Computer Science and Application 计算机科学与应用, 2022, 12(1), 233-251
Published Online January 2022 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2022.121024

Hans汉斯

基于主机的高级持续威胁检测技术综述

徐志强[1,2]，文　雨[2]

[1]中国科学院大学网络空间安全学院，北京
[2]中国科学院信息工程研究所，北京

收稿日期：2021年12月25日；录用日期：2022年1月21日；发布日期：2022年1月28日

摘　要

近年高级持续威胁(Advanced Persistent Threat, APT)已成为威胁国家安全、组织机构利益和个人隐私的严重网络空间安全危害。APT具有攻击过程复杂、隐蔽性高和破坏性强的特点，极难被检测和防御。而主机系统通常是APT活动的主要攻击目标。因此关注基于主机的APT检测技术的研究进展和未来趋势具有重要意义。本文首先总结了APT的生命周期和各攻击阶段特点及主机安全问题。接着介绍了主机实体类型及其行为数据类型。然后系统化总结了基于主机实体行为的APT检测技术。又归纳了威胁检测评价数据集和评价指标。最后总结了当前技术挑战并展望了未来研究方向。

关键词

高级持续威胁，主机实体，威胁检测，信息安全

A Survey of Host-Based Advanced Persistent Threat Detection Technology

Zhiqiang Xu[1,2], Yu Wen[2]

[1]School of Cyber Security, University of Chinese Academy of Sciences, Beijing
[2]Institue of Information Engineering, Chinese Academy of Sciences, Beijing

Received: Dec. 25th, 2021; accepted: Jan. 21st, 2022; published: Jan. 28th, 2022

Abstract

Recently, Advanced Persistent Threat (Advanced Persistent Threat, APT) has become a serious problem in cyber security that threatens national security, organizational interests and personal privacy. APTs are difficult to be defended against and detected because of their complex attack process, high concealment, and strong destruction. Host systems are often the primary target of APT activities. Therefore, it is of great significance to focus on the research progress and future

文章引用: 徐志强, 文雨. 基于主机的高级持续威胁检测技术综述[J]. 计算机科学与应用, 2022, 12(1): 233-251.
DOI: 10.12677/csa.2022.121024

trend of host-based APT detection. This paper first summarizes the life cycle of APT and characteristics of each attack stage and host security issues. It then introduces the types of host entities and the types of their behavior data. Then host entity behavior based APT detection techniques are systematically summarized. The evaluation methods of threat detection techniques are introduced, including data sets and evaluation metrics. Finally, the technical challenges and future research are concluded.

Keywords

Advanced Persistent Threat, Host Entity, Threat Detection, Cyber Security

1. 引言

高级持续威胁(APT)是一种高度复杂并极具隐蔽性和破坏性的网络空间安全威胁。早在 2005 年，英国和美国的计算机应急响应机构发布相关安全报告，提醒组织机构和个人用户注意某些针对性的网络钓鱼攻击。随即在 2006 年，“APT”一词出现并被用于指称这类攻击。随着近年信息技术和攻防对抗的不断演进，APT 已成为威胁国家安全、组织机构利益和个人用户隐私的严重网络空间安全危害。

2010 年曝出震惊全球的震网(Stuxnet)攻击[1]。攻击者利用定制化恶意软件，实现了对伊朗纳坦兹核电站基础设施入侵，导致上千台铀离心机无法正常工作且没有被及时发现。同年 Google 员工遭到极光(Aurora)攻击[2]。攻击者利用针对 Google 特定员工的恶意连接，诱骗员工电脑下载和安装远程控制木马，并被劫持用于暴力破解服务器管理员权限，最后攻击者在 Google 内网中窃取资料数据长达数月。2020 年，网络安全公司 FireEye 发现网管软件厂商 SolarWinds 的软件产品被攻击者植入了后门程序[3]。该后门程序可与攻击者服务器进行通信，而 SolarWinds 的重要客户通过购买和安装该软件产品而被成功渗透。最后经 SolarWinds 排查，全球至少 30 万家大型政府和企业机构受到影响。

由于组织机构(如政府部门、军工单位、商业公司等)内部网络中的主机系统(通常包括用户终端、应用服务器和网络设备等)通常处理或存储重要的机密数据，因此该主机系统通常成为 APT 的主要攻击目标。通过入侵主机系统，攻击者可窃取关键信息和数据、实施基础设施破坏或作为下一步攻击的跳板。如，利用服务器漏洞盗取机密数据，入侵网络设备篡改防火墙配置，或通过监听用户终端窃取用户口令等。

虽然采用安全加固的软硬件部件和基于已知攻击向量特征(如恶意软件签名)的检测技术，能够缓解主机入侵威胁，但是难以抵御 APT 攻击。APT 攻击通常在实现目标网络中主机入侵后，会进一步利用主机存在的漏洞(如零日漏洞)获取高级权限，或窃取和破解合法身份，来绕过系统常规防御机制。并继续利用合法权限渗透其它主机，以扩大其据点和入侵范围。此外，通过向外与攻击者取得联络，又持续获得高级攻击工具和外部支持，以保持对入侵主机长期隐蔽的攻击。

近年，基于主机的 APT 检测逐渐引起重视并成为研究热点，特别是大数据和人工智能技术的兴起，出现了许多新的检测技术和方法。本文在此背景下，旨在对该研究领域的研究现状进行系统化总结，对 APT 特点、主机行为数据类型、检测技术、评价方法等进行介绍和分析，并提出当前挑战和展望未来研究方向。本文主要贡献包括：1) 总结了 APT 生命周期，分类阐述了 APT 不同阶段攻击特点和主机安全

性问题；2) 全面介绍了主机实体类型及其行为数据类型；3) 系统化总结了基于实体行为的 APT 检测技术；4) 归纳了当前威胁检测评价数据集和评价指标；5) 对该研究领域的当前挑战和未来发展进行了展望。

2. APT 的生命周期

APT 是一种隐蔽性很强的新型攻击形式。具体来说，攻击者首先对信息网络中的某台主机进行渗透和劫持，然后通过提权或盗取身份凭证进行长期潜伏，并据此在网络中进行横向移动，最终锁定目标主机并实施攻击(如窃取数据)。虽然具体的 APT 不尽相同，且采用了各种不同的攻击手段，但是已有学者指出 APT 的发展过程具有一定规律性，并提出构建高 APT 生命周期的方法[4] [5]。Chen 等[4]提出六阶段划分：侦查和构建攻击工具、工具传送、初始入侵、命令和控制目标、横向移动、窃取数据。而 Zhu 等[5]则划分为诱饵、触发、安装和命令控制目标四个阶段。

基于对不同划分形式的梳理和对比，我们发现它们的差异主要在于各阶段的细化程度和表述。虽然具体的 APT 通常具备独特特征，然而其攻击阶段极为相似，不同之处在于每阶段所采用的具体攻击技术[4]。因此，本文综合这些已有工作，将 APT 的生命周期概括为八个阶段，如图 1 所示：1) 攻击部署，2) 工具投递，3) 初始入侵，4) 命令与控制(C&C)，5) 提权和获取身份证书，6) 横向移动，7) 窃取数据，8) 掩盖痕迹。这些基本涵盖了 APT 主要攻击阶段。

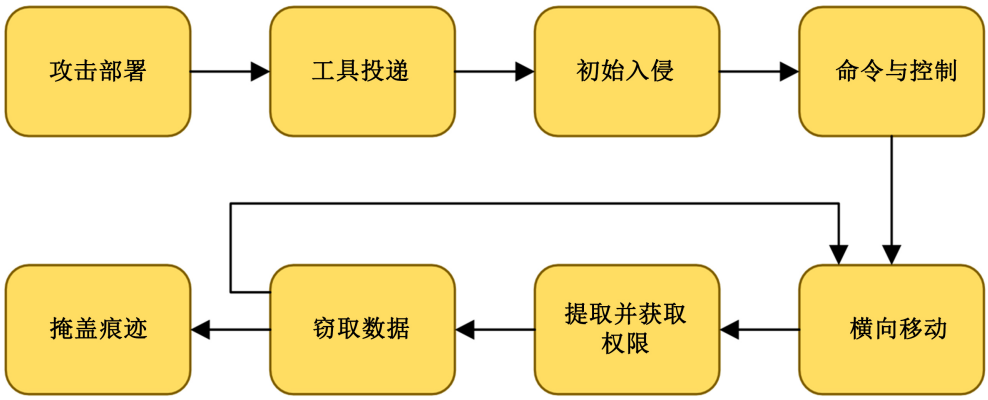

Figure 1. APT life cycle
图 1. APT 的生命周期

攻击部署阶段。攻击者旨在构建引诱主机用户下载攻击工具(如木马程序)的陷阱，包括恶意域名和恶意文档两种方式。通过构建恶意域名，攻击者将含有木马等恶意程序的恶意页面伪装成供用户访问的网络链接。恶意程序是指意图破坏系统机密性、完整性或可用性的恶意软件。早在 2012 年，卡巴斯基的调查报告[6]指出，87.4%的网络攻击与恶意域名相关。此外，攻击者还可将指向恶意域名的链接或可被文档编辑器执行的漏洞利用恶意代码植入供用户访问的文档。2018 年，卡巴斯基的季度安全报告[7]提到，70%的软件漏洞利用是针对微软 office 应用软件的恶意 word 文档。

工具投递阶段。攻击者通过欺骗方式使主机用户访问恶意域名或恶意文档，达到在未经允许情况下在主机系统中下载和安装恶意程序，实现入侵主机的目的。攻击者主要采用的攻击方式包括路过式下载和网络钓鱼攻击。在路过式下载攻击中，攻击者首先针对主机用户可能经常访问的正常网站进行渗透活动。在获得网站权限后，通过篡改网页，嵌入自动指向恶意域名的重定向代码。当主机访问该网页时，会自动跳转访问恶意域名指向的恶意网站。最后，恶意网站通过向主机发送含有恶意代码(如木马程序)的网络页面，通过主机系统漏洞利用，实现在主机下载和安装木马等恶意程序的目的。2015 年，Mcafee

实验室的安全威胁报告[8]提到，单个季度中发现的新增恶意 URL 超过 2000 万个。2019 年，赛门特克的互联网安全威胁报告[9]指出，2018 年全球新发现的恶意 URL 接近 3 亿个。此外，攻击者还可以通过邮件等形式，将含有指向恶意域名链接的邮件内容或含有恶意代码的附件，发送给主机用户，使得恶意程序被自动下载和安装到主机系统。2021 年，Verizon 的数据泄露调查报告(DBIR) [10]称，2020 年证实的 5250 起数据泄露事件中，有 36%与网络钓鱼攻击有关，排在所有攻击类型的第一位。

初始入侵阶段。由于恶意程序种类很多，在一般的主机入侵攻击中，当工具投递成功后，主机中可能被秘密安装了各种类型的恶意软件，包括病毒、蠕虫、特洛伊木马、勒索软件、间谍软件、Rootkit 程序等。对于 APT，本阶段主机入侵的主要方式包括木马等恶意程序。与病毒、蠕虫等以传播和感染为目的的恶意程序不同，木马程序一般通过伪装成正常进程，或通过提权获得更高系统权限，在主机系统的后台秘密执行各种后续的攻击任务，如与攻击者控制平台连接、窃取用户身份凭证、横向移动等。例如，臭名昭著的木马程序 Zeus [11] (也称为 Zbot)能够执行多种恶意任务，包括通过记录用户的击键操作窃取用户账户信息。2009 年 6 月，安全公司 Prevx 披露，Zeus 入侵了包括美国广播公司、亚马逊公司、商业周刊、思科公司、美国宇航局、甲骨文公司、美国银行等许多机构的网站主机，劫持了超过 74,000 个 FTP 帐户。

命令与控制阶段。恶意程序通过利用被劫持的主机系统与攻击者的控制平台取得联系，并在两者之间建立一个可被攻击者用来直接控制被劫持主机的通信通道(如远程访问工具)，以获取后续攻击活动指示和其它攻击工具(如间谍程序等)，同时向攻击者泄露窃取的信息和数据。攻击者的控制平台可以是一台集中式服务器，也可以是分布在僵尸网络中的多台服务器。命令与控制阶段主要涉及恶意域名、网络隐蔽信道、远程控制木马等攻击方式。恶意程序可以通过访问恶意域名直接与攻击者控制平台取得联系，也可以使用网络协议或网络数据包的协议字段、时间等特征与控制平台建立网络隐蔽信道，以绕过网络安全设备和系统(如网络防火墙)的监控，达到秘密传输信息的目的。远程控制木马允许攻击者远程、秘密地控制被入侵和劫持的主机。远程控制木马不仅可以远程下载或上传文件，监视键盘敲击和主机屏幕，还可以自动从被入侵主机向攻击者的控制平台发送连接请求，从而建立一条穿透防火墙连接双方的加密隧道。通过这种加密隧道，攻击者可使用控制平台发送命令来控制被入侵主机，如指使远程控制木马开展工具下载、网络探测、搜索敏感信息等活动，并将收集到的数据发送给攻击者等。由于远程控制木马可以像一般的网络会话一样使用系统的 80 至 443 端口，并对信道进行加密，因此具有绕过端口过滤和深度包检测[12] [13]等防御机制的能力。

提权并获取凭证阶段。攻击者通过控制平台向被入侵主机投放更强的攻击工具，或者远程指使木马程序，进一步获得更高的系统权限或合法的身份凭证，以使其攻击活动变得更加隐蔽，不易被系统防御机制所察觉。主要的攻击方式包括 Rootkit 攻击、间谍软件和零日攻击。Rootkit 是一种攻击隐藏技术，一般通过对应用编程接口(API)进行代码插桩来让恶意代码获得应用服务或系统的访问权限，或通过加载恶意内核模块和设备驱动程序来篡改操作系统内核的结构体和提升恶意代码的运行级别。间谍软件是一种对用户主机活动进行监视的恶意程序[13]。攻击者通常利用间谍软件偷窥和收集用户的键盘敲击、主机登录和账户口令等信息。例如前面提到的窃取了大量用户 FTP 账号的木马程序 Zeus。零日攻击是一种高级的攻击类型，该类攻击可以通过利用主机中尚未有修补方案的漏洞来入侵系统内核并进行提权。2010 年，对伊朗纳坦兹核电站造成严重影响的震网蠕虫病毒(Stuxnet) [1]，通过利用 Windows 操作系统存在的多个零日漏洞获取了该核电站内许多主机的系统控制权，致使该核电站基础设施发生了大面积的失控。

横向移动阶段。在持续攻击过程中，攻击者会在被入侵主机所在网络中横向移动，进一步扩大其在该网络中的立足点，并逐渐接近更高价值的攻击目标和扩大攻击战果。横向移动阶段的主要攻击方式包括内部攻击和蠕虫病毒等恶意程序。伪装攻击属于内部攻击的一种，指攻击者通过利用在前面攻击阶段

获得的合法权限(如主机用户账号和口令)伪装成普通用户，入侵和劫持网络中其它主机来扩大其控制范围，接着通过浏览、检索这些主机的文件和数据来发现和窃取高价值信息。内部威胁广泛存在于各种信息系统和网络中，包括社交网络[14]、拍卖系统[15]、网络服务器[16]、数据库系统[17]和应用服务[18]等。蠕虫病毒是一种能够通过在网络中复制和传播自身来感染网络中其它主机的独立运行的病毒程序。蠕虫病毒在网络中的传播力很强。例如，臭名昭著的 Love Gate、CodeRed、SQL Slammer、MyDoom、Storm Worm 等蠕虫病毒已经成功攻击了数亿台 Windows 主机。2001 年，CodeRed 蠕虫被释放的第一天，就 36 万台互联网主机被感染。蠕虫病毒甚至能够突破网络物理隔离，在不同网络中进行传播和扩散。例如，前面提到针对伊朗核电站攻击的震网病毒 Stuxnet，首先通过在互联网上传播，进而感染核电站工作人员的个人电脑，最终通过存储介质被带入物理隔离的核电站工控网络。

窃取数据阶段。当攻击者在网络中进行横向移动时，一旦成功入侵有攻击价值的主机后，会进一步窃取主机里的关键信息和数据。窃取数据阶段的攻击类型主要包括间谍软件、侧信道攻击和伪装攻击。与提权并获取凭证阶段类似，间谍软件会通过监视主机系统里的用户活动(如用户击键操作)来收集系统敏感信息(如用户账号和口令)。与横向移动阶段不同，这里的伪装攻击指攻击者利用获得的数据访问权限(如数据库账户和口令)伪装成合法用户进行数据访问操作。而侧信道攻击则能够利用系统底层共享部件(如处理器缓存等)的漏洞，通过观察这些底层部件的访问时间特征，间接的推断出存储在这些部件中的系统关键信息(如数据密钥)。

掩盖痕迹阶段。攻击者在持续攻击网络主机时，为了阻碍甚至避免被安全分析员追踪和溯源，会刻意抹掉遗留在系统中留下的各种攻击痕迹。例如，攻击者可以利用获得的系统权限伪装成系统管理员，卸载攻击工具和审计工具，或删除审计日志等。

需要指出的是，实际的 APT 攻击可能包含以上阶段的多次循环，例如多次提权并获取凭证、横向移动和窃取数据，其中也可能多次使用命令和控制阶段建立的秘密信道与攻击者控制平台进行数据传输，以实现持续的对网络中的主机进行攻击。

3. 主机实体行为数据

本节主要对 APT 检测所涉及的实体行为数据及其特点进行了归纳和总结。近期，已有一些工作提出了安全数据特点及其分类方法。例如，王琴琴等[19]指出安全数据具备 5V 特征：Volume (大量)、Variety (多样)、Value (低价值密度)、Velocity (高速)和 Veracity (真实)。Jing 等[20]将安全数据分为 packet-level data (包级数据)、flow-level data (流级数据)、connection-level data (连接级数据)和 host-level data(主机级数据)四个级别。虽然这些工作较好的概括了安全数据的特点和采集层次，但是仍不能全面体现 APT 胁检测数据的特点。

在第 2 节，我们对 APT 各阶段攻击目的和攻击方式进行了介绍，可以看出 APT 涉及网络和主机中的多种实体类型，主要包括恶意域名(包括恶意 URL 和网页)、恶意文档、恶意邮件、恶意程序和伪装用户。自然地，这些实体相关的信息和特征就成为 APT 检测任务必须考虑的数据采集来源。由于 APT 相关实体的信息和特征具有非常显著的差异性和多样性，因此我们需要对各种检测数据的特点进行更为细致的分类和梳理。通过相关文献调研和分析，我们围绕 APT 实体类型，从 3 个维度对需要的检测数据的特点进行了总结，如表 1 所示。首先，我们按照实体类型，将检测数据分为 5 类，包括恶意域名、恶意文档、恶意邮件、恶意程序和伪装用户。其次，我们从攻击阶段、实体特征和采集位置 3 个维度对各类检测数据的特点进行了分析。

第一个数据维度是检测数据覆盖 APT 攻击阶段的情况。其中，恶意域名、恶意文档和恶意邮件等实体数据主要用于检测 APT 的前期阶段，而恶意程序和伪装用户相关数据则是检测主要攻击阶段的关键数

据。实际上，APT 的前期阶段采用的攻击方式同样是许多普通主机入侵攻击的惯常方法，只是这些入侵攻击可能没有命令与控制、横向移动等高级阶段。虽然这些数据不能直观地反映出主机中是否存在 APT 攻击，但是对于后面阶段的攻击检测仍然有一定帮助。例如，某个被检测出的恶意域名，如果被某个主机进程所访问，则该进程很可能正在执行恶意代码。

Table 1. Classification of data for APT detection
表 1. APT 检测数据分类

数据类型	攻击阶段								实体特征		采集位置	
	攻击部署	工具投递	初始入侵	命令与控制	提权并获取凭证	横向移动	窃取数据	掩盖痕迹	静态	动态	网络侧	主机侧
恶意域名	√	√		√					√	√	√	
恶意文档	√								√	√	√	
恶意邮件		√							√		√	
恶意程序			√	√	√	√	√		√	√	√	√
伪装用户				√		√	√	√		√		√

第二个数据维度是检测数据所包含的实体特征，通常包括实体的静态特征和动态特征。静态特征指实体的内容特征。如恶意域名的网络 URL 字符串、网页源码等，恶意文档的内容和结构等信息，恶意邮件的发送和接收邮箱地址、发送和接收域名、源和目的 IP 地址、时区等信息[21]，恶意程序的二进制文件(包括字节序列、操作码序列、文件头部信息、动态链接库信息等信息)。而动态信息主要指实体的行为特征。如恶意进程的系统调用、运行状态[22]、网络访问[23]等信息，伪装用户的主机和应用审计日志[24]等。

第三个数据维度是检测数据在系统中的采集位置，包括网络侧和主机侧。网络侧主要指在网络上采集实体的各种静态和动态信息。如在 DNS 设备上采集恶意 URL，在网络交换机上从网络流量中采集恶意文档和恶意程序，从 web 服务器爬取恶意网页源码，在邮件服务器上采集恶意邮件等静态信息，或在网络安全设备上安装沙箱、虚拟机等模拟运行环境，通过执行恶意程序来采集恶意程序的各种动态信息。主机侧主要指在主机上采集系统实体的各种动态信息。如使用系统审计工具采集恶意进程和伪装用户的系统级行为信息[25] [26]。

实体静态信息对于 APT 的揭示作用有限。首先，在网络中发现攻击部署和工具投递等攻击阶段，不能说明主机中存在 APT。其次，恶意程序静态检测一般很难发现零日攻击等新型恶意程序。再者，伪装用户的发现依赖用户的主机行为信息。因此，接下来我们主要总结实体动态信息数据类型，包括系统行为数据和用户行为数据。

3.1. 系统行为数据

系统行为数据主要用于动态检测网络和主机中的恶意程序及其恶意行为，主要包括的类型有：进程运行过程中的各种调用行为和网络通信行为等。表 2 总结了各类系统行为数据所在的系统层次、针对的攻击阶段和采集方法。

Table 2. Classification of program behavior data
表 2. 系统行为数据分类

数据类型	系统层次		针对攻击阶段					采集方法
	系统内核	网络	初始入侵	命令与控制	提权并获取凭证	横向移动	数据窃取	
系统调用	√		√	√	√	√	√	操作系统采集工具(Linux Audit, ETW, DTrace 和 Sysdig)
网络数据		√	√	√		√		通过网络设备采集。

3.1.1. 系统调用

系统调用数据记录了系统实体(包括进程，文件和 IP 等)之间的交互关系(如读取，写入和创建等)。一条日志记录代表一个系统事件，例如进程P1写入文件F1 (P1→F1)，其中P1代表该事件的源实体(Subject)和 F1 代表该事件的目标实体(Object)。系统调用数据能较好的反映出应用程序或命令在系统内核中的整体运行逻辑，因此系统调用是最常见的程序动态分析数据。由于系统调用能够较全面的反映程序运行时内部状态，因此可用于多个攻击阶段和多种攻击类型的检测。如通过分析系统的内核网络模块调用，发现命令与控制、横向移动等高级持续攻击行为，通过分析系统的动态链接库调用，发现用户层 Rootkit 攻击。目前，系统调用的采集主要使用操作系统采集工具，如面向 Linux 系统的 Linux Audit [25]、DTrace [27]和 Sysdig [28]；面向 Windows 系统的 ETW [26]。

3.1.2. 网络数据

网络数据记录了程序的网络通信行为，包括网络数据包信息和网络流量信息。网络数据包信息包括各种网络协议类型的数据包字段信息。如 TCP/IP 数据包头中的标识符，IP 数据包头中的序列号，ICMP 数据包的数据负荷字段和 4 字节的包头字段，HTTP 数据包的请求字段长度和发送时间等。网络流量信息主要指网络流量的统计信息。如各类网络协议数据包的时间间隔、带宽等。在现代计算机网络中，许多网络设备(如交换机)都支持网络数据包和网络流量数据采集。网络数据可用于分析在命令与控制阶段实施网络隐蔽信道攻击的恶意程序。其中，网络数据包信息主要用于分析基于存储的隐蔽信道攻击，而网络流量信息主要用于分析基于时间的隐蔽信道攻击。

3.2. 用户行为数据

用户行为数据主要用于检测攻击者伪装成正常用户进行的横向移动、数据窃取等攻击行为，主要是用户在主机的日常行为，包括应用程序访问和操作、主机本地操作、远程主机登录和访问等。表 3 总结了这些用户行为数据类型所在的系统层次、针对的攻击阶段和采集方法。

Table 3. Classification of user behavior data
表 3. 用户行为数据分类

数据类型	系统层次		针对攻击阶段			采集方法
	应用程序	操作系统	横向移动	数据窃取	掩盖痕迹	
应用访问和操作	√		√	√	√	应用内部日志模块。
主机本地操作		√		√	√	用户进程系统调用。
远程登录和访问		√	√			用户进程系统调用。

3.2.1. 应用程序访问和操作

用户的应用程序访问和操作数据包括数据库服务访问日志[29]、邮件操作日志[30]、浏览器访问日志[31]、办公自动化软件操作日志等。其中，数据库服务访问日志记录了用户名、数据库查询请求(如 SQL 查询语句等)。邮件操作日志包括邮件主题、发送人和接收人等相关人员的邮箱地址、发送或接受操作、附件尺寸等信息。浏览器访问日志包括网页访问的历史记录、数据上传和下载操作等。这类数据主要用于分析伪装用户的数据窃取和删除等行为，通常需要在应用程序内部实现采集。

3.2.2. 主机本地操作

用户的主机本地操作数据包括主机文件系统访问日志[32]、主机命令行输入日志[33]、主机外设(如键盘、鼠标、U 盘等)操作日志[34]等。其中，文件系统访问日志记录了用户访问的文件路径和创建、读写、删除等操作，主要用于分析伪装用户的数据窃取、掩盖痕迹等行为。命令行输入日志是一组用户命令序列，记录了命令名和相关参数，主要用于基于行为异常识别伪装用户。外设操作日志主要包括用户的键盘输入记录、鼠标轨迹、U 盘的插拔和数据拷贝等操作，主要用于识别伪装用户及其数据窃取行为。这类数据的采集主要通过操作系统记录用户进程的系统调用来实现。

3.2.3. 远程主机登录和访问

用户的远程主机登录和访问数据主要指用户使用远程登录客户端(如远程桌面、SSH 客户端等)对远程主机的访问日志，包括本地主机名和 IP 地址、远程主机名和 IP 地址、登录用户名、认证协议、登录时间等，主要用于分析伪装用户的横向移动活动。这类数据的采集主要通过操作系统记录远程登录进程的系统调用来实现。

4. 基于主机的 APT 检测技术

本节主要介绍和总结已有的基于系统行为和用户行为的 APT 检测技术工作。为了保证工作的前瞻性和高质量，我们主要选取了近 5 年发表在 CCF-A/B [35]中的正式会议或期刊(即，不涉及 workshop 和 short page)上的文章。此外，为了使脉络清晰，我们也介绍了一些早期高引用的经典文章。

4.1. 基于系统行为的检测技术

目前已有的基于系统行为的 APT 检测技术主要分析系统调用日志，系统调用日志记录程序或命令在系统内核层面的行为活动，包括进程与文件，进程与 IP 和进程与进程之间的交互行为。一些工作基于已知的攻击事件追踪引发该攻击事件的源头，并构建攻击路径。例如，最早的工作 King 等[36]采用因果关系分析方法(causality analysis)将系统调用日志转换为系统依赖关系图。具体为，给定两个日志事件，如果一个事件的目标实体是另一个事件的源实体，则这两个事件存在因果依赖关系。系统依赖图中的节点代表系统实体(包括进程，文件和 IP)；边是有向边，代表两个实体之间的交互事件(如读取，写入和创建等)。之后，以一个被揭露的攻击事件为起始点，沿着节点的入射边迭代的搜索与该攻击事件存在依赖关系并且发生在该攻击事件之前的所有事件，搜索到的所有事件构成的依赖关系图为该攻击事件的溯源依赖图。如图 2(a)所示，该图为攻击事件 process B→file X 的溯源依赖图。接下来提出了 5 个规则，用于滤除非重要的节点和边，进一步减少溯源图的规模。处理后的溯源图包含了主要的 APT 攻击行为。在此基础上，为了寻找由攻击事件引发的前向攻击路径，King 等[37]提出了前向追踪算法：沿着节点的出射边迭代的搜索与该攻击事件存在依赖关系并且发生在该攻击事件之后的所有事件。此外，该工作实现了网络端的发送和接收行为追踪，以完成跨主机之间的攻击检测。为了进一步追踪异常的动态链接库，Wang 等[38]设计一种动态链接库跟踪技术，将链接库和对应的栈一起标识出来，从而可以准确地显示库的执行来源，

然后将其整合到溯源图，实现 APT 的追踪检测。目前，基于系统调用日志的检测技术都是基于以上工作展开。

然而，长时运行程序(如 firefox)的存在，使得活动进程接收大量输入并产生许多输出，而每个输出又可能与之前的所有输入有因果关系，导致依赖爆炸，使得攻击检测与推断几乎不可行。如图 2(b)所示，进程 A 存在大量输入 IP 节点，使得难以确定哪一个 IP 是攻击的一部分。为了解决这个问题，一些工作提出实体节点分割技术。例如 Sitaraman 等[39]额外记录文件偏移量，用以分割运行在不同文件段上的进程，只有在同一文件段上运行的进程才被认为是因果相关的。Goel 等[40]将一个进程分割为由两个连续的网络端口读取所限定的区段，以减少错误的依赖。属于不同进程区段的实体不被认为存在因果关系。考虑到上述两个方案在一些场景下的局限性，Lee 等[41]通过监控一个程序的事件处理循环，将长时运行进程划分为多个独立的单元，每一次迭代对应一个独立的输入/请求处理。该循环模式是从应用程序运行内存中提取。由于 Lee 等[41]的方法为了获取内存信息，需要对系统内核进行改写，具有较差的适用性，为此 Ma 等[42]基于特定应用的高级任务结构，提出了一种语义感知的程序标注和插装技术来划分进程。该方法允许开发人员/用户通过注释少量的数据结构来标记系统执行的任务/单元结构，然后通过静态程序分析获取注释和程序位置，这些位置代表了进程单元边界，最后根据边界对进程进行划分。该方法无需对系统内核进行改写，适用性较好。由于 Ma 等[42]需要对应用程序源代码进行改写，因此实用性依然不高，为此 Yang 等[43]利用容易获取的 UI 元素/事件对进程进行分割。具体为，首先对 UI 元素/事件(代表用户视角)和底层系统事件同时进行因果分析，然后基于时间戳和资源属性，将系统事件与 UI 事件关联起来，最后基于被关联的 UI 事件将长时运行进程进行划分，每一个进程区段代表一个 UI 事件。对于同样的问题，Hassan 等[44]则利用应用程序事件日志对进程进行划分。具体为，将应用程序事件日志与系统调用日志进行整合，利用应用程序日志恢复执行路径，可用于分区进程，避免依赖爆炸。特别地，一个应用程序日志实体序列可以用来恢复一个近似的程序路径，这种程序路径的重复出现表明一个程序正在处理一个独立的任务。然后将每个被识别的程序路径映射到相应的系统审计日志实体，然后对日志实体进行分割。Yu 等[45]同样提出了应用程序日志和系统调用日志相结合的解决方案。不同之处在于，该方案首先将应用程序日志和系统审计日志统一规范化为标准形式。这种标准化形式能够表达出所有类型的行为模式，特别是复杂的异步行为模式。然后，将标准化的日志实体被加载到 Datalog [46]中，根据预定义的规则推断新的关系。通过推断出的关系，可以很容易地构造出更加细粒度的依赖图。

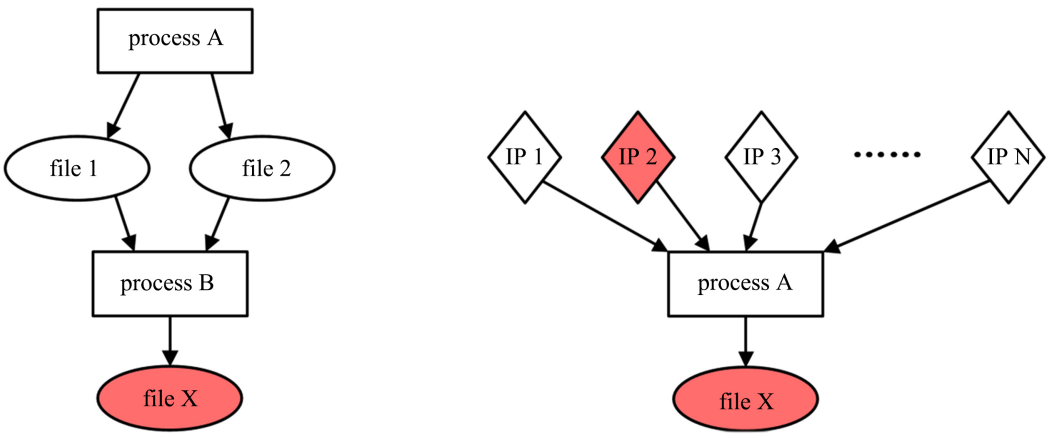

Figure 2. Provenance dependence graph. (a) Example, (b) Dependence explosion
图 2. 溯源依赖图。(a) 例子，(b) 依赖爆炸

为了解决依赖爆炸问题，一些工作采用信息流追踪技术自动搜索相关的攻击路径和滤除不相关的路径。例如 Ji 等[47]提出了一种可细化的攻击调查系统，该系统有选择地提供更细粒度的指令集日志，同时保持可接受的运行开销。该系统不断地监视和记录系统调用事件和指令集数据，以便在构建溯源图时进行回放。当在图中检测到任何异常事件时，它将对该事件执行基于回放的动态信息流追踪，以删除任何不需要的依赖。为了进一步解决跨主机之间的依赖爆炸问题，Ji 等[48]通过标签覆盖和标签切换技术将标签依赖(即主机之间的信息流)从分析中解耦出来，并使动态信息流追踪器独立于由通信所强加的任何顺序。该方法允许对多个主机上的多个进程并行动态信息流追踪，从而实现更高效的分析。为了更快速锁定攻击相关的路径，Liu 等[49]提出了一种向后和向前的因果跟踪器 PrioTracker，它在追踪过程中优先考虑异常的依赖。为了区分异常的系统事件，PrioTracker 建立了一个记录企业计算机系统中日常活动的参考模型，以此来量化每个事件的稀缺性。之后，根据事件在溯源图中的稀缺性和拓扑特征计算每个事件的优先级得分，最后搜索出高优先分数的信息路径。基于异常检测的路径搜索，通常会带来很高的误报，因为很多罕见的良性事件通常表现为异常特征，为此 Hossain 等[50]为了更准确的移除良性依赖，提出两个关键的概念 tag attenuation 和 tag decay。tag attenuation 用于刻画对象实体在接收恶意依赖输入后实体的 tag 衰减过程，tag decay 用于刻画良性实体间依赖在接收恶意实体输入后依赖的 tag 衰减过程。最后根据 tag 传播规则，移除衰减较弱的依赖。对于同样的问题，Fang 等[51]认为与攻击事件相关的依赖通常与不相关的依赖表现出不同的属性值(如数据流和时间)。基于此，他们提出了 DEPIMPACT 框架，该框架通过为边分配有区别的依赖权值，将代表攻击序列的边与非重要的依赖边区分开来，然后将依赖影响信息值从被揭露的攻击事件反向传播到入口节点，最后根据依赖影响信息值从排名靠前的入口节点执行前向因果分析，过滤掉没有发现的边。

此外，一些工作在 APT 检测过程中不依赖任何已知的攻击事件。例如，Hossain 等[52]提出一种基于系统依赖图标签传播的 APT 检测方法。首先，基于程序系统调用日志形成主机系统依赖关系图，并进一步根据实体关系、实体行为信息等先验知识对图结点设置标签，以及标签的信任级别和密级。然后利用设定的传播规则识别异常事件。最后采用后向和前向分析分别追踪该异常事件的攻击源头和分析攻击后果。Hassan 等[53]认为系统依赖图中每个事件的可疑性都与图中相邻事件有关。基于此，他们首先构建一个事件频率数据库，该数据库存储企业中以前发生的所有事件的频率。然后，基于该数据库为依赖图中每一个边设置异常分数。分配异常分数之后，采用网络传播算法，聚合告警事件相邻事件的异常分数，聚合后的异常分数作为该告警事件的最终异常分数。最后根据该异常分数确定告警事件是否为真正的攻击事件。考虑到已有的 APT 检测技术只是检测攻击事件或路径，无法映射到具体的攻击阶段(如初始入侵阶段，命令与控制阶段等)，Milajerdi 等[54]将 APT 生命周期特点作为重要参考建立了实时的 APT 检测系统 HOLMS。HOLMS 构建了一个高级场景图(HSG)，其中集成了 MITRE ATT&CK 的 TTP (Tactics, Techniques, and Procedures)，TTP 是描述 ATP 步骤的重要行为标准。最后，HOLMS 通过 HSG 将系统调用级日志事件实时的映射到对应的 APT 阶段。对于同样的目标，Milajerdia 等[55]将 APT 检测问题形式化为图模式匹配问题，目标是查询或搜索与查询图(query graph)中描述相似的系统实体依赖关系或信息流。这里的查询图描述了 IOC 及它们(Indicators of Compromise)之间的关系。IOC 及其之间的关系是从与已知攻击相关的 CTI 报告中提取，因此代表了 APT 完整的攻击阶段。在此基础上，为了更加精确的提取和量化 IOC 之间的关系，Zhao 等[56]采用图卷积神经网络学习 IOC 之间潜在的关系模式，并将该关系嵌入到低维向量空间，通过与该向量空间的相似性比较来检测 APT 攻击。此外，考虑到 CTI 报告具有非结构化特点，安全人员很难从文本中提取有效信息，Satvat 等[57]提出一个自动化提取工具 Extrator，利用自然语言处理算法从 CTI 报告中精确提取攻击行为，之后使用语言角色标注进行语义分析，理解攻击行为关系，最后将非结构化的文本转换为依赖关系图，用于 APT 攻击检测。考虑到现有的 APT 检测技术无法

检测未知攻击，Han 等[58]将 APT 检测问题看作是实时的依赖图异常检测问题。系统实时捕获的依赖图将与学习到的正常行为依赖图的演变模式进行比较，如果存在明显差别，则被认为存在攻击。其中每一时刻的依赖图具有固定大小和增量更新等特性可以有效减少内存的计算开销。Alsaheel 等[59]发现因果依赖图中不同攻击的关键阶段可能存在相似的模式，这些模式通过 NLP 技术转换为序列模型。在检测过程中，给定一个新的实体序列，将其与经过训练的序列模型进行比较，确定该序列是否为攻击。在 APT 生命周期中，横向移动是获取机密数据的关键阶段，Fang 等[60]提出了一个横向移动攻击检测系统 LMTacker。LMTacker 融合了系统审计日志和系统网络流量日志，构建了含有主机和用户实体节点的系统依赖图，然后采用随机行走算法提取路径，并学习路径向量，最后使用自编码网络实现基于异常的路径检测。

一些工作专注于减少 APT 检测所带来的运行时间和存储开销。例如 Ma 等[61]在 Lee 等[41]工作的基础上减少检测系统的运行时间和存储开销。具体为，对于在永久存储写文件或向外界发送数据包等事件，在溯源图中使用系统审计日志进行关联，而对于诸如读文件和接收网络数据包等事件，则采用 Lee 等[41]的细粒度实体进行事件关联。该方法的时间开销减少了 98.72%和空间开销减少了 93%。Xu 等[62]通过合并系统实体之间重复的依赖关系，以达到节省系统依赖图的存储空间，同时保证依赖分析不被影响。但是，他们的依赖压缩是基于节点的邻居边信息，在一些场景中效果有限。为此，Hossain 等[63]利用全局图信息(如可达性)对两节点间重复的依赖关系进行合并。具体为，如果两个实体之间的前向可达路径和反向可达路径相同，则对路径中的重复边进行合并。压缩效果相比于 Xu 等[62]进一步减少了 5 倍。不同于以上两个工作，Tang 等[64]发现系统链接库，资源和配置文件的读取或载入通常用于进程的初始化，因此它们并不包含有价值的依赖信息。此外由于它们是只读的，在系统依赖图中是末端节点，因此将其移除并不会破坏依赖分析。基于此，他们将具有相同访问模式的只读文件进行合并，以此减少存储开销。Fei 等[65]在以上三个工作的基础上，同时考虑实体节点与边的压缩。此外，他们考虑与 write 和 Execute 事件相关的实体节点的压缩。以上工作都是面向单主机的解决方案，Hassan 等[66]提出面向分布式系统的解决方案 Winnower。Winnower 背后的观点是：在分布式系统中，不同的主机节点通常会执行相同的应用程序，因此存在相同的依赖图结构，它们被看作是等价的(即高度冗余)。基于此，Winnower 采用 Deterministic Finite Automata (DFA)模型学习和识别这种重复结构，之后将其中一个图结构上传到中心主机，以此减少中心主机的存储开销。以上工作的研究目标是较少系统依赖图规模，但是在构图之前依然需要收集和存储大量原始的冗余事件，造成了不必要的时间和存储开销。Ma 等[67]专注去解决这个问题。他们受到硬件/软件缓存系统设计的启发，提出了一个内核级的、低开销的、基于内存缓存的审计日志系统(KCAL)。KCAL 缓存重要的依赖和系统事件，并实时的检测冗余性，一旦检测到事件是冗余的，将立刻被丢弃，只保留新引入的事件或依赖。

一些工作提出用于 APT 检测的辅助系统或工具，使检测变的更加容易和可行。例如，Bates 等[68]为 Linux 操作系统设计可实际部署的 APT 溯源模块(LPM)。LPM 能够实时监控进程行为活动、IPC、网络活动和内核状态，为 APT 溯源提供良好环境。LPM 也提供了主机之间的认证和数据通道，实现了分布式的溯源分析。此外，LPM 能够与系统中其它安全机制相互协同，为系统提供更强的安全保障。LPM 的实施仅增加了 2.7%的性能开销。Kwon 等[69]提出了因果推断模型(MCI)。MCI 模型是通过分析 LDX [70]的运行结果构建，其中 LDX 是一个基于双执行的因果关系分析系统，LDX 通过改变输入端的系统调用事件，然后观察输出的变化来推断因果关系。构建的 MCI 模型能够表达细粒度的因果依赖关系，包括基于内存推断的隐式依赖。在攻击推断阶段，MCI 模型直接作用于系统审计日志，识别事件之间的因果依赖关系，省去了 Lee 等[41]和 Ma 等[42]提出的复杂的隐式依赖推断过程，降低了分析难度。Gao 等[71]设计了一套用于 APT 攻击调查的查询语言(AIQL)，去帮助安全分析人员从系统审计日志中搜索攻击行为。AIQL 涵盖了 APT 的基本特征，包括多阶段性、因果依赖和异常性，从而支持相似性查询、时间关系查

询、因果依赖关系查询和频率查询。AIQL 已经部署到了已有的数据库，如 PostgreSQL 和 Greenplum 等。考虑到用于攻击调查的系统溯源图规模的庞大和结构的复杂性，其中的行为模式很难被安全分析人员理解，进而难以判断出其中的攻击行为，为此 Xu 等[72]提出了面向 APT 攻击调查的图概要技术(DEPCOMM)。DEPCOMM 为系统溯源图生成一个结构更加简洁和易理解的超图(即概要图)，用于描述溯源图中的行为结构。概要图的生成过程包括三个阶段：1) 根据定义的行为阶段划分社区子图；2) 压缩社区子图中的重复行为模式；3) 为社区子图提供行为摘要。通过分析概要图，安全分析人员能够快速找出攻击行为。

4.2. 基于用户行为的检测技术

相对于基于系统行为的检测技术，基于用户行为检测技术的研究工作相对较少，特别是近 5 年高质量的工作更为稀少。主要是由于用户行为事件丰富多样，事件之间的关系复杂多变，很难被刻画，这直接导致检测精度不高。但是，基于用户行为检测技术的优势在于：具有丰富的语义，能使安全分析人员容易理解攻击的本质。不管怎样，我们接下来将仅有的一些基于用户行为检测技术的高质量研究工作进行分析和总结。

一些工作将用户行为事件提取成序列形式，进而使用序列处理工具对事件进行分析。例如，Holgado 等[73]采用隐马尔科夫模型分析攻击行为序列，当新的事件序列输入到已经训练好的模型后，该模型输出序列中的事件是否为攻击。同样，Shen 等[74]采用深度学习模型 LSTM 对攻击事件序列进行建模，该模型能够预测新序列中的事件是否为攻击。到目前，已有的工作只是判断日志事件是否为攻击，无法推断事件被恶意利用的过程，为此 Shen 等[75]采用词嵌入技术将事件连同它的上下文编码为低维的向量空间，之后通过分析向量的变化过程，去识别事件的演变趋势和推断事件的利用过程。

而另一些工作将用户行为数据转化为图结构，然后采用图分析方法对行为进行分析。Siadati 等[76]利用主机登录和访问数据检测 APT 攻击中的横向移动攻击。该方法基于两个观点：1) 一个企业正常用户的登录行为是结构化和可预测的；2) 恶意的登录模式与正常登录模式通常是不一致的。基于此，他们为正常登录建立了网络登录结构，该结构描述了公司部门中主机之间的正常登录行为，然后使用异常检测方法来检测与企业网络登录结构不一致的恶意登录。Bohara 等[77]利用主机登录和访问数据检测 APT 攻击中的命令与控制和横向移动阶段。具体为，首先将登录和访问数据转换为有向图，其中节点代表访问的主机节点和边代表通信连接。然后，从图中提取与命令与控制和横向移动行为有关的特征。之后，对特征进行标准化处理，并生成节点表达。最后，采用基于机器学习的异常检测方法识别异常主机。Liu 等[78]提出基于图嵌入的非监督、细粒度 APT 检测系统 Log2vec。首先，Log2vec 根据用户行为日志的各种关系构建用户行为异构图。其次，采用图嵌入技术将异构图中行为节点表达成向量。最后，采用聚类算法将表达成向量的行为节点进行聚类，设置阈值将类成员数量低于阈值的类视为异常类，据此确定异常行为。此外，Liu 等[79]针对攻击者的横向移动攻击，使用图神经网络技术分析网络中用户主机登录行为。首先，设计的背景信息提取器可根据具体检测场景筛选出特定的登录属性作为用户主机登录的背景信息。最后，采用图神经网络对发起登录主机、被登录主机和提取的登录背景信息进行分析，检测用户异常登录行为并且确定可疑主机。

5. 评价方法

本节主要介绍可用于评价 APT 检测有效性的常用数据集和评价指标。

5.1. 数据集

APT 检测常用数据集可大致分为三大类：系统调用数据、网络数据、用户主机行为数据(主要为应用访问、网络访问、系统命令、外设操作等)。本文将其概括为表 4。

Table 4. Datasets for evaluation
表 4. 评价数据集

数据集	数据类型		
	系统调用	网络数据	用户主机行为
ADFA [80]	√		
NGIDS-DS [81]		√	
DARPA 98 [82]	√	√	
KDD CUP 99 [83]		√	
SEA [33]			√
WUIL [84]			√
CERT [31]			√
TWOS [34]		√	√
RUU [32]	√		√
LANL [85]	√	√	
Garg [86]	√		√
DARPA-TC [87]	√		

ADFA 数据集[80]是澳大利亚国防学院发布的一套主机级入侵检测数据集，主要包含 3 组从网页浏览到 Latex 文档编辑等正常活动的系统调用数据，以及被攻击的 Ubuntu 操作系统系统调用数据。NGIDS-DS 数据集[81]包含了从 2016 年 3 月 11 日至 2016 年 3 月 16 日在不同企业(电子商务、军事、学术界、社交媒体和银行)网络中采集的网络数据。Haider 等在收集数据过程中设置了 4 个准则来保证收集到的数据可靠性，包含企业网络中发生在关键网络设施上的正常网络活动，以及漏洞攻击、DoS、蠕虫、木马、侦察、恶意代码、后门等异常流量数据。DARPA 98 数据集[82]在 1998 年由美国国防部先进研究项目局公布，用于对入侵检测系统进行离线评估和实时评估，主要包括网络流量和系统审计日志，其中 7 周数据为训练集，而其它 2 周数据为测试集。KDD CUP 99 数据集[83]是由美国空军在一个模拟局域网上采集的 9 周网络连接数据。该数据集包括正常网络连接、拒绝服务攻击、网络监视和探测活动、远程机器非法访问等数据。其中，异常网络连接被细分为 39 种攻击类型，其中 22 种攻击在训练集中出现，而另外 17 种攻击仅出现在测试集中。SEA 数据集[33]是由 Schonlau 等于 2001 年首次公开。该数据级首次将内部攻击者分为叛徒和伪装者，记录了 70 个 UNIX 用户命令操作日志，每个用户有 15,000 条命令操作记录。其中随机挑选 50 位用户作为正常用户，并通过在剩余用户的数据中随机插入命令操作来模拟攻击。每个用户的日志数据以 100 条命令为间隔划分为 150 个块，前三分之一数据块用于代表用户正常行为，而后三分之二数据中随机插入攻击数据。WUIL 数据集[84]是在 2014 年由 Camiña 等公布。该数据集记录了 20 个背景各异的 Windows 用户在不同时间段的文件访问记录，由支持 Windows XP、7、8 和 8.1 等多个操作系统版本的专用工具收集。模拟攻击分为基础、中级和高级三个级别。CERT 数据集[31]由卡耐基梅隆大学应急响应中心在真实场景中采集，包含用户的主机登录和注销、可移动驱动器文件拷贝、电子邮件、网站访问、可移动驱动器插拔等。TWOS 数据集在文献[34]中被用到，该数据集模拟了公司内部的用户活动，包含键盘鼠标操作、程序和文件操作、网络流量、电子邮件、主机登录和注销等数据。内部

攻击者包括伪装者和叛徒。RUU 数据集在文献[32]中被提及，包括 34 个普通用户和 14 个伪装用户的 22 类主机日志数据。该数据集由作者开发的 Windows 主机感知模块采集，主要记录注册表活动、进程创建和销毁、窗口 GUI、文件访问、DLL 库活动等。LANL 数据集[85]是美国洛斯阿拉莫斯国家实验室在其内部计算机网络中连续 58 天收集的网络和系统数据，主要包含 Windows 主机认证日志、进程启动和停止日志、DNS 查找表、网络流量、红队攻击数据等，共涉及 12,425 个用户、17,684 台计算机。Garg 等的数据集[86]包括 3 名用户的键盘活动、鼠标移动和点击、进程运行、用户命令等数据，并通过在一个用户数据中注入其它用户数据来模拟伪装攻击。DARPA-TC 数据集[87]是美国国防部先进研究项目局于 2018 年公布，主要包括从 Linux、Windows 和 BSD 系统在两周内采集的系统调用数据。

5.2. 评价指标

可采用的检测性能指标包括准确率、召回率、误报率(FPR)、精度、F-分数(F-Score)、AUC 等。其中，准确率指正确预测的样本占总样本的比例，召回率指正确预测的正样本占总样本的比例，误报率表示被误分到正样本中的负样本占所有负样本的比例，精度表示正确预测的正样本占所有预测为正样本的比例，F-分数是准确率和召回率的调和值，AUC 是横坐标为 FPR，纵坐标为 TPR 所画出的 ROC 曲线下的面积，其值越大，分类效果越好。以上指标的计算方式如表 5，其中，TP 表示将正样本预测为正类，TN 表示将负样本预测为负类，FP 表示将负样本预测为正类，FN 表示将正样本预测为负类。

Table 5. Calculation formula of evaluation index
表 5. 评价指标计算公式

指标	计算公式
准确率	$(TP+TN)/(TP+TN+FP+FN)$
召回率	$TP/(TP+FN)$
误报率	$FP/(TN+FP)$
精度	$TP/(TP+FP)$
F-分数	$(1+\beta^2)*P*R/\beta^2(P+R)$

6. 总结与展望

本文系统概括分析了 APT 的生命周期、主机实体及其行为数据、基于主机 APT 检测技术和评价方法。现有的 APT 检测技术，虽然很好的解决了许多问题，但是由于实际环境的不确定性和网络空间攻防技术演进的此消彼长，仍面临许多挑战。我们从以下三个方面分析存在的挑战，并指出未来的研究方向。

1) 多日志融合。APT 攻击是一种多阶段攻击，其中攻击阶段中的具体攻击手段可能被不同的日志监控系统捕获，如应用程序的运行过程会被系统调用日志捕获，主机间通信信息会被网络流量日志捕获，U 盘的插拔和数据拷贝信息会被记录到外设操作日志中。因此，只对一种日志进行分析无法发现完整的 APT 过程，必然需要结合多种日志进行分析。一些工作[60] [88]已经实现了多种日志的融合，但是他们只考虑系统层面的日志(如，系统调用日志，网络流量日志和 DNS 日志等)。因此，实现多层面的日志(包括系统行为日志和用户行为日志)的融合，是未来重要的研究方向。

2) 日志数据完整性。系统日志是威胁检测的重要基础，因此日志信息的完整性也成为威胁检测有效性的重要保障。从攻击者视角，向目标系统发起的攻击活动极大可能被系统监测和采集，因此利用获得

的系统权限(通过攻击获得或者攻击者本身就是系统内部管理员等内部人员)，篡改或删除攻击活动相关的日志数据，使得攻击行为不能被检测系统所发现，成为攻击者必然采用的隐蔽手段。而现有的威胁检测工作通常假设日志数据可被有效保护或不被破坏，因此较少考虑日志数据完整性问题。由于理想的日志完整性防护方案在实际中并不可行，因此研究人员需要在未来进一步探索如何利用其它信息将缺失的部分恢复，或是在日志信息缺失的情况下如何保证检测模型不被影响。

3) 检测模型鲁棒性。近期，神经网络模型的成功使得它们被越来越广泛的应用于威胁检测领域。然而，神经网络模型被证明存在可被对抗学习利用的脆弱性。对抗学习利用神经网络反向传播原理，在原始样本上找到可导致模型分类错误的不易觉察微小改变，并产生这样的对抗样本。该攻击也被称为对抗攻击。此外，还有研究表明神经网络存在被投毒攻击的风险[89]，即攻击者通过向模型的训练数据中注入恶意训练样本，刻意影响模型训练并导致模型在测试数据上的分类精度。目前很多 APT 检测技术采用神经网络模型[56] [59] [74] [79]，因此分析这些模型的脆弱性，和提出具备能够不受对抗攻击和投毒攻击等规避攻击影响的高健壮检测模型是未来需要重点关注的。

基金项目

中国科学院战略重点研究计划(XDC02010300)资助。

参考文献

[1] Chen, T.M. and Abu-Nimeh, S. (2011) Lessons from Stuxnet. *Computer*, **44**, 91-93. https://doi.org/10.1109/MC.2011.115

[2] Lelli, A. (2018) The Trojan. Hydraq Incident: Analysis of the Aurora 0-Day Exploit. http://www.symantec.com/connect/blogs/trojanhydraq-inc

[3] Arquilla, J. and Guzdial, M. (2021) The SolarWinds Hack, and a Grand Challenge for CS Education. *Communications of the ACM*, **64**, 6-7. https://doi.org/10.1145/3449047

[4] Chen, P., Desmet, L. and Huygens, C. (2014) A Study on Advanced Persistent Threats. In: De Decker, B. and Zúquete, A., Eds., *Communications and Multimedia Security*. *CMS* 2014. *Lecture Notes in Computer Science*, Vol. 8735, Springer, Berlin, 63-72. https://doi.org/10.1007/978-3-662-44885-4_5

[5] Zhu, Z. and Dumitras, T. (2018) ChainSmith: Automatically Learning the Semantics of Malicious Campaigns by Mining Threat Intelligence Reports. 2018 *IEEE European Symposium on Security and Privacy* (*EuroSP*), London, 24-26 April 2018, 458-472. https://doi.org/10.1109/EuroSP.2018.00039

[6] Khonji, M., Iraqi, Y. and Jones, A. (2013) Phishing Detection: A Literature Survey. *IEEE Communications Surveys & Tutorials*, **15**, 2091-2121. https://doi.org/10.1109/SURV.2013.032213.00009

[7] Kaspersky (2021) Malware Reports: IT Threat Evolution Q3. https://securelist.com/it-threat-evolution-in-q3-2021-pc-statistics/104982/

[8] Christiaan, B., Douglas, F., Paula, G., *et al.* (2015) Mcafee Labs Threats Report. Technical Report, McAfee.

[9] Symantec Internet Security Threat Report (2019). https://www.phishingbox.com/downloads/Symantec-Security-Internet-Threat-Report-ISRT-2019.pdf

[10] Data Breach Investigations Report (2021). https://www.verizon.com/business/resources/reports/dbir/

[11] Kaspersky (2009). https://usa.kaspersky.com/resource-center/threats/zeus-virus

[12] Kocak, T. and Kaya, I. (2006) Low-Power Bloom Filter Architecture for Deep Packet Inspection. *IEEE Communications Letters*, **10**, 210-212. https://doi.org/10.1109/LCOMM.2006.1603387

[13] Borders, K. and Prakash, A. (2004) Web Tap: Detecting Covert Web Traffic. *Proceedings of the* 11*th ACM conference on Computer and Communications Security*, Washington DC, 25-29 October 2004, 110-120. https://doi.org/10.1145/1030083.1030100

[14] Eberle, W., Graves, J. and Holder, L. (2010) Insider Threat Detection Using a Graph-Based Approach. *Journal of Applied Security Research*, **6**, 32-81. https://doi.org/10.1080/19361610.2011.529413

[15] Kammueller, F., Kerber, M. and Probst, C.W. (2016) Towards Formal Analysis of Insider Threats for Auctions. *MIST*'16: *Proceedings of the* 8*th ACM CCS International Workshop on Managing Insider Security Threats*, Vienna, 28

October 2016, 23-34. https://doi.org/10.1145/2995959.2995963

[16] Myers, J., Grimaila, M.R. and Mills, R.F. (2009) Towards Insider Threat Detection Using Web Server Logs. *CSIIRW'*09: *Proceedings of the* 5*th Annual Workshop on Cyber Security and Information Intelligence Research*: *Cyber Security and Information Intelligence Challenges and Strategies*, **54**, 1-4. https://doi.org/10.1145/1558607.1558670

[17] Mathew, S., Petropoulos, M., Ngo, H.Q., *et al.* (2010) A Data-Centric Approach to Insider Attack Detection in Database Systems. In: Jha, S., Sommer, R. and Kreibich, C., Eds., *Recent Advances in Intrusion Detection*. *RAID* 2010. *Lecture Notes in Computer Science*, Vol. 6307, Springer, Berlin, Heidelberg, 382-401. https://doi.org/10.1007/978-3-642-15512-3_20

[18] Ben, J.W. and Kheir, N. (2016) A Grey-Box Approach for Detecting Malicious User Interactions in Web Applications. *MIST'*16: *Proceedings of the* 8*th ACM CCS International Workshop on Managing Insider Security Threats*, Vienna, 28 October 2016, 1-12.

[19] 王琴琴, 周昊, 严寒冰, 等. 基于恶意代码传播日志的网络安全态势分析[J]. 信息安全学报, 2019, 4(5): 14-24.

[20] Jing, X.Y., Zheng, Y. and Witold, P. (2018) Security Data Collection and Data Analytics in the Internet: A Survey. *IEEE Communications Surveys & Tutorials*, **21**, 586-618. https://doi.org/10.1109/COMST.2018.2863942

[21] Wardman, B., Warner, G., Mccalley, H., *et al.* (2010) Reeling in Big Phish with a Deep MD5 Net. *Journal of Digital Forensics*, *Security and Law*, **5**, 33-56. https://doi.org/10.15394/jdfsl.2010.1079

[22] 汪嘉来, 张超, 戚旭衍, 等. Windows 平台恶意软件智能检测综述[J]. 计算机研究与发展, 2021, 58(5): 977-994.

[23] 潘亚峰, 周天阳, 朱俊虎, 等. 基于 ATT&CK 的 APT 攻击语义规则构建[J]. 信息安全学报, 2021, 6(3): 77-90.

[24] Mayhew, M., Atighetchi, M., Adler, A., *et al.* (2015) Use of Machine Learning in Big Data Analytics for Insider Threat Detection. *MILCOM IEEE Military Communications Conference*, Tampa, FL, 26-28 October 2015, 915-922. https://doi.org/10.1109/MILCOM.2015.7357562

[25] S. Grubb. Redhat Linux Audit (2020). https://people.redhat.com/sgrubb/audit/

[26] Event Tracing for Windows (ETW) (2020). https://docs.microsoft.com/en-us/windows/win32/etw/

[27] Dtrace (2017). http://dtrace.org/blogs/

[28] Sysdig (2017). https://sysdig.com/

[29] Kamra, A., Terzi, E. and Bertino, E. (2008) Detecting Anomalous Access Patterns in Relational Databases. *The VLDB Journal*, **17**, 1063-1077. https://doi.org/10.1007/s00778-007-0051-4

[30] CALO Project (2015) Enron Email Dataset. https://www.cs.cmu.edu/~./enron/

[31] CERT Dataset (2016). http://www.cert.org/insider-threat/tools/index

[32] Ben. S. M. RUU Dataset. http://www1.cs.columbia.edu/ids/RUU/data/

[33] Masquerading User Data (1998). http://www.schonlau.net/intrusion.html

[34] Harilal, A., Toffalini, F., Castellanos, J., *et al.* (2017) Twos: A Dataset of Malicious Insider Threat Behavior Based on a Gamified Competition. *Proceedings of the* 2017 *International Workshop on Managing Insider Security Threats*, Dallas, TX, 30 October 2017, 45-56. https://doi.org/10.1145/3139923.3139929

[35] 中国计算机学会推荐国际学术会议和期刊目录[Z]. https://www.ccf.org.cn/ccf/contentcore/resource/download?ID=144845, 2019.

[36] King, S.T. and Chen, P.M. (2003) Backtracking Intrusions. *Proceedings of the ACM Symposium on Operating Systems Principles* (*SOSP*), Bolton Landing, NY, 19-22 October 2003, 223-236. https://doi.org/10.1145/945445.945467

[37] King, S.T., Mao, Z.M., Lucchetti, D.G., *et al.* (2005) Enriching Intrusion ALERTS through Multi-Host Causality. *Proceedings of the Annual Network and Distributed System Security Symposium* (*NDSS*), San Diego, CA, 20 January 2005, 1-12.

[38] Wang, F., Kwon, Y., Ma, S.Q., *et al.* (2018) Lprov: Practical Library-Aware Provenance Tracing. *Proceedings of the* 34*th Annual Computer Security Applications Conference* (*ACSAC*), San Juan, PR, 3-7 December 2018, 605-617. https://doi.org/10.1145/3274694.3274751

[39] Sitaraman, S. and Venkatesan, S. (2005) Forensic Analysis of File System Intrusions Using Improved Backtracking. *Third IEEE International Workshop on Information Assurance*, College Park, MD, 23-24 March 2005, 154-163. https://doi.org/10.1109/IWIA.2005.9

[40] Goel, A., Po, K., Farhadi, K., *et al.* (2005) The Taser Intrusion Recovery System. *ACM SIGOPS Operating Systems Review*, **39**, 163-176. https://doi.org/10.1145/1095810.1095826

[41] Lee, K.H., Zhang, X.Y. and Xu, D.Y. (2013) High Accuracy Attack Provenance via Binary-Based Execution Partition. *Proceedings of the Annual Network and Distributed System Security Symposium* (*NDSS*), San Diego, CA, 24-27 February 2013, 1-16.

[42] Ma, S.Q., Zhai, J., Wang, F., *et al.* (2017) MPI: Multiple Perspective Attack Investigation with Semantic Aware Execution Partitioning. 26*th USENIX Security Symposium* (*USENIX Security*), 1111-1128.

[43] Yang, R.Q., Ma, S.Q., Xu, H.T., *et al.* (2020) UISCOPE: Accurate, Instrumentation-Free, and Visible Attack Investigation for GUI Applications. *Proceedings of the Annual Network and Distributed System Security Symposium* (*NDSS*), San Diego, CA, 23-26 February 2020, 1-18. https://doi.org/10.14722/ndss.2020.24329

[44] Hassan, W.U., Noureddine, M.A., Datta, P., *et al.* (2020) OmegaLog: High-Fidelity Attack Investigation via Transparent Multi-Layer Log Analysis. *Proceedings of the Annual Network and Distributed System Security Symposium* (*NDSS*), San Diego, CA, 23-26 February 2020, 1-16. https://doi.org/10.14722/ndss.2020.24270

[45] Yu, L., Ma, S.Q., Zhang, Z., *et al.* (2021) ALchemist: Fusing Application and Audit Logs for Precise Attack Provenance without Instrumentation. *Proceedings of the Annual Network and Distributed System Security Symposium* (*NDSS*), Virtual, 21-25 February 2021, 1-18. https://doi.org/10.14722/ndss.2021.24445

[46] Jordan, H., Scholz, B. and Subotic, P. (2016) Souffle: On Synthesis of Program Analyzers. In: Chaudhuri, S. and Farzan, A., Eds., *Computer Aided Verification*. *CAV* 2016. *Lecture Notes in Computer Science*, Vol. 9780, Springer, Cham, 422-430. https://doi.org/10.1007/978-3-319-41540-6_23

[47] Ji, Y., Lee, S., Downing, E., *et al.* (2017) RAIN: Refinable Attack Investigation with On-Demand Inter-Process Information Flow Tracking. *Proceedings of the* 2017 *ACM SIGSAC Conference on Computer and Communications Security* (*CCS*), Dallas, TX, 30 October-3 November 2017, 377-390. https://doi.org/10.1145/3133956.3134045

[48] Ji, Y., Lee, S., Fazzini, M., *et al.* (2018) Enabling Refinable Cross-Host Attack Investigation with Efficient Data Flow Tagging and Tracking. 27*th USENIX Security Symposium* (*USENIX Security*), Baltimore, MD, 15-17 August 2018, 1705-1722.

[49] Liu, Y.S., Zhang, M., Li, D., *et al.* (2018) Towards a Timely Causality Analysis for Enterprise Security. *Proceedings of the Annual Network and Distributed System Security Symposium* (*NDSS*), San Diego, CA, 18-21 February 2018, 1-15. https://doi.org/10.14722/ndss.2018.23254

[50] Hossain, M.N., Sheikhi, S. and Sekar, R. (2020) Combating Dependence Explosion in Forensic Analysis Using Alternative Tag Propagation Semantics. *IEEE Symposium on Security and Privacy* (*SP*), San Francisco, CA, 18-21 May 2020, 1139-1155. https://doi.org/10.1109/SP40000.2020.00064

[51] Fang, P.C., Gao, P., Liu, C.L., *et al.* (2022) Back-Propagating System Dependency Impact for Attack Investigation. 31*st USENIX Security Symposium* (*USENIX Security*), Boston, MA, 10-12 August 2021, 1-18.

[52] Hossain, M.N., Milajerdi, S.M., Wang, J., *et al.* (2017) Sleuth: Real-Time At-Tack Scenario Reconstruction from Cots Audit Data. 26*th USENIX Security Symposium* (*USENIX Security*), Vancouver, BC, 16-18 August 2017, 487-504.

[53] Hassan, W.U., Guo, S., Li, D., *et al.* (2019) NODOZE: Combatting Threat Alert Fatigue with Automated Provenance Triage. *Proceedings of the Annual Network and Distributed System Security Symposium* (*NDSS*), San Diego, CA, 24-27 February 2019, 1-15. https://doi.org/10.14722/ndss.2019.23349

[54] Milajerdi, S.M., Gjomemo, R., Eshete, B., *et al.* (2019) HOLMES: Real-Time APT Detection through Correlation of Suspicious Information Flows. *IEEE Symposium on Security and Privacy* (*SP*), San Francisco, CA, 19-23 May 2019, 1137-1152. https://doi.org/10.1109/SP.2019.00026

[55] Milajerdi, S.M., Eshete, B., Gjomemo, R., *et al.* (2019) POIROT: Aligning Attack Behavior with Kernel Audit Records for Cyber Threat Hunting. *Proceedings of the* 2019 *ACM SIGSAC Conference on Computer and Communications Security*, London, 11-15 November 2019, 1795-1812. https://doi.org/10.1145/3319535.3363217

[56] Zhao, J., Yan, Q.B., Lin, X.D., *et al.* (2020) Cyber Threat Intelligence Modeling Based on Heterogeneous Graph Convolutional Network. 23*rd International Symposium on Research in Attacks*, *Intrusions and Defense*s (*RAID*), San Sebastian, 14-16 October 2020, 241-256.

[57] Satvat, K., Gjomemo, R. and Venkatakrishnan, V.N. (2021) EXTRACTOR: Extracting Attack Behavior from Threat Reports. *IEEE Symposium on Security and Privacy* (*SP*), Vienna, 6-10 September 2021, 598-615. https://doi.org/10.1109/EuroSP51992.2021.00046

[58] Han, X.Y., Pasquier, T., Bates, A., *et al.* (2020) UNICORN: Runtime Provenance-Based Detector for Advanced Persistent Threats. *Proceedings of the Annual Network and Distributed System Security Symposium* (*NDSS*), San Diego, CA, 23-26 February 2020, 1-18. https://doi.org/10.14722/ndss.2020.24046

[59] Alsaheel, A., Nan, X.Y., Ma, S.Q., *et al.* (2021) ATLAS: A Sequence-Based Learning Approach for Attack Investigation. 30*th USENIX Security Symposium* (*USENIX Security*), Vancouver, BC, 11-13 August 2021, 3005-3022.

[60] Fang, Y., Wang, C.S., Fang, Z.Y., *et al.* (2022) LMTracker: Lateral Movement Path Detection Based on Heterogeneous Graph Embedding. *Neurocomputing*, **474**, 37-47. https://doi.org/10.1016/j.neucom.2021.12.026

[61] Ma, S.Q., Zhang, X.Y., Xu, D.Y., *et al.* (2016) ProTracer: Towards Practical Provenance Tracing by Alternating Between Logging and Tainting. *Proceedings of the Annual Network and Distributed System Security Symposium* (*NDSS*), San Diego, CA, 21-24 February 2016, 1-15. https://doi.org/10.14722/ndss.2016.23350

[62] Xu, Z., Wu, Z.Y., Li, Z.C., *et al.* (2016) High Fidelity Data Reduction for Big Data Security Dependency Analyses. *Proceedings of the* 2016 *ACM SIGSAC Conference on Computer and Communications Security* (*CCS*), Vienna, 24-28 October 2016, 504-516. https://doi.org/10.1145/2976749.2978378

[63] Hossain, M.N., Wang, J.A., Sekar, R., *et al.* (2018) Dependence-Preserving Data Compaction for Scalable Forensic Analysis. 27*th USENIX Security Symposium* (*USENIX Security*), Baltimore, MD, 15-17 August 2018, 1723-1740.

[64] Tang, Y.T., Li, D., Li, Z.C., *et al.* (2018) NodeMerge: Template Based Efficient Data Reduction for Big-Data Causality Analysis. *Proceedings of the* 2018 *ACM SIGSAC Conference on Computer and Communications Security* (*CCS*), Toronto, 15-19 October 2018, 1324-1337. https://doi.org/10.1145/3243734.3243763

[65] Fei, P., Li, Z., Wang, Z.Y., *et al.* (2021) SEAL: Storage-Efficient Causality Analysis on Enterprise Logs with Query-Friendly Compression. 30*th USENIX Security Symposium* (*USENIX Security*), Vancouver, BC, 11-13 August 2021, 2987-3004.

[66] Hassan, W.U., Lemay, M., Aguse, N., *et al.* (2018) Towards Scalable Cluster Auditing through Grammatical Inference over Provenance Graphs. *Proceedings of the Annual Network and Distributed System Security Symposium* (*NDSS*), San Diego, CA, 18-21 February 2018, 1-15. https://doi.org/10.14722/ndss.2018.23141

[67] Ma, S.Q., Zhai, J., Kwon, Y.H., *et al.* (2018) Kernel-Supported Cost-Effective Audit Logging for Causality Tracking. 2018 *USENIX Annual Technical Conference* (*USENIX ATC*), Boston, MA, 11-13 July 2018, 241-254.

[68] Bates, A., Tian, D., Moyer, T., *et al.* (2015) Trustworthy Whole-System Provenance for the Linux Kernel. 24*th USENIX Security Symposium* (*USENIX Security*), Washington DC, 12-14 August 2015, 319-334.

[69] Kwon, Y., Wang, F., Wang, W.H., *et al.* (2018) MCI: Modeling-Based Causality Inference in Audit Logging for Attack Investigation. *Proceedings of the Annual Network and Distributed System Security Symposium* (*NDSS*), San Diego, CA, 18-21 February 2018, 1-15. https://doi.org/10.14722/ndss.2018.23306

[70] Kwon, Y.H., Kim, D., Sumner, W.N., *et al.* (2016) LDX: Causality Inference by Lightweight Dual Execution. *ASPLOS*'16: *Proceedings of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems*, **44**, 503-515. https://doi.org/10.1145/2872362.2872395

[71] Gao, P., Xiao, X.S., Li, Z.C., *et al.* (2018) AIQL: Enabling Efficient Attack Investigation from System Monitoring Data. 2018 *USENIX Annual Technical Conference* (*USENIX ATC*), Boston, MA, 11-13 July 2018, 113-125.

[72] Xu, Z.Q., Fang, P.C., Liu, C.L., *et al.* (2022) DEPCOMM: Graph Summarization on System Audit Logs for Attack Investigation. *IEEE Symposium on Security and Privacy* (*SP*), San Francisco, CA, 22-26 May 2021, 70-87.

[73] Holgado, P., Villagra, V.A. and Vazquez, L. (2020) Real-Time Multistep Attack Prediction Based on Hidden Markov Models. *IEEE Transactions on Dependable and Secure Computing*, **17**, 134-147. https://doi.org/10.1109/TDSC.2017.2751478

[74] Shen, Y., Mariconti, E., Vervier, P.A., *et al.* (2018) Tiresias: Predicting Security Events through Deep Learning. *Proceedings of the* 2018 *ACM SIGSAC Conference on Computer and Communications Security* (*CCS*), Toronto, 15-19 October 2018, 592-605. https://doi.org/10.1145/3243734.3243811

[75] Shen, Y., Stringhini, G., *et al.* (2019) ATTACK2VEC: Leveraging Temporal Word Embeddings to Understand the Evolution of Cyberattacks. 28*th USENIX Security Symposium* (*USENIX Security*), Santa Clara, CA, 14-16 August 2019, 905-921.

[76] Siadati, H., Memon, N., *et al.* (2017) Detecting Structurally Anomalous Logins within Enterprise Networks. *Proceedings of the* 2017 *ACM SIGSAC Conference on Computer and Communications Security* (*CCS*), Dallas, TX, 30 October-3 November 2017, 1273-1284. https://doi.org/10.1145/3133956.3134003

[77] Bohara, A., Noureddine, M.A., Fawaz, A., *et al.* (2017) An Unsupervised Multi-Detector Approach for Identifying Malicious Lateral Movement. *IEEE* 36*th Symposium on Reliable Distributed Systems* (*SRDS*), Hong Kong, 26-29 September 2017, 224-233. https://doi.org/10.1109/SRDS.2017.31

[78] Liu, F.C., Wen, Y., Zhang, D.X., *et al.* (2019) Log2vec: A Heterogeneous Graph Embedding Based Approach for Detecting Cyber Threats within Enterprise. *Proceedings of the* 2019 *ACM SIGSAC Conference on Computer and Communications Security* (*CCS*), London, 11-15 November 2019, 1777-1794. https://doi.org/10.1145/3319535.3363224

[79] Liu, F.C., Wen, Y., Wu, Y.N., *et al.* (2020) MLTracer: Malicious Logins Detection System via Graph Neural Network. *IEEE* 19*th International Conference on Trust, Security and Privacy in Computing and Communications* (*TrustCom*), Guangzhou, 29 December-1 January 2021, 715-726. https://doi.org/10.1109/TrustCom50675.2020.00099

[80] Creech, G. and Hu, J. (2013) Generation of a New IDS Test Dataset: Time to Retire the KDD Collection. 2013 *IEEE Wireless Communications and Networking Conference* (*WCNC*), Shanghai, 7-10 April 2013, 4487-4492. https://doi.org/10.1109/WCNC.2013.6555301

[81] Haider, W., Hu, J., Slay, J., *et al.* (2017) Generating Realistic Intrusion Detection System Dataset Based on Fuzzy Qualitative Modeling. *Journal of Network and Computer Applications*, **87**, 185-192. https://doi.org/10.1016/j.jnca.2017.03.018

[82] DARPA Intrusion Detection Evaluation Dataset (1998). https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset

[83] KDD Cup 1999 Data (1999). http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[84] Camina, J.B., Hernandez, G.C., Monroy, R., *et al.* (2014) The Windows-Users and Intruder Simulations Logs Dataset (WUIL): An Experimental Framework for Masquerade Detection Mechanisms. *Expert Systems with Applications*, **41**, 919-930. https://doi.org/10.1016/j.eswa.2013.08.022

[85] Kent, A.D. (2015) Comprehensive, Multi-Source Cyber-Security Events Data Set. https://csr.lanl.gov/data/cyber1/

[86] Garg, A., Rahalkar, R., Upadhyaya, S., *et al.* (2006) Profiling Users in GUI Based Systems for Masquerade Detection. *Proceedings of the* 2006 *IEEE Workshop on Information Assurance*, West Point, NY, 21-23 June 2006, 48-54. https://doi.org/10.1109/IAW.2006.1652076

[87] DARPA (2018) Transparent Computing Engagement 3 Data Release. https://github.com/darpa-i2o/Transparent-Computing

[88] Pei, K., Gu, Z.S. and Saltaformaggio, B. (2016) HERCULE: Attack Story Reconstruction via Community Discovery on Correlated Log Graph. *Proceedings of the* 32*nd Annual Conference on Computer Security Applications* (*ACSAC*), Los Angeles, CA, 5-8 December 2016, 583-595. https://doi.org/10.1145/2991079.2991122

[89] Xu, J.H., Wen, Y., Yang, C., *et al.* (2020) An Approach for Poisoning Attacks against RNN-Based Cyber Anomaly Detection. *IEEE* 19*th International Conference on Trust*, *Security and Privacy in Computing and Communications* (*TrustCom*), Guangzhou, 29 December-1 January 2021, 1680-1687. https://doi.org/10.1109/TrustCom50675.2020.00231