涉及多种类型的节点（如进程、文件、套接字）和多种类型的边（如读取、写入和创建）。此外，节点或边可以通过属性来描述（如文件可以通过文件扩展名和敏感级别等属性来描述）。

第二，溯源图的节点和边是系统实体和系统事件的实例，缺乏泛化性，使得检测模型几乎不可能从这些完全不同的实例中学习到共同模式。

第三，网络攻击检测是一项对效率很敏感的任务。然而，GNN 的计算复杂度和内存消耗很高。此外，由于 APT 攻击持续时间长，溯源图会随着时间的推移不断增长，这使得基于 GNN

的方法不可避免地会遇到效率和内存问题。

第四，APT 攻击通常是隐蔽的，因此仅通过查看单个系统实体很难检测到它们（如攻击可能由多个进程和文件的协作完成）。然而，考虑整个溯源图也是不切实际的，因为它通常太大并且基本由正常系统实体组成。

第五，检测模型必须不断处理新传入的系统实体，而 GNN 通过将新系统实体嵌入共享空间中的现有节点和边来学习新系统实体的表示。然而，由于高计算复杂性，实时更新系统实体的嵌入表征是不可行的。

针对上述挑战，本文提出了一种基于异构溯源图学习的智能 APT 攻击检测方法。对于第一个和第二个挑战，采用异构图[17]来表示溯源图中的复杂系统实体和事件，并使用基于元路径的方法[18]来提取系统实体之间的高级别语义交互。对于第三个挑战，使用图嵌入技术来学习异构图中每个节点的低维向量表示。图嵌入可作为预训练步骤执行，并使用学习到的向量表示来快速初始化检测模型。对于第四个和第五个挑战，将新传入的系统实体链接到现有的异构图，并提出一种子图采样策略，将新传入的系统实体和最相关的现有节点合并到一个小而紧凑的子图中，然后基于子图建立检测模型。

2 相关工作

现有 APT 攻击检测系统的主流为取证分析方法，旨在根据预定义的规则发现攻击事件和攻击路径。例如，PrioTracker[19]通过调查异常因果关系的优先级来实现 APT 攻击分析，其中系统事件的优先级通过预定义的规则来衡量。SLEUTH[7]通过基于标签的方法识别最有可能参与 APT 攻击的系统实体和事件，首先设计标签对代码和数据的可信度和敏感性进行编码，然后利用这些标签设计网络攻击检测规则。HOLMES[8]是一个用于 APT 攻击检测的分层框架，关键组件是一个中间层，根据领域知识（如 ATT&CK 模型）的规则将底层审计数据映射到可疑行为。CONAN[9]是一个基于状态的框架。每个系统实体（即进程或文件）都以类似 FSA（有限状态自动机）的结构表示，其中状态是基于预定义的规则推断出来的。然后，状态序列用于 APT 攻击检测和重建。基于精心设计的规则，上述取证分析方法具有准确性高、可解释性强、便于部署的优点，但是设计有效的规则在很大程度上依赖于深入的跨学科领域知识，如果环境发生变化，规则很容易过时。

为了突破取证分析方法的局限性，现有研究通过利用基于学习的技术（包括机器学习和深度学习技术）以自动方式构建 APT 攻击检测模型。例如，Barre 等[20]从溯源图中提取一组特征如写入的数据总量、使用的系统文件的数量等来构建分类器，以检测 APT 攻击。Berrada 等[21]从溯源图中提取布尔值特征，并通过无监督学习技术将 APT 攻击检测视为异常检测任务。Xiang 等[22]从 PC 平台和移动平台中提取不同的特征，并使用多种机器学习算法，基于组合特征检测 APT 攻击。Zimba 等[23]通过半监督学习框架识别表现出可疑恶意活动的主机，该框架基于网络聚类算法提取特征，并使用标签和无标签数据训练检测模型。上述研究基于传统机器学习技术检测 APT 攻击的核心步骤是特征提取，但是特征提取也非常依赖领域知识。

针对上述传统机器学习的问题，深度学习[24]能以完全自动的方式从大数据中发现潜在特征。因此，现有一些研究尝试使用深度学习技术进行 APT 攻击检测。APT 攻击最基本的特征之一是持久性，收集到的系统事件通常跨越很长一段时间，可以表示为时间序列。因此，循环神经网络

（Recurrent Neural Network， RNN），比如 LSTM（Long Short-Term Memory）和 GRU（Gated Recurrent Unit），被广泛应用于 APT 攻击[25-27]。然而， RNN 只能捕获系统事件之间的顺序关系，忽略其他重要的上下文信息（如系统实体之间不同类型的交互）。针对这个问题，Liu 等[28]提出了 log2vec——一种基于异构图嵌入的 APT 攻击检测模型：首先根据一组预定义的规则将日志数据转换为异构图，然后使用异构图嵌入技术将每条日志数据表示为一个低维向量，最后根据日志向量将恶意和正常日志分类。但是 log2vec 仍然存在以下限制：首先，异构图是基于预定义的规则构建的，因此 log2vec 仍然强烈依赖领域知识；其次，log2vec 是离线工作的，没有考虑新传入的系统实体。

3 方法概述

3.1 威胁模型

根据威胁模型（如杀伤链模型[10]、ATT&CK 模型[11]），攻击者实施一次 APT 攻击通常需要通过多个阶段，并且会在每个阶段使用多种技术。现有研究工作发现，在这些阶段中，APT 攻击通常具有以下不变部分[9]。第一，攻击者必须将他们的代码部署到受害者主机上。第二，APT 攻击的最终目标为窃取敏感信息或造成损害。第三，攻击者必须与服务器进行通信。因此，本文根据这些不变部分检测新的 APT 攻击。

3.2 系统架构

本文提出方法的系统架构如图 2 所示。

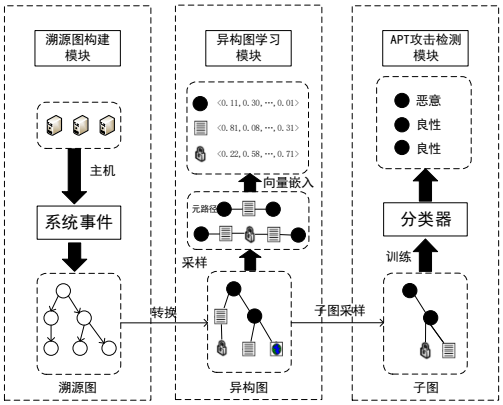


图 2 系统架构

Fig.2 System architecture

(1) 溯源图构建模块：利用操作系统审计工具（如 Auditd[29]和 ETW[30]）收集内核日志数据，并利用收集到的操作系统内核日志数据构建溯源图。

(2) 异构图学习模块：首先定义节点类型集和边类型集，并将溯源图转换为具有语义的异构图；然后通过元路径采样策略和层次注意力嵌入技术将每个异构图节点表征为一个低维向量（即异构图嵌入）。通过以上方法，异构图可以保留各种类型节点之间的语义和结构关系。

(3) APT 攻击检测模块：首先将新传入的系统实体 v_n 连接到现有的异构图，并以 v_n 为中心节点对子图进行采样；然后采用图卷积网络探索更大的图邻域，发现系统实体之间的因果关系和上下文关系；最后通过节点分类实现 APT 攻击检测。

4 方法详述

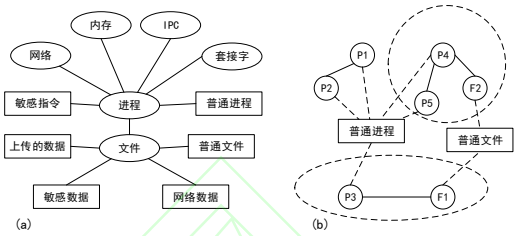
4.1 异构图构建

首先，本文利用操作系统审计工具收集内核系统事件来生成溯源图。其中，节点代表系统实体（如进程、文件），边代表系统事件（如写入文件及读取 IP 地址）。然后，采用如下定义的异构溯源图对系统事件数据进行建模。

定义 1（异构溯源图） 异构溯源图 $G = (V, E)$由表示系统实体的节点集 V 和表示系统事件的

边集 E 组成。此外，定义 A 和 R（分别表示节点类型集和边类型集）以及 $\varphi: V \rightarrow A$ 和 $\psi: E \rightarrow R$（分别表示节点类型映射和边类型映射），其中 $|A| + |R| > 2$。

根据定义，异构溯源图构造如下。首先，定义异构溯源图的节点类型和边类型。本文定义了 7 种主要的节点类型：进程、文件、套接字、进程间通信（Inter-Process Communication，IPC）、内存、网络和属性。节点类型“属性”用于描述“进程”和“文件”的特征，可进一步分为普通进程、敏感指令、普通文件、网络数据、敏感数据、上传数据 6 种子类型。图 3 显示了这些节点类型及其潜在关系。其中，“敏感指令”代表进程执行敏感指令，“网络数据”代表文件包含来自网络的数据，“敏感数据”代表文件包含敏感数据，“上传数据”代表有数据发送到服务器。图 3 中“普通进程”和“普通文件”是冗余属性，这两种属性用于防止异构溯源图被分割成许多不相连的小图。如图 3(b)所示，P_1，P_2，P_3，P_4，P_5 为普通进程，F_1 和 F_2 为普通文件。如果没有“普通进程”和“普通文件”属性（即图 3(b)中的虚线箭头），这些系统实体将被分成 3 个不连续的小图（如虚线椭圆所示），导致难以在同一个特征空间中学习 P_1 和 P_3、F_1 和 F_2 之间的语义相关性。由于绝大多数系统实体都是普通进程或者普通文件，如果没有冗余属性，可能严重影响实际学习效果。



(a) 异构图模式　　(b) “普通进程”和“普通文件”属性的必要性示例

图 3 异构图说明

Fig.3 Illustration of heterogeneous graph

然后定义 8 种边类型，如表 1 所列。边类型表示系统实体之间的关系，可以为检测 APT 攻击提供关键信息。一对系统实体类型可以具有多种类型的关系。例如，“进程”和“文件”之间可能具有“读取”“写入”“创建”等关系。

表 1 边类型总结

Table 1 Summary of edge types

系统实体类型对	ID	系统事件类型	描述
进程→进程	R_1	进程→op1→进程	“op1” 可以为 “派生”“执行”“退出”“克隆”“改变”
进程→文件	R_2	进程→op2→文件	“op2” 可以为 “读取”“打开”“关闭”“写入”“读取库”“创建”“删除”“更改”“截断”“重命名”“映射”“更新”
进程→网络	R_3	进程→op3→网络	“op3” 可以为 “连接”“发送”“接收”“读取”“关闭”“接受”“写入”
进程→内存	R_4	进程→op4→内存	“op4” 可以为 “保护”“映射”
进程→IPC	R_5	进程→op5→IPC	“op5” 可以为“写入”“关闭”“读取”“映射”
进程→套接字	R_6	进程→op6→套接字	“op6” 可以为 “连接”“发送”“接收”“读取”“关闭”“接受”“写入”
进程→属性	R_7	进程→包含→属性	一个进程包含一种属性,例如 “普通进程”“敏感指令”
文件→属性	R_8	文件→包含→属性	一个文件包含一种属性，例如 “普通文件”“网络数据”“敏感数据”“上传的数据”

异构溯源图对原始的系统日志进行了一定程度的泛化，使用语义来定义每个节点。例如，所有文件实例共享相同的节点类型“文件”。这便于模型学习到不同节点之间的语义关联。

4.2 异构溯源图嵌入

APT 攻击的高持久性特点导致生成的异构溯源图可能非常大。因此，处理异构溯源图的方法必须是高效、可扩展的，以满足实时检测的需

求。根据图神经网络领域的研究，图嵌入（Graph Embedding）是一种表示大规模图的有效方式[31]，因此本文采用图嵌入技术来表示异构溯源图。异构溯源图嵌入的定义如下所示。

定义 2（异构溯源图嵌入） 给定一个异构溯源图 $G = (V, E)$，G 的嵌入表示为 $f: V \rightarrow R^d$，并将每个节点 $v \in V$ 映射到一个 d 维向量 ($d << |V|$)， 保证向量空间能够保留 G 中的拓扑结构和语义关系。

由于异构溯源图包含多种类型的节点和边，因此不能直接使用传统的图嵌入技术（如 DeepWalk[32]，node2vec[33]和 LINE[34]）。针对上述问题，本文采用元路径来引导随机游走连接各种类型的节点[18]。基于元路径的随机游走可以为每个节点生成一组邻居，从而表示异构溯源图中的各种结构和语义信息。元路径定义如下。

定义 3（元路径） 元路径是在异构溯源图中定义的路径，形式为 $A_1 \rightarrow R_1 \rightarrow A_2 \rightarrow R_2 \ldots R_{L-1} \rightarrow A_L$，其中 $R = R_1 \rightarrow R_2 \rightarrow R_{L-1}$ 定义了节点类型 A_1 和 A_L 之间的复合关系。元路径中的 A_1 和 A_L 通常是相同的节点类型，因此在元路径上结束的随机游走可以立即在另一个元路径上开始。

表 2 定义了各种元路径来表示不同类型系统实体间的相关性。例如，MP_4 表示如果两个进程连接到具有相同属性的文件，则两个进程是相关的；MP_6 表示如果两个进程在网络上执行相同的操作，则两个进程是相关的。

表 2 元路径类型总结

Table 2 Summary of meta-path types

ID	元路径
MP_1	进程→op1→进程
MP_2	进程→包含→属性→包含$^{-1}$→进程
MP_3	进程→op2→文件→op2^{-1}→进程
MP_4	进程→op2→文件→包含→属性→包含$^{-1}$→文件→op2^{-1}→进程
MP_5	进程→op5→IPC→op5^{-1}→进程
MP_6	进程→op3→网络→op3^{-1}→进程
MP_7	进程→op4→内存→op4^{-1}→进程
MP_8	进程→op6→套接字→op6^{-1}→进程

在定义了所有元路径之后，本文采用一种基于分层注意力的异构图嵌入技术 HGAT[35]来学习每个节点的 d 维向量。相对于其他异构图嵌入技术（如 metapath2vec[18] 和 metagraph2vec[36]），HGAT 的优势在于它考虑了节点的不同邻居和元路径的重要性。因为不同的系统实体和行为通常具有不同的敏感和危险级别，HGAT 这一优势对于 APT 攻击检测任务尤为重要。

本文主要分两步将 HGAT 应用于异构溯源图嵌入。

第一步，通过元路径来引导随机游走，从异构溯源图中生成各种类型的系统实体序列。给定一个异构溯源图 $G = (V, E)$ 和一组元路径 MPS，随机游走的工作原理如下：首先，从 MPS 中随机选择一个元路径 MP_i（以 $A_1 \rightarrow R_1 \ldots A_t \rightarrow R_t \rightarrow A_{t+1} \ldots R_{L-1} \rightarrow A_L$ 的形式）；然后，按照 MP_i 在异构溯源图上随机游走，其中节点 v_j 到 v_{j+1} 在第 j 步的转移概率如式(1)所示，$\varphi(v_j)=A_t$，$N_{t+1}(v_j)$是实体类型为 A_{t+1} 的节点 v_j 的邻居集；最后，随机选择另一个以 A_L 开始的元路径 MP_{i+1}，并重复游走过程。

$$p(v_{j+1} \mid v_j, MP_i) = \begin{cases} \dfrac{1}{|N_{t+1}(v_j)|} & (v_j, v_{j+1}) \in E, \phi(v_{j+1}) = A_{t+1} \\ 0 & (v_j, v_{j+1}) \in E, \phi(v_{j+1}) \neq A_{t+1} \\ 0 & (v_j, v_{j+1}) \notin E \end{cases} \quad (1)$$

第二步，将生成的系统实体序列输入模型，用于学习系统实体嵌入。根据 HGAT 的分层注意力结构（包括节点级注意力和语义级注意力），对于给定的系统实体 v_i 和元路径 MP_j，节点级注意力从 MP_j 中学习生成的系统实体序列中 v_i 的每个邻居的重要性，并将这些有意义的邻居聚合起来形成 v_i 的候选嵌入，表示为 $\mathbf{z}_i^j$。然后，利用语义级注意力学习不同元路径对 v_i 的重

要性，并融合所有候选嵌入即 $\mathbf{z}_i^1, \mathbf{z}_i^2, \cdots \mathbf{z}_i^{|MPS|}$，从而获得 v_i 的最终嵌入。

4.3 子图采样与 APT 攻击检测

在获得所有系统实体嵌入后，检测 APT 攻击的最简单策略是训练分类器，直接将系统实体的 d 维嵌入向量作为特征，将每个“进程”分类为正常进程或恶意进程。但是，这种简单的策略存在以下问题。首先，APT 攻击通常是隐蔽的，因此仅考虑一个系统实体很难检测到它们。其次，APT 攻击检测系统必须不断处理新传入的系统实体，而频繁重新运行异构溯源图嵌入过程是不可行的。

为了解决上述问题，本文提出了一种基于子图采样的检测策略。给定一个新的传入系统实体 v_n 以及与 v_n 关联的系统实体（表示为 *NVS*），首先将 *NVS* 中的每个系统实体链接到现有的异构溯源图（新的图表示为 *CG*）。实体类型“属性”可以保证每个系统实体都将链接到现有的异构溯源图。然后，根据如下定义的 k 阶子图采样，从 *CG* 中为 *NVS* 采样一个子图。

定义 4（k 阶子图采样） 对于系统实体 v_i，其在 $G = (V, E)$上的 k 阶子图采样定义如式(2)所示。基于定义，每个系统实体可以基于 k 阶子图采样形成一个子图。对于一组系统实体 *VS*，其在 $G = (V, E)$上的 k 阶子图采样定义如式(3)所示。

$$SG^{(k)}(v_i, G) = \begin{cases} \{v_j \mid (v_i, v_j) \in E\} & k = 1 \\ \{SG^{(1)}(v_z, G) \mid v_z \in SG^{(k-1)}(v_i, G)\} & k > 1 \end{cases} \quad (2)$$

$$SG^{(k)}(VS, G) = \bigcup\nolimits_{v_i \in VS} SG^{(k)}(v_i, G) \quad (3)$$

如果没有任何约束，采样的子图可能会变得非常大。例如，如果子图包含一个类型为“普通文件”的内部节点 v_i，它将连接到大量类型为“文件”的节点。为避免这种情况，设置子图中节点出度的上限。首先，将七大实体类型分为两类：可实例化实体类型（即进程和文件）和概念型实体类型（即套接字、IPC、内存、网络和属性）。显然，概念实体类型的节点数量有限，因此本文只考虑可实例化实体类型的节点数量。对于子图中的每个节点，设置其连接的可实例化实体类型的节点数量不超过上限 λ。图 4 为子图采样策略的示例。子图采样的目的有两点：1）将来自所有系统实体的上下文信息聚合在子图中用于 APT 攻击检测，而不是仅利用一个系统实体的信息；2）对于新传入的未知系统实体，将根据从子图中的邻居传输的信息来推断其嵌入，从而省去重新训练异构溯源图嵌入的过程。

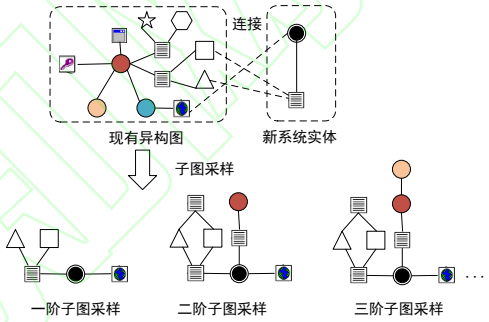


图 4 子图采样策略示例

Fig.4 Example of subgraph sampling strategy

经过子图采样之后，对子图中每个类型为“进程”的节点进行分类，从而实现 APT 攻击检测。本文将 关系图卷积网络（R-GCN）[37]应用于节点分类任务。R-GCN 是 GCN 对关系图的扩展，它基于式（4）聚合从邻居到中心节点（由 v_i 表示）的信息。更多的 R-GCN 层（即更大的 l 值），表明模型可以聚合来自更远节点的信息。

$$h_i^{(l+1)} = \sum_{r \in R} \sum_{j \in N_i^r} \frac{1}{|N_i^r|} W_r^{(l)} h_j^{(l)} + W_0^{(l)} h_i^{(l)} \quad (4)$$

其中，$h_i^{(l)}$ 是第 l 层 R-GCN 层中节点 v_i 的隐藏状态，N_i^r 是 v_i 的边类型为 r 的邻居的索引集，$W_0^{(l)}$ 与 $W_r^{(l)}$ 是可学习的参数。

R-GCN 可以看作是子图的编码器。它通过在子图上堆叠多个 R-GCN 层来学习子图中更高阶的上下文信息。然后，本文通过最小化所有标

签节点的交叉熵损失，并在最后一层的每个节点的输出隐藏状态上添加 softmax 激活函数来训练节点分类器。

5 实验

5.1 实验设置

5.1.1 数据集

本文使用以下两个 APT 攻击数据集进行实验。

Lab 数据集：该数据集由本文研究人员通过在真实 Linux 主机上模拟 APT 攻击采集得到，详细信息如表 3 第一行所列。“持续时间”一栏指采集数据集的时间跨度，包括正常活动（如观看在线视频、网站浏览、文档编辑等）和攻击相关的活动。“#攻击”一栏指 APT 攻击的数量，一次 APT 攻击涉及多个恶意进程。

在此数据集的收集过程中，主要模拟发动了 3 种类型的 APT 攻击：1）“webshell 攻击”，使用 webshell 脚本，通过利用 web 漏洞来维持对目标系统的长期访问；2）“RAT 攻击（Remote Access Trojan）”，通过钓鱼攻击将木马植入目标系统，然后利用木马窃取敏感信息；3）“LotL（Living off the Land）攻击”，直接在内存中运行恶意 shellcode。

接下来，通过溯源图构建工具 SPADE[38]将收集到的系统日志数据转换为溯源图。最终经过处理的数据集包含 19659 个进程、10073 个文件和 2700912 个系统事件。

DARPA 数据集：该数据集是由 DARPA 组织团队从 3 台安装了 Windows 10 系统的主机上执行 APT 攻击采集得到，详细信息如表 3 第二行所列。

表 3 数据集的详细信息

Table 3 Dataset details

数据集	持续时间 (时：分：秒)	读取文件/%	写入文件/%	进程/%	网络/%	其他/%	#攻击
Lab	10:41:33	45.11	18.41	1.94	14.47	20.07	34
DARPA	24:32:15	45.10	27.78	0.91	9.04	17.17	7

在该数据集的收集过程中，主要进行了 3 种类型的 APT 攻击：敏感信息的收集和泄露，利用 FireFox 漏洞的内存攻击，以及恶意文件下载和执行。可以看出，DARPA 数据集中的 APT 攻击与 Lab 数据集中的有很大不同，但所有这些攻击都包含一个或多个不变的阶段（如部署恶意代码、窃取敏感信息与服务器通信）。最终经过处理的数据集包含 53084 个进程、219706 个文件和 26751468 个系统事件。

实验在配备 Intel Xeon E5-2680 v4 CPU（具有 14 ×2.4 GHz 内核）、128GB 内存和 4 ×RTX 2080Ti GPU 的服务器上进行，并在 Ubuntu 20.04 上运行。

5.1.2 评估策略

首先介绍本文实验中使用的两种评估策略，如下所示。

(1) In-Sample：该评估策略在整个数据集上构建异构溯源图，然后再将其划分为训练集和测试集（比例为 3:1）。在采样的过程中，保持训练集和测试集中恶意、正常进程的原始比例。

(2) Out-Sample：该评估策略用于评估对新进程的检测性能。先将原始数据集划分为训练集和测试集（比例为 3:1），然后在训练集上训练异构溯源图嵌入和 APT 攻击检测模型，并对测试集中的进程进行检测。

其次，将 ACC，Precision，Recall 和 Macro-F1 作为性能指标。其中，ACC 指准确度，Precision 和 Recall 为检测恶意样本的精确率和召回率（Precision 与误报率成反比），Macro-F1 用于考虑恶意和正常样本的不平衡性。

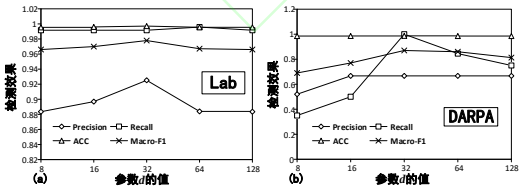
5.1.3 训练策略

由于数据集高度不平衡，即大多数训练样本是正常进程，因此标准的训练策略往往会被正常样本淹没而无法检测恶意样本。针对这种情况，本文采用了代价敏感的交叉熵损失函数，为每个样本分配一个权重 δ，每个恶意样本的 $\delta = 1$，每个正常样本的 $\delta = 0.1$。

5.2 调参实验

本文的 APT 攻击检测方法有 3 个关键参数：节点嵌入的维度 d、子图中节点出度的阈值 λ 和子图采样的跳数 k。本实验主要测试这 3 个参数对方法检测性能的影响。本实验中，使用 In-Sample 评估策略。

（1）参数 d 的实验结果如图 5 所示。实验中固定 $k = 8$，$\lambda = 10$，并在[8，128]范围内调整 d 的数值。从图中可知，检测性能整体表现为先上升再下降。因为当 d 太小时，节点嵌入很难编码获得足够的上下文信息。当 d 太大时，节点嵌入更容易受到噪声的影响并导致过拟合。总体上看，当 $d = 32$ 时整体检测性能最佳。

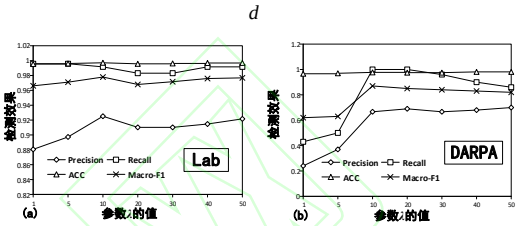


(a) Lab 数据集上的实验结果　　(b) DARPA 数据集上的实验结果

图 5 参数 d 的影响

Fig. 5 Effect of parameter d

（2）参数 λ 的实验结果如图 6 所示。实验中固定 $d = 32$，$k = 8$，并在[1, 50]范围内调整 λ 的数值。当 $\lambda < 10$ 时，增大 λ 带来显著的检测性能提升，这说明对邻居节点采样数量过少会显著影响检测性能。当 $\lambda > 10$ 时，继续增大 λ 无法带来显著的检测性能提升，说明考虑太多可实例化实体类型的系统实体（即进程和文件）对 APT 攻击检测性能的影响有限。对检测性能更重要的是系统实体间的上下文语义。此外，随着 λ 增长，计算复杂度也显著增长。因此，在检测性能和计算复杂度之间进行权衡后，设置 $\lambda = 10$。

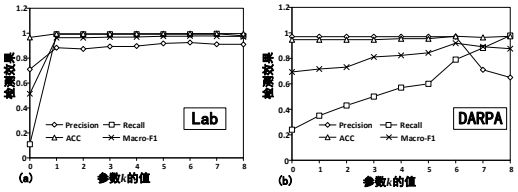


(a) Lab 数据集上的实验结果　　(b) DARPA 数据集上的实验结果

图 6 参数 λ 的影响

Fig.6 Effect of parameter λ

（3）参数 k 的实验结果如图 7 所示。实验中固定 $d = 32$，$\lambda = 10$，并在[0, 8]范围内变化 k 的数值。当 k 增加时，检测性能最初有一个明显的上升阶段，说明 APT 攻击检测任务应该考虑多个系统实体及其上下文信息。此外，Lab 数据集中的上升趋势与 DARPA 数据集中的上升趋势有很大不同。在 Lab 数据集中，当 k 从 0 增加到 2 时，会有一个显著的上升阶段，然后逐渐趋于稳定。在 DARPA 数据集中，当 k 从 0 增加到 6 时，有一个显著上升的阶段。原因是 DARPA 数据集中的攻击更隐蔽，跨越时间更长，因此需要考虑更远距离的系统实体来捕获它们的上下文信息。根据综合实验性能，设置 $k = 8$。



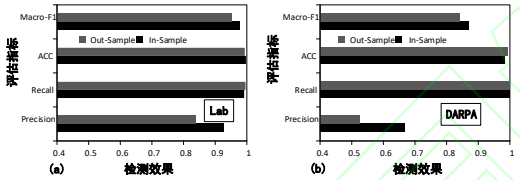
(a) Lab 数据集上的实验结果　　(b) DARPA 数据集上的实验结果

图 7 参数 k 的影响

Fig.7 Effect of parameter k

5.3 消融实验

第一个实验评估本方法对新传入的未知进程进行分类的能力，即比较在 In-Sample 和 Out-Sample 评估策略下的检测性能。实验结果如图 8 所示。与 In-Sample 评估策略相比，在 Out-Sample 评估策略下的检测性能有较低的 Precision，这意味着对未知进程的误报率有一定程度的增加，说明如果没有全面了解目标进程和其他系统实体之间的关系，检测性能就会受到影响。然而与 In-Sample 评估策略相比，在 Out-Sample 评估策略下的 ACC，Recall 和 Macro-F1 并没有明显下降，说明本方法仍然可以保证对未知进程较好的检测性能。



(a) Lab 数据集上的实验结果 (b) 在 DARPA 数据集上的实验结果

图 8 对新传入的未知进程进行分类方法的评估

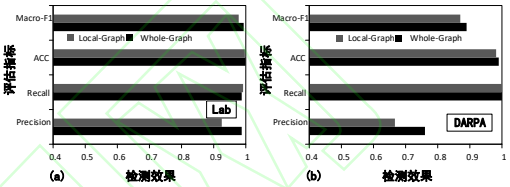
Fig. 8 Evaluation of our method for classifying new incoming unknown processe

第二个实验验证子图采样策略的有效性。在本次实验中，使用 In-Sample 评估策略评估以下两种采样策略的检测性能和计算开销。

(1) Whole-Graph：将整个异构溯源图视为子图。

(2) Local-Graph：采样 8 阶子图作为子图。

检测性能如图 9 所示。可以看出，Whole-Graph 采样策略稍优于 Local-Graph 采样策略。这意味着一些 APT 攻击是非常隐蔽的，它们在进入系统后不会立即进行恶意活动，而是长时间隐藏自己并由其他系统实体触发攻击，导致恶意进程与在异构溯源图中执行恶意活动的被操纵系统实体之间存在很长的距离。



(a) Lab 数据集上的实验结果 (b) DARPA 数据集上的实验结果

图 9 子图采样策略的评估

Fig. 9 Evaluation of subgraph sampling strategies

然而，处理整个异构溯源图会导致极高的计算开销。为了验证这一假设，在整个异构溯源图上测试了 APT 攻击检测的执行时间和内存使用。首先，通过随机插入噪声节点和边来模拟非常大的异构溯源图。测试结果如表 4 所列，其中“执行时间”包括对图进行卷积运算和对单个节点进行分类的时间。随着节点数量的增加，执行时间略有延长，但内存使用量却呈现显著增长。这说明子图采样策略在实际检测中能以较低的计算开销保证检测性能。

表 4 whole-graph 检测的计算开销

Table 4 Computation overhead of whole-graph detection

	Whole-Graph					Local-Graph
	50 节点	500 节点	5000 节点	50000 节点	90000 节点	
内存使用/GB	0.821	0.981	1.439	6.361	10.263	0.845
执行时间/s	0.1558	0.1602	0.1669	0.6568	2.1332	0.1598

第三个实验验证异构溯源图嵌入模块的有效性。使用 In-Sample 评估策略在 Lab 数据集上测试了从 MPC_1 到 MPC_6 不同元路径组合的检测性能。

(1) MPC_1：只包含进程和文件（即 MP_1 和 MP_3）的元路径。

(2) MPC_2：只包含进程、文件、网络和套接字的元路径（即 MP_1，MP_3，MP_6 和 MP_8）。

(3) MPC_3：只包含进程、文件、IPC 和内存的元路径（即 MP_1，MP_3，MP_5 和 MP_7）。

(4) MPC_4：只包含进程、文件和文件属性（即 MP_1，MP_3 和 MP_4）的元路径。

(5) MPC_5：只包含进程、文件和进程属性（即 MP_1，MP_2 和 MP_3）的元路径。

(6) MPC_6：包含所有元路径。

实验结果如表 5 所列。1）MPC_6 的整体检测性能最好，表明所有元路径都有助于 APT 攻击检测任务，这也意味着 APT 攻击非常复杂，仅考虑某些类型的系统实体和系统事件是不可能检测到所有 APT 攻击的。2）MPC_4 和 MPC_5 的检测性能优于 MPC_1，这表明进程和文件属性可以提供更丰富的信息来提高检测性能。3）MPC_4 检测性能优于除 MPC_6 之外的所有其他方法，意味着文件属性是 APT 攻击检测的最重要信息。这一结果与现有的研究经验一致，即从文件中窃取敏感数据或对文件执行敏感操作是 APT 攻击最重要的原因之一[9]。4）*MPC* 检测性能优于 MPC_3 和 MPC_5。这意味着在 APT 攻击中访问网络比操纵进程和内存更重要。

表 5 不同元路径组合的评估

Table 5 Evaluation of different meta-path combinations

	Precision	*Recall*	*ACC*	*Macro-F*1
MPC_1	0.8514	0.9916	0.9940	0.9565
MPC_2	0.8902	0.9916	0.9957	0.9679
MPC_3	0.8708	0.9958	0.9950	0.9633
MPC_4	0.9046	**1.0000**	0.9965	0.9740
MPC_5	0.8551	0.9958	0.9943	0.9586
MPC_6	**0.9252**	0.9916	**0.9971**	**0.9779**

5.4 对比实验

第一个实验通过与以下 4 个方法进行比较来评估本方法的检测性能。所有方法都使用 In-Sample 评估策略，并考虑了样本不平衡问题，在训练过程中为样本分配了不同权重（恶意样本和良性样本为 10:1）。

(1) SVM：利用向量空间模型创建特征向量（每一项代表异构溯源图中目标进程节点的一阶邻居中某类节点的个数），并将 SVM 作为分类器。

(2) GAT：同构图神经网络[39]。首先构造一个与异构溯源图具有相同拓扑结构的同构图，但忽略节点的类型，然后基于 GAT 模型进行节点嵌入和分类。

(3) HGAT：异构图神经网络。首先构建异构溯源图，然后基于 HGAT 模型进行节点嵌入和分类，无需子图采样和 R-GCN 分类步骤。只需关注单一进程，可以将其视为本方法的一种变体。

(4) CONAN：基于规则的 APT 攻击检测模型[9]。从源代码、行为、特征和网络 4 个方面手动定义大量规则，进行 APT 攻击检测。

(5) THREATRACE：基于溯源图的智能检测模型[40]。首先，基于溯源图中节点周围其他不同类型节点的数量构建初始特征向量。然后，采用 GraphSage 算法对溯源图进行嵌入学习。最后，基于节点的嵌入表征向量对齐进行分类。

实验结果如表 6 所列。1）GAT 完全无法检测 APT 攻击，因为 GAT 只考虑系统实体之间的交互，而忽略了所有对于检测恶意活动至关重要的语义（如系统实体的类型、系统事件的上下文等）。2）SVM 的检测性能也很差，因为在实际攻击中，攻击者往往在执行恶意活动之前会隐藏自己一段时间。而 SVM 只考虑与目标进程相邻的系统实体的类型，忽略了这些系统实体的下游活动。3）本方法优于 HGAT。这再次表明，仅通过关注一个系统实体很难检测到 APT 攻击。

4）CONAN 在 Lab 数据集上比本方法略有优势，而本方法在 DARPA 数据集上优于 CONAN。这说明本方法的检测性能与 CONAN 接近。然而，CONAN 是在对攻击样本进行深入分析之后人工设计规则而实现的，而本方法完全是数据驱动的，因此本方法在实现代价和泛化能力上相比 CONAN 具有明显的潜在优势。5）THREATRACE 在 Lab 数据集上的检测性能略优于本方法，而在 DARPA 数据集上本方法比 THREATRACE 有优势。THREATRACE 通过 1 阶邻居生成初始特征，并基于 GraphSage 捕捉多阶上下文关联。而本方法通过节点属性生成初始特征，并基于图嵌入和子图卷积捕捉多阶上下文关联。由于 THREATRACE 是在整张溯源图上捕捉上下文关联，因此考虑的阶数受到计算复杂度的限制，而本方法在子图上捕捉上下文关联，因此可以考虑较高的阶数。综上，本方法在 APT 攻击隐蔽性更高的 DARPA 数据集上表现较好。

表 6 不同检测方法在 In-Sample 策略上的比较

Table 6 Comparison of different detection methods in In-Sample .strategy

	Precision	***Recall***	***ACC***	***Macro-F1***
Lab 数据集:				
SVM	**1.0000**	0.0457	0.9641	0.5041
GAT	0.0000	0.0000	0.9670	0.4916
HGAT	0.7102	0.1093	0.9663	0.5124
CONAN	0.9758	**0.9981**	**0.9983**	**0.9915**
THREATRACE	0.9673	0.9750	0.9978	0.9892
OUR	0.9252	0.9916	0.9971	0.9779
DARPA 数据集:				
SVM	1.0000	0.1253	0.9237	0.5709
GAT	0.0000	0.0000	0.9715	0.4925
HGAT	**1.0000**	0.2491	0.9732	0.6932
CONAN	0.5895	1.0000	0.9759	0.8395
THREATRACE	0.5657	0.8334	0.9746	0.8107
OUR	0.6667	**1.0000**	**0.9820**	**0.8703**

5.5 案例分析

本节通过一个真实的攻击案例来说明本文方法。给定如图 1 所示攻击场景的溯源图（其中菱形代表网络，椭圆形代表进程，矩形代表文件），攻击者首先入侵服务器并留下后门，当触发后门时会创建一个进程“Webshell”。然后，“Webshell”会派生几个进程来执行敏感命令，例如通过“netstat”来查看详细的网络信息并将收集到的信息写入文件“mailer.log”。最后，“Webshell”执行一个进程“mailman”从“mailer.log”和另一个敏感文件“shadow”收集敏感信息，并发送到远程 C&C 服务器。

图 1 生成的异构溯源图如图 10 所示，其中虚线边框矩形代表属性。根据现有研究[9]，APT 攻击的最终目的是窃取敏感信息或造成损害，因此它们通常具有一些共同的特征（例如执行敏感指令、读取敏感文件等）。这些共同特征可以通过基于元路径的异构溯源图嵌入来学习。其次，原始恶意进程与执行恶意活动的系统实体之间通常存在一定的距离（如“Webshell”和“netstat”，“Webshell”和“reading from shadow”）。因此，仅关注一个系统实体无法检测到此类 APT 攻击，而本方法可以基于子图采样和卷积操作对多阶系统实体进行建模。以“Webshell”作为待分类进程，如果从中采样一个 3 阶子图，则可以通过卷积操作将“敏感指令”特征传播给它。如果从中采样一个 4 阶子图，“敏感数据”特征也可以传播给它。这些特征可以潜在地帮助模型确定“Webshell”是一个恶意进程。

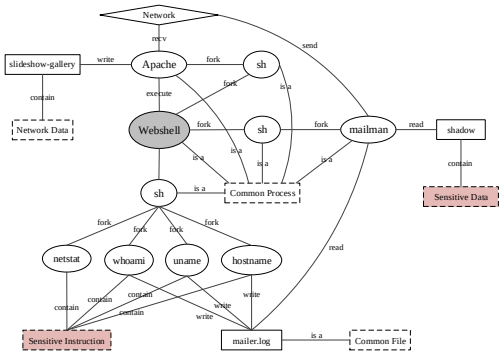


图 10 真实攻击案例的异构溯源图

Fig. 10 Heterogeneous graph of a real attack case

结束语 本文研究了基于溯源图的 APT 攻击检测问题，提出了一种基于深度学习技术的智能 APT 攻击检测方法，采用异构溯源图对所有系统实体和事件进行建模，并学习一个低维向量以可扩展的方式表示每个系统实体；然后通过从溯源图中采样一个小的子图来重建攻击场景，并检测该子图上的 APT 攻击。基于上述设计，本文方法可以有效地检测具有持久性、隐蔽性和多样性特征的 APT 攻击。 基于包含真实 APT 攻击的数据集的一系列实验，证明了本文方法优于当前其他基于机器学习和深度学习的模型以及基于规则的方法。

参考文献

[1] GHAFIR I,HAMMOUDEH M,PRENOSIL V, et al. Detection of Advanced Persistent Threat Using Machine-Learning Correlation Analysis[J]. Future Generation Computer Systems, 2018, 89(DEC.):349-359.

[2] BRIDGES R A, GLASS-VANDERLAN T R, IANNACONE M D, et al. A survey of intrusion detection systems leveraging host data[J]. ACM Computing Surveys (CSUR), 2019, 52(6): 1-35.

[3] SINGLA A , BERTINO E , VERMA D . Preparing Network Intrusion Detection Deep Learning Models with Minimal Data Using Adversarial Domain Adaptation[C]// ASIA CCS '20: The 15th ACM Asia Conference on Computer and Communications Security. ACM, 2020.

[4] HAN X, PASQUIER T, SELTZER M. Provenance-based intrusion detection: opportunities and challenges[C]//10th USENIX Workshop on the Theory and Practice of Provenance (TaPP 2018). 2018.

[5] JENKINSON G, CARATA L, BYTHEWAY T, et al. Applying Provenance in APT Monitoring and Analysis: Practical Challenges for Scalable, Efficient and Trustworthy Distributed Provenance[C]//9th USENIX Workshop on the Theory and Practice of Provenance (TaPP 2017). 2017.

[6] HAN X, PASQUIER T, BATES A, et al. Unicorn: Runtime provenance-based detector for advanced persistent threats[C]//Network and Distributed System Security Symposium.2020.

[7] HOSSAIN M N, MILAJERDI S M, WANG J, et al. SLEUTH: Real-time attack scenario reconstruction from COTS audit data[C]//26th USENIX Security Symposium (USENIX Security 17). 2017: 487-504.

[8] MILAJERDI S M, GJOMEMO R, ESHETE B, et al. Holmes: real-time apt detection through correlation of suspicious information flows[C]//2019 IEEE Symposium on Security and Privacy (SP). IEEE, 2019: 1137-1152.

[9] Xiong C, Zhu T, Dong W, et al. Conan: A Practical Real-Time APT Detection System With High Accuracy and Efficiency[J]. IEEE Transactions on Dependable and Secure Computing,2022,19(1): 551-565.

[10] YADAV T, RAO A M. Technical aspects of cyber kill chain[C]//International Symposium on Security in Computing and Communication. Cham: Springer, 2015: 438-452.

[11] MITRE ATT&CK[OL]. https://attack.mitre.org/.

[12] YE Y, LI T, ADJEROH D, et al. A survey on malware detection using data mining techniques[J]. ACM Computing Surveys (CSUR), 2017, 50(3): 1-40.

[13] ZHANG X, ZHANG Y, ZHONG M, et al. Enhancing state-of-the-art classifiers with api semantics to detect evolved android malware[C]//Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. 2020: 757-770.

[14] WENG H, LI Z, JI S, et al. Online e-commerce fraud: a large-scale detection and analysis[C]//2018 IEEE 34th International Conference on Data Engineering (ICDE). IEEE, 2018: 1435-1440.

[15] BRANCO B, ABREU P, GOMES A S, et al. Interleaved sequence RNNs for fraud detection[C]//Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020: 3101-3109.

[16] WU Z, PAN S, CHEN F, et al. A comprehensive survey on graph neural networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 32(1): 4-24.

[17] SUN Y, HAN J. Mining heterogeneous information networks: principles and methodologies[J]. Synthesis Lectures on Data Mining and Knowledge Discovery, 2012, 3(2): 1-159.

[18] DONG Y, CHAWLA N V, SWAMI A. metapath2vec: Scalable representation learning for heterogeneous networks[C]//Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017: 135-144.

[19] LIU Y, ZHANG M, LI D, et al. Towards a Timely Causality Analysis for Enterprise Security[C]//NDSS. 2018.

[20] BARRE M, GEHANI A, YEGNESWARAN V. Mining data provenance to detect advanced persistent threats[C]//11th International Workshop on

Theory and Practice of Provenance (TaPP 2019). 2019.

[21] BERRADA G, CHENEY J, BENABDERRAHMANE S, et al. A baseline for unsupervised advanced persistent threat detection in system-level provenance[J]. Future Generation Computer Systems, 2020, 108: 401-413.

[22] XIANG Z, GUO D, LI Q. Detecting mobile advanced persistent threats based on large-scale DNS logs[J]. Computers & Security, 2020, 96: 101933.

[23] ZIMBA A, CHEN H, WANG Z, et al. Modeling and detection of the multi-stages of advanced persistent threats attacks based on semi-supervised learning and complex networks characteristics[J]. Future Generation Computer Systems, 2020, 106: 501-517.

[24] LECUN Y, BENGIO Y, HINTON G. Deep learning[J]. Nature, 2015, 521(7553): 436-444.

[25] DU M, LI F, ZHENG G, et al. Deeplog: Anomaly detection and diagnosis from system logs through deep learning[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017: 1285-1298.

[26] SHEN Y, MARICONTI E, VERVIER P A, et al. Tiresias: Predicting security events through deep learning[C]//Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018: 592-605.

[27] EKE H N, PETROVSKI A, AHRIZ H. The use of machine learning algorithms for detecting advanced persistent threats[C]//Proceedings of the 12th International Conference on Security of Information and Networks. 2019: 1-8.

[28] LIU F, WEN Y, ZHANG D, et al. Log2vec: A heterogeneous graph embedding based approach for detecting cyber threats within enterprise[C]//Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. 2019: 1777-1794.

[29] Linux Auditd[OL]. https://linoxide.com/auditd-tool-security-auditing/.

[30] Windows ETW[OL]. https://docs.microsoft.com/en-us/windows-hardware/drivers/devtest/event-tracing-for-windows--etw-.

[31] WANG J, HUANG P, ZHAO H, et al. Billion-scale commodity embedding for e-commerce recommendation in alibaba[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018: 839-848.

[32] PEROZZI B, AL-RFOU R, SKIENA S. Deepwalk: Online learning of social representations[C]//Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2014: 701-710.

[33] GROVER A, LESKOVEC J. node2vec: Scalable feature learning for networks[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016: 855-864.

[34] TANG J, QU M, WANG M, et al. Line: Large-scale information network embedding[C]//Proceedings of the 24th International Conference on World Wide Web. 2015: 1067-1077.

[35] WANG X, JI H, SHI C, et al. Heterogeneous graph attention network[C]//The World Wide Web Conference. 2019: 2022-2032.

[36] ZHANG D, YIN J, ZHU X, et al. Metagraph2vec: Complex semantic path augmented heterogeneous network embedding[C]//Pacific-Asia Conference on Knowledge Discovery and Data Mining. Cham: Springer, 2018: 196-208.

[37] SCHLICHTKRULL M, KIPF T N, BLOEM P, et al. Modeling relational data with graph convolutional networks[C]//European Semantic Web Conference. Cham: Springer, 2018: 593-607.

[38] GEHANI A, TARIQ D. SPADE: Support for provenance auditing in distributed environments[C]//ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing. Berlin: Springer, 2012: 101-120.

[39] VELIČKOVIĆ P, CUCURULL G, CASANOVA A, et al. Graph attention networks[C] // International Conference on Learning Representations. 2018.

[40] WANG S, WANG Z, ZHOU T, et al. threa Trace: Detecting and Tracing Host-based Threats in Node Level Through Provenance Graph Learning[J]. arXiv:2111.04333, 2021.

[41] ZHU X J,ZOU B, GHAHRAMAN I. Learning from labeled and unlabeled data with label propagation: Tech. Rep., Technical Report:CMU-CALD-02--107[R]. Carnegie Mellon University,2002.

DONG Chengyu, born in 1996, postgraduate. His main research interests include data mining and graph neural networks.

LÜ Mingqi, born in 1982, Ph. D, associated professor, is a member of China Computer Federation. His main research interests include data mining and ubiquitous computing.

（责编：柯颖）