计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2021.20200942
58(5):1006-1020,2021

隐私保护的基于图卷积神经网络的攻击溯源方法

李 腾[1] 乔 伟[2] 张嘉伟[1] 高怿旸[3] 王申奥[1] 沈玉龙[2] 马建峰[1]
[1](西安电子科技大学网络与信息安全学院 西安 710071)
[2](西安电子科技大学计算机科学与技术学院 西安 710071)
[3](西安电子科技大学人工智能学院 西安 710071)
(litengxidian@gmail.com)

Privacy-Preserving Network Attack Provenance Based on Graph Convolutional Neural Network

Li Teng[1], Qiao Wei[2], Zhang Jiawei[1], Gao Yiyang[3], Wang Shenao[1], Shen Yulong[2], and Ma Jianfeng[1]
[1](*School of Cyber Engineering*, *Xidian University*, *Xi'an* 710071)
[2](*School of Computer Science and Technology*, *Xidian University*, *Xi'an* 710071)
[3](*School of Artificial Intelligence*, *Xidian University*, *Xi'an* 710071)

Abstract APT(advanced persistent threat) attacks have a long incubation time and a vital purpose. It can destroy the inside's enterprise security fortress, employing variant Trojans, ransomware, and botnet. However, the existing attack source tracing methods only target a single log or traffic data, making it impossible to trace the complete process of multi-stage attacks. Because of the complicated log relationship, serious state explosion problems will occur in the log relationship graph, making it difficult to classify and identify attacks accurately. Simultaneously, data privacy protection is rarely considered in using log and traffic data for attack tracing approaches. We propose an attack tracing method based on a Graph Convolutional Network (GCN) with user data privacy protection to solve these problems. Supervised learning solves the state explosion caused by multiple log relationship connections, optimizing the Louvain community discovery algorithm to improve detection speed and accuracy. Moreover, using map neural networks to attack classification effectively and combining privacy protection scheme leveraging CP-ABE (Ciphertext-Policy Attribute Based Encryption) properties realize log data secure sharing in public cloud. In this paper, the detection speed and efficiency of four APT attack testing methods are reproduced. Experimental results show that the detection time of this method can be reduced by 90% at most, and the accuracy can reach 92%.

Key words attack provenance; graph convolutional neural network; privacy preserving; data access control; attribute-based encryption

收稿日期:2020-11-09;**修回日期**:2021-03-12

基金项目:国家自然科学基金青年科学基金项目(61902291);中国博士后基金项目(2019M653567);陕西省自然科学基金项目(2019JM-425);中央高校基本科研业务费专项资金(JB191507)
This work was supported by the National Natural Science Foundation of China (61902291), the China Postdoctoral Science Foundation (2019M653567), the Natural Science Foundation of Shaanxi Province of China (2019JM-425), and the Fundamental Research Funds for the Central Universities (JB191507).

通信作者:张嘉伟(zjw8512@126.com)

摘　要　APT(advanced persistent threat)攻击潜伏时间长,目的性强,会通过变种木马、勒索病毒、组建僵尸网络等手段从内部瓦解企业安全堡垒.但现有攻击溯源方法都只针对单一日志或流量数据,这导致了无法追溯多阶段攻击的完整过程.并且因为日志条目间关系复杂,日志关系图中会产生严重的状态爆炸问题,导致难以对攻击进行准确的分类识别.同时,在利用日志及流量数据进行攻击溯源过程中,很少考虑到数据隐私保护问题.为解决这些问题,提出了一种具有隐私保护的基于图卷积神经网络的攻击溯源方法.通过监督学习解决了因多日志关系连接导致的状态爆炸,对 Louvain 社区发现算法进行优化从而提高了检测速度及准确性,利用图卷积神经网络对攻击进行有效的分类,并结合属性基加密实现了日志数据的隐私保护.通过复现 4 种 APT 攻击测试方法来检测速度和效率.实验结果表明:该方法的检测时间最多可有 90%的缩减,攻击溯源准确率可达 92%.

关键词　攻击溯源;图卷积神经网络;隐私保护;数据访问控制;属性基加密

中图法分类号　TP18; TN915.08

在各类网络安全问题中,高级持续性威胁(advanced persistent threat, APT)现在是各大企业、政府所要面对的最大问题.智能系统通常“外部安全堡垒”建设完善,但是,APT 攻击多以商业信息和政治安全为攻击目的,具有高度的隐蔽性,往往经过了长期的经营与策划.APT 攻击已在世界范围产生了巨大的危害,2019 年 4 月俄罗斯背景组织 APT28(Fancy Bear)攻击乌克兰 2019 年大选活动①,6 月 14 日美国 E-ISAC 发现 Triton 针对美国境内智能电网的探测活动②.检测并对 APT 攻击进行溯源,已成为了国家、企业所要解决的重要问题.如果安全攻防信息持续不对等,系统方将很难发现 APT 攻击的蛛丝马迹.由于 APT 攻击表现行为复杂多变,使用传统人工规则分析方式将存在很多局限,人工智能技术是解决此类问题的重要手段.

APT 攻击持续时间长,空间跨度大,前期采用社会工程学等手段对目标进行反复侦察并获取有用信息,其行为往往难以探测并具有伪装性.当攻击者获取目标的信任并能够以不被发现的身份潜藏在目标网络中时,便会盗取信息并不断扩大其感染范围,严重的会使整个目标网络瘫痪.一次完整的攻击事件由多个阶段构成,主要包括信息搜集、漏洞利用、建立据点、权限提升、权限维持、横向移动、痕迹清除 7 步,期间涉及了多个攻击程序和步骤,这为实现对攻击的快速溯源、构建攻击溯源图、及准确判断 APT 攻击的类型带来了很大的挑战.传统的攻击溯源会将日志字段进行关键字相似度关联匹配,由于攻击者的隐秘性以及在系统中攻击事件所产生日志的关联性,使得溯源事件关联分析时会产生状态爆炸问题.除此之外,传统的访问控制,流量监控等技术手段针对 APT 攻击的检测和溯源早已不能适用,针对越来越智能的攻击者和丰富的攻击手段来说,检测和溯源技术必须引入人工智能的手段来帮助攻击事件之间的关系查找及具体攻击类型的鉴别[1].此外,由于日志及流量数据中往往包含大量的重要数据和敏感信息,因此,在检测攻击时必须对日志数据的访问进行限制从而保护数据隐私.

为解决这一问题,现有的工作[2-5]对日志记录展开了分析,然而现有的溯源方案仍存在着大量问题.当下很多的攻击检测都基于单个日志,他们对 DNS,HTTP 等单一日志进行流量分析[1-2],然而 APT 攻击多是多步攻击,其攻击踪迹隐藏在多个日志之中,单一日志分析很难完整地提取攻击社区.一次完整的 APT 攻击过程中不只有异常事件,而传统单点检测的方法只能发现某阶段的局部异常,没有关注日志间联系,无法从局部异常还原出整体攻击过程.同时,异常检测技术存在着严重的误报情况.并且,因为日志间关系错综复杂,连接日志关系时会产生严重的状态爆炸[6-7],为解决因此产生的状态爆炸问题,现有的工作[8-11]很多集中在了数据的筛选与精简,然而其实际表现并不一定可以很好地优化数据关系,而且可能会影响数据的有效性.MARD[12]和 Holmes[13]可以实时构造 APT 的攻击图谱,但是没有对攻击进行特定类型的识别,这不利于后续 APT 攻击的防范和分析.

针对 APT 攻击的溯源和分类问题,需要进行

① https://www.fifthdomain.com/international/2019/02/13/ukrainian-official-hacking-intensifies-as-election-nears/

② https://www.wired.com/story/triton-hackers-scan-us-power-grid/

多日志的关系连接、解决状态爆炸和隐私保护问题，本文提出了具有隐私保护的图卷积神经网络攻击溯源方法.本文首先通过正则匹配处理多日志信息，将日志解析为格式化条目.然后通过设定的特征关系，构建多维特征向量，从而构造条目关系网.对构建好的关系网进行权重优化，输入社区检测模块，提取出需要的攻击社区.同时，在外包日志数据时，采用一种日志隐私保护方法来实现多用户安全数据共享.在数据共享方案中，我们不仅实现了半隐藏访问策略来保护隐私信息，还实现了基于时间的细粒度访问控制，任何用户只有持有有效的属性密钥才可以访问共享的日志数据.最后本文将构造好的溯源图利用图卷积神经网络进行特定类型的分类，告诉管理方具体的攻击类型.

本文模拟了目前流行的4种APT攻击，在Linux，Windows系统平台都有实验，过程中收集了可能会留有痕迹的多种日志，依据方案给出的特征维度，构建了事件之间的关系，并通过基于TensorFlow构造的GCN框架对攻击进行了有效的分类.实验结果表明了多日志分析可以更完善的展示攻击社区，并且通过对Louvain的优化，运算时间最多可有90%的缩减，准确率可达92%.所使用的日志隐私保护方法，不仅支持大规模属性集，而且通过引入高性能的质数群，计算开销减少了83%以上，密文的存储开销也缩减了50%以上.

本研究主要有4个创新点：

1) 提出基于监督学习的攻击关系图优化方法，解决了状态爆炸问题，首次利用剪枝判断、连接优化的方法，提升社区检测的速度与准确性.

2) 通过隐私保护的大规模属性基加密方法，实现了日志文件在服务器存储的多用户安全共享和细粒度访问控制.

3) 利用图卷积神经网络对构造的溯源图进行基于人工智能的有效分类，准确告诉安全管理员攻击的类型，方便后续的攻击防御.

4) 实验过程复现了流行的多种APT攻击，收集产生的多种日志数据.展示了本方法通过日志分析构造出完整的攻击社区，并对分析结果做出了评估，很好地证明了系统的有效性.

1 相关工作

人工智能攻击检测技术：现有方法[14-16]使用具有代表性的训练数据和序列处理技术，从过去的事件学习并预测接下来的事件.其中文献[14]识别日志条目之间的关系，但忽视了事件之间的关系.然而，日志级别的粒度不能反映攻击者的真实意图，使得安全人员不能有效地识别和防范相应的攻击.Deeplog[14]和Tiresias[15]使用LSTM算法学习攻击日志并且对攻击进行预测，APT自动分类器[16]研究真实APT的已有行为，NoDoze[17]学习异常的攻击事件的特征从而对攻击进行检测，PrioTracker[18]学习异常流量数据特征并对流量数据进行检测.但这些方法忽略了41%的APT攻击技术和使用的软件都是未知，它们不能简单地从历史数据中学习并且进行预测.当攻击者使用新型的恶意软件或者未知方法时，攻击就能避开检测.大多数方法只考虑多个日志之间的空间相关性或单个日志的事件相关性，没有融入人工智能手段，不能进行有效的分类.

通常攻击者在发动一个APT攻击破坏前，就已经开始在进入系统之前为其做准备，他们通常扫描并收集情报，以更好地了解一个目标如何工作及其潜在的弱点.对于APT的溯源，Log2vec[19]使用系统审计日志来检测内部威胁，UIScope[20]利用Windows事件追踪(ETW)日志也只能追踪到进入系统内部的攻击事件.这2个方法关注系统内部威胁而忽略了破坏系统之前的准备工作.在后续的因果分析工作中，会确立事件之间的关联关系，由于是通过字段匹配来进行关联分析，当前事件依赖于前面所有事件，即会产生状态爆炸问题[21-22]，这会导致溯源图构造的分析结果不准确.为解决这一问题，SLEUTH[23]和MORSE[24]利用了编码可信度和数据敏感性的评估标签，但在APT场景中，我们不能从原始审计日志中找出标签.OmegaLog[25]构造了一个带有应用程序日志的通用起源图(UPG)，但跨应用程序的事件因果关联增加了解决问题的难度.UIScope[22]将系统事件属性归为高级UI元素，这种方法不能普遍适用于系统.

最近，策略隐藏的密文策略属性基加密(Hidden Policy Ciphertext-Policy Attribute Based Encryption，HP-CP-ABE)方案[26]得到了广泛的关注和扩展.然而，这类方案中的访问策略本身可能会包含用户隐私，这给用户隐私带来很大威胁.因此，Nishide等人[27]在2008年提出了第1个部分隐藏访问策略的属性基加密方案，但是仅支持与逻辑访问策略；2018年文献[28]中提出了完全隐藏访问策略的方案以支持访问策略中的属性完全隐藏，多属性机构和多种访问策略，为了降低该方案的开销，文献[28]扩展了

该方案并引入外包解密和可验证机制;由于完全隐藏策略存在高计算复杂度等缺点,文献[29]基于合数阶群提出了一种具备解密测试的部分隐藏访问策略方案;之后,文献[30]提出了基于质数阶的部分隐藏访问策略方案以提高效率;2020 年 Han 等人[31]提出了一种支持可撤销可追踪功能的部分隐藏策略的属性基加密方案.然而,该方案无法提供基于时间的用户密钥有效期的功能.

本文提出基于时间的细粒度访问控制.只有具有足够权限的用户在有效访问时间范围内才可以访问存储在服务器的共享日志数据.同时,共享日志数据的机密性需要得到保护.由于访问策略是随着密文一起在服务器存储和共享,因此,访问策略的值能够隐藏,保证用户隐私不被泄露.

2　攻击示例

一次完整的 APT 攻击涉及了多个程序、多个文件,其攻击痕迹分布在了多个日志之中,如图 1 所示.阶段 1 用户点击了钓鱼邮件中的链接,通过链接下载了含有恶意命令的文档;阶段 2 用户运行文档后,通过执行恶意命令,打开了 PowerShell 后门;阶段 3 便可访问用户数据、植入病毒.攻击针对性强,隐蔽能力高,完整的 APT 攻击过程不仅仅有异常行为,还有很多正常的程序过程,而且各阶段攻击采用的手段极为丰富,现有的异常检测系统并不能很好地对此进行分析.对此,本文给出了基于多日志的攻击社区提取方法.

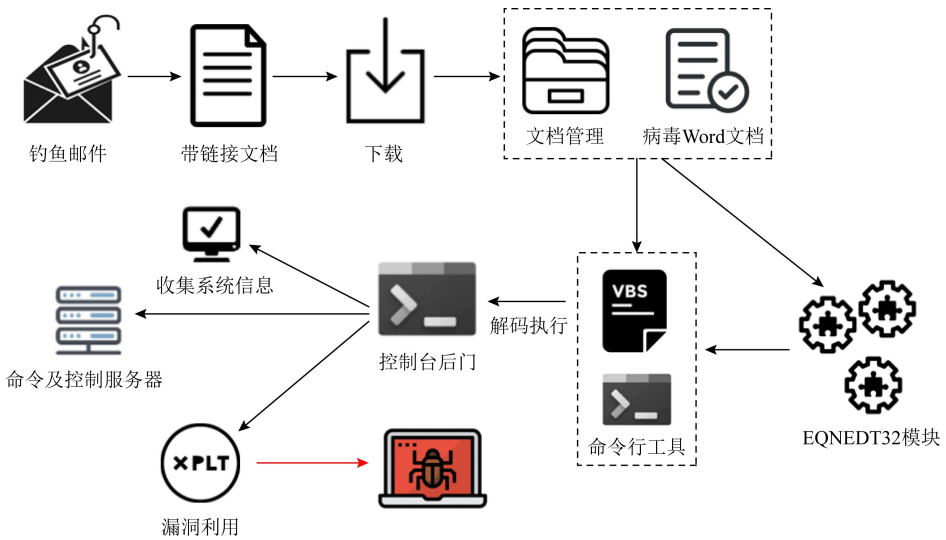


Fig. 1　Microsoft Office memory corruption vulnerability

图 1　Microsoft Office 内存损坏漏洞

3　系统概览

完整的系统概览,如图 2 所示.整个系统包含了 4 个实体:密钥产生中心、日志生产者、日志分析者和服务器.其中①密钥产生中心,该实体负责初始化整个系统,生成系统公共参数和主密钥,并为系统中的用户分发密钥;②日志生产者在运行过程中生产大量的日志数据,为了减轻系统的本地负担,需要将日志数据加密后上传至服务器进行共享;③日志分析者负责各种日志数据的分析,并检测其中的攻击.在日志分析之前,该实体需要从服务器下载具体的加密日志文件并进行解密;④服务器提供海量日志存储和共享服务,并根据用户的请求返还对应的日志文件.

本文系统运行过程分为 4 个部分,通过多个处理模块,提取所需要的攻击社区,从而更好的分析与解决 APT 攻击,并对攻击进行分类.阶段 1,本文方法中的日志生产者收集的多种日志信息(如 firewall 日志、网络日志、系统日志)加密存储到服务器中.在需要日志分析时,合法的日志分析者从服务器中获取日志密文并解密.然后,日志分析者依据预设的提取字段,解析为信息完整且易于系统操作的结构化日志条目.将条目节点依据特征关系连接,构成关系图.因日志关系连接紧密,关系图的构建会产生的严重的状态爆炸,所以,阶段 2 本文方法采用了监督

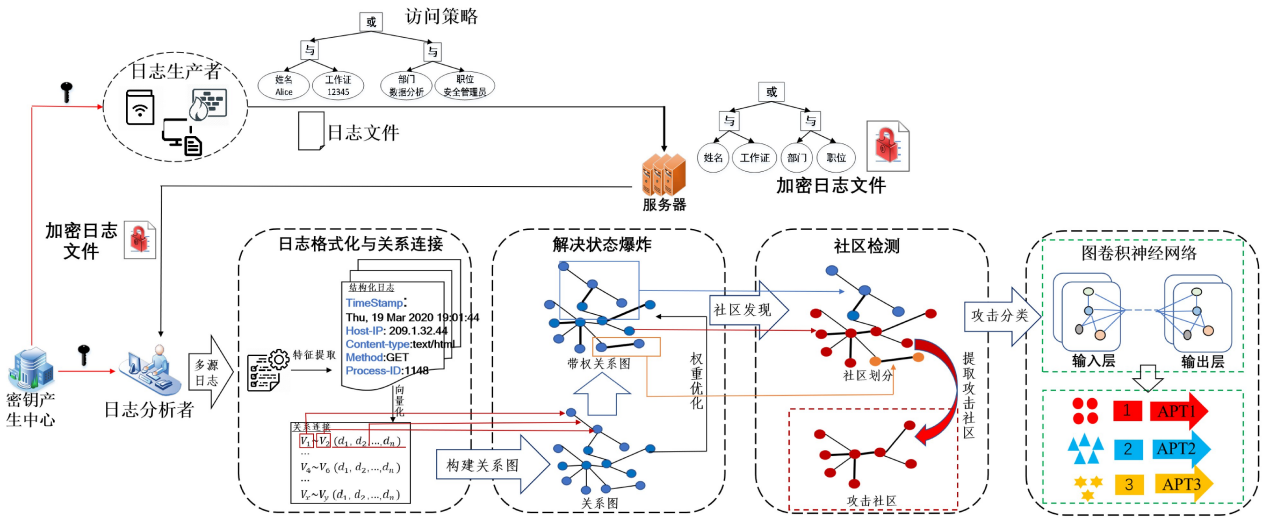


Fig. 2 Atlack provenance system framework

图 2 攻击溯源系统框架

学习的方式,重构关系图,生成带权关系图.在阶段 3 进行社区检测,划分社区.通过社区发现提取所需的攻击社区.最后,本文利用图卷积神经网络对划分的社区进行攻击分类.

4 系统设计

4.1 日志格式化与关系链接方法

我们通过爬虫、抓包等方式收集系统、网络(HTTP, DNS, UDP/TCP)、应用(Chrome, Email, Office)等产生的日志,输入解析程序.解析程序中,将所需的特征量表达为正则式通过正则匹配方式,解析日志,如将系统的 DNS 日志"26-June-2020 09:18:46.056 queries: client 192.168.1.106 # 39164 (www.baidu.com): query: www.baidu.com IN AAAA+(45.116.155.55)"解析为表 1 所示的结构化实体:

Table 1 Structured Entity

表 1 结构化实体

序号	样例	名称
1	26-June-2020	日期
2	09:18:46.056	时间戳
3	192.168.1.106	源 IP 地址
4	39164	端口号
5	www.baidu.com	域名
6	AAAA	解析记录
7	45.116.155.55	域名系统地址

表 2 给出了解析中预设提取的条目特征,如进程号、端口号等,我们在表 3 中给出了这些特征的关系连接.

Table 2 Feature Description

表 2 特征描述

特征	描述
TimeStamp	时间戳
Pid	进程号
P-pid	父进程号
Pname	进程名称
H-ip	主机 IP 地址
D-ip	源 IP 地址
H-port	主机端口号
D-port	目的端口号
Path	绝对路径
Objname	对象名称
Ctype	网络媒体类型
Status	网络状态

Table 3 Feature Relation Connection

表 3 特征关系连接

维度	描述
d_1	$(u.timestamp\text{-}v.timestamp)<t$
d_2	$u.Pid=v.Pid$
d_3	$u.P\text{-}pid=v.P\text{-}pid$
d_4	$u.Pname=v.Pname$
d_5	$u.H\text{-}ip=v.H\text{-}ip$

续表 3

维度	描述
d_6	$u.D\text{-}ip=v.D\text{-}ip$
d_7	$u.H\text{-}port=v.H\text{-}port$
d_8	$u.D\text{-}port=v.D\text{-}port$
d_9	$u.Type=v.Type$
d_{10}	$u.Status=v.Status$
d_{11}	$u.Path=v.Pname$
d_{12}	$u.Path=v.Objname$
d_{13}	$u.Objname=v.Objname$

d_1:在一个阈值时间内的事件往往有一定的相关,所以我们在该维度通过时间设定了关系的连接.如我们接受钓鱼邮件、下载病毒文档、执行文档会发生在同一时间段下.

$d_2\sim d_4$:APT 攻击在靶机中多是由多个进程多次攻击完成,所以 $d_2\sim d_4$ 维度表示了进程的相关性.如一个进程的溢出,从而产生新的进程,发起真正的攻击,所以进程号、名称、进程的父子关系都可以很好地联系 2 个日志事件.

$d_6\sim d_{10}$:网络攻击中检测 IP 地址、端口号、传输类型、传输状态是否相同,可以很重要地反映攻击来源的相关性、攻击方式的相关性,所以为其建立了特征联系.

$d_{11}\sim d_{13}$ 反映了进程与网络路径对文件的访问,如某一网络事件 u 下载恶意程序,另一进程 v 执行了该程序,则 u 和 v 应属于同一攻击路径.或者是进程 A 创建了恶意对象 obj,而进程 B 访问了 obj,进程 A,B 间也应存在攻击联系.

将解析后的格式化数据输入,通过日志条目的特征关系连接节点构成未加权无向图.形成一个 n 维网络图 $G(V,E,D)$.其中 V 是一节点事件集合,表示日志条目,E 是日志条目间关系构成的边,D 是 n 维的特征关系.最终形成了 $V\times V\times D$ 的 3 维矩阵 $\boldsymbol{M}$,i 和 j 表示 2 个节点条目,则 $M_{i,j,k}=1$ 表示节点 i 和节点 j 间存在第 k 维的关系,$M_{i,j,k}=0$ 则表示节点 i 和节点 j 间没有关系.E 中的边 e 可以表示为 $\{(i,j,d_1,d_2,\cdots,d_n)\mid i,j\in V,d_k\in D\}$.

4.2　解决状态爆炸问题

关系构图后,因日志条目间的关系错综复杂,不对特征量进行权重分配,会出现很明显的状态爆炸现象,而现有的很多攻击分析没有解决状态爆炸问题,直接对构成的关系网络图检测,所得结果会因攻击社区与非攻击社区间的关系连接产生很大的影响.为了解决这一问题,我们给出了使用监督学习优化权重的方法.我们将日志条目分为 A,B 社区,其中 A 社区包含攻击相关的日志条目,B 社区包含攻击无关的日志条目.则 e_A,e_B 分别表示 A,B 社区内部的边,e_{AB} 为 2 个社区间的边,为了更好地区分攻击社区与非攻击社区,我们需要使 $e_A\gg e_{AB}$ 且 $e_B\gg e_{AB}$,所以我们使用了 Logistic Regression,通过学习找到一个权重向量 $\boldsymbol{\alpha}$,可以使权重关系满足不等式条件:

$$\begin{aligned}&\sum_{e\in e_A}\sum_{i=1}^{k}\alpha_i\times e_i>\sum_{e\in e_{AB}}\sum_{i=1}^{k}\alpha_i\times e_i,\\&\sum_{e\in e_B}\sum_{i=1}^{k}\alpha_i\times e_i>\sum_{e\in e_{AB}}\sum_{i=1}^{k}\alpha_i\times e_i.\end{aligned}\tag{1}$$

为了防止 $\boldsymbol{\alpha}$ 对边加权后出现负权,我们通过函数设置将权重范围映射到[0,1],

$$w=S\Big(\sum_{i=1}^{k}\alpha_i\times e_i\Big)=\frac{1}{1+\mathrm{e}^{-\sum_{i=1}^{k}\alpha_i\times e_i}}.\tag{2}$$

对 m 条训练边,定义 $E=(x_i,y_i)$,$i\in[1,m]$,x_i 表示第 i 条训练边 e,若 $e\in e_{AB}$,则 $y_i=1$,否则 $y_i=0$,构造函数 g:

$$h_{\boldsymbol{\alpha}}(x_i)=g(\boldsymbol{\alpha}^{\mathrm{T}}x_i)=\frac{1}{1+\mathrm{e}^{-\boldsymbol{\alpha}x_i}}.\tag{3}$$

我们用 $h_{\boldsymbol{\alpha}}(x_i)$ 表示 $e\in e_{AB}$,$y_i=1$ 的概率,反之,概率为 $1-h_{\boldsymbol{\alpha}}(x_i)$,我们用对数似然法最小化成本函数:

$$-\frac{1}{m}\Big[\sum_{i=1}^{m}y_i\ln h_{\boldsymbol{\alpha}}(x_i)+\sum_{i=1}^{m}(1-y_i)\ln(1-h_{\boldsymbol{\alpha}}(x_i))\Big].\tag{4}$$

4.3　社区检测方法

关系图加权后,我们给出了增强型 Louvain 算法对节点进行分类,找出攻击社区.

社区发现中通过模块度评估一个社区网络划分好坏的程度.其定义为

$$\begin{aligned}&Q=\frac{1}{2m}\sum_{i,j\in\mathbf{Z}}\left[A_{ij}-\frac{k_ik_j}{2m}\right]\delta(c_i,c_j),\\&\delta(u,v)=\begin{cases}1,&\text{若 }u=v,\\0,&\text{其他},\end{cases}\end{aligned}\tag{5}$$

其中,A_{ij} 为节点 i,j 间边的权重,$k_i=\sum_{j\in\mathbf{Z}}A_{ij}$ 表示所有与节点 i 相连的边的权重之和,c_i 表示节点 i 所属的社区,$m=\frac{1}{2}\sum_{i,j\in\mathbf{Z}}A_{ij}$ 表示所有边的权重之和.基于模块度,我们给出了增强型 Louvain 算法.

4.3.1 社区划分

将图中的每个节点初始化为独立的社区,然后对每个节点 i,依次尝试分配到其邻居节点所在的社区,计算移动前后的模块度变化 ΔQ,并记录 ΔQ 最大的邻居节点,若 $\max \Delta Q>0$,则把节点 i 分配给 ΔQ 最大的居节点所在的社区,否则不移动社区.实验发现,节点 i 从 A 移动到另一社区 B 后,我们只需将不在社区 B 中节点 i 的邻居节点加入节点判读队列即可,无需判断全部节点.这样可以大幅提升算法速度,且对分区准确率影响很小.

算法 1 中,先将每个节点初始化为单节点社区,设置一个空队列,节点以随机顺序添加到队列中,然后我们从队首取出节点,通过质量函数,判断该节点是否应该移动至其他社区,若该节点被移动,则会将不属于新社区且尚未加入队列的所有邻居节点加入队列,然后我们不断从队列中取出节点进行移动操作,一直持续到队列为空.

算法 1. 快速移动节点算法.

输入:图 G;

输出:节点所在社区 C.

① $FastMoveNodes(G)$;

② $C \leftarrow Initial(G)$;

③ $Q \leftarrow Queue(V(G))$;

④ while $Q \neq \varnothing$ do

⑤ $v \leftarrow Q.pop()$;

⑥ $best\_commnuity \leftarrow SelectBeatCommunity()$;

⑦ $N \leftarrow \{u \mid (u,v) \in E(G), u \notin (best\_community \cup Q)\}$;

⑧ $Q.push(N)$;

⑨ end while

⑩ Return C.

4.3.2 连接优化与节点聚合

社区划分过程中,每次仅考虑单个节点的移动,因此可能会出现当 1 个节点离开社区后,社区内节点断开的情况,为了解决在社区划分过程产生的连接不良,我们对划分的后社区进行了进一步的连接优化.

在划分社区后,对每个社区进行精炼,然后基于精炼社区聚合网络,避免聚合网络时会将断开的社区聚合为新的节点.并且在精炼划分过程中,使用贪婪移动限制太大,以至于不能达到最优划分,而通过随机性的选择社区允许更广泛地探索分区空间.所以我们让节点可以与使模块度增加的任意社区随机合并,模块度增加越大,移动可能性便越大.其中社区的随机选择程度通过参数 $\theta>0$ 影响.

分区精炼算法中,首先将 G 中的每个节点初始化为一个单节点社区,对于本地移动后划分好的社区,我们通过判断其节点的连接度是否良好,选择连接良好的节点进一步分析,对于良好连接的节点集合中未被合并的点,我们挑选连接良好的社区,将节点依模块度的提升随机移入,未被并入社区节点单独成为社区.

基于精炼分区,我们对整个网络图进行压缩,将一个社区的节点压缩为一个新的节点,社区内节点间的边权转换为新节点的自环权重,社区间连接的边权转换为新节点间的边权.然后重新执行社区划分.

4.3.3 基于 GCN 的网络威胁分类

图是一种常用的数据结构类型,它由若干个顶点以及边组成,通过边可以将顶点之间的连接关系进行直观的表示.CNN 卷积神经网络能对目标的特征信息进行提取,从而学习到输入与输出之间的映射关系.但由于拓扑图中每个顶点的度都不同,因此 CNN 在非欧几里得数据上无法保持平移不变性.而 GCN 可以较好地解决这一问题,提取拓扑图的空间特征.

GCN 是一种基于图的深度神经卷积网络,它能利用拉普拉斯矩阵对图的特征进行分析,从而完成分类等工作.GCN 结构通常可分为输入层、若干隐藏层和输出层,其中隐藏层用于完成图上的卷积操作.

假设无向图 $G=(V,E)$,其中 V,E 分别代表顶点和边的集合.矩阵 $\boldsymbol{X},\boldsymbol{A}$ 分别为节点的特征矩阵和邻接矩阵.则 GCN 的层间传播规则为

$$\boldsymbol{H}^{(l+1)}=\sigma(\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}), \tag{6}$$

其中,$\widetilde{\boldsymbol{A}}=\boldsymbol{A}+\boldsymbol{I}_N$ 为无向图的带自环邻接矩阵,$\boldsymbol{I}_N$ 为单位矩阵,$\widetilde{D}_{ii}=\sum_{j\in \mathbf{Z}}\widetilde{A}_{ij}$,$\boldsymbol{W}^{(l)}$ 为可训练权重矩阵,$\sigma(\cdot)$ 为该层使用的激活函数,$\boldsymbol{H}^{(l)}\in\mathbb{R}^{N\times D}$ 为第 l 层的激活矩阵,且 $\boldsymbol{H}^{(0)}=\boldsymbol{X}$.

对于一个 2 层的 GCN 模型,其前向传播公式为

$$Z=f(\boldsymbol{X},\boldsymbol{A})=\operatorname{softmax}(\hat{\boldsymbol{A}}\,ReLU(\hat{\boldsymbol{A}}\boldsymbol{X}\boldsymbol{W}^{(0)})\boldsymbol{W}^{(1)}), \tag{7}$$

其中,$\hat{\boldsymbol{A}}=\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}$,$\boldsymbol{W}^{(0)}\in\mathbb{R}^{C\times M}$ 为输入层到隐藏层的权重矩阵,该隐藏层共用 M 个特征映射,$\boldsymbol{W}^{(1)}\in\mathbb{R}^{M\times F}$ 为隐藏层到输出层的权重矩阵,F 为输出层的特征映射数,对应检测的类别数.

对于图 3 所示的 2 层 GCN 网络,分别使用 $ReLU$ 和 softmax 作为激活函数.其中 s 为输入,$ReLU$ 将

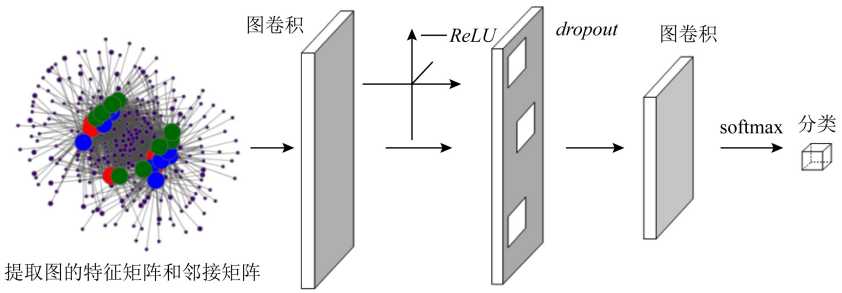


Fig. 3　Two layer GCN model

图 3　2 层 GCN 模型

输入 s 与 0 取最大值后输出,对输入具有单侧抑制的作用,使神经元具有稀疏激活性.

使用 $ReLU$ 激活函数可以解决非线性问题,有助于神经网络的快速收敛,$ReLU$ 表达式为

$$ReLU(s)=\max(0,s). \tag{8}$$

在输出层我们使用 softmax 作为激活函数,归一化后每个元素的输出为

$$\alpha_i=\frac{e^{z_i}}{\sum_{j=1}^{F}e^{z_j}},\forall i=1,2,\cdots,F, \tag{9}$$

softmax 函数可以将输出映射到(0,1)区间内的某个值,且各输出的加和为 1.若将映射后的输出视为概率,则可以选取概率最大的节点作为预测目标,也就是预测标签.

对于本文所需要解决的分类问题,我们使用交叉熵误差作为损失函数:

$$L=-\sum_{i\in\mathbf{Z}}\sum_{c=1}^{M}y_{ic}\mathrm{lb}(p_{ic}), \tag{10}$$

其中,M 为类别的数量;y_{ic} 为指示变量(取值为 0 或 1),若类别与样本 i 类别相同则取 1,否则取 0;p_{ic} 为观测样本 i 属于类别 c 的预测概率.

在训练过程中,我们使用梯度下降法对权重矩阵 $\boldsymbol{W}^{(0)}$ 和 $\boldsymbol{W}^{(1)}$ 进行训练,每次训练迭代对整个数据集进行批量梯度降.并通过 $dropout$ 向训练过程中引入随机性,防止模型过拟合,提高泛化能力.

在验证模型的有效性时,只需把样本送入训练完成的网络中即可得到预测标签.例如输入样本图的邻接矩阵 $\boldsymbol{A}$ 和特征矩阵 $\boldsymbol{X}$,根据 GCN 网络的前向传播公式式(7)以及训练好的权重矩阵,可以得到网络输出,并用 softmax 激活函数对输出进行归一化处理,选择最大概率的输出所对应的标签作为预测标签,并与真实标签进行对比,验证模型预测的准确性.

4.4　日志隐私保护方法

任何一种日志分析方法,都需要大量的日志数据作为支撑,这需要各种系统,包括资源受限的系统,不停产生大量的日志数据.因此,需要将日志数据外包存储到服务器中进行共享,以减轻系统的本地负担.

在图 2 所示的系统框架中,根据文献[30],整个系统运行时,共享服务器被认为"半可信"的,该实体按照指定协议提供服务,但是可能会通过分析用户日志数据获取隐私和机密信息.日志生产者被认为是可信的且不与服务器进行串谋.密钥产生中心被认为是可信的并且不与任何一方串谋.日志分析者被认为是不可信任的,存在一些恶意的日志分析者非法访问共享日志并且泄露其中的敏感和隐私信息,甚至篡改日志文件以影响其他日志分析者的分析结果.而日志文件包含大量的用户隐私和敏感信息,如果共享的日志文件被恶意日志分析者攻击或者被服务器分析和窃听,将会对用户隐私造成重大危害.因此,我们设计了一种新颖且高效的日志隐私保护方法.本文方法主要用于保护日志文件中敏感和隐私数据的安全共享.

首先,密钥产生中心生成系统公共参数和主密钥,密钥产生中心运行一个双线性群[30]生成算法生成 2 个阶为素数 p 的乘法循环群 G_1 和 G_2,以及一个双线性映射 $\hat{e}:G_1\times G_1\to G_2$.同时,选取一个随机对称加密算法$(Enc,Dec)$,产生一个群 G_1 中的生成元 g 以及几个随机的元素 $h,\mu\in G_1$.密钥产生中心同时负责属性管理和系统时间管理,每个属性包括属性名称和属性值 2 部分,系统时间的表示方式采取层次化时间树[32],此处我们假设时间树的深度为 ϕ,密钥产生中心为系统时间树随机选取 $T_0,T_1,\cdots,T_\phi\in G_1$ 和并用 2 个随机数 $\alpha,\beta\in\mathbb{Z}_p$ 作为系统主密钥同时计算系统公共参数.最后,密钥产生中心公开系统公共参数 PP 并在本地安全存储系统主密钥 MSK:

$$PP=\{p,G_1,G_2,\hat{e},g,h,\mu,T_0,T_1,\cdots,T_\phi,\hat{e}(g,g)^\alpha,g^\beta,(Enc,Dec)\},MSK=\{\alpha,\beta\}. \quad (11)$$

其次,密钥产生中心注册并校验用户的合法性,同时根据用户的属性集合生成对应的私钥,并指定该用户私钥的有效时间.对于一个用户 u,系统选取一个随机数 $r\in\mathbb{Z}_p$ 对其进行标识.而该用户的属性集合 S_u 中的每个属性 $a_i\in S_u$,计算对应的私钥元素 $D_i=g^{a_i r}\mu^{-\beta r}$.然后,系统对该用户设定一个密钥有效时间 T_v,并且根据该有效时间 T_v 的形式化表示:

$$\boldsymbol{T}=(\delta_1,\cdots,\delta_m,\cdots,\delta_k),$$

对于第 m 个时间元素 δ_m,选择一个随机数 $\sigma_m\in\mathbb{Z}_p$ 计算对应的私钥元素:

$$D_{m,1}=g^{\sigma_m},D_{m,2}=g^\alpha h^{\beta r}\left(T_0\prod_{j=1}^{\phi_m}T_j^{\delta_{m,j}}\right)^{\sigma_m}. \quad (12)$$

最后,密钥产生中心生成用户 u 的私钥为

$$SK_u=\{D_0=g^{\beta r},D_1=g^r,\{D_i\}_{a_i\in S_u},\{D_{m,1},D_{m,2}\}_{m\in[k]}\}. \quad (13)$$

当日志生产者产生日志数据 m 后,指定一个具体的访问策略 $\boldsymbol{A}_0=(\boldsymbol{A},\rho)$[31],其中 $\boldsymbol{A}$ 是一个 $l\times n$ 的秘密生成矩阵,ρ 是一个从 A 的每一行到属性名称索引的映射,随机生成一个对称会话密钥 $ssk\in G_2$,并使用系统中的随机对称加密算法 Enc 对 m 加密得到 $CT_1=Enc(ssk,m)$.接着,随机获取一个随机秘密值 $s\in\mathbb{Z}_p$,计算 $C_0=ssk\times\hat{e}(g,g)^{\alpha s}$ 从而对会话密钥 ssk 进行细粒度访问控制保护.之后,计算 $C_1=\left(T_0\prod_{j=1}^{\phi_m}T_j^{\delta_{m,j}}\right)^s$,将解密持续时间变量嵌入到密文中.选择一个随机元素 $t_x\in\mathbb{Z}_p$ 和随机向量 $\boldsymbol{\gamma}=(s,\gamma_1,\cdots,\gamma_n)$,其中,$\gamma_2,\cdots,\gamma_n\in\mathbb{Z}_p$.对于秘密生成矩阵 $\boldsymbol{A}$ 的每一行 $\boldsymbol{A}_x$,计算 $\lambda_x=\boldsymbol{A}_x\times\boldsymbol{\gamma}$.得到密文元素:

$$C_{x,1}=h^{\lambda_x}\mu^{t_x},C_{x,2}=g^{-t_x v_x},C_{x,3}=g^{t_x}, \quad (14)$$

其中,v_x 是 $\rho(x)$ 对应的属性值.最后,算法隐藏访问策略中的属性值得到 $\overline{\boldsymbol{A}}$,并将其嵌入到密文中进行输出:

$$\boldsymbol{CT}=(\overline{\boldsymbol{A}},CT_1,C_0,C_1,C_2,(C_{x,1},C_{x,2},C_{x,3})_{x\in\{1,2,\cdots,l\}}). \quad (15)$$

日志密文生成之后,日志生产者将该密文上传到服务器中,以节省本地的存储空间.

当日志分析者需要分析日志检测攻击时,从服务器请求得到加密日志数据 $\boldsymbol{CT}$,用自己的密钥 SK_u 按照步骤进行计算:

$$F=\hat{e}(D_{m,2},C_2)=\hat{e}(g^\alpha h^{\beta r}(T_0\prod_{j=1}^{\phi_m}T_j^{\delta_{m,j}})^{\sigma_m},g^s)$$
$$e(g,g)^{\alpha s}\hat{e}(g,h)\hat{e}(g,(T_0\prod_{j=1}^{\phi_m}T_j^{\delta_{m,j}}))^{\sigma_m s}, \quad (16)$$

$$P_x=\hat{e}(D_0,C_{x,1})\hat{e}(D_1,C_{x,2})\hat{e}(D_{\rho(i)},C_{x,3})=$$
$$\hat{e}(g^{\beta r},h^{\lambda_x}\mu^{t_x})\hat{e}(g^r,g^{-t_x v_x})\hat{e}(g^{a_{\rho(i)}r}\mu^{-\beta r},g^{t_x})=$$
$$\hat{e}(g,h)^{\beta r\lambda_x}\hat{e}(g,\mu)^{\beta r t_x}\hat{e}(g,g)^{a_{\rho(i)}r t_x}\hat{e}(g,g)^{-\beta r t_x}=$$
$$\hat{e}(g,h)^{\beta r\lambda_x}. \quad (17)$$

当且仅当日志分析者拥有足够权限时,即用户私钥中嵌入的属性值 $a_{\rho(i)}$ 和密文中嵌入的访问策略值 v_x 一致($a_{\rho(i)}=x$),我们可以得到结果:

$$P=\prod_{x=1}^{l}(P_x)^{\omega_x}=\prod_{x=1}^{l}(\hat{e}(g,h)^{\beta r\lambda_x})^{\omega_x}=$$
$$\hat{e}(g,h)^{\beta r\sum_{x=1}^{l}\lambda_x\omega_x}=\hat{e}(g,h)^{\beta rs}, \quad (18)$$

$$W=\hat{e}(D_{m,1},C_1)=\hat{e}(g^{\sigma_m},(T_0\prod_{j=1}^{\phi_m}T_j^{\delta_{m,j}})^s)=$$
$$\hat{e}(g,(T_0\prod_{j=1}^{\phi_m}T_j^{\delta_{m,j}}))^{\sigma_m s}, \quad (19)$$

当且仅当日志分析者所持有的密钥在有效范围内.

最后,对称会话密钥 ssk 可以通过计算获得

$$ssk=\frac{C_0\times P\times W}{F}. \quad (20)$$

同时,明文数据可以通过 $m=Dec(ssk,CT_1)$ 进行恢复.日志分析者即可对日志明文进行相关攻击检测分析.

5 实验与结果

我们使用恶意软件数据集 CCCS-CIC-AndMal-2020 对增强型 Louvain 算法的计算速度提升进行了验证,并且对目前 4 种常见的 APT 攻击进行了仿真复现实验,以此来评价本文方法在图卷积神经网络攻击溯源中的检测效果.

5.1 数据集

在时间复杂度优化效果的验证部分,我们选用了恶意软件数据集 CCCS-CIC-AndMal-2020.该数据集包括 20 万个良性和 20 万个恶意软件样本,总计 40 万个 android 应用程序,其中包含 14 个突出的恶意软件类别和 191 个恶意软件家族.

5.2 攻击溯源性能分析

在检测效果的对比部分,我们选取了表 4 所示的 4 种常见 APT 攻击漏洞来进行仿真复现实验,分别用本文方法与 HERCULE 方法进行攻击溯源,并将 2 种方法的检测准确性进行对比.

Table 4　Four Common APT Attack Vulnerabilities

表 4　4 种常见 APT 攻击漏洞

序号	漏洞名称	CVE
APT1	IE 脚本引擎内存损坏漏洞	CVE-2019-1367
APT2	Chrome 0-day 漏洞	CVE-2019-13720
APT3	WinRAR 远程代码执行漏洞复现	CVE-2018-20250
APT4	Office 远程代码执行漏洞	CVE-2019-18570

增强型 Louvain 算法通过对社区划分的移动过程进行优化,大幅提升了算法的执行速度.为了更直观地展示该算法对于整个社区检测过程时间复杂度的优化效果,我们对比了本文方法和 HERCLUE 在同一数据集下进行社区检测所消耗的时间成本.

图 4 是 2 种算法的运行时间对比图,可以明显看出,HERCULE 方法在该数据集上的平均执行时间为 80.6 s,而本文方法的执行时间平均为 8.2 s.图 4 中的实验结果证明:相比于 HERCULE 方法,本文方法时间优化率大约在 90%左右.这种优化效果在算法检测大量节点或者进行多次迭代的过程中体现的尤为明显.

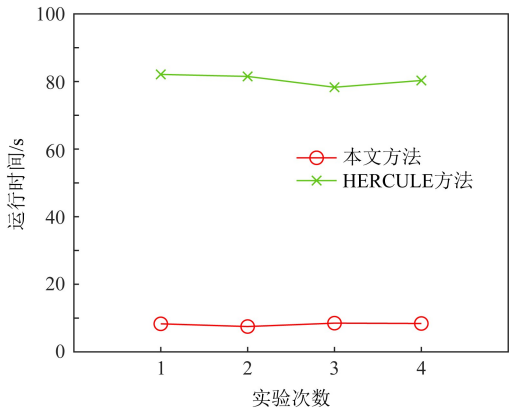


Fig. 4　Running time comparison between our method and HERCULE method

图 4　本文方法和 HERCULE 方法运算时间对比

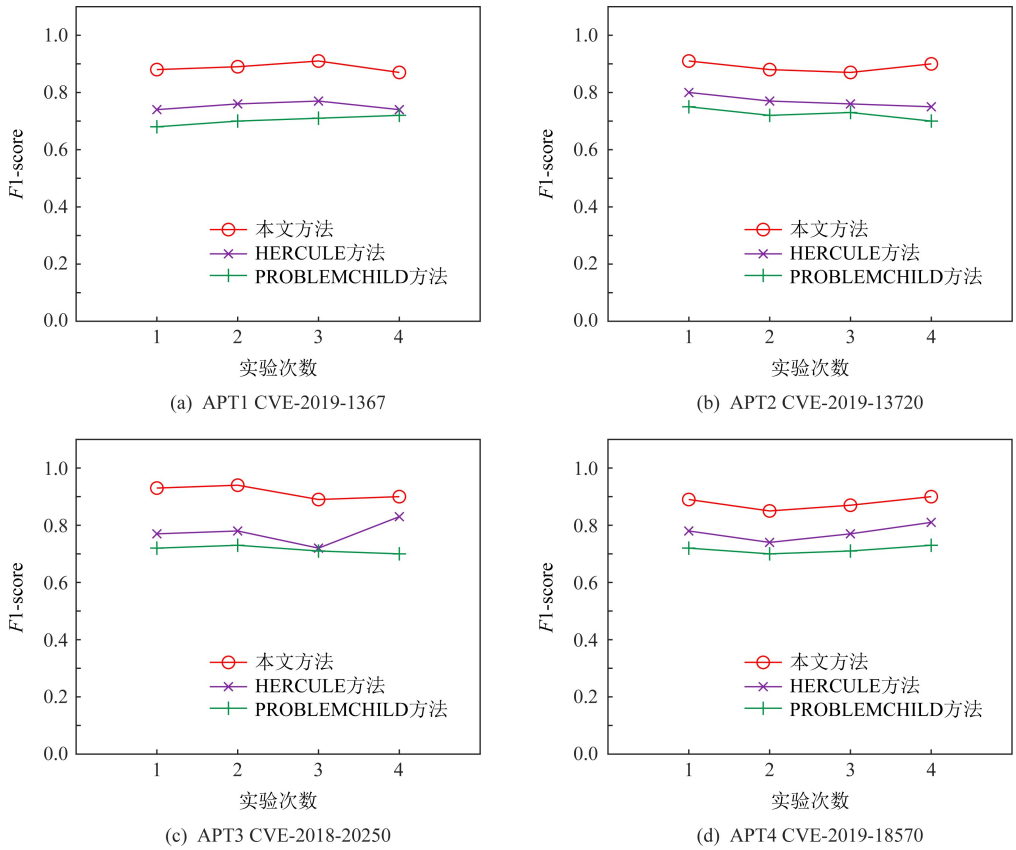
另外,本文方法对社区划分过程中产生的连接不良的社区进行了优化,因此大大减少了无关日志节点在社区检测中的错误连接,提高了社区发现算法的精准度.为了对比方法在社区检测中的执行效果,我们用 $F1$ 分数对其识别质量进行衡量.把正确分类为攻击日志的节点数量,错误分类为攻击日志的节点数量,正确分类为无关日志的节点数量和错误分类为无关日志的节点数量分别记为 tp,fp,tn,fn.

定义精确率 *precision* 来表示所有预测为攻击日志节点中实际与攻击相关的节点的比例,$precision=\frac{tp}{tp+fp}$.定义召回率 *recall* 表示实际与攻击相关的日志节点中被正确划分到攻击日志社区中的节点的比例,$recall=\frac{tp}{tp+fn}$.$F1$ 分数是 *precision* 和 *recall* 的调和平均数 $F1=\frac{2\times precision\times recall}{precision+recall}$.当 *precision* 和 *recall* 都接近 1 时,$F1$ 分数也更接近 1,即其对应的方法检测效果更优.

我们对比了该方法、HERCULE 方法和 PROBLEMCHILD 方法.如图 5 所示,对于 4 种 APT 攻击的检测效果.在最坏情况下,如 APT1 检测中,本文方法的 $F1$ 分数也可达到 0.87,而 HERCULE 方法仅为 0.74,PROBLEMCHILD 方法更是低至 0.68;而最优情况下,如 APT3 检测中,本文方法的 $F1$ 分数可达 0.94,HERCULE 方法和 PROBLEMCHILD 方法分别为 0.83 和 0.73.因为本文方法在检测中优化了连接不良的社区,所以可以大幅减少攻击相关的事件和与攻击无关的事件之间的错误连接.更好的事件划分有效地提高了检测系统的性能,对比 HERCULE 方法和本文方法的平均 $F1$ 分数,分别为 0.77 和 0.89.本文方法的检测效率平均提升了 16%.对比 PROBLEMCHILD 方法,检测效率平均提升了 20%.在图 6 中,我们展示了图卷积神经网络攻击溯源的可视化输出.方形节点所在的社区是与攻击相关的社区,圆形节点所在的社区则是与攻击无关的社区.通过对攻击社区进行分析,我们能够完成对攻击者恶意行为的重构和复现.

以图 6 中 CVE-2019-1367 所构建出的攻击社区为例,这是一个 IE 浏览器脚本引擎内存损坏漏洞,攻击者可以利用此漏洞破坏内存,获得用户的当前权限,并执行任意代码.如果当前用户拥有管理权限,攻击者可以在系统上执行各种操作,从创建具有完全权限的新账户到安装程序甚至修改数据.

在仿真实验中,我们利用本文方法对攻击行为进行了日志溯源,成功重构了 3 种攻击行为:1)攻击者发送电子邮件,并诱导目标打开 Gmail 中的链接,查看攻击者提前利用该漏洞构造的恶意网页.2)攻击者成功获取了目标客户机的管理员用户权限并与 C&C 服务器的 IP 地址 192.168.2.15 建立了反向 TCP 连接.3)攻击者浏览了不同的文件夹和文件.找到目标文件后,在命令行中启动 FTP 客户端,将文件上传到 IP 地址 192.168.2.15.

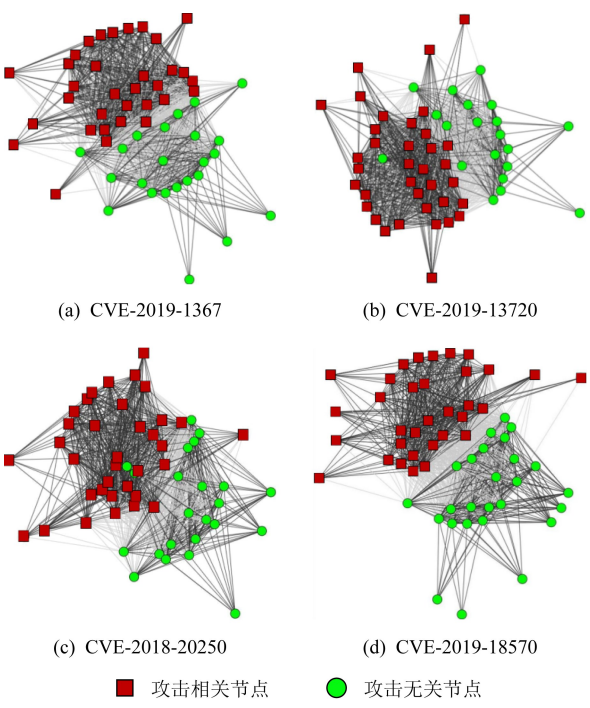


(a) APT1 CVE-2019-1367 (b) APT2 CVE-2019-13720

(c) APT3 CVE-2018-20250 (d) APT4 CVE-2019-18570

Fig. 5 Comparison of the detection effect of this method with HERCULE method and PROBLEMCHILD method

图 5 本文方法与 HERCULE 方法、PROBLEMCHILD 方法的检测效果对比



(a) CVE-2019-1367 (b) CVE-2019-13720

(c) CVE-2018-20250 (d) CVE-2019-18570

Fig. 6 Visual community graph of APT attack

图 6 APT 攻击的可视化社区图

5.3 GCN 分类准确率对比

为验证本文使用 GCN 作分类的有效性，我们将其与 4 种其他的方法进行比较验证：

1）深度游走（DeepWalk）[32] 使用随机游走（random walk）的方法在图中对节点进行采样，产生相应的节点序列.然后用 skip-gram 模型对序列进行向量学习.

2）半监督节点嵌入（semi-supervised embedding，SemiEmb）[33] 可以将图数据映射为低维稠密向量，从而获取图的拓扑结构等相关信息.

3）独立成分分析（ICA）[34] 利用源信号的独立性和非高斯性，从多维统计数据中分离出独立分量.ICA 可以在源信号和线性变换未知的情况下，从观测的混合信号中估计出源信号.

4）BiLSTM-GCN[35] 将网络流量数据分解为基于 IP 的网络流，并基于网络流对 IP 信息进行重构，最后利用 GCN 模型从所有 IP 中检测出 APT 攻击的异常 IP.

模型的测试结果如表 5 所示，我们使用本文方法对不同的威胁类型进行了分类测试，并用 DeepWalk，

SemiEmb,ICA,BiLSTM-GCN 4 种方法作为对照.结果表明:对于 4 种不同类型的攻击,本文所采用的 GCN 模型的分类准确率均显著优于其他 4 个模型.

Table 5　Comparison of Classification Accuracy

表 5　分类准确率对比　%

方法	分类准确率			
	APT1	APT2	APT3	APT4
DeepWalk	66.4	67.5	44.9	57.8
SemiEmb	59.1	58.7	60.2	26.5
ICA	74.7	72.5	67.1	24.2
BiLSTM-GCN	71.4	68.5	67.2	54.7
本文方法	80.0	79.1	68.6	59.2

相对于传统方法,本文结合 GCN 能有效对图的特征进行提取并充分利用节点的特征关系,从而提高分类精度,得出较好的分类结果.

5.4　隐私保护性能分析

图 7(a)描绘了 3 个方案的文件加密时间消耗.可以很明显看出,本文的方案在加密过程中所需要的时间损耗远小于文献[29,36]所述方案,而且随着访问策略复杂度的增长,本文中方案所需的加密时间增长较为缓慢.图 7(b)展示了 3 个方案中文件解密时间随着密文个数的变化情况.在同样的访问策略设置下对同样个数的密文进行解密的时间长度,

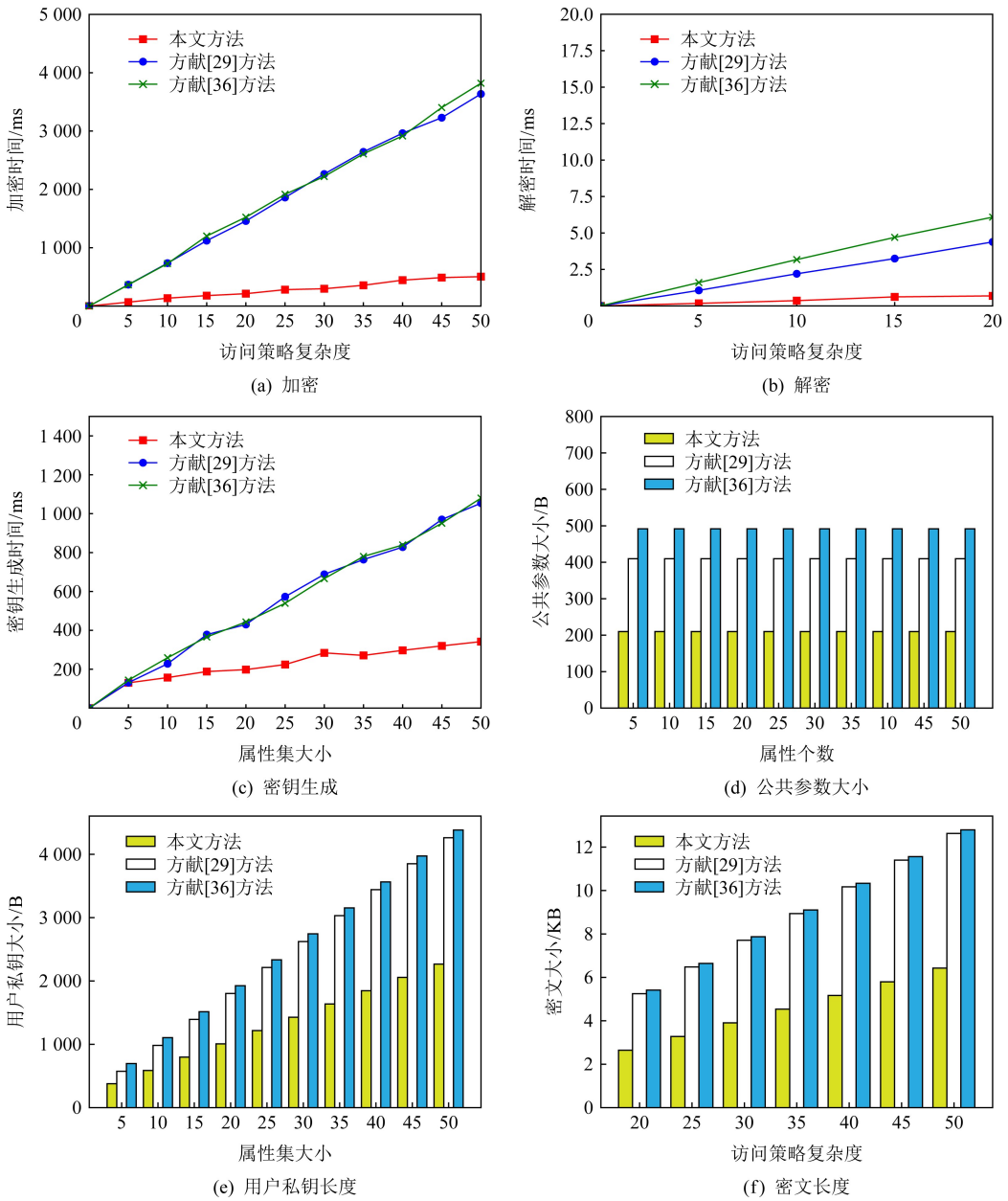


Fig. 7　Scheme performance evaluation

图 7　方案性能评估

本文中的方案所需要的时间消耗要远低于另外 2 个文献[29,36]方案.图 7(c)显示了 3 个方案的密钥生成时间.可以看出,在同样的时间元素个数条件下,密钥生成时间也在随着用户属性集合大小缓慢增大.而文献[29,36]方案的密钥生成时间远高于本文方法中的时耗.

图 7(d)描绘了系统公共参数的大小随系统属性集合的变化情况.很明显,3 个方案中,系统公共参数的大小均不受系统属性全集大小的变化,也就是说,这 3 个方案均事实上支持大规模属性集合.在本文方案中,由于要支持时间有效性而引入了时间相关的公共参数,因此,会增大公共参数的存储开销.图 7(e)显示了用户私钥大小在 3 个方案中随着用户属性集合大小的变化情况.本文方案的用户私钥大小会随着时间元素个数的增加而缓慢增长.而在同样的时间元素个数设置下,本文方案和文献[29,36]方案中的用户私钥大小都随着用户属性集合大小的增大而增长,可以很明显看到,本文方案的变化较为缓慢,同时,在同样的用户属性集合大小情况下,文献[29,36]方案中的用户私钥存储占用要远高于本文方案.图 7(f)展示了密文大小在 3 个方案中的变化情况.如图 7 所示,3 个方案中的密文大小均受访问策略的复杂度影响.而且,随着访问策略的复杂度的增大,密文大小也在增加,所占用的存储空间也会加大.然而,由于未采用双系统方案,本文方案的密文占用存储开销远低于文献[29,36]方案.因此,本文的密文所需要的存储空间更小.

本文方案由于在系统公共参数中,引入时间相关的一些固定参数,因此在公共参数的存储开销中有所增加.然而,在方案的计算性能和其他存储空间开销方面,本文方法都远超文献[29,36]方案.因此,本文方法更具有实用性和可操作性.

6 总 结

本文介绍了一种具有隐私保护的基于图卷积神经网络的攻击溯源方法.本文方法将收集的多种日志解析为计算机可以操作的格式化条目,通过权重优化解决了多日志关系连接导致的状态爆炸,并且通过对 Louvain 算法的调整,很好地优化了社区检测过程,最后基于 GCN 对多重 APT 攻击进行精确分类.相较于现有的单日志分析方法,本文方法可以更加完整地提取攻击社区.同时,本文方法结合属性基加密实现了日志数据的隐私保护.实验通过在不同操作系统上复现的多种 APT 攻击,显示了我们重构后的攻击社区信息更加完整且有效.实验同时表明:优化后的社区检测很好地提升了算法的检测速度和社区划分的准确度.而且,所提出的隐私保护方法在实验中也表现出了高效性和可用性.

参 考 文 献

[1] Zhao Guodong, Xu Ke, Xu Lei, et al. Detecting APT malware infections based on malicious DNS and traffic analysis [J]. IEEE Access, 2015, 3: 1132-1142

[2] Oprea A, Zhou Li, Yen Tingfang, et al. Detection of early-stage enterprise infection by mining large-scale log data [C] //Proc of the 45th Annual IEEE/IFIP Int Conf on Dependable Systems and Networks (DSN). Piscataway, NJ: IEEE, 2015: 45-56

[3] Hassan W U, Lemay M, Aguse N, et al. Towards scalable cluster auditing through grammatical inference over provenance graphs [C/OL] //Proc of the 25th Network and Distributed System Security Symp. Reston, VA: The Internet Society, 2018 [2021-02-23]. https://par.nsf.gov/biblio/10047685

[4] Pasquier T, Han Xueyuan, Moyer T, et al. Runtime analysis of whole-system provenance [C] //Proc of the Computer and Communications Security Conf. New York: ACM, 2018: 1601-1616

[5] Li Teng, Ma Jianfeng, Pei Qingqi, et al. AClog: Attack chain construction based on log correlation [C/OL] //Proc of the 2019 IEEE Global Communications Conf (GLOBECOM). Piscataway, NJ: IEEE, 2020 [2021-02-23]. https://ieeexplore.ieee.org/abstract/document/9013518

[6] Lee K H, Zhang Xiangyu, Xu Dongyan. High accuracy attack provenance via binary-based execution partition [C/OL] //Proc of the 20th Annual Network and Distributed System Security Symp. Reston, VA: The Internet Society, 2013 [2021-02-23]. https://www.ndss-symposium.org/wp-content/uploads/2017/09/03_1_0.pdf

[7] Kwon Y, Wang Fei, Wang Weihang, et al. MCI: Modeling-based causality inference in audit logging for attack investigation [C/OL] //Proc of the 25th Network and Distributed System Security Symp. Reston, VA: The Internet Society, 2018 [2021-02-23]. https://www.ndss-symposium.org/wp-content/uploads/2018/02/ndss2018_07B-2_Kwon_paper.pdf

[8] Fredrikson M, Christodorescu M, Giffin J, et al. A declarative framework for intrusion analysis [M] //Proc of the Cyber Situational Awareness. Berlin: Springer, 2010: 179-200

[9] King S T, Mao Z M, Lucchetti D G, et al. Enriching intrusion alerts through multi-host causality [C/OL] //Proc of the Network and Distributed System Security Symp. Reston, VA: The Internet Society, 2005 [2021-02-23]. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.332.71

[10] Goel A, Po K, Farhadi K, et al. The taser intrusion recovery system [J]. ACM SIGOPS Operating Systems Review, 2005, 39(5): 163-176

[11] Kim T, Xi W, Zeldovich N, et al. Intrusion recovery using selective re-execution [C] //Proc of the 9th Symp on Operating Systems Design and Implementation (OSDI 10). Berkeley, CA: USENIX Association, 2010: 89-104

[12] Alam S, Horspool R N, Traore I, et al. A framework for metamorphic malware analysis and real-time detection [C] //Proc of the 28th IEEE Int Conf on Advanced Information Networking and Application. Piscataway, NJ: IEEE, 2014: 480-489

[13] Milajerdi S M, Gjomemo R, Eshete B, et al. Holmes: Real-time APT detection through correlation of suspicious information flows [C] //Proc of the 40th IEEE Symp on Security and Privacy. Piscataway, NJ: IEEE, 2019: 1137-1152

[14] Du Min, Li Feifei, Zheng Guineng, et al. Deeplog: Anomaly detection and diagnosis from system logs through deep learning [C] //Proc of the 24th ACM SIGSAC Int Conf on Security, Audit and Control. New York: ACM, 2017: 1285-1298

[15] Shen Y, Maricont Ei, Vervier P A, et al. Tiresias: Predicting security events through deep learning [C] //Proc of the 25th ACM SIGSAC Int Conf on Security, Audit and Control. New York: ACM, 2018: 592-605

[16] Barre M, Gehani A, Yegneswaran V. Mining data provenance to detect advanced persistent threats [C/OL] //Proc of the 11th Int Workshop on Theory and Practice of Provenance. Berkeley, CA: USENIX Association, 2019 [2021-02-23]. https://www.usenix.org/conference/tapp2019/presentation/barre

[17] Hassan W U, Guo Shengjian, Li Ding, et al. Nodoze: Combatting threat alert fatigue with automated provenance triage [C/OL] //Proc of the 26th Network and Distributed System Security Symp. Rosten, VA: The Internet Society, 2019[2021-02-23]. https://par.nsf.gov/biblio/10085663

[18] Liu Yushan, Zhang Mu, Li Ding, et al. Towards a timely causality analysis for enterprise security [C/OL] //Proc of the 25th Network and Distributed System Security Symp. Rosten, VA: The Internet Society, 2018 [2021-02-23]. https://www.princeton.edu/~pmittal/publications/priotracker-ndss18.pdf

[19] Liu Fucheng, Wen Yu, Zhang Dongxue, et al. Log2vec: Aheterogeneous graph embedding based approach for detecting cyber threats within enterprise [C] //Proc of the 26th ACM SIGSAC Int Conf on Security, Audit and Control. New York: ACM, 2019: 1777-1794

[20] Yang Runqing, Ma Shiqing, Xu Haitao, et al. UiScope: Accurate, instrumentation-free, and visible attack investigation for gui applications [C/OL] //Proc of the 27th Network and Distributed System Security Symp. Rosten, VA: The Internet Society, 2020 [2021-02-23]. https://www.ndss-symposium.org/wp-content/uploads/2020/02/24329-paper.pdf

[21] Li Bo, Vadrevu P, Lee K H, et al. JSgraph: Enabling reconstruction of web attacks via efficient tracking of live in-browser JavaScript executions [C/OL] //Proc of the 25th Annual Network and Distributed System Security Symp. Reston, VA: The Internet Society, 2018 [2021-02-23]. https://www.researchgate.net/profile/Phani-Vadrevu/publication/323248874_JSgraph_Enabling_Reconstruction_of_Web_Attacks_via_Efficient_Tracking_of_Live_In-Browser_JavaScript_Executions/links/5c8fc4ce45851564fae68400/JSgraph-Enabling-Reconstruction-of-Web-Attacks-via-Efficient-Tracking-of-Live-In-Browser-JavaScript-Executions.pdf

[22] Ma Shiqing, Zhai Juan, Wang Fei, et al. MPI: Multiple perspective attack investigation with semantic aware execution partitioning [C] //Proc of the 26th USENIX Security Symp (USENIX Security 17). Berkeley, CA: USENIX Association, 2017: 1111-1128

[23] Hossain M N, Milajerdi S M, Wang Junao, et al. SLEUTH: Real-time attack scenario reconstruction from COTS audit data [C] //Proc of the 26th USENIX Security Symp(USENIX Security 17). Berkeley, CA: USENIX Association, 2017: 487-504

[24] Hossain M N, Sheikhi S, Sekar R. Combating dependence explosion in forensic analysis using alternative tag propagation semantics [C] //Proc of the 41st IEEE Symp on Security and Privacy. Piscataway, NJ: IEEE, 2020: 1139-1155

[25] Hassan W U, Noureddine M A, Datta P, et al. OmegaLog: High-fidelity attack investigation via transparent multi-layer log analysis [C/OL] //Proc of the 27th Annual Network and Distributed System Security Symp. Reston, VA: The Internet Society, 2020 [2021-02-23]. https://par.nsf.gov/biblio/10146531

[26] Goyal V, Pandey O, Sahai A, et al. Attribute-based encryption for fine-grained access control of encrypted data [C] //Proc of the 13th ACM Conf on Computer and Communications Security. New York: ACM, 2006: 89-98

[27] Nishide T, Yoneyama K, Ohta K. Attribute-based encryption with partially hidden encryptor-specified access structures [C] //Proc of the 6th Int Conf on Applied Cryptography and Network Security. Berlin: Springer 2008: 111-129

[28] Zhong Hong, Zhu Wenlong, Xu Yan, et al. Multi-authority attribute-based encryption access control scheme with policy hidden for cloud storage [J]. Soft Computing, 2018, 22(1): 243-251

[29] Zhang Yinghui, Zheng Dong, Deng R H. Security and privacy in smart health: Efficient policy-hiding attribute-based access control [J]. IEEE Internet of Things Journal, 2018, 5(3): 2130-2145

[30] Cui Hui, Deng R H, Lai Junzuo, et al. An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures, revisited [J]. Computer Networks, 2018, 133: 157-165

[31] Han Dezhi, Pan Nannan, Li K C. A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection [J/OL]. IEEE Transactions on Dependable and Secure Computing, 2020 [2021-02-23]. https://ieeexplore.ieee.org/abstract/document/9020182

[32] Perozzi B, Al-Rfou R, Skiena S. DeepWalk: Online learning of social representations [C/OL] //Proc of the 20th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2014 [2021-02-23]. https://dl.acm.org/doi/abs/10.1145/2623330.2623732

[33] Weston J, Ratle F, Mobahi H, et al. Deep learning via semi-supervised embedding [M] //Neural Networks: Tricks of the Trade. Berlin: Springer, 2012: 639-655

[34] Getoor L. Link-based classification [M] //Advanced Methods for Knowledge Discovery from Complex Data. Berlin: Springer, 2005: 189-207

[35] Li Qi, Zhang Yinghui, Zhang Tao, et al. HTAC: Fine-grained policy-hiding and traceable access control in mHealth [J]. IEEE Access, 2020, 8: 123430-123439

[36] Do Xuan C, Nguyen H D, Dao M H. APT attack detection based on flow network analysis techniques using deep learning [J]. Journal of Intelligent & Fuzzy Systems, 2020, 39(3): 4785-4801

Li Teng, born in 1991. PhD, lecture. His main research interests include network security, system log analysis, attack detection, data security and privacy preserving.

李 腾, 1991 年生.博士, 讲师.主要研究方向为网络安全、系统日志分析、攻击检测, 数据安全和隐私保护.

Qiao Wei, born in 1998. Undergraduate. His main research interest is network security.

乔 伟, 1998 年生.本科生.主要研究方向为网络安全.

Zhang Jiawei, born in 1985. PhD candidate. His main research interests include applied cryptography, authentication, access control, blockchain, cloud computing security, privacy computing and network security.

张嘉伟, 1985 年生.博士研究生.主要研究方向为应用密码学、认证、访问控制、区块链、云计算安全、隐私计算、网络安全.

Gao Yiyang, born in 2000. Undergraduate. His main research interests include machine learning and deep learning.

高怿旸, 2000 年生.本科生.主要研究方向为机器学习和深度学习.

Wang Shenao, born in 2001. Undergraduate. His main research interest is advanced persistent threats.

王申奥, 2001 年生.本科生.主要研究方向为高级持续性威胁.

Shen Yulong, born in 1978. PhD, professor. Member of CCF. His main research interests include network security, system security, physical layer security and IoT security.

沈玉龙, 1978 年生.博士, 教授.CCF 会员.主要研究方向为网络安全、系统安全、物理层安全和物联网安全.

Ma Jianfeng, born in 1963. PhD, professor. Member of CCF. His main interests include network security, system security, data security and unmanned aerial vehicle (UAV) security.

马建峰, 1963 年生.博士, 教授.CCF 会员.主要研究方向为网络安全、系统安全、数据安全和无人机安全.