基于系统溯源图的威胁发现与取证分析综述

冷涛[1,2,3], 蔡利君[1], 于爱民[1,2], 朱子元[1,2], 马建刚[1],李超飞[1,2], 牛瑞丞[1,2], 孟丹[1,2]

（1. 中国科学院信息工程研究所 北京 中国 100093；2. 中国科学院大学网络空间安全学院 北京 中国 100049；
3. 四川警察学院智能警务四川省重点实验室，四川 泸州 646000）

摘 要：通过调研溯源图研究相关的文献，提出了基于系统溯源图的网络威胁发现和取证分析研究框架。详细综述了基于溯源图的数据采集、数据管理、数据查询和可视化方法；提出基于规则、基于异常和基于学习的威胁检测分类方法；概括了基于威胁情报或基于战略、技术、过程驱动的威胁狩猎方法；总结出基于因果关系、序列学习、特殊领域语言查询和语义重建的取证分析方法；最后指出未来的研究趋势。

关键词：溯源图；高级持续威胁；威胁发现；取证分析；图神经网络

中图分类号：TP391

文献标识码：A

doi:

A Review of Threat Discovery and Forensic Analysis Based on System-level Provenance Graphs

LENG Tao[1,2,3],CAI lijun[1],YU Aimin[1,2],ZHU ziyuan[1,2], MA Jiangang[1],LI Caofei[1,2],NIU Ruicheng[1,2],MENG Dan[1,2]

1. Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
2. School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049，China
3. Intelligent Policing Key Laboratory of Sichuan Province，SiChuan Police College, Sichuan 646000, China

Abstract: By investigating works of literature related to provenance graph research, a research framework for network threat discovery and forensic analysis based on system-level provenance graph is proposed. A detailed overview of data collection, data management, data query, and visualization methods based on provenance graphs is provided; proposed rule-based, anomaly-based, and learning-based threat detection classification methods; summarized threats based on threat intelligence or based on strategy, technology, and process-driven threats Hunting method; summarized forensic analysis methods based on causality, sequence learning, language query and semantic reconstruction in special fields; finally pointed out future research trends.

Key words: provenance graph; advanced persistent threats; threat discovery; forensics analysis; graph neural network

1 引言

当前，政府和企业面临着高级持续威胁（APT, advanced persistent threat）[1]。震网攻击、极光漏洞先后发生，世界各国开始重视 APT 攻击。传统的 APT 攻击检测方法主要聚焦单步攻击检测，无法捕获系统长期运行行为,而 APT 攻击大量应用零日漏洞，导致检测困难。2015 年，美国国防部高级研究计划局提出 4 年透明计算计划[2]，希望找到一种高保真和可视化的方法来抽象出系统中的攻击活动。研究人员发现依靠系统监控日志数据,构造具有较强抽象表达能力的溯源图，进行因果关系分析，能有效表达威胁事件的起因、攻击路径和攻击影响，为威胁发现和取证分析提供较高的检测效率和稳健性[3]。在综述方面，Han 等[3]介绍基于溯源图的入侵检测的机遇和挑战。Zafar 等[4]描述了安全溯源的生命周期，提出了现有安全溯源方案的分类方法，指出它们的优缺点。Tan 等[5]讨论了网络攻击溯源中数据源优化和数据关系分析两类文献，并围绕安全性、有效性（效能）、效率进行对比分析。Li 等[6]侧重讨论利用系统级溯源图,构建攻击模型，进行威胁检测和调查。潘亚峰等[7]重点综述了 APT

收稿日期：2022−01−05；**修回日期**：2022-04-20

通信作者：于爱民，yuaimin@iie.ac.cn

基金项目：中科院战略先导研究项目（XDC02040200），智能警务四川省重点实验室资助项目（ZNJW2022ZZQN002）

Foundation Items: The Strategic Priority Research Program of Chinese Academy of Sciences (No. XDC02040200.) ,Intelligent Policing Key Laboratory of Sichuan Province, No. ZNJW2022ZZQN002

攻击场景重构方法。本文重点综述基于溯源图的数据采集、数据管理（图构建，缩减图，图存储和图查询）、数据分析（威胁检测、威胁狩猎、取证分析）等工作。

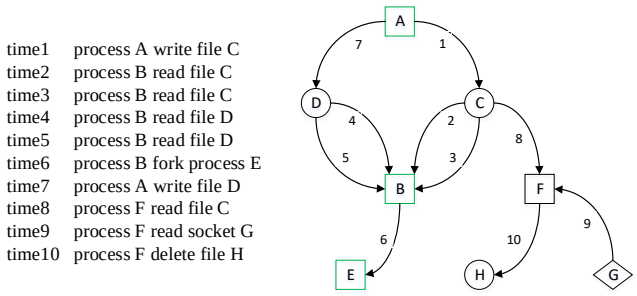
通过搜索溯源图研究相关文献，获得 CCF A 类 44 篇，CCF B 类 25 篇，CCF C 类 8 篇，以及其它 SCI、博士论文等文献。经分析总结，本文贡献可概括为：1）提出了基于溯源图的威胁发现和取证分析框架；2）总结了多种场景下日志采集，数据缩减和存储方案；3）分类总结威胁检测、威胁狩猎、取证分析的研究方法和模型；4）对下一步研究方向进行了展望。

2 背景知识

2.1 溯源图

管理员通过系统审计或配置服务器可以获得多个层级的日志事件，如应用程序级、网络级、系统级和指令级[8]。应用程序级日志是应用程序产生的日志，如网站服务器等应用程序产生的日志。网络级日志可通过监控系统的网络访问获得，如 zeek 捕获网络流量日志。指令级日志是指机器指令产生的日志，可提供完整信息，但很难理解。系统级审计日志是一串按时间顺序排列的事件元组，表示了不同时间，某进程（或线程）访问某个文件或网络连接的方式。ETW、Auditd 等内核级框架审计工具可获取系统调用事件日志。研究者们将系统级审计日志事件抽象为溯源图（provenance graph）表示[8]。

定义 1.溯源图.溯源图 $G =< S, O, E, T >$，S 表示主体（指进程或线程）的集合，主体属性包括进程 id、pid、命令行、所有者、代码和数据的标签等；O 表示客体（文件、管道、网络连接等）集合，客体属性包括名字，类型，所有者和标签；E 表示系统调用事件方式集合，如 read、write、fork、open、create 等，它表示了实体（主、客体）间的信息流；T 表示时间戳，指主客体的访问时间。在溯源图中，主体和客体以顶点表示，事件类型表示为边，在不同的时间，两个顶点之间可以有多条边。这种表示了实体（主体和客体）间关系方向的图被称为溯源图。由于溯源图表达了系统日志的起源，有些也将溯源图翻译为起源图，依赖图，本文统一称为溯源图。如图 1 所示，其中 A，B，E 进程表示存活状态，F 进程状态为死亡。



（a）事件日志　　（b）溯源图

图 1 审计日志与溯源图

图 1（a）表示事件日志，图 1（b）是通过事件日志形成的溯源图。图中的顶点代表系统中的实体，连接两个顶点的边代表信息流。箭头代表两个实体之间的数据内容或控制信息的流动。边上的数字代表操作发生的时间（数字越小事件发生越早）。

2.2 威胁检测

威胁检测是分析整个安全生态系统，识别可能危及网络的任何恶意活动。威胁检测方法主要包括基于误用的检测和基于异常的检测[9]，误用检测一般通过构建恶意样本特征进行检测，只能检测已知攻击，基于异常的检测通过构建合理行为的边界，设置异常阈值，超过阈值判断为异常。虽然基于异常检测可判断未知攻击，但也导致了较高的误报率。APT 攻击是一种复杂攻击，跨度时间长，一般可达半年潜伏期，具有多步骤，隐蔽性等特点，单步检测效果不佳，研究者探索基于系统日志构造溯源图，利用规则、异常检测、图学习等方法实现 APT 攻击威胁检测。

2.3 威胁狩猎

徐嘉涔等[10]定义威胁狩猎为主动持续地在网络中搜索可以绕开安全检测或产生危害的威胁的过程。Valentina 等[11]定义威胁狩猎是人为主动的活动，通过反复搜索组织环境（网络、端点和应用程序）的妥协指标(IoC, indicator of compromise)，以缩短停留时间并最大限度地减少入侵对组织的影响。常见的妥协指标如恶意文件/进程名、病毒特征、僵尸网络的 IP 地址和域名等。停留时间是指攻击侵入系统到被检测发现的时间。威胁狩猎的方法包括数据驱动，情报驱动，实体驱动，战略、技术、过程（TTP, tactic technique procedure）驱动，混合驱动五种类型[13]。数据驱动是指查看已有数据来寻找内容，如利用代理日志查看不常见用户代理发现异常。情报驱动是指分析师利用威胁情报数据集，通

过搜索和匹配威胁指标。实体驱动表示搜索关键知识产权和网络资源等高风险、高价值实体。TTP 驱动是指通过了解攻击者使用的战术、技术和过程，通过搜索已知的 TTP，实现威胁狩猎。混合狩猎是上述方法的融合。图 2 展示了威胁狩猎的过程。

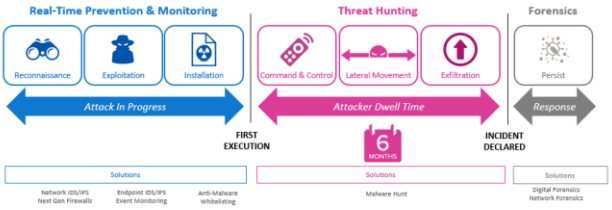


图 2 威胁狩猎[12]

2.4 取证分析

取证分析的概念包含较广，本文所指取证分析是指用户在发现遭受网络攻击后，调查人员根据告警或攻击症状开展攻击溯源，进行攻击场景重建分析等。基于溯源图的取证分析的一般过程是在溯源图上找到攻击症状节点，在图上执行后向查询，找到攻击的入口点，然后根据攻击入口点执行前向查询，查看攻击的影响，关联出攻击事件路径。此外，还考虑攻击场景重构，即从大量日志数据中，根据特定的攻击行为模式和语义知识，通过分析数据之间的的关联关系，还原包含数据层攻击行为的语义信息和攻击战略战术、过程语义知识的完整攻击行为视图的过程[7]。

定义 2.后向查询. 边 e 的后向追踪是溯源图 G 的子图，表示从 e 可到达的目的顶点的边的集合。

定义 3.前向查询. 边 e 的前向追踪是溯源图 G 的子图，表示从 e 作为源顶点可到达边的集合。

BackTracker[8]第一次引入构建溯源图用于入侵检测，开辟了终端主机攻击溯源的工作，他们通过定义终端主机进程之间、进程与文件之间以及进程与文件名之间的依赖关系来构造溯源图。攻击的入口点是通过给定告警事件后向追踪分析确定的，当系统中的一个实体被标记为可疑时，在图中反复搜索其他实体对目标实体的历史作用，直到该实体没有流入的边。

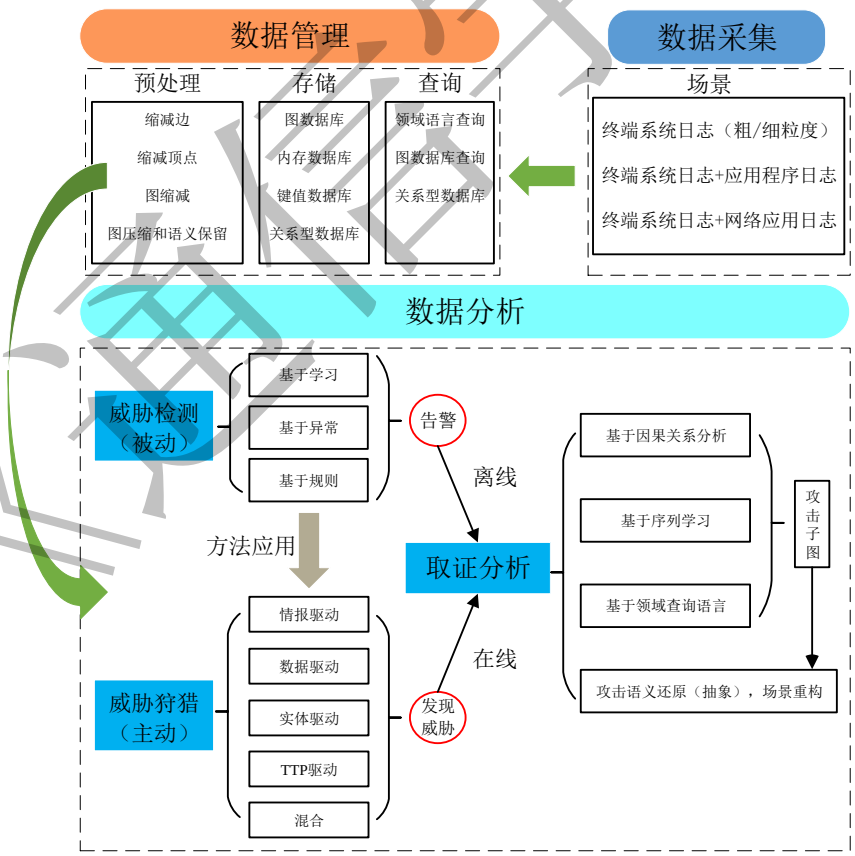


图 3 框架图

3 研究框架

基于系统溯源图的威胁发现与取证分析包括数据采集，数据管理，数据分析三个模块。数据采集包含不同场景下数据采集；数据管理包括数据预处理，溯源图的存储和查询可视化；数据分析模块包括威胁检测、威胁狩猎和取证分析等内容。威胁检测的方法可应用于威胁狩猎的不同框架中，取证分析是基于已发现的威胁开展取证调查和重建分析，整体研究框架如图 3 所示。下面详细介绍各模块的内容和方法。

4. 数据采集

4.1 数据采集方式

日志采集主要包括终端侧系统审计日志，应用程序日志和网络侧日志等。

4.1.1 基于终端侧系统级日志采集

常见的终端侧系统内核级日志采集工具如 Auditd，ETW，Dtrace 等。Lineage[14]是系统级溯源的首次尝试，该系统通过修改 linux 内核调用，用户进程从 printk 缓冲区中读取捕获的内容并存储到 Sql 数据库中。PASS[15]在虚拟文件系统层捕获溯源数据，PASSV1[15]提供了进程 I/O 交互的函数，而 PASSV2[16]提供了一个跨语义层溯源集成的 API，这些方案随着系统版本的升级，很难扩展。SPADE[17]是一个分布式系统日志审计工具，可支持跨平台。Hi-Fi[18]是第一个完整的全系统溯源，可收集完整溯源记录，除了内核和应用程序行为，还包括网络连接等，Hi-Fi 采用 LSM hook 实现数据监控，不支持安全模式堆栈，因此容易受到攻击。Linux 溯源模块[19] 创建了一个可信赖的起源感知执行环境，解决了此限制，可收集整个系统的起源数据。Bates 等[20]提出 DAP 捕获 web 服务组件的详细数据源，它是 Linux 溯源模块[19]的附加服务。CamFlow[21]是一个严格意义上的独立实现，它使用标准的内核功能，并且容易扩展。

4.1.2 基于终端侧系统级日志+应用程序日志采集

虽然系统级日志展现了进程、文件、网络连接之间的依赖关系，但与应用程序日志相比，从系统层面挖掘系统行为的因果依赖关系没有考虑应用层语义，存在语义鸿沟问题，而对于攻击调查取证和场景重建，应用程序日志能提供大量的攻击相关信息，因此 OmegaLog[22]和 ALchemist[23]尝试融合系统级日志记录和应用程序日志记录，实现语义还原。

4.1.3 基于终端侧+网络侧数据采集

由于 APT 攻击通常跨越多个主机，基于终端侧系统级日志和应用程序日志不能完全捕获数据，因此研究者探索将系统监控审计数据与网络侧数据相结合[24]-[27]。虽然 PASS[15]可以支持使用网络文件系统来收集溯源日志，但不能支持收集访问本地机器的套接字信息。例如 PASS 不能记录通过远程攻击破坏或窃取本地 IP 地址和端口号的行为。PDMS[24]对 PASS 进行了扩展，通过监控和记录每一个网络会话，捕获连接到本地主机的每一个网络套接字，它将网络套接字视为文件对象，收集文件、管道、进程和网络套接字之间的依赖关系，可以准确地跟踪系统的数据流入和流出。Haas 等[25]提出了开源平台 zeek-osquery，将操作系统级的日志与网络侧日志实时关联，实现实时入侵检测，然而这种级别的跨主机攻击溯源依然会因为套接字的不确定性而存在大量的错误关联。Ji 等[26]综合了多种技术，提出了一种有效的跨主机追踪溯源方法 RTAG，可以从一定程度上解决当前网络侧与终端侧数据无法关联溯源的问题。

4.2 数据采集粒度

根据系统日志收集数据粒度不同，分为粗粒度和细粒度[5]。粗粒度采集是指仅追踪系统级对象（进程，文件），它是一种进程级调用监控或对内核模块安装钩子进行数据拦截的方法。系统级溯源可通过系统内置审计组件监控获得。细粒度收集的目标是实现精确依赖关系，比系统进程级追踪粒度更细，常采用划分进程执行分区[28]，污点分析追踪变量变化等。一个进程可以被“分割”成多个单元，每个单元分区是一个进程的执行段，它处理一个特定的对象，例如浏览器进程可根据打开的网页窗口进行划分。Lee 等[28]首先提出基于进程执行单元分区的方法，由于追踪变量的污点分析粒度太细，不适合构造因果溯源图，因此他在进程级粗粒度和变量级细粒度之间，提出“单元”的概念。Protracer[29]利用基于单元的执行分区来提高压缩率，他们将程序划分为多个单元，以实现细粒度的污点跟踪，其中一个单元对应一个循环模式。MPI[30]要求软件开发者对应用程序中的重要数据结构进行注释，通过注释的语义感知实现单元划分。这些技术都依赖于源代码或二进制工具[31]。LogGC[32]引入程序工具，

输出细粒度的依赖信息，不仅可以将一个进程分成多个可执行单元，还可把一个数据文件分成多个逻辑数据单元，但它对每个应用程序的定制成本太高。UIScope[33]也借鉴了单元分区的方法。

4.3 数据集

4.3.1 开源数据集

研究 APT 攻击检测和取证分析的常用开源数据集有 Streamspot[34]，CERT[35]，LANL[36]，DARPA TC 系列[37]-[38]，Operationally Transparent Cyber（OpTC）[39]等。Manzoor 等[34]开源了 Streamspot 数据集，该数据集包含 1 个攻击和 5 个良性场景组成，数据集较小，常用于对比实验[40]。CERT[35]数据集是内部威胁检测数据集，该数据集模拟恶意内部人员实施系统破坏、信息窃取、内部欺诈等攻击行为数据。LANL[36]数据集描述了一个红队所进行的恶意活动，该数据集在威胁检测场景中主要用于模拟 APT 攻击检测[40]-[43]。OpTC 数据集是 DARPA TC 的最新迭代，SK-Tree[44]是该数据集的首次应用。DARPA 透明计算系列提供对高级持续威胁的实时检测和取证分析[23][44]-[53]。目前开源了 DARPA3 和 5 两个数据集。Berrada 等[51]从 DARPA TC2 和 3 中选择部分数据构造了 adaptdata 数据集，Benabderrahmane 等[52]基于此数据集提出基于规则的高级威胁检测方法。DAPT 2020[54]提供了 APT 攻击的详细阶段，并打了标签，但检测模型的准确率很低，需要研究新的检测模型。

4.3.2 复现实验捕获数据集

由于 APT 攻击复杂，有关 APT 检测的开源数据集较少。以往的研究除了在开源数据集上检测模型，还通过自主设计实验或利用安全企业采集的数据进行分析，自主实验一般复现 APT 攻击报告，自主采集日志生成数据集。常见的复现实验有数据窃取，钓鱼邮件[23][53][55]，破壳漏洞[56]，后门[29]，文件传送[57]，哈希传递攻击[53]，错误配置[29]等。

5. 溯源图数据管理

APT 攻击潜伏期较长，企业需要保留半年以上的日志数据，据统计每天每台电脑监测产生的日志超过 1G [58]，存储负担重，不利于后续查询和分析工作，因此需要对数据做预处理，减少数据存储，同时也考虑保证攻击语义的完整性。

5.1 溯源图缩减与压缩

溯源图缩减主要围绕边、节点、图缩减和语义保留等进行研究，主要文献见表 1。

表 1 溯源图缩减方法

类型	论文	方法	缩减大小倍数
缩减边	Xu 等[57]	CPR	1.27~3.56x
		CPR+PCAR	1.43~5.59x
		CPR+PCAR+DOM	1.46~10.2x
	FD-SD[49]	完全依赖保留	4.46x~91.5x
		源依赖保留	4.54x~122.5x
缩减点	Nodemerge[59]	基于模板	4.2x~33.7x
	Loggc[32]	垃圾收集理念	0~77x
图缩减	NoDoze[53]	减掉普通行为	2x
	Rapsheet[60]	两个规则	1.5x
图形压缩	SEAL[61]	友好查询压缩	2.63x~12.94x
语义保留	GS-SS[62]	基于全局语义保留	4.36x~13.18x
		基于可疑语义压缩	7.86x~26.99x

5.1.1 缩减边

徐等[57]根据系统事件之间因果关系的等同性来减少日志条目的数量，提出了因果关系保全缩减(CPR, causality-preserving reduction)、以进程为中心的因果关系逼近缩减（PCAR, process-centric causality approximation reduction）和基于领域知识的缩减方法。CPR 聚合相同依赖性的事件，虽然能保留图的网络拓扑结构，但会丢失统计信息，如访问频率等。某些系统行为会导致对象和其相关邻居形成密集连接的依赖图，因此他们提出一种保留因果关系的单跳图缩减技术 PCAR，这种方法会删除与目标文件无关的重复读/写操作。基于领域知识的缩减主要是删除临时文件等。临时文件是指在其生命周期中只与一个进程有信息交换，在攻击取证中也不引入任何明确的信息流，因此可以从数据中删除所有临时文件的事件。CPR 保留了语义信息，但缩减效率有限，为进一步压缩，Hossain 等[49]提出了完全依赖保留（FDPR, full dependence preserving reduction）和源依赖保留（SDPR, source dependence preserving reduction）。FDPR 在缩减数据的同时保留完全依赖性，而 SDPR 算法在 FDPR 算法基础上只考虑保留前向依赖关系，进一步提高缩减效率。但 FDPR，SDPR 放宽了因果关联的条件，使得更多的重复事件可以被修剪，同样当查询有时间限制时，也可能引入假阴性。

5.1.2 缩减顶点

LogGC[32]关注对象的生命周期，引入垃圾收集

理念。由于许多应用程序在执行期间创建并操作临时文件，这些文件在应用程序终止后会被销毁，未来的系统行为不会受到这些文件的影响，因此可将这些对象和事件当作垃圾收集以节省空间。但如果删除的临时文件与网络套接字有关，攻击者所做的数据渗透可能会被遗漏。Nodemerge[59]提出了在线数据缩减方法，通过自动学习固定的库和运行程序的只读资源集作为模板，并进一步使用这些模板来缩减系统事件数据。Nodemerge 在大数据分析处理等只读事件多的缩减任务很有效，对那些没有加载很多文件，或在初始阶段有文件访问模式的应用程序时,缩减效果不明显。

5.1.3 图缩减

Priotracker[55]优先考虑异常依赖关系的边，Nodoze[53]将该方法推广到针对异常路径而不是单个边，可将原始图的大小减小两个数量级，加快了调查速度，同时不会丢失攻击的重要信息。然而基于异常的方法需要有代表性的训练数据，训练数据的问题可能导致假阳性、假阴性。Rapsheet[60]提出图缩减的两条规则，为保证完成警报规则匹配，时间间隔必须足够长，该研究提出了可以获得更好压缩效果的一般方法，但该压缩算法需要将整个依赖图作为输入，不能处理实时流数据。

5.1.4 图形压缩

LogGC[32]，FD-SD[49]，CPR[57]，Nodemerge[59]等方法都是采用匹配预定义的模式去掉日志，实现有损缩减，虽然通过案例实验表明其因果关联的有效性，但不能保证所有任务都能得到正确结果。无损压缩是一种保存所有信息支撑因果关系分析。SEAL[61]通过系统日志生成依赖图，并对图的结构（顶点和边）进行无损压缩，然后对边的属性（如时间戳）进行无损压缩，确保每次查询都能得到正确的回答，同时保证查询效率。

5.1.5 语义保留压缩

Zhu 等[62]提出基于通用、高效、实时的数据压缩方案，包含维护全局语义和可疑语义压缩两种压缩策略。维护全局语义的数据压缩（GS）策略是确定并删除不影响全局依赖的冗余事件。GS 的思想是假设在源顶点的语义没有改变的情况下，信息流对同一目标的影响是等价的，等价的事件可以作为冗余被删除，只需保留对目标顶点有影响的第一个事件。在溯源图中，一个顶点没有传入边时，可以认为该顶点的语义没有发生变化，其传出边的语义也没发生变化。基于可疑语义的数据压缩（SS）策略是根据取证分析的目的恢复攻击链, SS 的思想是通过使用实体上下文，自动判断该事件是否与攻击有关，与攻击无关的事件可以被删除。SS 策略默认维护两个表，一个是高价值文件目录表，另一个是敏感进程命令行表，并定义一套可扩展的可疑语义转移规则。Michael 等[58]首次提出了取证有效性度量，形式化定义了无损取证，因果保全取证，攻击保全取证 3 个衡量标准。并提出了针对攻击的优化近似方法 logApprox。

5.2 数据存储模型

常见图存储的方法主要有图数据库，内存数据库，键值数据库和关系数据库等。

5.2.1 图数据库

图数据库是一种非关系型数据库，常用来存储和表示图的数据结构，快速执行图的相关算法等。Graalf[47]和彭等[63]使用图数据库进行存储,彭等[63]将系统进程、文件、网络连接存储为顶点，事件存储为边，并根据关键属性建立索引，提高查询速度。一般存储在图数据库之前需进行数据缩减，但政府和企业往往拥有成千上万台机器，其原始数据量很容易达到 PB 级别[62]，即使经过预处理，溯源图仍然较大，每次使用时，需要将保存在图数据库中的溯源图加载进入内存，这造成巨大的开销和内存负载，而且图数据库支持的算法有限。

5.2.2 内存数据库

内存数据库是一种内存数据结构，SLEUTH[44]，Holmes[48]，FD-SD[49]和 Poirot[64]设计利用内存数据库，将整个因果关系数据存储在主内存中进行取证分析。SLEUTH[44]采用数据压缩和编码技术，使用可变长度编码事件特征，但增加了复杂性，降低运行效能，FD-SD [49]提出依靠版本图和优化算法实现紧凑性，在图构建任务时，比 SLEUTH[44]快 3 倍，Holmes 利用上述方案，平均需要不到 5 字节表示[48]。SWIFT[65]采用分层存储系统，设计了一个异步缓存驱逐策略，计算出因果图中最可疑的部分，并只将该部分缓存在主内存中，而将其余部分放到磁盘上。KCAL[66]提出了一种内核级缓存，以消除冗余的因果事件，并减少日志从内核到用户空间的传输开销。Graalf[47]使用内存存储作为事件的缓冲区，然后再送入关系数据库或图数据库存储，并在内存存储之前提供了“不压缩，无损压缩，保持取证的准确性，有损压缩”四种处

理模式。

5.2.3 键值数据库

PIDAS[67]使用 BerkeleyDB 数据库来存储缩减后的溯源图，pnode 号码唯一标识每个对象，IdentityDB 存储每个对象的身份信息（例如文件节点号，进程 ID），ParentDB 和 ChildDB 分别存储一个对象与其父节点和子节点之间的依赖关系，NameDB 存储一个对象的名称和它的 pnode 编号之间的映射关系，RuleDB 存储发生的事件。PDMS[24]采用了同样的设计方法。Pagoda[68]使用 Redis 键值数据库存储。

5.2.4 关系型数据库

PostgreSQL 是开源关系型数据库，同时支持 json 等非关系数据类型。Graalf[47]，彭等[63][69]使用 PostgreSQL 关系数据库进行后端存储，文献[63]将日志提取出的系统实体和系统事件存储在不同的表中，AIQL UI[69]还支持 Greenplum 开源数据库。

5.3 数据查询与可视化

溯源图的构建、存储为查询系统的开发奠定基础，研究者们先后发表了基于溯源图的查询系统如 CamQuery[70]、AIQL[71]、SAQL[72]、ThreaRaptor[73]，可视化应用如 ThreatRaptor webui[63] 、AIQL UI[69]、SAQL UI[74]、GrAALF[47]。CamQuery[70]提供了一个可编程的图形处理框架，实现以顶点为中心的查询 API。AIQL、SAQL 都是特定领域的查询语言，AIQL 提出了一个建立在现有监测工具和数据库之上，实现持久性存储，可以支持即时的攻击调查，SAQL 是基于流的查询系统，将企业中多个主机的实时事件反馈作为输入，并提供一个异常查询引擎，可实时检测基于指定异常模型的异常行为，还可以查询实时攻击足迹。ThreaRaptor [63][73]利用开源威胁情报自动构建威胁行为图，实现威胁狩猎穷举搜索和可视化。GrAALF[47]实现一个图形化的取证分析系统，可有效加载、存储、处理、查询和显示从系统事件中提取的因果关系，以支撑取证，与类似系统相比，GrAALF[47]提供了关系数据库、图形数据库和内存存储三种后端存储方式，实现存储、直观查询和实时跟踪更长事件序列的能力。

经过对系统审计日志的采集，构造系统溯源图，利用各种算法实现对溯源图的缩减，设计数据存储模型完成溯源图的高效存储和查询，下一步将介绍利用系统溯源图数据进行数据分析，主要包括威胁发现和取证分析两大模块，第 6,7 节将分别介绍相关研究。

6. 基于系统溯源图的威胁发现

基于溯源图的威胁发现主要包括威胁检测和威胁狩猎。威胁检测覆盖整个攻击阶段,是被动的检测，而威胁狩猎是假设攻击者已经进入系统,还没有被发现,利用威胁情报驱动等方法主动发现威胁。

6.1 威胁检测

威胁检测主要任务是检测给定网络场景的威胁，触发网络告警。MITRE ATT & CK 框架提出 14 个阶段的 APT 知识库来描述 APT 攻击战略和战术。Holmes[48]根据 APT 攻击杀伤链七个阶段设计了检测指标。Li 等[40]和 CONAN[75]提出相似的三阶段划分：(1) 渗透和恶意代码执行，(2) 内部侦察和横向移动，(3) C&C 通信和数据渗出。APT 攻击威胁检测研究较多，如鱼叉式钓鱼邮件检测[76]、横向移动检测[42]-[43]，利用 DNS 检测妥协主机[40]等。Log2vec[42]，MLTrace[43]通过构建异构图，分别利用图嵌入，图神经网络检测异常。近年来基于系统溯源图检测 APT 攻击成为研究热点（表 2 所示）。

6.1.1 基于规则的检测

基于规则的检测是指根据已知攻击制定规则策略。SLEUTH[44]结合攻击者的动机和手段，定义了 5 条触发警告的规则，提出基于标签的方法，如果一段数据或代码有未知标签，就是不受信任的源。Morse[46]将本地检测结果存储在标签中，并通过标签在溯源图中的传播对攻击链进行关联，并定义了二进制代码内存执行、恶意文件执行、进程注入、修改文件权限、文件崩溃、提权、可信数据泄露等 7 条规则对应操作和条件来实现攻击检测，但是如果对标签没有控制，标签会过度传播并导致依赖性爆炸问题。Holmes[48]通过安全专家构建的威胁子图作为知识图，采用层次化的策略模板，将底层实体行为映射为 ATT&CK 矩阵中的 TTP，并定义了 APT 攻击 7 个阶段 16 条 TTP 规则类型，然后利用图匹配算法计算与系统溯源图中相匹配的攻击，实现有语义的威胁检测，并区分攻击所处的阶段。POIROT[64] 利用 APT 攻击报告手动构建威胁查询图，基于图对齐匹配溯源图检测威胁。SAQL[72]提供基于规则，利用特定领域语言查询威胁。Patrol[77]通过捕获操作系统对象之间的依赖关系，通过入侵症状执行向前和向后搜索，构建零日攻击路径规则，识别出可疑的候选入侵传播路径，然后进一步

识别路径中未知漏洞利用的指标（如一些内核函数），从而识别出这些路径中高度可疑的候选者。对于 APT 攻击，这个方法可以捕获不同时间跨度的入侵传播路径，但无法将他们关联起来。

表 2 威胁检测

方法	论文	检测内容	检测方法/模型	数据集	实时/离线
基于规则	SLEUTH[44]	APT 检测	溯源图标签+自定义策略规则	DARPA TC	实时
	Morse[46]	APT 检测（告警）	标签传播，自定义策略规则	DARPA TC	实时
	Holmes[48]	APT 检测（多步）	Kill-chain，TTP	DARPA TC	实时
	POIROT[64]	APT 检测（告警）	图模式匹配	DARPA TC	离线
	SAQL[72]	企业系统异常检测	基于规则查询	采集实时数据	实时
	Patrol[77]	Zero-Day 攻击路径	规则匹配	真实企业网络数据	实时
基于异常	Steamspot[34]	APT 检测	聚类：异常分数	Steamspot	实时
	Pagoda[68]	检测进程	异常分数（路径异常+图异常）	17 个正常和漏洞应用	实时
	SAQL[72]	企业系统异常检测	统计异常：基于时间序列、不变值、离群值异常查询	采集实时数据	实时
	Frappuccino[78]	PaaS 错误检测	统计异常：时间窗口	Camflow 采集 Paas 实例	实时
	P-Gaussian[79]	检测入侵行为变体	统计异常：（基于证据的高斯分布）	17 个正常和漏洞应用+DARPA APT trace	实时
	unicorn[80]	APT 检测	聚类：图形草图聚类	DARPA TC，StreamSpot	实时
	ZePro[81]	Zero-Day 攻击路径	异常路径贝叶斯网络推理	CVE-2008-0166,CVE-2009-2692,CVE-2011-4089	实时
基于学习	Zitong li 等[40]	APT 检测	注意力图神经网络，深度自编码	LANL，streamspot	离线
	SIGL[82]	安全软件安装	graph lstm 深度自编码	NEC 实验室数据	离线
	Gbadebo 等[83]	Zero-Day 攻击	在线度量学习，基于距离的学习	Camflow 采集攻击数据	实时
	ProvDetector[84]	隐秘的恶意软件	图嵌入，局部离群因子	恶意样本（约 15000 个）	实时

6.1.2 基于异常的检测

异常行为检测方法是建立正常活动的轮廓，对违反正常活动的行为判定为异常。Steamspot[34]建模主机级 APT 检测问题为在流异构图中基于聚类的异常检测任务，作者考虑了图中不同子结构出现的频率，提出了一种基于 shingling 的带时间戳类型图的相似函数来表示异构有序图，并设计 streamhash 维护这些摘要，采用基于质心的在线聚类和异常检测方案。Frappuccino[78]分析了系统级的溯源图，为平台即服务的应用行为建模，它使用动态滑动窗口算法来持续监测和检查应用实例是否符合所学模型。Gao 等人[72]设计了一种特定领域的查询语言 SAQL 分析大规模的溯源数据，但它最终需要专家的领域知识来确定与查询相匹配的元素/模式。PIDAS[67]是一个基于溯源路径的入侵检测和分析系统，它使用溯源图信息作为在线入侵检测的数据源，由于溯源图代表一个对象的历史，记录了入侵发生时受感染的文件、进程和网络连接的依赖关系。它通过计算由一系列依赖关系组成的一定长度路径的异常程度，并与预定的阈值相比较，可以实时判断入侵是否已经发生，但这种方法的缺点在于只使用一条路径来检测入侵，不能反映整个溯源图的行为。Pagoda[68]不仅分析单一路径的异常程度，还分析整个溯源图的异常程度。它首先寻找可能导致入侵的入侵路径，如果找到就不遍历整个溯源图，否则计算出每条路径的异常度，乘以路径长度，得到每条路径的权重值，最后将这些权重值的总和除以所有路径的长度之和。这种方法可以快速识别出入侵过程中只对系统中的一个敏感文件或一个小的文件子集造成损害。P-Gaussian[79]检测入侵行为及其打包加密的变体，他们将入侵行为变体的检测抽象为比较序列顺序或不同序列之间长度的变化。Han 等人[80]设计了一个实时的异常检测系统 Unicorn 来分析从系统活动中产生的流式溯源图。

该检测系统随着主机系统的发展学习动态执行模型，从而捕捉模型中的行为变化，这种学习方法使其适用于检测长期运行的持久性威胁。UNICORN 使用 graph sketching 技术，可以在长时间运行的系统中分析包含丰富上下文和历史信息的溯源图，从而识别未知、慢速攻击。ZePro[81]提出了一种概率方法来识别零日攻击路径。作者构建了一个基于实例图的贝叶斯网络，通过利用入侵起源，贝叶斯网络可以定量计算对象实例被感染的概率，具有高感染概率的对象实例暴露自己并形成零日攻击路径。

6.1.3 基于学习的检测

Li 等[40]基于正样本和负样本经过自编码重构后，异常样本的重构误差更大，提出了基于深度自编码检测系统异常，在 LANL 数据集上验证了 APT 攻击检测的有效性。SIGL[82]是第一个基于溯源图的异常软件安装检测系统，可以在没有事先攻击知识的情况下保证软件安装的安全，他们通过对图中的异常进程节点进行分流，减轻负担。Gbadebo 等[83]提出在线度量学习解决 0-day APT 攻击检测问题，他们首先模拟 APT 攻击，利用 CamFlow 记录日志数据，然后利用 CamQuery 将记录的日志转换为溯源图，再过滤溯源图，生成只包含系统命令执行的子图，最后构造在线度量学习分类器检测区分新型的 APT 攻击、已存在的 APT 攻击和良性事件，在特征提取上，利用图嵌入方法(node2vec)将图转化为向量。ProvDetector[84]采用图嵌入，基于概率密度的局部离群因子来检测隐蔽恶意软件，它使用一种新的基于稀有度的路径选择算法来识别溯源图中表示进程潜在恶意行为的因果路径，然后使用 doc2vec 嵌入模型和离群检测模型确定这些路径是否恶意，实现隐藏的恶意进程检测。

6.2 威胁狩猎

已有基于溯源图的威胁狩猎主要采用基于威胁情报或基于 TTP 驱动的方法。

6.2.1 基于威胁情报驱动

开源威胁情报(OSCTI，open-source cyber threat intelligence)是一种基于证据的知识形式，主要关注妥协指标（IoC）。常见的威胁情报有结构化的情报如 STIX 情报，半结构化数据如 MISP，OpenIoC，非结构化情报数据如安全博客，APT 报告。

（1）威胁情报提取

POIROT[64]采用手动提取威胁情报，构造威胁行为查询图，查询图的顶点表示进程、文件、套接字等，边表示系统调用关系，然后利用图对齐算法匹配基于审计日志构造的溯源图，实现威胁狩猎。该实验数据集主要来源于 STIX，MISP 等结构化或半结构化情报。非结构化的 OSCTI 报告不仅包含妥协指标，还描述了它们之间的关系，如进程和文件之间的读取关系，这种威胁行为可以与攻击步骤联系起来，因此 ThreatRapter[73] 基于 OSCTI 报告，提出了无监督 NLP 管道提取结构化威胁行为图，图的顶点表示 IoCs，边表示 IoC 之间的关系，实现了初始特征和关系的自动提取，其实体的提取精确率:96%，召回率:97.3%，F1 值:96.64% ，关系提取的精确率:96%, 召回率： 89%， F1 值:92% 。EXTRACTOR[85]提出一种新的文本总结方法，他们将攻击行为与其它文本区分开来，使用语义角色标记方法提取攻击行为和句子的主体、客体和行动，并以图形的形式呈现攻击步骤和相关实体之间的因果信息流，通过从非结构化 APT 报告、公开数据集 Darpa TC 3 以及微软等公司的 CTI 报告中提取攻击行为图，并与报告的真实活动（威胁行为图中的边）进行对比，评价精确率、召回率和 F1 值，然后采用 POIROT 系统验证自动生成的攻击行为图可用于威胁狩猎。以上三种方法提取威胁图，都是为了匹配系统溯源图，或查询系统日志，实现威胁狩猎。HINTI[86]首次提出了基于多粒度注意的 IoC 识别方法，其 IoC 包括攻击者、漏洞、设备、平台、恶意文件和攻击类型六种类型，并从开源威胁报告中提取描述 IoC 的关系，构造异质信息网络（HIN），并提出了一个基于图卷积网络的威胁情报计算框架进行知识识别。HINTI 的威胁情报来源于安全博客，黑客论坛等社交网络。作者只对实体提取情况进行了评估，其准确率为 98.59%,精确率 98.72%, 微观 F1 值 98.69%。SecurityKG[87]是一个自动收集和管理 OSCTI 的系统，它通过从各种来源收集 OSCTI 报告，使用人工智能和 NLP 技术的组合来提取威胁行为，并构建一个安全知识图，但没有对提取的准确率进行评价，后两种方法表示了较为丰富的威胁知识，但没有表达系统底层日志行为，不能直接和系统溯源图进行匹配检测。

表 3 威胁情报提取模型

论文	实体类型及关系	提取方法	威胁情报查询	数据集	实验评价
POIROT [64]	实体：进程、文件、套接字、管道等；关系：系统调用	手动提取查询图	图对齐	MISP，STIX	——
ThreatRaptor[73]	实体：进程、文件、套接字；关系：描述关系	利用 spaCy 自动提取 IoC 和 IoC 关系，构建威胁行为图	TBQL 查询	DARPA TC 3，其他 CVE 案例	实体提取精确率:96%，召回率:97.3%，F1 值:96.64% 关系提取的精

					确率:96%,召回率：89%，F1值:92%
EXTRACTOR [86]	实体：进程、文件、套接字； 关系：系统调用	自动提取攻击行为图 实体提取：文本摘要； 关系提取：在依赖解析基础上，考虑语义角色标签，对应系统审计日志	图对齐	1 非结构化真实 CTI 报告，2.Darpa tc（公开数据集），3.微软等 CTI 报告	1.平均精度:0.9 召回率:0.958 F1 值:0.928 2.精度：0.96 召回率:0.94 F1 值:0.95
HINTI [85]	实体：攻击者、漏洞、设备、平台、恶意文件和攻击类型； 关系：描述关系	异构信息网络（HINTI） 实体提取：Xpath 提取，基于注意的多粒度识别 关系提取：定义关系模板	GCN	安全博客，黑客论坛，cve 数据库等	IOC 实体识别准确率为 98.59%,精确率 98.72%,微观 F1 值 98.69%。
SecurityKG[87]	实体：威胁者，技术，工具，软件，多种类型的 IOCs; 关系：来自文本描述	自动提取安全知识图;实体提取：IOC 保护，CRF 模型;关系提取：依赖解析	——	OSCTI 报告	——

（2）基于威胁情报的图匹配

POIROT[64]将威胁狩猎建模为一个不精确的图模式匹配问题，将 STIX，MISP 等格式的威胁情报转化为攻击行为查询子图，进而主要解决威胁情报子图与系统级溯源图的节点概念对齐及匹配问题，其对齐算法包含节点对齐和图对齐，通过计算出查询图和溯源图之间的图形对齐分数，结果显示能在包含数百万节点的图内进行搜索，而且根据查询图中的信息流搜索出溯源图中的对齐节点，可在几分钟内准确定位攻击。DeepHunter[88]也采用威胁情报驱动，手动提取开源报告中的 IoC 关系，然后基于图神经网络将溯源图数据与已知攻击查询图匹配，其网络架构包括属性网络和图形神经网络，属性嵌入网络纳入 IoCs 的信息，图嵌入网络捕获 IoCs 之间的关系。通过五个真实和合成的 APT 攻击场景测试，DeepHunter[88]可以检测所有的攻击行为，而且其准确性和稳健性超过了 POIROT。这两种方法的局限在于威胁子图的构建需要依赖专家知识，而且对未知威胁无能为力。

(3)基于威胁情报的特定领域语言查询

特定领域语言是一种非过程化语言，研究者先后提出了 CyQL[89], τ -calculus[90]和 TBQL[73]等。CyQL 是基于 MITRE CyGraph 多源异构图架构，τ -calculus 是基于 IBM 威胁情报计算时序图分析引擎的静态图查询，Shu 等[90]提出威胁情报计算的方法，将威胁发现作为一个图计算问题。ThreatRaptor[73]通过自动解析开源威胁情报报告，提取 IoC 实体和关系，构建威胁行为图，提出了基于 TBQL 对系统审计日志进行威胁查询，发现恶意的系统活动。该系统首次通过查询合成机制，自动合成一个 TBQL 查询威胁行为，也支持安全分析人员对威胁查询行为进行修改，经过攻击案例评估，证明了在实际威胁猎取中的准确性（精确率:100%，召回率:96.74%），但该系统不能狩猎针对 windows 注册表项的攻击；另外如果自动提取的 OSCTI 文本不可用或几乎不包含有用的 IoC 信息，将限制其应用。WILLE[91]提出利用自然语言处理技术来自动提取和翻译已知的威胁描述，他们采用自动生成特定领域语言（DSL）进行威胁狩猎，此外还考虑使用基于进化论的遗传编程方法增加 IoCs 的遗传扰动，以扩大识别威胁的变体家族。

6.2.2 基于 TTP 驱动

Holmes[48]和 RapSheet[60]都采用基于 TTP 驱动的模式，Holmes 基于攻击链构建高级溯源图，弥合低级系统调用视角和高级攻击链视角之间的语义差距，构建了一个高级别场景图(HSG, high-level scenario graph)作为中间层。HSG 节点是 TTP 中的实体,边表示 TTP 之间的信息流。Holmes[48]是通过专家实现了由底层日志数据到 TTP 的映射，但是该方法完全依赖于专家知识撰写规则，因此，检测结果的好坏严重依赖于安全专家。Rapsheet[60]从战术角度构建攻击溯源，利用攻击行为到 ATT&CK 相关战术映射，实现攻击行为战术溯源，大大减少溯源图规模，基于战术容易获取攻击意图。

7. 基于系统溯源图的取证分析

基于系统溯源图的取证分析主要方法包括基于因果关系、序列学习、特定领域语言查询和语义重建等。表 4 对比分析了近年基于溯源图的取证分

析文献。

表 4 取证分析

分类方法		论文	方法	案例研究	评价方案（结果）
基于因果关系分析	执行单元	Ma 等[31]	解析 ETW 日志为单元，执行后向追踪/前向追踪	错误配置，钓鱼攻击，信息泄露，间谍软件	有效性:匹配正确
		ProPatrol[92]	应用程序执行单元划分（浏览器，邮件等客户端）	远程访问木马，挂马，CSRF and DNS 重定向，即时消息客户端	找出根源，给出还原的因果关系图
	通用溯源	Priotracker[55]	优先考虑异常依赖边前向追踪	3 个案例攻击图，（数据窃取，钓鱼邮件，Shellshock 后门）	给出缩减版前向溯源图
		NoDoze[53]	考虑整个事件链条异常后向追踪/前向追踪	10 个攻击，举例 2 个：数据窃取，Shellshock 后门	完整性：通过溯源图能找到攻击依赖图的比率。生成精确的警告依赖图（1 个 88%，9 个 100%）
		OmegaLog[22]	修改整个系统溯源图，增加 app 日志顶点，形成富含语义，执行分区的通用溯源图	信息泄露攻击，钓鱼邮件	给出传统溯源图和基于 Omegalog 的语义溯源图对比
	记录和重放	RTAG[26]	跨主机调查	6 个攻击场景	信息流匹配事实真相（100%）
	污点分析	Morse[46]	标签传播，重构场景图	Firefox 后门，浏览器扩展，恶意 http 请求，Ccleaner，勒索软件，横向移动，内核恶意软件	给出攻击案例的溯源场景图（入口点和前向分析）
	标签传播	Newsome J[94] Yin H 等[95]	入口点识别（后向追踪），前向分析		
	模型推断	LDX[97],MCI[98]	代码双执行，MCI 利用 LDX 方法	钓鱼邮件和伪装的 FTP 服务器利用 InfoZip 进行信息窃取	给出利用 MCI 模型生成的因果图
基于序列学习		**ATLAS[102]**	序列词法化，采样，序列嵌入，模型学习 调查：攻击实体识别，关联攻击事件	10 个攻击场景(单主机和跨主机两种场景)，1 个案例调查（Pony campaign）	实体识别和事件识别的精确率、召回率和 F1 值，案例调查给出恢复攻击序列和溯源图
基于语义还原的场景重建（抽象行为）		**WATSON[50]**	上下文语义聚合抽象行为	4 个数据集，其中 DARPA TC 3 trace，恶意数据集（8 个实例） 调查案例：配置泄露，内容销毁	行为抽象：F1：92.8%，精确率：92.8%，召回率：94.2% 可视化总结行为实例的信息流
		OmegaLog[22]	多层日志，执行分区	信息泄露攻击，钓鱼邮件	给出 Omegalog 的语义溯源图
		UIScope[33]	关联系统事件和 UI 事件	6 个真实攻击:钓鱼邮件，远程代码执行，office 宏病毒，基于凭证的攻击，水坑攻击，内部攻击	判断是否找到入侵的根源（均正确），提供 1 个案例（远程代码执行）的溯源图
		ALchemist[23]	应用程序日志和系统审计	14 个攻击实例(含 darpa tc) 攻击案例调查：渗出，Azazel attack	攻击取证有效性：审计级日志：精度 92.8%，召回率：99.6%；应用级日志：精度 97.7%，召回率：100% 攻击案例调查的 ALchemist 因果图
基于特定领域语言查询		AIQL[71]	AIQL 查询语言	APT 攻击（包含 5 个步骤）	完整攻击：查询次数，事件匹配，调查时间
		Graalf[47]	Graalf 查询语言	3 个攻击调查案例（darpa tc 3）	查询结果图
		APTrace[56]	BDL 查询语言	5 个攻击实例，其中钓鱼邮件和恶意 excel 宏病毒作为攻击案例	攻击的分析时间，BDL 查询语句，生成依赖图

（注：加粗文献，表示该文献评价方案较细，以节点，边为评价指标，其他较粗，为图级粗略比较）

7.1 基于因果关系的取证分析

BackTracker[8]首次使用溯源图分析入侵，以确定入侵的入口点，为加速调查取证分析，提高准确率和性能，以往研究主要基于两种思路，一种是通过图形压缩和数据缩减减少分析日志，5.1 节已做了详细介绍。另一种思路是解决依赖爆炸和高存储负载[50]，依赖爆炸问题是由于在因果关系分析中，当一个长期运行的进程与许多输入和输出对象相互作用时，每个输出对象都被认为是对所有前面的输入对象存在因果依赖。为缓解依赖爆炸，研究者提出了执行单元分区，污点分析，记录和重放，模型推断等多种方法。

7.1.1 基于执行单元分区的依赖剪枝

Ma 等[31]基于 ETW 审计日志，并对 ETW 进行扩展，记录重要的非系统事件，然后提出一种算法，将日志分析和二进制程序分析结合起来，推导出可以用来解析日志到单元的模型。通过单元分区，精确识别事件之间的因果关系。ProPatrol[92]系统利用了企业应用程序如浏览器，邮件开放式分区设计，该方法不需要应用源二进制工具。而是利用了一些面向互联网的应用程序设计中固有的执行分隔来减轻依赖爆炸，确定真正的依赖关系。Mnemosyne[93]基于浏览器层级划分单元分区调查水坑攻击。

7.1.2 污点分析

污点分析可以精确追踪进程内的信息流，有效防御信息泄露和 0-day 攻击。Newsome J 等[94]提出了自动检测和分析覆盖攻击的动态污点分析方法。Yin H 等[95]提出全系统细粒度污点分析，以辨别未

知代码的细粒度信息访问和处理行为，然而污点分析也带来了负载。Morse[46]为了缓解依赖爆炸，提出了标签衰减（tag attenuation）和标签衰变（tag decay），他们设计构建一个紧凑的场景图，捕捉绝大多数攻击，同时排除良性背景活动，可以将虚警率降低一个数量级以上。

7.1.3 基于记录和重放

Rain[96]使用记录回放技术实现按需细粒度信息流跟踪，他们通过合并进程内溯源分析和进程间的分析可以精确追踪信息流,帮助重建低级别的攻击步骤。使用粗粒度数据如系统调用，开销低，准确度低，而使用细粒度，如指令执行，准确度高，但开销大。RTAG[26]综合二者优势，使用记录和重放，在记录程序运行时，执行高级别的日志记录和分析；在重放程序运行时，执行低级别的日志记录和分析，实现了一种有效的数据流标记和追踪机制，可用于跨主机环境下的攻击调查。

7.1.4 模型推断

LDX[97]是一个双向执行因果推断模型，它通过改变系统调用的输入，观察输出的状态变化来推断系统调用的关系。MCI[98]将可执行文件输入因果推理引擎 LDX，获得程序的因果模型，然后根据解析后的系统日志和相应的模型，得出事件之间的细粒度依赖关系，但该方法的压缩效果取决于大量的软件模型，而实际系统运行许多未知软件，使得该方法覆盖率难以保证，同时软件更新可能导致原始模型失效。

7.1.5 通用溯源技术

Priotracker[55]和 Nodoze[53]提出了基于统计特征的攻击调查方法，他们通过对异常事件和因果依赖进行优先级排序，排序度量指标包括频率和拓扑特性。Priotracker 通过优先探索涉及罕见或可疑事件的路径，加快前向和后向分析，但 Priotracker 仅仅考虑了单个事件的异常，优先考虑表示异常依赖关系的边，而 NoDoze[53]考虑了整个事件链条的异常，提出目标异常路径的方法，使用统计低频路径挖掘的方法解决依赖爆炸问题，从而更准确还原告警产生对应的溯源数据子图，但不能精确定位异常传输的 IP 地址,这种基于统计的方法可能导致不稳定的结果。UIScope[33]利用低层系统事件和 UI 事件相关联，将系统事件归结为单个 UI 元素避免依赖爆炸。

7.2 基于序列学习的取证分析

Hercule[99],Tiresias[100],Attack2vec[101],ATLAS[102]都使用了机器学习技术来建模攻击事件，其中 Hercule[99]使用了社区检测算法来对攻击事件进行关联，他们认为威胁事件与正常事件之间有较明确的社区化行为划分，通过将多源日志融合，以自动化的方式完成异常行为社区发现，归并其对应的攻击步骤。后三者研究均采用了词嵌入将文本信息（序列）转换为向量，Tiresias[100]，Attack2vec[101]仅限于识别和报告日志中的单个日志中的攻击事件，ATLAS[102]的目标是定位攻击，它基于序列学习，在已知攻击症状的情况下，通过邻居图构造序列，经过序列学习获得攻击和非攻击序列，确定所有的攻击实体，重构攻击路径，但只支持 windows 平台，而且无法检测使用类似正常事件序列的隐藏攻击行为，比如模拟攻击。

7.3 基于特定领域语言查询的取证分析

传统基于关系数据库和图形数据库的查询系统缺乏语言结构来表达主要攻击行为的关键属性，而且由于语义无关的设计无法利用系统监测数据的属性来加速查询的执行，所以往往执行查询的效率很低。CamQuery[70]提供了一个可编程的图形处理框架，实现以顶点为中心的查询 API；AIQL[71]通过持久性存储实现取证查询，提出了一个建立在现有监测工具和数据库之上的新型查询系统，实现攻击调查查询语言(AIQL)支持即时的攻击调查。APTrace[56]利用 BDL 语言，实现企业级因果分析查询。它通过给定安全异常警告，利用 BDL 执行向后查询，基于执行窗口分区算法解决依赖爆炸问题，输出溯源子图。但基于执行窗口分区的时间选择是一个难点，时间选择的不同，将影响依赖图的大小和后续的分析。Graalf[47]提供图形化查询系统，可有效的加载，存储，处理，查询和显示计算机取证的系统事件，实时追踪较长事件序列，帮助识别攻击。

7.4 基于语义重建的取证分析

基于特定领域语言的查询取证分析可以呈现系统级的因果关系，不能完全恢复从用户的角度发生的事情。基于语义还原的取证分析包括常规语义还原，实现程序行为动作还原，如将审计日志与应用日志相结合，解决语义鸿沟；也包括攻击场景下，识别告警日志数据中的攻击行为，还原 TTP 语义。TGMiner[103]将感兴趣的行为中挖掘出辨别性的图形模式，并将其作为模板来识别类似行为。Holmes[48]和 RapSheet[60]将多阶段攻击视为符合 TTP 规格的因果事件链。WATSON[50]利用基于系统审计日志知识图的上下文信息来实现语义推断，通过向量表示不同的行为语义，并利用语义相似行为进行聚类。该研究结果表明可以准确抽象出良性和恶意的行为。OmegaLog[22]通过识别和模拟应用层的日志行为，使应用事件与系统层访问准确协调，然后，拦截应用程序的运行时日志活动，并将这些事件移植到系统层溯源图上，使调查人员能够更精确的推断攻击的性质。ALchemist[23]将应用程序日志和审计日志结合起来，基于关系推理引擎 Datalog，推理关键攻击信息，实验证明其性能优于 Nodoze 和 Omegalog。UIScope[33]采集用户界面元素和事件

收集器以及系统事件收集器，将低层次的因果关系分析与高层次的用户界面元素和事件分析相结合，以获得两者的优势。潘亚峰等[104]提出基于ATT&CK 的 APT 攻击语义规则构建，通过将攻击语义文本中的语义知识抽取成对溯源图的检测规则，实现底层审计日志数据到上层 TTP 语义知识的映射，在语义规则匹配过程中设置了最小路径长度和最大路径长度，该方法只能检测出 APT 攻击生命周期中的局部行为。RATScope[105]开发了一个远程访问木马取证分析系统，由于 ETW 不提供任何底层数据的输入参数，导致两个不同的程序调用触发相同的底层系统调用行为，为解决这个语义冲突问题，作者提出了聚合 API 树记录图，利用低级别的系统调用和高级别的应用程序调用栈相结合来建立细粒度的程序行为，因为两个不同的应用程序在应用程序调用栈是明显不同的，从而可以区分 RAT 的潜在恶意功能。

8. 结束语

随着网络攻击的日益复杂，无文件攻击等新型攻击手法越来越隐蔽，从大规模、多源异构日志数据中有效识别复杂攻击及其意图，仍然面临许多挑战。

（1）隐蔽性威胁检测：由于 APT 攻击复杂多变，开源数据集很难获得，目前常用的是 DARPA TC 系列，但文档并不完善，因此研究具有多种新的 APT 攻击，完善文档的开源数据集具有实际意义。无文件攻击手法多样，探索无文件攻击机理和实时未知威胁检测方法成为热点，另外通过多源多模态事件图谱构建，实现可解释的异常检测与威胁定位也是未来研究的一个方向。

（2）自动化威胁狩猎：威胁情报提取主要面向结构化和非结构化威胁情报进行提取，已有研究证明了自动提取威胁情报的准确性和用于威胁狩猎的可行性，但应用开源威胁情报报告中自动提取 IoC 及其关系进行威胁狩猎仍然面临一些问题，如报告中记录的结构形式不统一，记录错误，省略攻击详细步骤等体现出与审计日志不相同的情况。面临这些非规范化格式情况下威胁情报行为提取的准确性等问题，需要进一步研究 NLP+语义辅助威胁行为图进行高精度提取，探索基于学习的方法识别一些特定名词（如 mimikatz）等，进一步拓展 IOC 实体提取，构建更加丰富的威胁行为图。此外基于 TTP 行为图的构建，进化的 IoCs 生成也是可以探索的研究热点。自动化威胁狩猎方法中考虑研究基于自动生成的威胁查询图与自动生成的系统日志溯源图进行图节点对齐，子图匹配等新算法的准确度和效率等问题，以及针对这些技术的评估也是重点。

（3）基于攻击语义的取证分析：基于系统溯源图的取证分析有效关联前向和后向查询，目前主要从缩减日志和降低依赖爆炸两种思路来开展因果取证分析，已提出的基于序列学习的取证分析方案需要大量学习已知的攻击序列，针对底层日志与上层之间的语义鸿沟问题，目前探索了系统日志与应用日志、UI 日志相结合，针对攻击语义，利用 TTP 关联系统审计日志，在取证分析的结果评价方面，主要从告警点出发，基于溯源图来执行后向和前向查询，评价比较粗糙，只评价给出取证的溯源子图。仅有 ATLAS[102]，WATSON[50]和 ALchemist[23]的评价粒度较细。由于关联缺失，跨网络与终端数据难以有效同步日志触发条件，导致多源日志之间很难有效关联；另外分析语义缺失，统计规律很难反映攻击者底层的攻击意图和战术方法；探索知识图谱解决语义鸿沟问题，相比较溯源图，知识图谱的表示形式能支持更复杂的异常图，同时还能包含更多的实体与边的属性，通过知识图谱挖掘事件元信息及上下文，进而进行关系推理，实现攻击路径溯源与取证，是将来的可探索的方向。此外，取证分析的有效性度量也是考量的一个因素。

参考文献：

[1] Binde B E,MccRee R,O'Connor TJ.Assessing Outbound Traffic to Uncover Advanced Persistent Threat [R]. Maryland:SANS Technology Institute,2011.

[2] Eshete B, Gjomemo R, Hossain M N, et al. Attack analysis results for adversarial engagement 1 of the darpa transparent computing program[J]. arXiv preprint arXiv:1610.06936, 2016.

[3] 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.

[4] Zafar F, Khan A, Suhail S, et al. Trustworthy data: A survey, taxonomy and future trends of secure provenance schemes[J]. Journal of Network and Computer Applications, 2017, 94: 50-68.

[5] Tan C, Wang Q, Wang L, et al. Attack Provenance Tracing in Cyberspace: Solutions, Challenges and Future Directions[J]. IEEE Network, 2018, 33(2): 174-180.

[6] Li Z, Chen Q A, Yang R, et al. Threat detection and investigation with system-level provenance graphs: a survey[J].Computers & Security, 2021: 102282.

[7] 潘亚峰,朱俊虎,周天阳.APT 攻击场景重构方法综述[J].信息工程大学学报,2021,22(01):55-60+80.
Pan Yafeng, Zhu Junhu, Zhou Tianyang. Overview of APT attack scenario reconstruction methods[J]. Journal of Information

Engineering University, 2021, 22(01):55-60+80.

[8] King S T, Chen P M. Backtracking intrusions[C]//Proceedings of the nineteenth ACM symposium on Operating systems principles. 2003: 223-236.

[9] 蹇诗婕,卢志刚,杜丹,姜波,刘宝旭.网络入侵检测技术综述[J].信息安全学报,2020,5(04):96-122.
Jian Shijie, Lu Zhigang, Du Dan, Jiang Bo, Liu Baoxu. A review of network intrusion detection techniques[J]. Journal of Information Security,2020,5(04):96-122.

[10] 徐嘉涔,王轶骏,薛质.网络空间威胁狩猎的研究综述[J].通信技术,2020,53(01):1-8
Xu jiacen, Wang Yijun, Xue Zhi. A review of cyberspace threat hunting [J]. Communication technology, 2020,53 (01): 1-8

[11] Palacín, Valentina.Practical Threat Intelligence and Data-Driven Threat Hunting[M]. Packt Publishing,2021:398

[12] secjuice.Breach Detection | Controlling Dwell Time Is About Much More Than Compliance[EB/OL]. http://medium.com/secjuice/controlling-dwell-time-its-about-much-more-than-compliance-23a2149e590e.2021-07-10

[13] secjuice.5 TYPES OF THREAT HUNTING[EB/OL]. http://www.cybersecurity-insiders.com/5-types-of-threat-hunting/（2021-07-10）

[14] Can Sar and Cao Pei. Lineage File System[EB/OL]. http://crypto.stanford.edu/~cao/lineage.2021-07-10

[15] Muniswamy-Reddy K K, Holland D A, Braun U, et al. Provenance-aware storage systems[C]//Usenix annual technical conference, general track. 2006: 43-56.

[16] Muniswamy-Reddy K K, Braun U J, Holland D A, et al. Layering in provenance systems[C]//Proceedings of the 2009 USENIX Annual Technical Conference (USENIX'09). USENIX Association, 2009.

[17] 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. Springer, Berlin, Heidelberg, 2012: 101-120.

[18] Pohly D J, McLaughlin S, McDaniel P, et al. Hi-fi: collecting high-fidelity whole-system provenance[C]// Proceedings of the 28th Annual Computer Security Applications Conference. 2012: 259-268.

[19] Bates A, Tian D J, Butler K R B, et al. Trustworthy whole-system provenance for the linux kernel[C]//24th {USENIX} Security Symposium ({USENIX} Security 15). 2015: 319-334.

[20] Bates A, Butler K, Dobra A, et al. Retrofitting Applications with Provenance-Based Security Monitoring[J]. arXiv preprint arXiv:1609.00266, 2016.

[21] Pasquier T, Han X, Goldstein M, et al. Practical whole-system provenance capture[C]//Proceedings of the 2017 Symposium on Cloud Computing. 2017: 405-418.

[22] Hassan W U, Noureddine M A, Datta P, et al. OmegaLog: High-fidelity attack investigation via transparent multi-layer log analysis[C]///Network and Distributed System Security Symposium. 2020.

[23] Yu L, Ma S, Zhang Z, et al. ALchemist: Fusing Application and Audit Logs for Precise Attack Provenance without Instrumentation[J]. 2021.

[24] Xie Y, Feng D, Liao X, et al. Efficient monitoring and forensic analysis via accurate network-attached provenance collection with minimal storage overhead[J]. Digital Investigation, 2018, 26: 19-28.

[25] Haas S, Sommer R, Fischer M. Zeek-Osquery: Host-Network Correlation for Advanced Monitoring and Intrusion Detection[C]//IFIP International Conference on ICT Systems Security and Privacy Protection. Springer, Cham, 2020: 248-262.

[26] Ji Y, Lee S, Fazzini M, et al. Enabling refinable cross-host attack investigation with efficient data flow tagging and tracking[C]//27th {USENIX} Security Symposium ({USENIX} Security 18). 2018: 1705-1722.

[27] Ji Y. Efficient and refinable attack investigation[D]. Georgia Institute of Technology, 2019.

[28] Lee K H, Zhang X, Xu D. High Accuracy Attack Provenance via Binary-based Execution Partition[C]//NDSS. 2013.

[29] Ma S, Zhang X, Xu D. Protracer: Towards Practical Provenance Tracing by Alternating Between Logging and Tainting[C]//NDSS. 2016.

[30] Ma S, Zhai J, Wang F, et al. {MPI}: Multiple perspective attack investigation with semantic aware execution partitioning[C]//26th {USENIX} Security Symposium ({USENIX} Security 17). 2017: 1111-1128.

[31] Ma S, Lee K H, Kim C H, et al. Accurate, low cost and instrumentation-free security audit logging for windows[C]//Proceedings of the 31st Annual Computer Security Applications Conference. 2015: 401-410.

[32] Lee K H, Zhang X, Xu D. LogGC: garbage collecting audit log[C]//Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. 2013: 1005-1016.

[33] Yang R, Ma S, Xu H, et al. Uiscope: Accurate, instrumentation-free, and visible attack investigation for gui applications[C]//Network and Distributed Systems Symposium. 2020.

[34] Manzoor E, Milajerdi S M, Akoglu L. Fast memory-efficient anomaly detection in streaming heterogeneous graphs[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016:

1035-1044.

[35] The CERT Division. 2018. Insider Threat Tools. http://www.cert.org/insider-threat/tools/

[36] Kent A D. Comprehensive, multi-source cyber-security events data set[R]. Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2015.

[37] Transparent computing engagement 5 data release.http://drive.google.co-m/drive/folders/1okt4AYElyBohW4XiOBqmsvjwXsnUjLVf, 2019

[38] Angelos Keromytis. Transparent computing engagement 3 data release.http://github.com/darpa-i2o/Transparent-Computing,2018

[39] Anjum M M, Iqbal S, Hamelin B. Analyzing the Usefulness of the DARPA OpTC Dataset in Cyber Threat Detection Research[C]//Proceedings of the 26th ACM Symposium on Access Control Models and Technologies. 2021: 27-32.

[40] Li Z, Cheng X, Sun L, et al. A Hierarchical Approach for Advanced Persistent Threat Detection with Attention-Based Graph Neural Networks[J]. Security and Communication Networks, 2021, 2021.

[41] Li M, Li Q, Xuan G, et al. Identifying compromised hosts under APT using DNS request sequences[J]. Journal of Parallel and Distributed Computing, 2021, 152: 67-78.

[42] 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.

[43] Liu F, Wen Y, Wu Y, et al. MLTracer: Malicious Logins Detection System via Graph Neural Network[C]//2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, 2020: 715-726.

[44] Cochrane T, Foster P, Chhabra V, et al. SK-Tree: a systematic malware detection algorithm on streaming trees via the signature kernel[C]//2021 IEEE International Conference on Cyber Security and Resilience (CSR). IEEE, 2021: 35-40.

[45] 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.

[46] Hossain M N, Sheikhi S, Sekar R. Combating dependence explosion in forensic analysis using alternative tag propagation semantics[C]//2020 IEEE Symposium on Security and Privacy (SP). IEEE, 2020: 1139-1155.

[47] Setayeshfar O, Adkins C, Jones M, et al. Graalf: Supporting graphical analysis of audit logs for forensics[J]. Software Impacts, 2021, 8: 100068.

[48] 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.

[49] Hossain M N, Wang J, Weisse O, et al. Dependence-preserving data compaction for scalable forensic analysis[C]//27th {USENIX} Security Symposium ({USENIX} Security 18). 2018: 1723-1740.

[50] Zeng J, Chua Z L, Chen Y, et al. Watson: Abstracting behaviors from audit logs via aggregation of contextual semantics[C]//Proceedings of the 28th Annual Network and Distributed System Security Symposium, NDSS. 2021.

[51] 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.

[52] Benabderrahmane S, Berrada G, Cheney J, et al. A Rule Mining-Based Advanced Persistent Threats Detection System[J]. arXiv preprint arXiv:2105.10053, 2021.

[53] Hassan W U, Guo S, Li D, et al. Nodoze: Combatting threat alert fatigue with automated provenance triage[C]//Network and Distributed Systems Security Symposium. 2019.

[54] Myneni S, Chowdhary A, Sabur A, et al. Dapt 2020-constructing a benchmark dataset for advanced persistent threats[C]//International Workshop on Deployable Machine Learning for Security Defense. Springer, Cham, 2020: 138-163.

[55] Liu Y, Zhang M, Li D, et al. Towards a Timely Causality Analysis for Enterprise Security[C]//NDSS. 2018.

[56] Gui J, Li D, Chen Z, et al. APTrace: A Responsive System for Agile Enterprise Level Causality Analysis[C]//2020 IEEE 36th International Conference on Data Engineering (ICDE). IEEE, 2020: 1701-1712.

[57] Xu Z, Wu Z, Li Z, et al. High fidelity data reduction for big data security dependency analyses[C]//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 2016: 504-516.

[58] Michael N, Mink J, Liu J, et al. On the Forensic Validity of Approximated Audit Logs[C]//Annual Computer Security Applications Conference. 2020: 189-202.

[59] Tang Y, Li D, Li Z, et al. Nodemerge: Template based efficient data reduction for big-data causality analysis[C]//Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018: 1324-1337.

[60] Hassan W U, Bates A, Marino D. Tactical provenance analysis for endpoint detection and response systems[C]//2020 IEEE Symposium on Security and Privacy (SP). IEEE, 2020: 1172-1189.

[61] Fei P, Li Z, Wang Z, et al. {SEAL}: Storage-efficient Causality Analysis on Enterprise Logs with Query-friendly Compression[C]//30th {USENIX} Security Symposium ({USENIX} Security 21). 2021.

[62] Zhu T, Wang J, Ruan L, et al. General, Efficient, and Real-time Data Compaction Strategy for APT Forensic Analysis[J]. IEEE Transactions on Information Forensics and Security, 2021.

[63] Gao P, Shao F, Liu X, et al. A system for efficiently hunting for cyber threats in computer systems using threat intelligence[C]//2021 IEEE 37th International Conference on Data Engineering (ICDE). IEEE, 2021: 2705-2708.

[64] Milajerdi S M, Eshete B, Gjomemo R, et al. Poirot: Aligning attack behavior with kernel audit records for cyber threat hunting[C]//Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. 2019: 1795-1812.

[65] Hassan W U, Li D, Jee K, et al. This is Why We Can't Cache Nice Things: Lightning-Fast Threat Hunting using Suspicion-Based Hierarchical Storage[C]//Annual Computer Security Applications Conference. 2020: 165-178.

[66] Ma S, Zhai J, Kwon Y, et al. Kernel-supported cost-effective audit logging for causality tracking[C]//2018 {USENIX} Annual Technical Conference ({USENIX}{ATC} 18). 2018: 241-254.

[67] Xie Y, Feng D, Tan Z, et al. Unifying intrusion detection and forensic analysis via provenance awareness[J]. Future Generation Computer Systems, 2016, 61: 26-36.

[68] Xie Y, Feng D, Hu Y, et al. Pagoda: A hybrid approach to enable efficient real-time provenance based intrusion detection in big data environments[J]. IEEE Transactions on Dependable and Secure Computing, 2018, 17(6): 1283-1296.

[69] Gao P, Xiao X, Li Z, et al. A query system for efficiently investigating complex attack behaviors for enterprise security[J]. arXiv preprint arXiv:1810.03464, 2018.

[70] Pasquier T, Han X, Moyer T, et al. Runtime analysis of whole-system provenance[C]//Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018: 1601-1616.

[71] Gao P, Xiao X, Li Z, et al. {AIQL}: Enabling efficient attack investigation from system monitoring data[C]//2018 {USENIX} Annual Technical Conference ({USENIX}{ATC} 18). 2018: 113-126.

[72] Gao P, Xiao X, Li D, et al. {SAQL}: A stream-based query system for real-time abnormal system behavior detection[C]//27th {USENIX} Security Symposium ({USENIX} Security 18). 2018: 639-656.

[73] Gao P, Shao F, Liu X, et al. Enabling efficient cyber threat hunting with cyber threat intelligence[C]//2021 IEEE 37th International Conference on Data Engineering (ICDE). IEEE, 2021: 193-204.

[74] Gao P, Xiao X, Li D, et al. Querying Streaming System Monitoring Data for Enterprise System Anomaly Detection[C]//2020 IEEE 36th International Conference on Data Engineering (ICDE). IEEE, 2020: 1774-1777.

[75] 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, 2020.

[76] Ding X, Liu B, Jiang Z, et al. Spear Phishing Emails Detection Based on Machine Learning[C]//2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD). IEEE, 2021: 354-359.

[77] Dai J, Sun X, Liu P. Patrol: Revealing zero-day attack paths through network-wide system object dependencies[C]//European Symposium on Research in Computer Security. Springer, Berlin, Heidelberg, 2013: 536-555.

[78] Han X, Pasquier T, Ranjan T, et al. Frappuccino: Fault-detection through runtime analysis of provenance[C]//9th {USENIX} Workshop on Hot Topics in Cloud Computing (HotCloud 17). 2017.

[79] Xie Y, Wu Y, Feng D, et al. P-Gaussian: Provenance-Based Gaussian Distribution for Detecting Intrusion Behavior Variants Using High Efficient and Real Time Memory Databases[J]. IEEE Transactions on Dependable and Secure Computing, 2019.

[80] Han X, Pasquier T, Bates A, et al. Unicorn: Runtime provenance-based detector for advanced persistent threats[EB/OL]. arXiv preprint arXiv:2001.01525, 2020.

[81] Sun X, Dai J, Liu P, et al. Using Bayesian networks for probabilistic identification of zero-day attack paths[J]. IEEE Transactions on Information Forensics and Security, 2018, 13(10): 2506-2521.

[82] Han X, Yu X, Pasquier T, et al. {SIGL}: Securing Software Installations Through Deep Graph Learning[C]//30th {USENIX} Security Symposium ({USENIX} Security 21). 2021.

[83] Ayoade G, Akbar K A, Sahoo P, et al. Evolving Advanced Persistent Threat Detection using Provenance Graph and Metric Learning[C]//2020 IEEE Conference on Communications and Network Security (CNS). IEEE, 2020: 1-9.

[84] Wang Q, Hassan W U, Li D, et al. You Are What You Do: Hunting Stealthy Malware via Data Provenance Analysis[C]//NDSS. 2020.

[85] Satvat K, Gjomemo R, Venkatakrishnan V N. EXTRACTOR: Extracting Attack Behavior from Threat Reports[J]. arXiv preprint arXiv:2104.08618, 2021.

[86] Zhao J, Yan Q, Liu X, et al. Cyber Threat Intelligence Modeling Based on Heterogeneous Graph Convolutional Network[C]//23rd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2020). 2020: 241-256.

[87] Gao P, Liu X, Choi E, et al. A System for Automated Open-Source Threat Intelligence Gathering and Management[EB/OL]. arXiv preprint arXiv:2101.07769, 2021.

[88] Wei R, Cai L, Yu A, et al. DeepHunter: A Graph Neural Network

Based Approach for Robust Cyber Threat Hunting[EB/OL]. arXiv preprint arXiv:2104.09806, 2021.

[89] Noel S, Harley E, Tam K H, et al. CyGraph: graph-based analytics and visualization for cybersecurity[M].Handbook of Statistics. Elsevier, 2016, 35: 117-167.

[90] Shu X, Araujo F, Schales D L, et al. Threat intelligence computing[C]//Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018: 1883-1898.

[91] Karuna P, Hemberg E, O'Reilly U M, et al. Automating Cyber Threat Hunting Using NLP, Automated Query Generation, and Genetic Perturbation[J]. arXiv preprint arXiv:2104.11576, 2021.

[92] Milajerdi S M, Eshete B, Gjomemo R, et al. Propatrol: Attack investigation via extracted high-level tasks[C]//International Conference on Information Systems Security. Springer, Cham, 2018: 107-126.

[93] Allen J, Yang Z, Landen M, et al. Mnemosyne: An Effective and Efficient Postmortem Watering Hole Attack Investigation System[C]//Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. 2020: 787-802.

[94] Newsome J, Song D X. Dynamic Taint Analysis for Automatic Detection, Analysis, and SignatureGeneration of Exploits on Commodity Software[C]//NDSS. 2005, 5: 3-4.

[95] Yin H, Song D, Egele M, et al. Panorama: capturing system-wide information flow for malware detection and analysis[C]//Proceedings of the 14th ACM conference on Computer and communications security. 2007: 116-127.

[96] Ji Y, Lee S, Downing E, et al. Rain: Refinable attack investigation with on-demand inter-process information flow tracking[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017: 377-390.

[97] Kwon Y, Kim D, Sumner W N, et al. Ldx: Causality inference by lightweight dual execution[C]//Proceedings of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems. 2016: 503-515.

[98] Kwon Y, Wang F, Wang W, et al. MCI: Modeling-based Causality Inference in Audit Logging for Attack Investigation[C]//NDSS. 2018.

[99] Pei K, Gu Z, Saltaformaggio B, et al. Hercule: Attack story reconstruction via community discovery on correlated log graph[C]//Proceedings of the 32Nd Annual Conference on Computer Security Applications. 2016: 583-595.

[100] 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.

[101] Shen Y, Stringhini G. Attack2vec: Leveraging temporal word embeddings to understand the evolution of cyberattacks[C]//28th {USENIX} Security Symposium ({USENIX} Security 19). 2019: 905-921.

[102] Alsaheel A, Nan Y, Ma S, et al. {ATLAS}: A Sequence-based Learning Approach for Attack Investigation[C]//30th {USENIX} Security Symposium ({USENIX} Security 21). 2021.

[103] Zong B, Xiao X, Li Z, et al. Behavior query discovery in system-generated temporal graphs[J].arXiv preprint arXiv:1511.05911, 2015.

[104] 潘亚峰,周天阳,朱俊虎,曾子懿.基于 ATT&CK 的 APT 攻击语义规则构建[J].信息安全学报,2021,6(03):77-90.
Pan Yafeng,Zhou Tianyang,Zhu Junhu,Zeng Ziyi. Semantic rule construction for APT attacks based on ATT&CK[J]. Journal of Information Security,2021,6(03):77-90.

[105] R. Yang et al.RATScope: Recording and Reconstructing Missing RAT Semantic Behaviors for Forensic Analysis on Windows[J].IEEE Trans. Dependable and Secure Comput..2020: 1–1

[作者简介]

冷涛（1986 年– ），男，四川合江人，中国科学院大学博士生，四川警察学院副教授，主要研究方向为 APT 攻击检测、取证分析。

蔡利君（1988 年– ），男，河南汝南人，博士，中科院信息工程研究所助理研究员，主要研究方向为攻击检测、内部威胁检测。

于爱民（1980 年– ），男，山西临汾人，博士，中国科学院信息工程研究所正高级工程师、博士生导师，主要研究方向为可信软件测评、基于大数据的行为异常检测。

朱子元（1980 年– ），男，河南汝州人，博士，中国科学院信息工程研究所研究员、博士生导师，主要研究方向为处理器安全技术、系统安全理论与技术等。

马建刚（1990 年—），男，河北衡水人，硕士，中科院信工所高级工程师，主要研究方向为对抗网络高仿真、数据安全。

李超飞（1994 年– ），男，河南汝州人，中国科学院大学博士生，主要研究方向为加密流量、深度学习等。

牛瑞丞（1994 年– ），男，云南昆明人，中国科学院大学博士生，主要研究方向为恶意代码检测、深度学习等。

孟丹（1965 年– ），男，黑龙江人，博士，中国科学院信息工程研究所所长、研究员、博士生导师，主要研究方向为计算机系统安全、云计算安全等。