软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2024,35(1):333−355 [doi: 10.13328/j.cnki.jos.006891]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

加密恶意流量检测及对抗综述*

侯 剑[1,2], 鲁 辉[2], 刘方爱[1], 王兴伟[3], 田志宏[2]

[1](山东师范大学 信息化工作办公室, 山东 济南 250014)

[2](广州大学 网络空间安全学院, 广东 广州 510799)

[3](东北大学 计算机科学与工程学院, 辽宁 沈阳 110169)

通信作者: 田志宏, E-mail: tianzhihong@gzhu.edu.cn

摘 要: 网络流量加密在保护企业数据和用户隐私的同时, 也为恶意流量检测带来新的挑战. 根据处理加密流量的方式不同, 加密恶意流量检测可分为主动检测和被动检测. 主动检测包括对流量解密后的检测和基于可搜索加密技术的检测, 其研究重点是隐私安全的保障和检测效率的提升, 主要分析可信执行环境和可控传输协议等保障措施的应用. 被动检测是在用户无感知且不执行任何加密或解密操作的前提下, 识别加密恶意流量的检测方法, 其研究重点是特征的选择与构建, 主要从侧信道特征、明文特征和原始流量等 3 类特征分析相关检测方法, 给出有关模型的实验评估结论. 最后, 从混淆流量特征、干扰学习算法和隐藏相关信息等角度, 分析加密恶意流量检测对抗研究的可实施性.

关键词: 加密流量; 恶意流量检测; 中间盒; 可搜索加密; 机器学习

中图法分类号: TP309

中文引用格式: 侯剑, 鲁辉, 刘方爱, 王兴伟, 田志宏. 加密恶意流量检测及对抗综述. 软件学报, 2024, 35(1): 333–355. http://www.jos.org.cn/1000-9825/6891.htm

英文引用格式: Hou J, Lu H, Liu FA, Wang XW, Tian ZH. Detection and Countermeasure of Encrypted Malicious Traffic: A Survey. Ruan Jian Xue Bao/Journal of Software, 2024, 35(1): 333–355 (in Chinese). http://www.jos.org.cn/1000-9825/6891.htm

Detection and Countermeasure of Encrypted Malicious Traffic: A Survey

HOU Jian[1,2], LU Hui[2], LIU Fang-Ai[1], WANG Xing-Wei[3], TIAN Zhi-Hong[2]

[1](Informatization Office, Shandong Normal University, Jinan 250014, China)

[2](Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510799, China)

[3](School of Computer Science and Engineering, Northeastern University, Shenyang 110169, China)

Abstract: Network traffic encryption not only protects corporate data and user privacy but also brings new challenges to malicious traffic detection. According to different ways of processing encrypted traffic, encrypted malicious traffic detection technology can be divided into active and passive detection. Active detection technology includes detection after traffic decryption and that based on searchable encryption technology. Its research focuses on privacy protection and detection efficiency improvement, and mainly analyzes the application of trusted execution environments and controllable transmission protocols. Passive detection technology is a method of identifying encrypted malicious traffic without perception for users and without performing any encryption or decryption operations. The research focuses on the selection and construction of features. It analyzes relevant detection methods from three types of features such as side channel features, plaintext features, and raw traffic, and then the experimental evaluation conclusions of relevant models are given. Finally, the feasibility of the research on the countermeasures of encrypted malicious traffic detection is analyzed from the perspectives of obfuscating traffic characteristics, interference learning algorithms, and hiding relevant information.

* 基金项目: 国家自然科学基金 (U20B2046); 广东省高校创新团队项目 (2020KCXTD007); 广州市高校创新团队项目 (202032854); 山东自然科学基金 (ZR2020KF021)
收稿时间: 2021-09-27; 修改时间: 2022-02-20; 采用时间: 2022-05-28; jos 在线出版时间: 2023-07-28
CNKI 网络首发时间: 2023-07-31

Key words: encrypted traffic; malicious traffic detection; middlebox; searchable encryption; machine learning

恶意流量通常指通过未经许可的方式侵入、干扰、抓取他方业务或数据的网络流量, 主要由网络入侵、业务攻击、恶意程序等网络威胁产生.《2020 年中国互联网网络安全报告》[1]指出: 全年捕获恶意程序样本数量超过 4 200 万个, 日均传播次数为 482 万余次; 利用安全漏洞针对境内主机进行扫描探测、代码执行等的远程攻击行为日均超过 2 176.4 万次. 恶意流量仍然是目前互联网用户面临的较常见、影响较严重的网络安全威胁之一. 与此同时, 随着网络中加密流量占比的持续增加, 恶意流量为了隐藏行为, 也逐渐选择流量加密的方式来逃避检测. Zscaler 威胁研究报告[2]表明, 2020 年, 在针对医疗、金融及制造业等 5 大行业的网络攻击中, 基于加密 (SSL) 的恶意流量占网络安全威胁总流量的平均 17.95% 左右, 并且该指标在前 9 个月中增长了 260%. 加密恶意流量增速之快, 导致其引发的安全威胁日趋严重.

为了有效应对互联网安全威胁, 我国不断完善网络安全法律法规体系. 在《网络安全法》基础上, 颁布了《数据安全法》和《个人信息保护法》保护数据安全和用户个人信息安全. 同时, 在国家“十四五”规划中, 网络安全和数据安全也作为经济发展的重要保障贯穿于整个规划之中. 建立健全网络综合治理体系, 依法治理网络空间已经取得一定成效. 在相关法律和政策的指引下, 绝大多数企业互联网服务也都采用加密协议访问, 网络用户对隐私保护和数据保护的意识也愈加强烈. 以 DDoS 为代表的网络入侵行为持续降低, 由此引发的网络安全事件也不断走低. 然而, 网络安全漏洞数量的持续升高以及安全产品自身的安全等问题, 也带来了网络威胁和安全环境的不确定因素.

随着网络安全环境的变化, 恶意流量传播与治理对抗性也在不断加剧. 网络安全威胁的主要形态不断变化, 其相关检测研究重点也正在发生转变. 主要表现在以下几个方面: 一是加密的恶意流量占比增加. 加密是恶意流量规避检测的主要途径之一, 互联网加密流量的持续增加和网络安全漏洞趋势的上升, 往往会更有利于恶意流量的传播. 二是恶意流量向应用层发展. 随着传统防护措施的完善和流量检测技术的发展, 网络安全的研究重点不再仅限于网络层的入侵检测系统, 安全人员更加关注包含应用层信息的恶意流量分析. 三是有限检测权限. 加密流量恶意检测应在保障用户隐私和数据安全的前提下来完成, 尽量避免检测系统可保存或传输更多的明文信息. 四是人工智能技术的推动. 以机器学习为代表的人工智能技术已经在模式识别领域取得了巨大的成就, 目前, 在流量特征识别方面, 机器学习已是最受关注的加密恶意流量检测研究方法. 我们以“加密”“恶意流量”为关键词, 在 Scopus 和 WoS 数据库中进行了检索, 筛选了近 5 年的研究成果, 并依此构建了频繁词知识图谱. 如图 1 所示, 加密恶意流量检测是由入侵检测研究发展而来, 近期的技术研究热点主要围绕在密码学和机器学习两大知识领域开展. 在密码学相关领域, 涉及了网络协议、数据加密和密钥管理等技术. 在机器学习领域, 则涉及协议分析、特征提取和流量分析等相关技术. 深度学习作为机器学习领域最新的研究方向, 近年来在加密恶意流量检测中也有较多代表性成果.

根据上述网络安全威胁及其检测的研究趋势, 本文着重分析加密恶意流量的检测方法, 并根据是否解密流量, 把加密恶意流量的检测方法分为主动检测和被动检测[3,4]. 主动检测是指利用密码学知识, 通过对加密流量进行解密或半解密后检测流量数据. 该方法一般能获得较高的准确率指标, 但解密后的流量往往会带来进一步的隐私安全问题, 同时解密过程也有较高的网络性能损耗. 因此, 加密恶意流量主动检测的研究内容主要是解密过程的隐私安全和解密效率的有效提升. 被动检测方法不需要解密网络流量, 而是通过提取加密恶意流量的特征, 构建带标签的恶意特征集, 并使用机器学习或深度学习等人工智能技术通过训练模型来标注流量. 该类方法能最大程度保护用户隐私, 其准确度取决于流量特征和检测模型的构建. 恶意流量的逃避与检测是一个对抗的过程. 对于主动检测方案, 解密后的恶意流量检测通常可以利用 DoS 攻击、数据包拆分及载荷突变等传统入侵检测逃避技术实现对抗检测. 文献 [5,6] 综述了传统的入侵检测逃避技术, 本文主要面向加密的恶意流量, 因此该部分内容不做重点讨论. 本文第 1 节从可明文检测技术和可搜索加密技术两方面总结加密恶意流量主动检测的主要研究工作及代表性成果, 分析相关研究的技术路线及优势. 第 2 节总结加密恶意流量被动检测流程框架, 并从特征工程入手, 详细讨论基于人工智能技术的被动检测技术, 选取近年来有代表性的或取得显著效果的方法进行比较和分析. 第 3 节参照加密流量检测的对抗技术, 从网络流量审查规避和人工智能技术对抗等方面, 分析加密恶意流量检测对抗的可行性技术方法.

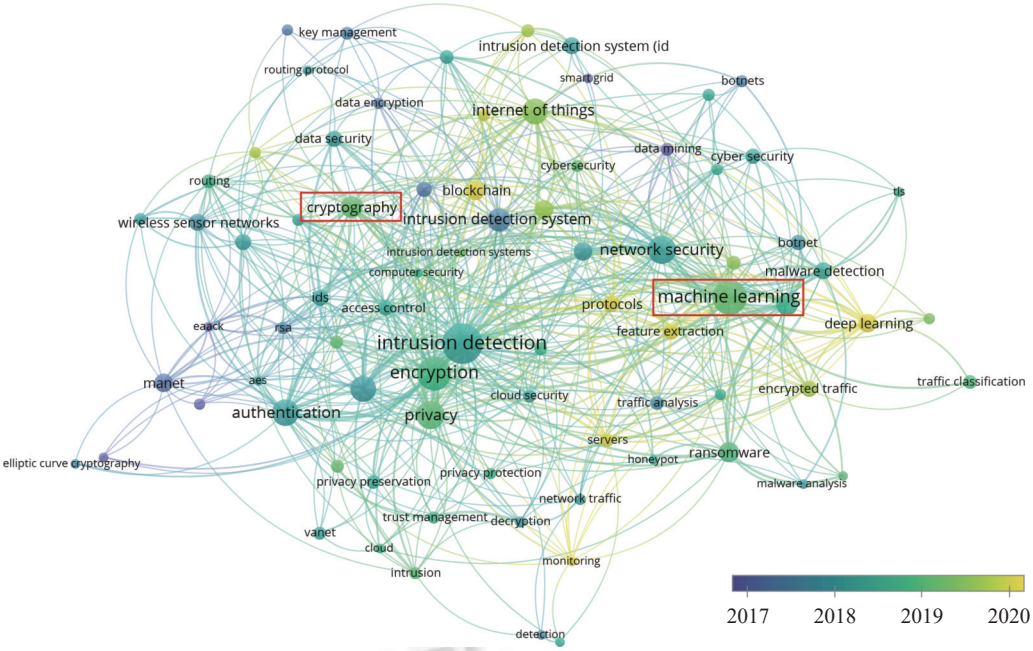


图 1　加密恶意流量检测研究热点关键词知识图谱

1　加密恶意流量主动检测

加密恶意流量主动检测的基本思想是利用中间盒 (middelbox, MB)[7]设备对原始加密流量执行解密或部分解密操作, 使加密流量的载荷信息明文可见或关键词可见, 从而可以使用深度包检测 (DPI) 或规则匹配识别恶意流量. 最常用的解决方案是在网络中增加一个代理设备并下发合法证书, 所有经过的加密流量由代理设备解密检查后再加密转发. 然而, 解密后的信息却带来了另一层面的网络安全隐患, 因此, 加密恶意流量主动检测的研究目标是, 在考虑加密流量隐私保护的同时, 高效、准确地实现恶意流量的检测. 目前, 安全企业和学者已从不同角度开展了相关的研究工作, 并取得了一定的成果. 根据从加密流量中获取信息方式的不同, 将加密恶意流量主动检测的方法分为可明文检测技术和可搜索加密技术分别总结.

1.1　可明文检测技术

可明文检测技术指在服务器和客户端之间部署中间盒设备, 直接对传入中间盒的加密流量进行解密分析, 并使用签名、关联或正则表达等 DPI 方法检测恶意流量. 在可明文检测技术中, 最早是利用中间人 (MitM) 技术开展解密检测[8]. 该技术是通过部署 MitM 代理, 将服务器与客户端的加密连接分成“客户端-代理”和“代理-服务器”两个部分, 代理能够对所有经过的流量进行解密, 并能通过深度包检测获取流量中所有的信息. 经过 MitM 代理的流量不需要修改协议类型, 重新加密后的流量能最大程度地保持原有流量的特性, 而且检测开销相对较低, 因此, 作为防病毒、防火墙和企业级网关等安全设备的底层技术架构, 该技术衍生了许多研究成果和商业解决方案[9], 同时也有较多的开源实现工具[10]. MitM 技术一直保持更新, 最近有网络安全企业基于该技术研发了可应用于 TLS 最新版协议的恶意流量检测产品[11]. 然而, 对于 MitM 技术一直以来研究人员最关注的还是其相关安全问题, 文献 [12] 分析了 14 个基于 MitM 技术的防病毒产品, 提出了一个评估客户端 TLS 代理的框架, 揭示了该类产品在 TLS 版本和证书映射、证书验证等方面的漏洞, 并提出了对类似安全产品的质疑; 在此研究的基础上, 文献 [13] 通过在 12 个网络设备和 13 个客户端 TLS 代理上应用早期框架的测试, 也发现了证书验证、密码套件、TLS 版本和已知 TLS 攻击中的设计缺陷. 综上所述, MitM 技术在安全和便捷性的交织中不断更新.

随着计算机软硬件技术的发展, MitM 代理设备的处理能力大幅提升, 已能基本满足用户对其网络及其检测性

能的需求. 然而, 另一方面由于 MitM 技术的开放性, 也不断给用户带来对该类中间盒设备安全性的担忧, 中间盒设备的可信性是值得用户关注的问题. Intel SGX 是 intel 架构一项新的安全扩展, 不依赖于周边固件和软件的安全状态, 以硬件安全为强制性保障, 为用户提供可信的应用执行环境. 文献 [14] 基于 SGX 的可信执行环境提出了可安全检测加密流量的 SGX-Box 系统, 流量的解密和深度包检测等应用运行在 SGX 提供的安全容器 enclave 中. enclave 通过实现计算和内存的隔离来确保敏感数据受到保护, 同时 SGX-Box 还允许用户远程验证检测应用的完整性和可信性. SGX-Box 提供了完整的流量解密接口, 安全人员可以直接获得解密后的流量, 从而更专注于恶意流量的深度包检测工作. 另外, 作者还指出 SGX 技术也逐步向云端环境发展. 云计算的发展也推动着中间盒设备的云化部署[15], 但是对用户来说, 公有云环境一般会被当作不受信的环境. 为了在不受信的云环境中部署高性能中间盒设备, 文献 [16] 基于 Scone 提出了 ShieldBox 框架, Scone 是一个基于 SGX 的安全容器. ShieldBox 在安全容器执行解密和检测应用, 并部署 click 软路由和 HyperScan 模块实现了 IDS 功能. 不是所有用户都对安全云服务持信任态度, 为了满足日益增长的网络安全中间盒设备需求, 并且避免使用云服务, 文献 [17] 提出了一种部署在网络边缘的分布式中间盒系统 EndBox, 为客户端提供基于可信硬件技术 SGX 的安全服务. EndBox 设计了 Snort 规则集和字符串匹配算法, 并作为 click 的一个模块, 来检测解密后的恶意流量. 为加密恶意流量检测提供可信执行环境是上述检测方法的基本思路, 同 MitM 技术一样, 该类方法可以对加密恶意流量开展深度检测; 然而, 受限于可信执行环境的性能和规则, 该类方法的检测效率普遍不高.

我们把上述中间盒部署方案归纳为半感知的方法, 也就是说客户端和服务器至少有一方不必了解中间盒的行为, 中间盒作为独立运行的设备不受客户端或服务器的约束. 为了提升中间盒设备对于网络端点的状态可见性和服务可控性, 文献 [18] 提出了建立在 TLS 协议之上的 mcTLS 协议, 端点设备通过 mcTLS 协议可以控制中间盒的功能和分配中间盒的权限. 中间盒与两个端点设备使用 mcTLS 协议建立安全和可验证的通道, 并按照策略授权访问加密流量. 基于 mcTLS 协议的中间盒设备可部署为 IDS, 按照客户端和服务器的需求有选择的解密流量并进一步检测恶意流量, 从而提升加密恶意流量的检测效率. 基于 mcTLS 协议的方法需要将网络中 TLS 协议替换为 mcTLS 协议, 相对于使用成熟的标准 TLS 协议体系, 这样的替换可能会带来一定的开销, 尤其是在日益发展的云计算环境下, 网络基础协议仍需以成熟的协议体系为主. 考虑到对通用标准的需求情况, 欧洲电信标准化协会 (ETSI) 推动了以 mcTLS 协议为基础的中间盒安全协议 (MSP) 的完善, 并于 2021 年 2 月发布了其首个版本[19]. 同时, 基于 mcTLS 思想新方法也陆续被提出. 文献 [20] 提出的 mbTLS 不用修改底层 TLS 协议, 并且更适合云环境的部署. 在 mbTLS 中, 客户端、中间盒和服务器每两个设备之间维护一个不同的 TLS 会话, mbTLS 作为应用层协议负责加密数据的封装、安全和验证. 另外, 为了满足在云部署环境下的安全, mbTLS 中间盒的应用被设计在 SGX enclave 环境下执行. 文献 [21] 提出的 maTLS 协议也同样适用于云环境, 可实现多中间盒设备部署, 同样 maTLS 不采用共享密钥的方法, 而是在中间盒之间协商单独的 TLS 会话, 并且客户端和服务器都能够监督所有中间盒行为. 上述文献没有给出相关方法在加密恶意流量检测中的案例实现, 但阐述了其在 IDS 应用场景下部署的灵活性. 同云环境一样, 物联网环境中的中间盒技术也至关重要. 文献 [22] 基于 TLS1.3 提出了 ME-TLS 协议, 设计思想同 mcTLS 基本一致, 另外提供了一种隐式的版本协商机制来实现与传统的 TLS 协议兼容. 中间盒能够在握手信息中识别连接对象是否支持 ME-TLS 协议, 并根据协商情况选择是否使用该协议, 并在握手完成后利用获得的密钥可执行恶意流量检测等任务. 另外, 文献还评估了执行 ME-TLS 的设备功耗, 指明了物联网环境部署的可行性.

综上所述, 加密恶意流量的可明文检测技术是在保障用户隐私安全的前提下, 并对完全解密后的流量明文信息开展深度包检测的方法. 由于加密后明文信息开展的深度包检测仍采用非加密恶意流量检测方法, 因此, 可明文检测技术的研究重点是为加密恶意流量检测构建相对安全的网络规则和运行环境. 基于 SplitTLS[8]技术的中间盒是被广泛应用的商业化技术, 但是其解密和检测环境的隐私安全仍是被网络用户和应用服务商最关注的问题之一. 目前, 以 Intel SGX 为代表的可信执行环境的相关研究, 正在试图为流量解密和恶意流量检测寻找可靠的运行环境; 基于 mcTLS 相关技术的研究, 也正在开展从客户端到服务器整条路经的信任链打造工作, 力图实现安全透明的通信会话. 我们用图 2 展示了基于可明文检测技术的加密恶意流量检测研究代表性成果框图.

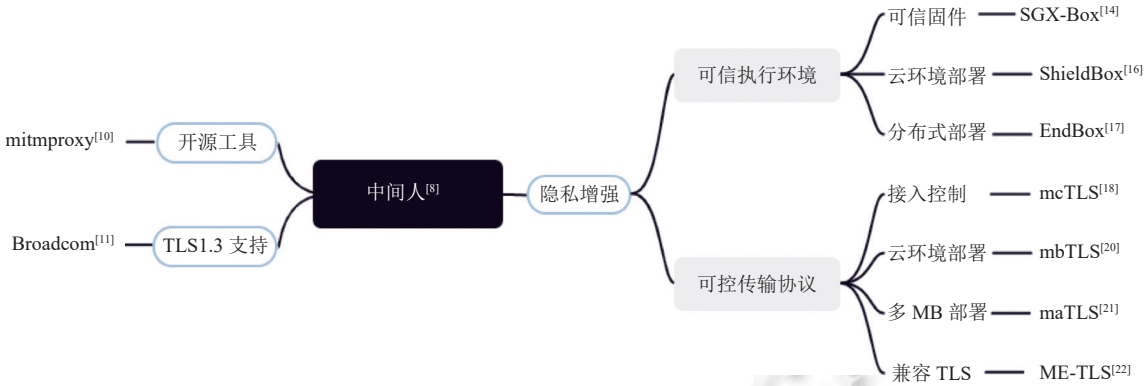


图 2　可明文检测技术典型研究成果

1.2　可搜索加密技术

可搜索加密是一种支持用户在密文上进行关键词查找的密码学原语. 与可明文检测技术不同, 使用可搜索加密技术的恶意流量主动检测方法不需要完全解密网络流量, 而是在流量仍是加密的状态下, 实现恶意信息关键词的检索. 基于可搜索加密技术的中间盒设备往往没有解密流量的权限或密钥, 因此, 无法直接对加密的 TLS 网络会话的数据解密, 中间盒会通过在客户端和服务器上安装代理的方式, 与两者之间维护一条用来传输加密令牌化数据的链路. 端点使用 TLS 协议加密前的网络流生成令牌化数据, 并使用由原有 TLS 会话密钥派生的密钥加密后传输. 中间盒设备使用与派生密钥相同的密钥或密钥对维护着一个加密的恶意流量规则集, 并以此对加密令牌化数据开展恶意关键词检索来实现加密恶意流量的识别. 数据传输到达后, 端点代理会验证两条链路数据的一致性. 图 3 展示了可搜索加密技术的基本流程图.

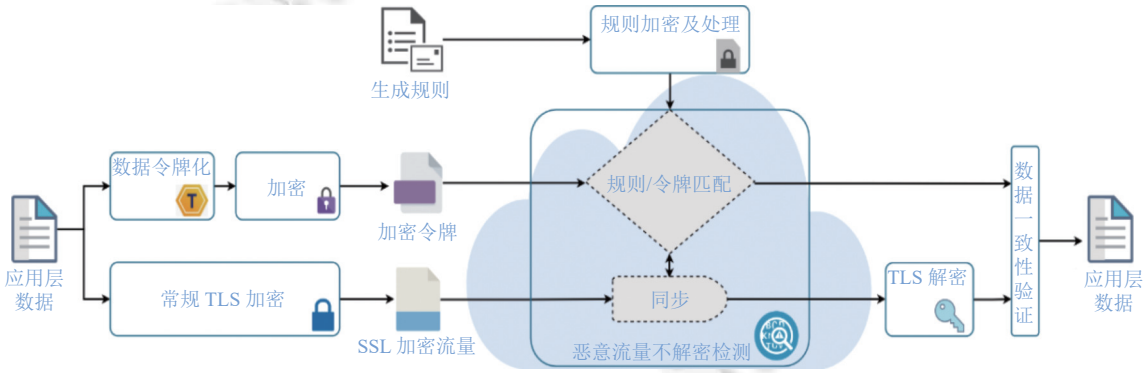


图 3　可搜索加密技术检测流程图

根据使用密钥的不同, 可搜索加密技术一般分为对称可搜索加密和公钥可搜索加密. 对称可搜索加密技术中的数据和关键词必须使用同一个密钥进行加密, 一般是单用户使用的场景, 因此, 考虑到中间盒设备往往运行在不可信环境中等因素, 对称可搜索加密通常不作为恶意流量主动检测的重点技术研究. 基于公钥可搜索加密技术的检测方案则对网络隐私安全环境要求相对较低, 端点设备用公钥来加密令牌数据, 中间盒设备使用私钥和关键词生成的陷门进行密文搜索. 作为公钥可搜索加密技术的重要组成, 密文检索和密文计算两方面内容, 在文献 [23] 已经做了详细的综述. 本文接下来将结合已有研究, 总结加密恶意流量检测中公钥可搜索加密技术的应用. 我们用图 4 表示基于可搜索加密技术的加密恶意流量检测研究典型成果.

BlindBox[24]在考虑了隐私保护需求的前提下, 基于可搜索加密技术建立中间盒系统, 是第 1 个相对成熟的基于可搜索加密技术的加密流量检测方案. 考虑到实际部署环境, 中间盒通常被认为是在不可信的网络环境中, 为确

保中间盒不会获得可以解密流量的密钥, BlindBox 使用乱码电路来执行规则集的加密. 文献中评估了该系统使用通用 IDS 规则集作为关键词的检索结果, 在数据集 ICTF2010 的攻击规则检测实验中最高达到了 99% 的检测率. 由于 BlindBox 需要对每个会话分别执行规则集加密, 所以系统的计算成本很高, 尤其在会话握手的准备阶段. 文中给出了该系统的时间开销, 对 3 000 条规则的准备时间达到了 97 s. 该团队在接下来的研究中考虑了中间盒在云环境部署的情况, 提出了 Embark 系统[25], 与 BlindBox 只能检测载荷数据相比, Embark 可以通过范围查询或前缀匹配实现数据包头信息的检测; 在时间开销方面, 与 BlindBox 相比, 每条链接的准备时间虽然没有缩短, 但网关和云可以保持一个长期持久连接, 因此, 只有在初始化网关配置是才会产生较高的时间开销. 文献实验中, 针对恶意流量识别的结果表明 Embark 在系统性能和安全性上相比 BlindBox 都有一定的提升. 中间盒的云部署推动了更多研究的产生, 文献 [26] 提出了一种在云端使用高性能加密规则过滤器的方法, 大幅提升了该类方法对恶意流量的检测效率, 文中使用了 Snort 和 ETOpen 规则集, 在 DARPA99 和 ICTF2008 数据集上开展了详细的模拟实验, 性能结果与 BlindBox 对比有大幅提升. 同样, 文献 [27] 也提出了一种基于云虚拟机的分布式架构, 将中间盒流量检测的功能分发到云计算多个节点上, 并设计了 SplitBox 系统通过协同计算实现整个分布式节点的网络检测功能. 在这种条件下每个节点并不能获取完整的网络信息, 从而保证了整体网络的隐私安全和可靠性. 除了云安全部署的需求, 可加密搜索中间盒系统的运行效率也是该研究关注的热点. 文献 [28] 提出的 SPABox 与 BlindBox 相比, 检测速度提升了 5 个数量级, 带宽占用少 3 个数量级; 文献 [29] 提出的 BlindIDS 在时间开销和内存开销上分别比 BlindBox 低 3 个数量级和 6 个数量级, 而 ES-DPI[30]通过减少初始化设置阶段的计算和通信开销, 与 BlindIDS 相比又有大幅提升. 在最近的研究中, 有研究人员发现 BlindBox 在处理独立短连接网络流时效率极差, 并针对此问题提出了 PrivDPI 系统[31], 作者放弃了使用乱码电路生成规则, 而是采用了一种可重用的混淆规则生成技术, 使第 1 个会话生成的规则可以在后续会话中重用, 这大大缩短了初始化握手阶段的计算时间, 尤其是当处理大量独立短连接会话时, 整体检测时间开销将得到有效改善; 在接下来的研究中, 考虑到第三方云部署场景和规则的隐私安全, 该团队提出了支持规则隐藏的 Pine 协议[32], 并优化了 PrivDPI 的预处理步骤, 进一步减少了连接建立时间和通信开销.

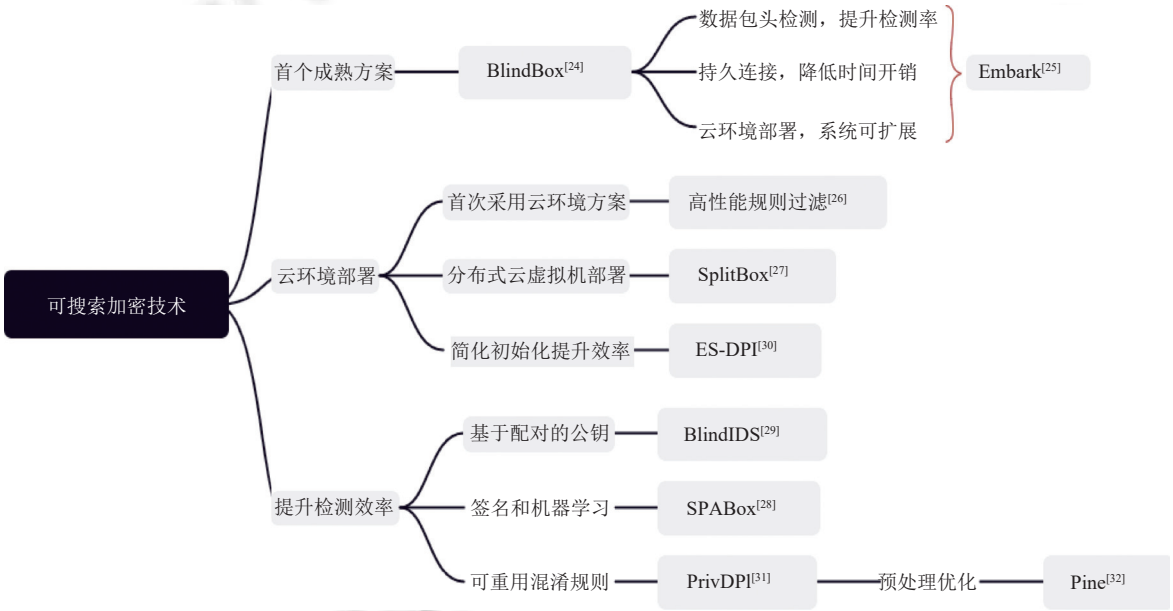


图 4 可搜索加密检测技术典型研究成果

综上所述, 基于可搜索加密技术的恶意流量检测方法不直接检测 TLS 会话流量, 其分析对象是加密的令牌化数据, 该数据应与 TLS 会话具有原始信息的一致性. 在可搜索加密的检测方法中, 中间盒不应该知道用于加密规

则集和生成令牌的密钥, 因为如果获得密钥, 中间盒将能解密令牌化流量中的所有信息, 因此, 隐私保护应是该类方法考虑的问题之一. 另外, 该类方法检测的完整过程包含了密钥生成与分发、令牌化数据的形成、规则加密与匹配等一系列步骤, 而涉及的密文检索和密文计算往往有较高的复杂性, 因此, 基于可搜索加密技术的恶意流量检测方法的效率问题也成为研究的关注重点.

1.3 小　结

加密恶意流量主动检测是中间盒系统主动获取网络流量, 并通过解密流量或生成陷门的方法, 使加密流量转换为可以开展明文检测或密文检索形式. 可明文检测技术中加密流量会被中间盒系统解密, 并采用传统恶意流量检测方法识别, 输出结果后流量被再次加密后继续传输, 因此, 中间盒系统环境的安全性是影响该类检测方法的关键因素. 采用可搜索加密技术的恶意流量检测系统不直接解密网络流量, 而是通过构造令牌化加密数据, 使用密文检索技术根据既定规则发现恶意流量; 由于不用解密网络流量, 可搜索加密技术的隐私安全性相对有一定保障, 然而其密钥安全的管理、加密规则的保护以及检测效率等问题有待深入研究.

相对于采用流量“拦截”方式的主动检测技术, 以镜像流量为分析对象的被动检测方法则更容易保障网络隐私安全, 且更具检测效率. 加密恶意流量被动检测是以恶意流量的通信模式或流量特征为检测依据, 利用机器学习技术建立识别模型的检测方法. 机器学习技术已在加密流量分类识别研究领域显示出优越的性能[33−36], 基于机器学习的加密恶意流量被动检测也成为网络安全领域的热门研究方向.

2 加密恶意流量被动检测

2.1 检测体系

早期的恶意流量检测研究以僵尸网络检测为主. 文献 [37,38] 提出了基于流量模式的被动检测方法, 利用僵尸网络 C&C 通道网络流量的时空相关性和相似性, 发现了僵尸网络的 C&C 通信模式和恶意行为模式, 并在真实网络环境下检测验证, 提高了僵尸网络的检测效果. 文献 [39] 借鉴了 Alshammari 等人关于加密流量应用识别的研究成果[40−45], 构建了可以检测加密恶意流量的系统框架. 该方法从网络流量文件中提取了 29 个数据包头特征构成特征集, 基于机器学习算法构建了网络级别的恶意流量分类器, 该成果首次系统性对比了 5 种机器学习算法在恶意流量检测研究中的应用情况. 在进一步的研究中[46], 作者又全面对比了 IP 数据包头特征分类器检测和使用全部数据包信息的深度包检测 (DPI) 两种恶意流量检测方法, 并从特征集选择、检测准确率和算法复杂度等几个方面做了评估, 结果显示使用基于特征分类的方法可以获得与 DPI 同样良好的检测率. 基于恶意流量模式检测方法的提出与发展, 为加密恶意流量被动检测提供了研究基础.

加密恶意流量被动检测的主要研究方法是基于恶意流量行为模式构建加密流量特征集, 利用机器学习或深度学习模型对其进行识别, 并通过模型设计与参数调优等方法获得理想的准确率. 该类方法具有准确率高、鲁棒性好和适用性广等优点, 因此也成为加密恶意流量检测的研究热点.

2016 年, Anderson 等人在文献 [47] 中首次较全面地阐述了加密恶意流量被动检测方法, 该研究提取了数据包长度及到达时间间隔、TLS 协议信息、上下文 DNS 信息及 HTTP 相关信息等特征, 详细分析了恶意流量与正常流量在不同特征维度上的区别, 首次增加了 TLS 握手数据包中提取的版本号、密钥长度等非加密信息来辅助检测恶意流量. 作者使用逻辑回归分类器, 在自建数据集上测试了不同特征组合的检测结果, 实验证明当使用全部特征识别恶意流量时可获取非常的检测效果, 并且在后续的研究中[48], 同步分析了该方法的在线检测性能. 该成果的公布代表着加密恶意流量检测已成为一个新的网络安全专项研究课题.

在已研究基础上, Anderson 等人还考虑了实际网络环境中存在噪音标签及加密网络服务的非平稳性等因素, 参考已有的恶意流量及加密流量分类特征提取成果[49,50], 进一步完善了加密恶意流量特征库, 并在检测实验中对比了 6 种常用的机器学习算法[51]. 实验结果显示, 随机森林 (RF) 集成分类器在检测加密恶意流量时具有较好的鲁棒性, 同时实验也证明了特征选择是影响加密恶意流量检测性能的重要因素.

从现有典型的研究成果来看, 加密恶意流量被动检测研究方法通常具有性能好、效率高及适用性广等特点,

我们将其研究体系归纳为 4 部分 (图 5), 首先根据检测任务选择数据集, 一般使用公开数据集或自建数据集; 其次对获取的数据集实施预处理及特征工程, 预处理主要包括流量拆分、重组及转换等操作, 特征工程一般包括特征提取和特征选择; 然后构建合适的机器学习和深度学习模型, 并将获取的流量特征作为输入, 训练识别恶意流量从而得到最终模型; 最后会依据准确度、精度及查全率等参数评估模型的性能. 接下来, 本文将从加密恶意流量的特征工程出发, 对加密恶意流量被动检测已有研究成果进行总结分析.

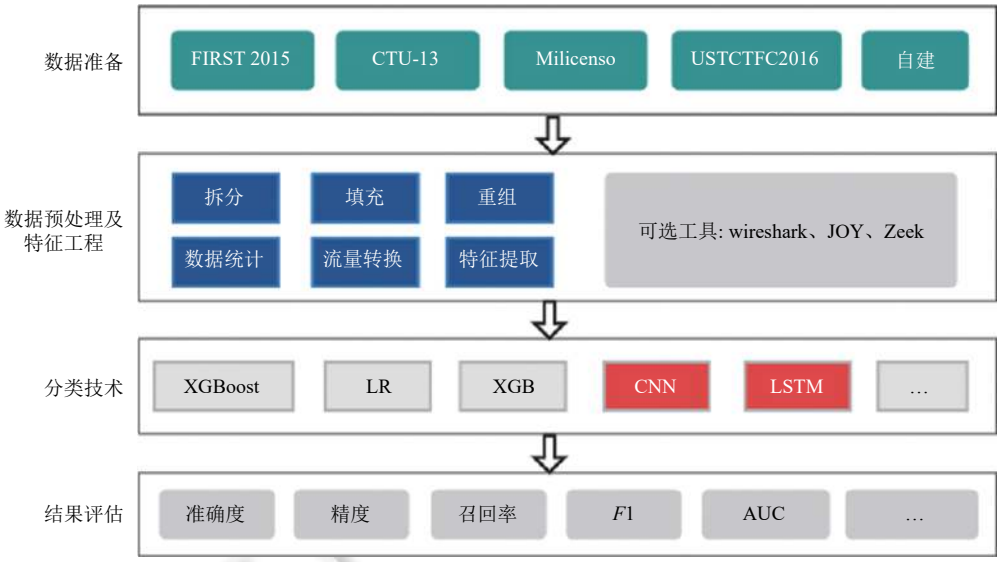


图 5 加密恶意流量被动检测框架

2.2 特征工程

加密恶意流量被动检测使用网络流元数据特征、侧信道特征、明文特征和原始流量作为检测模型的输入. 网络流元数据特征包括网络流包数及字节数、源/目的端口号和流持续时间, 该类特征通常不单独作为加密流量分类识别的特征, 而是与其他流量特征共同使用. 侧信道特征流量侧信道特征来源于不检查加密流量内容而直接观测的数据, 包括数据包长度特征、时间特征及相关统计特征. 明文特征是指可以从数据包中直接获取的密码套件、扩展信息等明文信息及其统计信息特征. 原始流量是检测模型输入的一种特殊形式, 使用原始流量作为输入的模型通常是以数据包字节为输入单元, 将其实际值作为模型的输入.

在加密恶意流量被动检测的研究中, 特征工程一直是被研究人员关注的对象. 加密流量特征往往可以作为一种可插拔 (pluggable) 单元应用于检测模型中, 不同特征在不同分类器上的性能表现有差异, 不同的检测任务, 譬如是否在线监测, 需要选择不同流量特征. Shekhawat 等人以数据集 CTU-13 和 MCFP 为例[52], 使用开源的流量分析软件从数据集中提取了流量日志文件, 包含加密流量的连接、会话和证书等 3 大类共 38 个加密流量特征. 作者选择了 SVM, RF, XGBoost 等 3 种机器学习算法对所提取的特征进行检测, 在取得较高检测率的同时, 对所有特征按信息增益进行排序, 由此来确定所选特征的重要性. 同时, 作者分析了所有特征的 Pearson 相关系数, 展示了不同特征之间的相关性. 通过综合分析, 作者认为在加密恶意流量检测研究中, 少量的特征足以得到理想的准确率和检测效率. 本文将加密恶意流量典型特征分为 3 种类型 (后文图 6), 下面将对已有的相关研究成果按照不同特征构建方式进行详细介绍.

2.2.1 侧信道特征

加密网络流量中的侧信道特征已被证明了可用来获取用户网络行为等重要信息[53]. 在文献 [54] 中, Stergiopoulos 等人通过分析 TCP 数据包的侧信道特征, 选择使用了数据包大小、有效载荷大小、有效载荷率、当前包与之前包的大小比率, 以及当前包与之前包时间差等特征, 并在 CTU-13 等 3 个公开数据集上测试了 6 种机器学习算法识别恶意流量. 实验结果显示, 仅使用加密恶意流量的侧信道信息, 就可以获取较高的准确率和查全率.

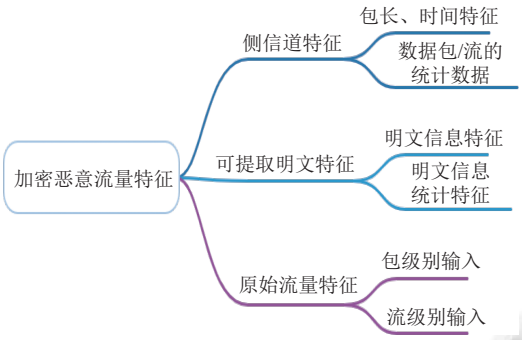


图 6 加密恶意流量特征组成

侧信道特征是加密恶意流量检测研究中最早应用, 也是最常使用的特征, 在加密恶意流量检测中有广泛的应用场景. 文献 [55] 将上下行数据流字节、链接持续时长及会话请求间隔时长等信息分成若干个区间, 组成了共 114 个特征的四元组向量, 使用度量索引优化的 KNN 算法进行检测, 该研究是首次专门针对 HTTPS 流量检测恶意软件; 在该研究基础上, 文献 [56] 完善了上述四元组向量对加密恶意流量通信模式的表达, 并应用神经网络 (NN)、RF 及 XGBoost 分类器上验证了检测结果. 恶意流量行为往往具有相似的特征, 通过构建流内或流间的特征表达模型可以实现新型恶意流量的检测识别. 文献 [57] 提出了基于距离的加密恶意流量检测方法, 通过分析具有恶意软件的相似网络行为构建识别框架. 该研究以 TCP/IP 包头、时间相关、长度相关及数据包变化等 4 大类 83 个侧信道特征为基础, 采用序列前向选择 (SFS) 的启发式特征选择方法, 使用 GMM 算法获得恶意流量样本中心点并使用 OPTICS[58]进行聚类分析, 从而挖掘出不同恶意流量的网络行为, 进一步提升恶意流量检测准确度. 侧信道特征同样能够提供精细化的检测, 可以实现恶意软件家族的分类识别. MalClassifier[59]是一个恶意软件家族分类系统, 能够自动提取恶意流量网络行为特征的检测系统, 该系统使用连接特征构建网络流的表示序列, 并考虑到恶意软件家族在真实网络流上的语义行为表达, 通过计算流序列相似度建立了恶意流量的“n-flow”特征集, 使用 KNN 或 RF 构建的分类器实现恶意软件家族的分类检测, 该研究同时考虑了单个流网络行为和连续流联合行为的表达. 加密挖矿流量也是较新的恶意流量行为, 相关检测研究已成为网络安全领域最新关注的问题之一, 文献 [60] 研究了加密挖矿流量的侧信道特征, 并对比分析了两组不同粒度的加密流量统计特征集对于分类结果的影响, 一组是使用 Tstat[61]工具提取并筛选的 51 个流量特征, 另一组是 IETF 标准的 8 个特征, 实验结果表明使用较细致的特征表达时能够获得比较高的检测精度和准确率.

侧信道特征属于一种网络通信特征, 具有容易获取、处理速度快等优势, 因此在大规模在线流量检测时, 常选择侧信道特征组成特征集, 在早期关于在线检测的研究中, 文献 [62] 选择了上下行字节、流持续时间及流间隔四元组表示流特征, 并使用 MapReduce 生成查询对象, 然后在检测阶段应用 KNN 完成分类任务, 该研究在近 1.5 亿条的日志中构建了实验数据集. 大规模流量带来的另一个问题就是数据不平衡和粗粒度的信息获取, 这为加密恶意流量的检测研究带来了一定挑战. 针对此问题, 文献 [63] 提出了一种能够在大规模网络流量中检测恶意流量的识别系统 MalAlert, 该系统以 IP 地址对为索引聚合网络流, 从中提取 441 个统计特征, 并使用相对互信息 (RMI) 度量来选择最具代表性的特征形成该聚合流的特征向量, 最后通过 RF 分类器识别恶意流量.

2.2.2 明文特征

相对于侧信道特征, 明文特征属于加密流量的显著特征, 通常可以为分类器提供有效的检测证据[64], 因此, 在加密恶意流量检测研究中, 明文特征通常会作为构建恶意流量特征集的重要组成部分, 基于明文特征的加密恶意流量检测方法成为研究热点[65−68]. 在早期典型的研究中[69], 作者基于 HTTPS 连接请求中获取的主机域名信息, 使用神经语言模型创建了低维表示特征, 并结合主机 IP、端口及时间戳等网络流特征, 构建了流量特征集, 最终选择 LSTM 模型完成恶意流量分类检测. 该方法有对未知恶意流量检测的能力, 但也受限于域名信息库的是否完善. TLS 是恶意流量常用的加密方法, TLS 证书内容也可作为典型的明文特征用于检测. 文献 [70] 从 TLS 证书中提取

特征, 并使用深度神经网络来确定恶意证书, 该方法在识别恶意软件和钓鱼软件证书问题上都有较高的检测准确率. 文章指出攻击者经常使用自签名证书加密流量, 是因为自签名证书生成速度快且免费, 因此在较严格的环境中, 自签名证书有时可以作为黑名单用于过滤恶意流量. Bro(Zeek) 是一个开源的流量分析软件, 在加密恶意流量研究中常被用来提取加密流量的连接、会话和证书等信息, 研究人员以上述信息为基础构建特征, 开展加密恶意流量分类研究. 在文献 [71] 中, 作者详细阐述了 Bro 提取和构建特征的过程, 使用 HTTPS 证书特征和流量链接特征构建特征集, 并提出了一种基于 XGBoost 的恶意软件检测模型. 文献 [72] 提出的 MalDetect 架构, 能够仅使用每条加密流量的前 8 个数据包即可实现恶意流量的检测. MalDetect 通过 Libpcap 捕获数据包并提取了 7 个包特征、8 个协议特征和 8 个证书特征, 同时为了提升模型训练效率并缩短检测响应时间, MalDetect 采用了在线随机森林模型作为分类器. 最新提出的 TLSmell 框架[73]同样是以上述 3 类特征为基础, 使用 Fisher Score、Select K Best 和 RF 算法从 33 个初始特征中选择了 16 个组成特征集, 文献对比了 LSTM、CNN 和 SVM 的检测准确率, 实验结果表明在 MCFP 数据集上, 基于 LSTM 或 CNN 的深度学习算法模型性能表现更好.

包含侧信道特征和明文特征的融合特征, 已成为加密恶意流量被动检测模型特征的常见构成方式, 不同的特征构成也可以为模型提供不同的检测能力, 在文献 [74] 提出的两层检测框架中, 首先仅使用 TLS 握手包中的明文特征对加密流量分类, 完成了恶意流量的二分类识别, 然后从分离出来的恶意流量中再提取上述融合特征实现了恶意软件家族的精细化分类. 文献 [75] 提出的基于密度峰值聚类算法的 3 层采样模型 (THS-IDPC) 同样提取了加密流量的共 36 个特征, 在聚类检测过程中, 算法依次基于链接、会话和证书等 3 类特征对加密流量聚类, 实现了不同分类粒度需求的流量检测; 同时, 作者针对大数据环境下的计算复杂度及数据结构差异等问题提出了基于网格筛选和自定义中心决策值的密度峰值聚类算法, 有效地降低了计算成本. 明文信息特征一般出现在加密网络连接建立的 TLS 会话握手阶段, 必须获取完整的 TLS 握手流量才能提取出有效的明文特征; 而在 TLS 会话恢复情况下, TLS 流量里无法提取有效的明文信息, 且在最新版 TLS 协议中证书也被加密, 因此利用上下文 DNS 信息成为获取域名等相关信息的一个渠道[47].

2.2.3 原始流量特征

深度学习应用的发展为加密恶意流量检测提供了新的方向, 同时计算机算力的提高也使得检测模型能够直接处理更高维度的原始流量, 使用深度学习算法直接从原始加密流量中提取特征并开展检测成为加密恶意流量被动检测的另一个热点研究课题. 文献 [76] 首次提出了使用原始流量数据作为特征, 并将表示学习方法应用于加密恶意流量检测; 基于不同类型的网络流由原始流量字节拼成的灰度图有明显的视觉差异这一现象, 作者提出了使用 CNN 对原始流量生成的图像进行分类, 实现了端到端的加密恶意流量识别; 文献中对比了 4 种原始流量预处理方案, 实验结果表明, 由双向流的会话和数据包的所有层数据构建的特征图像能够为检测提供更多的信息. 文献 [77] 提出了数据包级别的恶意流量样识别框架, 使用词嵌入技术开展数据包特征的表达; 首先将数据包头部信息视为句子, 其中每 2 个字节视为一个词, 使用 Word2Vec[78]构建了数据包头的词向量表达, 然后使用基于 LSTM 的学习模型进行二分类检测; 在多个数据集上实验结果显示该框架具有非常高的检测准确率和精度, 即使训练和验证阶段使用不同的数据集也能获得较好的检测准确率.

同数据包级别的原始流量一样, 流级别的原始流量也能够提取有效的特征表达, 文献 [79] 提出了以深度学习算法为基础的恶意流量检测框架 DeepMAL, 并针对两种输入分别设计了检测方法; 在数据包级别的检测方法中, 使用了前 1024 字节作为输入, 并基于 CNN 和 LSTM 设计了较深层的检测模型; 流级别的方法则从每个流中截取了前 2 个数据包, 每个数据包固定长度为 100 字节, 然后使用较浅层的 CNN 构成检测模型, 文献对比了两种检测方法, 并与浅层 ML 模型做了性能对比. 实验结果表明, 两种基于 DL 的检测方法性能均优于浅层的 ML 模型; 而与数据包级别的 DeepMAL 框架相比, 以原始流作为深度学习模型输入的 DeepMAL 获得了更好的检测效果. 同加密流量分类识别研究一样, 端到端模型的提出让该类型研究更关注模型的构建和输入数据的预处理, 而不再完全依赖专家知识的特征提取, 因此, 近期大多数使用原始流量作为特征的加密恶意流量检测主要研究内容就是, 探索能够充分表达加密流量时间和空间特征的深度学习模型[80−82]. 图 7 展示了原始流量特征的检测流程框架图.

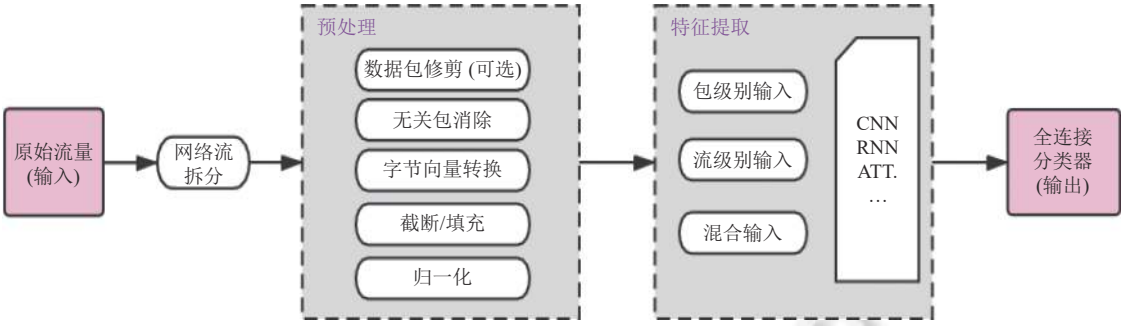


图 7 原始流量特征检测流程框架

2.3 数据集和评价方法

为了能够直观地对比各个检测方法之间的性能, 加密恶意流量被动检测研究通常会在公开数据集上给出实验结果. 由 Stratosphere 实验室提供的 CTU-13 数据集[83]和 MCFP 数据集[84]是最常用的两个数据集, 其中前者有 13 个文件, 对应着不同僵尸网络样本的 13 个场景, 每个场景执行了一个特定的恶意软件, 并使用多种协议执行了不同的操作, 每个场景都使用手动分析并标注了恶意流量、正常流量和背景流量; MCFP 数据集同样也包含并标注了上述 3 种流量, 但区别在于后者是对恶意流量长时间捕获生成, 其主要构成是不同恶意软件生成的 HTTPS 流量, 该流量能被长期捕获并存储. 另外一个常用数据集 USTC-TRC2016 是为了满足端到端检测模型对原始流量特征的需求而建立的[76], 该数据集由两部分组成, 一部分是筛选了 CTU 研究人员从真实网络中获取的 10 种恶意软件流量, 另一部分则是网络流量模拟设备生成的 10 种正常应用流量.

其他数据集还包括: ISCX2012 数据集[85]是生成较早的数据集, 它由 7 个捕获文件组成, 其中由 4 个文件包含了正常流量和 Brute Force SSH、DDoS、HttpDoS 和 Infiltrating 等 4 类恶意流量, 在采用侧信道或原始流量特征的检测模型中, 该数据集可以作为补充使用[77]; CIC-IDS2017 数据集[86]捕获了正常流量和常见攻击流量, 其特点是具有类似真实网络环境中的背景流量, 数据集包含了 Brute Force SSH、Heartbleed、渗透及僵尸网络流量在内的 8 中恶意流量; ISCXVPN2016 数据集[87]由 7 种常规加密流量和 7 种协议封装流量组成, 通常被作为加密流量应用分类数据集使用, 在加密恶意流量分类研究中可以作为二分类识别训练集或良性流量补充使用[80]. 另外, 在新的网络环境下开展研究或者现有数据集不能满足检测任务需求时, 研究人员则需要使用 JOY[88]、Bro(Zeek) 等网络工具自建数据集.

加密恶意流量检测模型通常采用准确率、精度及召回率等指标来进行性能评价, 除此之外, $F1$ 值是综合查准率和召回率的指标, 其指标越高表明模型分类性能越好; 而特异性表达了模型能够准确分类多少个负样本, 通常用于评估模型预测每个可用类别的真实负类的能力; ROC 曲线是分类模型性能的直观表达, 是分类器召回率和特异性的综合指标, 曲线横坐标是特异性, 纵坐标是召回率. 曲线下面积被定义为 AUC, 其值代表了分类器效果, 值越大分类器效果越好.

2.4 小 结

综上所述, 由于各项研究的主要任务或评估方式不一致, 且没有完全使用统一数据集, 因此, 本节将以特征分类和技术特点为切入点, 对各项代表性工作进行对比分析. 侧信道信息一直以来都是加密流量的代表性特征, 在已有的研究成果中, 绝大多数加密恶意流量被动检测方法都使用了侧信道特征. 众多研究成果显示, 侧信道特征是加密恶意流量的最典型特征, 在二分类任务中仅需要少量的侧信道特征便可实现较好的效果, 文献 [54] 使用 5 个特征在 CTU-13-1 等数据集上实现了 99.85% 的准确率, 文献 [52] 使用了 6 个特征在 MCFP 等数据集上实现了 98.5% 的准确率, 代表性特征的选择都是根据不同的机器学习算法学习过程中, 各自模型的特征贡献度排名获得, 不同的机器学习算法模型, 特征的重要性排名不同. 在针对侧信道特征的检测技术中, 通常以传统的机器学习算法为主, 而决策树类算法往往会获得较好的检测准确率, 如 RF、XGBoost 算法等. 明文特征是加密流量中比较特殊的一种, 恶意流量通常采用 SSL/TLS 加密传输, 因此, 在会话握手包中通常可以获取明文的 TLS 信息和证书信息. 现有

研究表明, 在二分类任务中只使用明文信息的检测方法无法获得最优的检测准确率[64], 因此, 明文特征通常被作为识别恶意软件家族的多分类应用场景[69], 或者新型恶意流量检测场景[72]. 目前, 大多数研究是结合了侧信道和明文两种特征, 譬如在文献 [75] 中, 针对不同的流量特征采用聚类算法, 实现了不同粒度级别的加密恶意流量检测分类. 原始流量能够直接作为流量特征使用, 得益于深度学习算法的发展和应用, 能够从复杂的流量中自动学习特征表达. 使用原始流量作为特征的方法主要依赖特征的预处理和模型的构建, 特征预处理过程通常是选择包含丰富流量信息的字节, 如数据包头部信息等[77].

表 1 总结了加密恶意流量被动检测的代表性研究成果, 从检测算法、特征工程、数据集及结果评估等几方面做了对比研究. 传统机器学习和深度学习均可作为检测模型的基础算法, 传统机器学习算法适合轻量、在线检测模型的构建[72]; 而当处理高维度特征时, 基于深度学习的检测模型性能则相对较优[69]. 在特征选择和特征处理方面, 侧信道特征处理的时间开销较小[54], 而明文特征能够提供更精细的加密恶意流量类型检测[75], 侧信道特征和明文特征的结合往往可以获得更高的准确率. 在使用常用公共数据集 (CTU-13 数据集或 MCFP 数据集) 作为评估依据的研究成果中, 以 RF 为代表的传统机器学习算法在混合使用侧信道特征和明文特征时往往会得到较高的检测性能, 在文献 [72] 中, 作者执行恶意流量二分类任务时使用随机森林算法构建模型获得了 99.2% 的召回率, 此时的漏检率仅为 0.09%, 考虑到正负样本的平衡, 这相当于高达 99.55% 的准确率. 在文献 [52] 的研究中, 同样是使用随机森林算法, 作者仅使用了 6 个上述混合特征参与分类便达到了 98.5% 的准确率. 深度学习算法则更适合在使用原始流量为特征的模型中, 文献 [76] 使用的恶意流量数据同样来自该公共数据集, 使用 CNN 构建的多分类检测模型获得了平均 99.41% 的准确率. 考虑到深度学习算法的执行效率, 文献 [89] 将原始流量特征图像维度缩减至 40, 并使用自组织启发式聚类算法加速参数寻优, 提升了约 20% 的学习速度. 在近期的研究中, 文献 [79] 提出的 DeepMAL 模型针对原始流量的预处理方法做了一定改进, 并使用归一化混淆矩阵作为评价指标, 在多分类任务中对比了使用相同输入特征的 RF 算法模型, 结果显示深度学习算法较大的优势.

表 1　加密恶意流量被动检测研究小结

文献	模型/算法	特征类别			特征选择	数据处理	数据集	评估	特点
		侧信道信息	明文信息	原始流量					
[55]	KNN	■	—	—	上/下行字节, 链接持续时长, 请求间隔时长	每特征做11维bins	CISCO-SWA-2014	FP-50: 2%	分类速度快
[64]	L1-LR	■	■	—	流基本信息: 数据包长度及间隔序列(SPLT); TLS握手信息; 载荷字节分布	SPLT组成10×10的状态转移矩阵; TLS握手获得198维特征	沙箱自建(JOY)	准确率: 二分类99.6%; 多分类93.2%	有超耦合的特征
[51]	线性回归; L1/L2-LR; DT; RF; SVM; MLP	■	■	—	22个标准特征(包长/时间及其统计); 319个增强特征 (细化的包长, TLS握手)	增强特征含: 载荷字节状态转移概率、密码套件和扩展的独热编码	企业网环境沙箱自建 (JOY)	含噪音标签准确率89% (RF)	探索噪声和非平稳性对恶意软件识别的影响
[69]	LSTM, RF对比	■	■	—	包长及持续时间等流特征; 域名特征: 域名URL字符特征、N-gram特征和W2V特征	每客户端取10个流一组	自建3个数据集	P-R曲线	恶软家族检测; LSTM较优
[76]	LeNet-5	—	—	■	4种组合方式, 输入前784字节: 流+ALL、流+L7、会话+ALL、会话+L7	USTC-TK2016: 流量拆分、图像生成、IDX转换等	USTC-TRC2016	准确率: 99.41% (平均值)	首次采用表征学习识别恶意流量
[56]	NN, RF, XGBoost	■	—	—	C2S字节数; S2C字节数; TLS通道持续时间; 同对S/C建立通道的时间间隔	一个由4元组表达的TLS通道: 使用直方图和GMM描述样本	自建Proxy日志	60%召回率时, 精度大于90%	研究加密恶意流量的通信模式 (数据不平衡)
[54]	CART, KNN	■	—	—	包大小, 有效载荷大小, 有效载荷率, 当前包与前一个包大小的比率及时间差	无	FIRST 2015; Milicenso; CTU-13-1	检测率: 99.85% (加密恶意流量二分类)	只使用侧信道信息: 特征少、时间开销小

表 1 加密恶意流量被动检测研究小结 (续)

文献	模型/算法	特征类别			特征选择	数据处理	数据集	评估	特点
		侧信道信息	明文信息	原始流量					
[70]	LSTM	—	■	—	基于证书内容的特征: 证书数量、证书发行者和主题字符数等40个特征	特征采用独热编码	Vaderetro2 censys.io	恶软: 94.87% 钓鱼: 88.64%	恶意TLS证书检测
[77]	LSTM+词嵌入模型	—	—	■	数据包头部信息	包头信息每2字节视为一个word, 字典大小为65 536, 应用W2V	ISCX2012; IoT数据集; USTC-TFC2016;	准确率: 97.22% (训练和验证采用不同IoT数据集)	包级别恶意流量分类
[72]	MalDetect: online RF	■	■	—	包特征 (7个)、协议特征 (8个)、证书特征 (8个);	检测到change_cipher_spec时停止抓包	CTU-13; MCFP	新恶意流量: FNR: 3.1%	在线早期检测模型; 可检测恶意流量类型; 可检测新型恶意流量
[52]	SVM, RF, XGBoost	■	■	—	Bro-IDS-logs提取conn.log, ssl.log, x509.log的38个特征	每种算法分别排序提取15个最重要性特征用于检测	CTU-13; MCFP	AUC: 0.9988 (XGBoost)	加密恶意流量的特征分析, 且数据不均衡
[75]	THS-IDPC: SVM, RF, XGBoost	■	■	—	用Bro从Pcap文件中提取conn.log, ssl.log, X509.log. 36个特征分3类: 连接特征; SSL特征; 证书特征	基于网格筛选、自定义中心决策值和互邻近度的密度峰值聚类算法的三阶采样模型 (THS-IDPC)	CTU-13	检测率最高98.3% (XGBoost)	恶意软件家族识别 (8个)
[79]	DeepMAL: CNN, LSTM	—	—	■	原始流量数据包+原始流量网络流	包级别: 取1 024字节截断或填充; 流级别: 每流2包, 每包100字节	CTU-13; USTC-TFC2016	Bi-AUC: 99.8% Rbot: 99.9% Virut: 54.7%	探讨了深度学习算法在恶意流量检测中的应用
[73]	TLSmell: SVM, LSTM, CNN (对比)	■	■	—	Zeek获得conn.log, ssl.log, X509.log, 33个初选特征	使用Fisher Score, Select K Best, RF特征选择. 最终保留16个特征	MCFP	准确率: SVM: 90.37% (最低) LSTM: 93.6% (最高)	在线恶意流量二分类: 小数据集均衡样本
[89]	自组织启发式聚类算法, 2DCNN	—	—	■	原始流量: 会话+ALL	26×26图像, 使用PCA缩减到40维	CTU-13	深度学习速度快20%以上; 精度大于等于98%	蚁群聚类加速CNN恶意流量识别

3 加密恶意流量检测对抗

检测对抗是指恶意流量绕过网络流量审查的规避技术, 流量加密是恶意流量逃避检测的一种方式[90]. 目前, 恶意流量检测对抗技术大多是基于入侵防御系统 (IPS) 的角度, 利用 IPS 技术缺陷或检测盲点等因素开展研究工作[5]. 目前, 尚无针对加密恶意流量检测对抗研究, 但是恶意软件的迭代正不断朝着规避检测的方向发展. 为了能够全面提升加密恶意流量的检测技术, 本节将分类总结现有的加密流量检测对抗技术. 加密恶意流量主动检测是使用 DPI 方法对解密或半解密后的数据包信息进行检测分析, 相关检测对抗的研究工作也主要集中在非加密流量的检测对抗方法[91−93], 因此, 不属于本文讨论的重点. 本节将以被动检测所依赖的特征为角度, 分类总结现有的加密流量检测对抗技术. 依据加密恶意流量被动检测中所述特征工程内容, 通过分析加密恶意流量检测对抗技术的应用范畴和实施方法, 就相应的检测对抗技术做可行性研究.

3.1 流量混淆

流量混淆是指将目标流量置于观测流量集中而达到无法识别的状态, 其目的是隐藏流量特征并规避网络审查. 针对加密恶意流量被动检测技术中侧信道信息的特点, 流量变形或填充等技术均可实现对加密流量的混淆操作.

流量变形[94]是较早提出针对加密流量分析检测的对抗技术, 该技术是从网络层使用数学模型处理加密流量, 使其数据包大小分布最大程度地与预期目标协议流量的相似, 从而逃避网络检测. 传统的流量确定性填充技术往往网络开销较大, 而流量变形技术考虑了传输效率和检测对抗之间的平衡, 使用转换矩阵通过拆分或填充数据包来更改数据包大小. 流量变形需要修改服务端和客户端, 但不需要修改目标应用程序, 因此也不参与原始数据包的生成过程. 由于只改变数据包大小分布, 该算法只适用于以数据包大小作为特征的检测对抗.

侧信道信息的泄露为网络隐私保护提出了挑战[53], 流量变形的对抗方法也无法完全保护网络隐私[95], 而且流量变形方法需要修改服务器和浏览器, 因此也会影响 Web 服务性能. 与流量变形方法不同, HTTPOS[96]是一个用户浏览器端的检测对抗方法, 不需要在服务端部署, 具有可扩展性的优势. HTTPOS 通过修改客户端 HTTP 请求中, 网络流在 TCP 和 HTTP 层上的数据包大小、数据包时间、Web 对象大小和流大小等 4 个基本特征来实现加密流量混淆的操作, 同时 HTTPOS 还能够有选择的混淆流量从而减少对浏览器性能的影响. 实验证明, HTTPOS 可以实现有效的、低开销的加密流量检测对抗. 图 8 总结了基于流量混淆的加密流量检测对抗研究代表性成果.

Xiang 等人在网站指纹的攻击与防御方面充分利用侧信道特征开展多项研究工作, 指出 HTTPOS 只修改了数据包大小特征, 无法防御基于包序列特征的攻击[97], 于是通过对缓冲固定长度 (BUFLO) 方法的优化, 提出了拥塞敏感的 BUFLO (CS-BUFLO) 流量检测防御方法[98]和 Tamaraw 防御框架[99]; 同时, 该团队提出的 GloVe[100]方法是将网络流量的聚类中心变形, 使其与其他网站无法区分, 作者并就减少带宽开销给出了信息论的解释. 同 GloVe 一样, Tao 等人提出了 Supersequences[101]方法, 采用聚类变形的方法将网络流量智能地分组到相似的集合中. WTF-PAD 方法[102]是一个轻量级的防御方法, 该方法特点是将从其他类的流量特征中采样的伪突发特征插入到原始类的流量特征中, 实现了原始流量的混淆. 在文献 [103] 中, 作者提出的 Walkie-Talkie (W-T) 防御模型是浏览器采用半双工的模式与服务器通信, 同时还添加了虚拟包和延迟信息使得流量之间的特征基本相同, 该模型能够防御包含时间特征和包序列特征的指纹攻击, 具有较小的网络带宽开销; 同样是该作者参与的工作, 在文献 [104] 中提出了两种轻量的防御模型 FRONT 和 GLUE, 其中 FRONT 模型使用随机化的虚拟数据包混淆流量前端, 防止攻击者分析提取流量头部特征, 而 GLUE 模型是在不同流量间添加虚拟包, 使攻击者无法分辨流量的起始和结束位置, 两种方案均没有延迟开销. 自相关属性是流量的重要特征之一, 文献 [105] 提出了控制流量自相关属性的网站指纹防御模型, 该模型消除了固定数据包长度及序列等特征, 并以最小的虚拟包插入流量获得较小的开销, 同时, 对于加载的每个网站能够随机更改数据包长度. 针对只使用时间相关信息的网站指纹攻击[106], 作者在文献 [107] 中提出了一种低开销的隧道模型, 通过模拟数据包序列使加密流量在时间特征上不可分. 由上述研究看出, 网站指纹攻击防御模型大多基于数据包大小、时间及分布等侧信道特征, 通过链路填充、数据包延迟等操作实现流量混淆[108−113], 因此可以为加密恶意流量被动检测的对抗研究提供借鉴.

作为加密流量的典型特征, 针对侧信道特征的攻击防御模型在物联网环境下也有较多研究. 数据包长度是侧信道特征的常用信息, 文献 [114] 中, 作者提出了一种轻量级的数据包填充机制, 通过在二层流量中插入随机选择大小的数据包, 插入随机数后的数据包大小不超过最大传输单元 (MTU), 因此该方法需要部署在网络边缘设备并获取局域网内设备流量. 文献 [115] 分析了物联网网关与云服务器之间的加密网络流量, 证明了使用 DNS 查询、连接信息和加密流量的统计信息等特征可以实现物联网设备及其活动状态的识别; 作者同时提出了一种基于流量变形技术的检测对抗方法, 通过插入与真实流量特征相似的虚拟流量, 降低攻击者对真实流量的检测率. 检测对抗和通信开销之间的平衡是该研究的关注点之一, 文献 [116] 提出了自适应网络流量状态的数据包动态填充方案, 由 SDN 网络控制填充数据的大小, 流量小填充多保证隐私, 流量大填充少减小开销, 该方案设计了四级填充策略用以动态平衡资源开销和检测率的关系; 作者选择了 4 种分类算法开展了对抗检测实验, 实验结果证明了该方案的可行性. 文献 [117] 通过评估 22 种现有检测方案, 提出了一种基于生成欺骗流量来隐藏目标物联网设备真实状态的检测对抗方案. 差分隐私[118]被作为一种衡量隐私风险的框架, 用于加密流量检测对抗的研究中, 文献 [119] 应用了较严格的本地差分隐私 (LDP) 模型, 基于数据包大小及其概率分布提出了流量混淆方案, 该文献同时给出了如何选择最佳数据包大小的方法及依据; 同样, 文献 [120] 针对智能家居的流量检测对抗, 提出了一种基于社区整体流量的差分隐私框架, 该框架首先使用效用感知差分隐私机制选择互联网流量网关, 然后通过在多个协作的智能家居网关之间发送流量实现流量混淆, 该框架同时使用了一种差分私有机制以线性优化的方式降低了互联网资源开销.

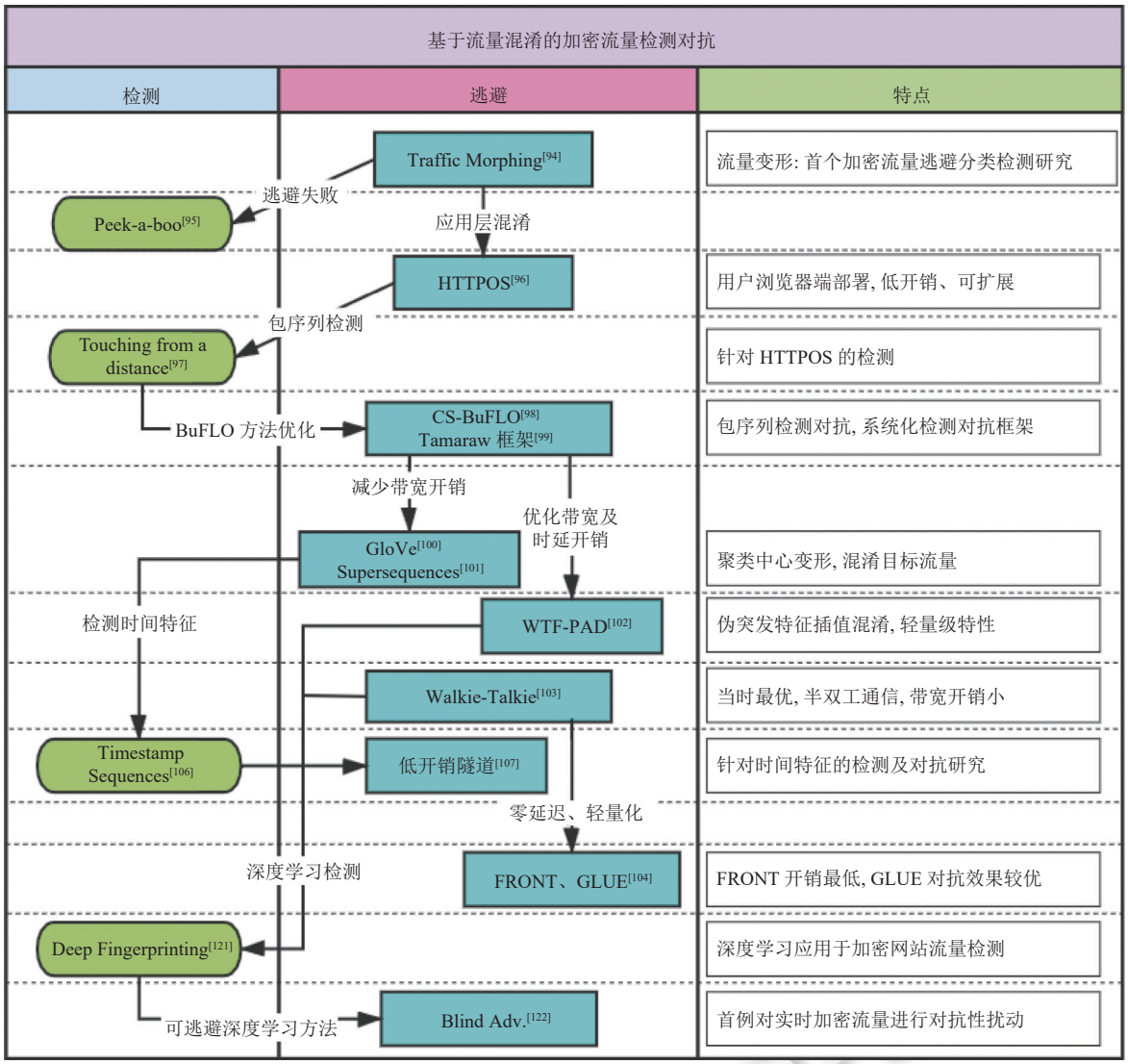


图 8　基于流量混淆的加密流量检测对抗研究

综上所述, 流量混淆技术是加密流量检测对抗、网站指纹检测对抗和物联网隐私保护等研究内容的主要解决方案, 流量填充、变形和数据包延迟是流量混淆的基本操作模式, 其目的是通过改变加密流量的侧信道信息使基于此类特征的分类器无法区分流量类别. 同样, 作为加密恶意流量检测所依赖的重要信息, 侧信道特征也应值得加密恶意流量检测对抗研究的关注.

3.2　深度学习算法对抗

近年来, 随着人工智能技术的不断发展, 基于深度学习的加密流量检测研究取得了一定的研究成果, 同样也对加密流量防御模型的鲁棒性提出了新的考验. 文献 [121] 提出了 deep fingerprinting (DF) 是当时最优的指纹识别模型, 在针对 WTF-PAD 和 W-T 等防御模型的攻击中达到了较好的效果; 文献 [122] 针对 DF 模型开展了实时加密流量的对抗性扰动研究; 针对物联网环境下变形的加密流量, 同样使用对抗性的深度学习方法也取得了较好的成果[123]. 包括深度学习算法在内大多数人工智能技术通常具有一定的线性性质, 这种性质可以使优化后的网络具有一定的预测能力[124]. 然而, 正是这种线性性质使得通过精心构造的对抗样本可以轻易地欺骗分类模型. 有研究指出大多

数机器学习算法都会受到对抗样本的影响[125], 对抗样本的存在是人工智能技术的固有弱点[126]. 随着深度学习对抗技术的不断发展, 在加密流量检测领域中, 主要围绕有效构建对抗样本以逃避检测和使用对抗样本提升检测模型性能两方面开展相关研究工作, 有以下代表性研究成果.

文献 [127] 探讨了对抗性机器学习算法和差分隐私在流量分析领域中技术应用的适应性, 作者首先提出了快速梯度符号方法 (FGSM) 生成对抗样本来混淆 CNN 分类器, 并成功降低了流量分类的准确率, 但是实验表明该对抗技术的鲁棒性不好, 无法抵抗其他分类器或使用对抗样本训练的攻击方法; 另外, 作者还提出采用差分隐私技术实现了流量分析对抗. 文献以流媒体流量分析对抗为例, 初步探索了对抗样本技术在流量分析中的应用.

随着深度学习技术的发展, 参照目标流量特征自动化生成伪装流量的研究已有多项研究成果. 文献 [128] 首次提出了使用生成对抗网络 (GAN) 动态生成伪装流量的方法, 所提出的 flowGAN 技术能够自动学习目标流量特征, 使用 GAN 模型生成变形流量, 并引入了不可区分性概念, 来衡量目标流量和变形流量的不可区分性; 文献 [129] 同样采用了 GAN 模型生成伪装流量, 称为 GAN 隧道, 而隐私保护流量则被封装在 GAN 隧道中, 从而实现加密流量的检测对抗.

文献 [130] 以网站指纹识别对抗为例提出了 WF-GAN 方法, 通过自动学习并生成对抗样本实现指纹识别防御. WF-GAN 使用基于流量突发的特征作为生成器模型的原始输入, 并使用梯度信息来优化对抗样本, 该方法实现了有目标和无目标两种防御模型. 实验证明, 该方法使用 5%–15% 的开销获得了 90% 的对抗成功率, 优于 W-T 模型.

文献 [131] 同样是在探讨网站指纹识别防御问题中, 提出了基于对抗样本的思想提出了 Mockingbird 算法. 作者首先证明了将对抗样本简单直接地映射到网络流量的做法是不合适的, 因为攻击者可以根据对抗样本的相关信息设计分类器, 并实现指纹的成功识别[132]. Mockingbird 算法的最终目标是生成非目标防御的混淆流量, 但是其训练过程则是生成有目标的对抗样本. 与 FGSM 不同, 该算法不关注检测器的损失函数, 而是随机搜索一个目标流量, 并逐渐减小与目标流量之间的距离, 因此, 产生的对抗样本是随机的, 而不是始终遵循相同的生成过程, 使算法具有一定的鲁棒性.

文献 [122] 首次提出了针对实时网络流量生成对抗样本的方法, 该方法能够应用于各种类型的流量分类器. 作者以防御基于深度神经网络的检测技术为例, 从两个方面考虑了算法的设计思想, 首先该方法是针对实时的未知流量, 无法获得该流量的相关先验知识, 因此算法的关键就在于如何优化独立于输入流量的“盲”扰动; 其次, 加入了对抗扰动的网络流量应遵循网络知识的约束, 譬如流量特征之间的关系等, 因此设计了映射函数和正则化器来满足此约束条件. 算法通过改变数据包大小、信息或插入虚拟数据包的方式生成对抗样本. 作者在 Tor 网络上开发了 BLANKET 以验证所提出的流量检测对抗算法, 实验表明, 该算法针对基于深度学习的 DF 等指纹检测方法具有较好的防御力, 并且相对于之前的对抗算法更具有鲁棒性.

目前, 加密恶意流量被动检测工作大多是基于机器学习或深度学习的人工智能方法. 该类方法的识别检测效果严重依赖于样本的数量和质量, 且简单的分类器模型往往会受到混淆或变形流量的欺骗, 因此, 基于人工智能技术固有弱点, 使用精心构造的对抗样本可有效防御加密恶意流量的检测. 而且, 随着深度学习技术的发展, 自动化的生成对抗样本和构建更鲁棒的防御模型将会成为未来研究的方向.

3.3 其他对抗方法

如前文所述, 明文特征是指在加密流量中能分离出来的明文及其统计信息, 包括未加密的 TLS 握手信息、由上下文推断出的 DNS 信息和 HTTP 头等信息. 我们推断, 如果能实现上述信息的加密或隐藏, 则也能有效对抗检测工作, 从而实现更严格的隐私保护. 包括 TLS1.2 及之前的加密协议版本, 服务器证书均在握手信息中以明文形式传输. 然而, 为了提升 TLS 协议的安全性和性能, IETF 于 2018 年发布了命名为 TLS1.3 的新版本协议[133], 该协议改变了加密策略和握手方式, 有效减少了会话握手数据包中明文信息的泄露. 另外, 谷歌公司设计的 QUIC[134]协议也于 2021 年由 IETF 作为标准化版本发布, 该协议不但提升了 Web 应用程序的性能, 还通过修改客户端与服务端的会话握手方式, 增强了会话信息的安全性. 因此, 过去使用握手明文信息的加密恶意流量检测方法会受到不同程度影响, 这也为加密恶意流量逃避检测提供了新的网络通道.

多路径路由是一种允许计算机网络通信通过多个备用路径发送的技术. 随着各种多路径路由支持协议的出现, 越来越多的终端设备和应用服务都开始支持多路径访问, 同时由于无法在一条路由上获取完整的流量, 也给网络流量分析带来了一定的困难. 文献 [135] 研究了多宿主设备和多路径协议对网站指纹攻击的影响, 首先设计了一种多路径调度器 (HyWF) 来防止指纹检测, 在不增加网络开销的前提下实现了良好的防御效果; 然后, 作者尝试将 HyWF 分别与其他防御模型结合, 进一步提高了网络流量的隐私性, 显著降低了指纹攻击的准确性.

4 总结与展望

加密恶意流量检测是当前网络安全领域最具挑战性的问题之一, 主动检测的本质是对加密流量开展深度包检测, 隐私安全和检测效率是关注的焦点, 而被动检测方法凭借着良好的性能和可实施性已经成为加密恶意流量检测领域内最有潜力的方法之一. 本文从可明文检测技术和可搜索加密技术两方面总结了主动检测的研究现状, 并详细综述了基于人工智能技术的代表性研究成果. 作为加密流量研究的一个重要分支, 加密恶意流量被动检测在继承上述人工智能技术基础上, 针对其自身特征开展了相对独立的研究方法. 本文第 2 节以特征工程为分类角度, 详细综述了人工智能技术结合流量特征在加密恶意流量被动检测研究中的应用实现与研究成果. 最后, 从网络流量混淆技术入手, 结合加密恶意流量被动检测所依赖的特征, 围绕加密流量分类识别的研究历程开展了系统性综述.

从相关研究可以看出, 基于人工智能的加密恶意流量被动检测研究近几年发展迅速, 取得了一定的研究成果, 也成为网络安全领域的研究热点之一. 随着计算机网络技术的发展和网络安全形势的变化, 加密恶意流量检测研究也将面临一定的机遇和挑战. 我们认为, 一方面计算机硬件性能的提升加速了算法的检测效率, 但是伴随着互联网流量的增加和网络安全集中管理的需求, 高速、大容量网络环境下加密恶意流量实时检测的研究将是一个重点方向; 另一方面 QUIC/HTTP3 等新型协议的出现, 使得加密协议进一步强化, 在提升了用户隐私保护的同时, 也增加了加密恶意流量被动检测技术实现的难度, 充分分析新型加密网络协议, 探索新型加密网络协议环境下恶意流量的检测方法也是该领域的研究趋势之一; 另外随着联网智能设备等新型网络环境的推广, 恶意流量特征也在发生着变化, 及时分析协议特征、尽快构建真实标注的样本集是检测研究的前提, 也是加密恶意流量检测的研究专题之一; 最后, 随着技术的发展, 恶意软件在不断快速更新, 恶意软件家族不断扩大, 良好的检测模型需要有能力主动发现新型的恶意流量, 因此, 如何构建自适应的敏捷检测模型也是加密恶意流量检测的研究热点之一.

References:

[1] 2020 Annual Report. 2021 (in Chinese). https://www.cert.org.cn/publish/main/upload/File/2020%20Annual%20Report.pdf

[2] Zscaler. 2020 State of Encrypted Attacks. 2020. https://www.zscaler.com/resources/industry-reports/state-of-encrypted-attacks-summary-report.pdf

[3] Wagner D, Schneier B. Analysis of the SSL 3.0 protocol. In: Proc. of the 2nd Conf. on USENIX Workshop on Electronic Commerce. Oakland: USENIX Association, 1996. 4.

[4] Poh GS, Divakaran DM, Lim HW, Ning JT, Desai A. A survey of privacy-preserving techniques for encrypted traffic inspection over network middleboxes. arXiv:2101.04338, 2021.

[5] Cheng TH, Lin YD, Lai YC, Lin PC. Evasion techniques: Sneaking through your intrusion detection/prevention systems. IEEE Communications Surveys & Tutorials, 2012, 14(4): 1011–1020. [doi: 10.1109/SURV.2011.092311.00082]

[6] Corona I, Giacinto G, Roli F. Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues. Information Sciences, 2013, 239: 201–225. [doi: 10.1016/j.ins.2013.03.022]

[7] Carpenter B, Brim S. Middleboxes: Taxonomy and issues. RFC, 2002. [doi: 10.17487/RFC3234]

[8] Jarmoc J. SSL/TLS interception proxies and transitive trust. In: Black Hat Europe, 2012.

[9] Huang LS, Rice A, Ellingsen E, Jackson C. Analyzing forged SSL certificates in the wild. In: Proc. of the 2014 IEEE Symp. on Security and Privacy. Berkeley: IEEE, 2014. 83–97. [doi: 10.1109/SP.2014.13]

[10] Aldo Cortesi, Maximilian Hils, Thomas Kriechbaumer, and contributors. mitmproxy: A free and open source interactive HTTPS proxy [Version 7.0]. 2010. https://mitmproxy.org/

[11] Broadcom. Manage encrypted traffic with SSL visibility appliance. 2020. https://www.broadcom.com/products/cyber-security/network/encrypted-traffic-management

[12] de Carné de Carnavalet X, Mannan M. Killed by Proxy: Analyzing client-end TLS interce. In: Proc. of the 23rd Annual Network and Distributed Systems Security Symp. San Diego: NDSS, 2016.

[13] Durumeric Z, Ma Z, Springall D, Barnes R, Sullivan N, Bursztein E, Bailey M, Halderman JA, Paxson V. The security impact of HTTPS interception. In: Proc. of the 24th Annual Network and Distributed Systems Security Symp. San Diego: NDSS, 2017.

[14] Han J, Kim S, Ha J, Han DS. SGX-Box: Enabling visibility on encrypted traffic using a secure middlebox module. In: Proc. of the 1st Asia-Pacific Workshop on Networking. Hong Kong: ACM, 2017. 99–105. [doi: 10.1145/3106989.3106994]

[15] Sherry J, Hasan S, Scott C, Krishnamurthy A, Ratnasamy S, Sekar V. Making middleboxes someone else's problem: Network processing as a cloud service. ACM SIGCOMM Computer Communication Review, 2012, 42(4): 13–24. [doi: 10.1145/2377677.2377680]

[16] Trach B, Krohmer A, Gregor F, Arnautov S, Bhatotia P, Fetzer C. ShieldBox: Secure middleboxes using shielded execution. In: Proc. of the 2018 Symp. on SDN Research. Los Angeles: ACM, 2018. 2. [doi: 10.1145/3185467.3185469]

[17] Goltzsche D, Rüsch S, Nieke M, Vaucher S, Weichbrodt N, Schiavoni V, Aublin PL, Cosa P, Fetzer C, Felber P, Pietzuch P, Kapitza R. EndBox: Scalable middlebox functions using client-side trusted execution. In: Proc. of the 48th Annual IEEE/IFIP Int'l Conf. on Dependable Systems and Networks (DSN). Luxembourg: IEEE, 2018. 386–397. [doi: 10.1109/DSN.2018.00048]

[18] Naylor D, Schomp K, Varvello M, Leontiadis I, Blackburn J, López DR, Papagiannaki K, Rodriguez PR, Steenkiste P. Multi-Context TLS (mcTLS): Enabling secure in-network functionality in TLS. ACM SIGCOMM Computer Communication Review, 2015, 45(4): 199–212. [doi: 10.1145/2829988.2787482]

[19] ETSI. Middlebox Security Protocol—Part 2: Transport layer MSP, profile for fine grained access control. ETSI TS 103 523-2 V0.1.0. 2019, https://portal.etsi.org/webapp/workProgram/Report_WorkItem.asp?wki_id=52930

[20] Naylor D, Li R, Gkantsidis C, Karagiannis T, Steenkiste P. And then there were more: Secure communication for more than two parties. In: Proc. of the 13th Int'l Conf. on Emerging Networking Experiments and Technologies. Incheon: ACM, 2017. 88–100. [doi: 10.1145/3143361.3143383]

[21] Lee H, Smith Z, Lim J, Choi G, Chun S, Chung T, Kwon T. maTLS: How to make TLS middlebox-aware? In: Proc. of the 2019 Network and Distributed Systems Security (NDSS) Symp. San Diego: NDSS, 2019. [doi: 10.14722/ndss.2019.23547]

[22] Li J, Chen RM, Su JS, Huang XY, Wang XF. ME-TLS: Middlebox-enhanced TLS for Internet-of-Things devices. IEEE Internet of Things Journal, 2020, 7(2): 1216–1229. [doi: 10.1109/JIOT.2019.2953715]

[23] Zeng Y, Wu ZY, Dong LH, Liu ZH, Ma JF, Li Z. Research on malicious traffic identification technology in encrypted traffic. Journal of Xidian University, 2021, 48(3): 170–187 (in Chinese with English abstract). [doi: 10.19665/j.issn1001-2400.2021.03.022]

[24] Justine S, Lan C, Popa RA, Ratnasamy S. BlindBox: Deep packet inspection over encrypted traffic. ACM SIGCOMM Computer Communication Review, 2015, 45(4): 213–226. [doi: 10.1145/2829988.2787502]

[25] Lan C, Sherry J, Popa RA, Ratnasamy S, Liu Z. Embark: Securely outsourcing middleboxes to the cloud. In: Proc. of the 13th USENIX Conf. on Networked Systems Design and Implementation. Santa Clara: USENIX Association, 2016. 255–273.

[26] Yuan XL, Wang XY, Lin JX, Wang C. Privacy-preserving deep packet inspection in outsourced middleboxes. In: Proc. of the 35th Annual IEEE Int'l Conf. on Computer Communications. San Francisco: IEEE, 2016. 1–9. [doi: 10.1109/INFOCOM.2016.7524526]

[27] Asghar HJ, Melis L, Soldani C, De Cristofaro E, Kaafar MA, Mathy L. SplitBox: Toward efficient private network function virtualization. In: Proc. of the 2016 Workshop on Hot Topics in Middleboxes and Network Function Virtualization. Florianopolis: ACM, 2016. 7–13. [doi: 10.1145/2940147.2940150]

[28] Fan JY, Guan CW, Ren K, Cui Y, Qiao CM. SPABox: Safeguarding privacy during deep packet inspection at a middlebox. IEEE/ACM Trans. on Networking, 2017, 25(6): 3753–3766. [doi: 10.1109/TNET.2017.2753044]

[29] Canard S, Diop A, Kheir N, Paindavoine M, Sabt M. BlindIDS: Market-compliant and privacy-friendly intrusion detection system over encrypted traffic. In: Proc. of the 2017 ACM on Asia Conf. on Computer and Communications Security. Abu Dhabi: ACM, 2017. 561–574. [doi: 10.1145/3052973.3053013]

[30] Ren H, Litt H, Liu DX, Shen XS. Toward efficient and secure deep packet inspection for outsourced middlebox. In: Proc. of the 2019 IEEE Int'l Conf. on Communications (ICC). Shanghai: IEEE, 2019. 1–6. [doi: 10.1109/ICC.2019.8761954]

[31] Ning JT, Poh GS, Loh JC, Chia J, Chang EC. PrivDPI: Privacy-preserving encrypted traffic inspection with reusable obfuscated rules. In: Proc. of the 2019 ACM SIGSAC Conf. on Computer and Communications Security. London: ACM, 2019. 1657–1670. [doi: 10.1145/3319535.3354204]

[32] Ning JT, Huang XY, Poh GS, Xu SM, Loh JC, Weng J, Deng RH. Pine: Enabling privacy-preserving deep packet inspection on TLS with rule-hiding and fast connection establishment. In: Proc. of the 25th European Symp. on Research in Computer Security. Guildford: Springer, 2020. 3–22. [doi: 10.1007/978-3-030-58951-6_1]

[33] Zhu YC, Zheng Y. Retracted article: Traffic identification and traffic analysis based on support vector machine. Neural Computing & Applications, 2020, 32(7): 1903–1911. [doi: 10.1007/s00521-019-04493-2]

[34] Yao ZJ, Ge JG, Wu YL, Lin XS, He RK, Ma YX. Encrypted traffic classification based on Gaussian mixture models and hidden Markov models. Journal of Network and Computer Applications, 2020, 166: 102711. [doi: 10.1016/j.jnca.2020.102711]

[35] Shen M, Liu YT, Zhu LH, Du XJ, Hu JK. Fine-grained webpage fingerprinting using only packet length information of encrypted traffic. IEEE Trans. on Information Forensics and Security, 2021, 16: 2046–2059. [doi: 10.1109/TIFS.2020.3046876]

[36] Liu C, He LT, Xiong G, Cao ZG, Li Z. FS-Net: A flow sequence network for encrypted traffic classification. In: Proc. of the 2019 IEEE Conf. on Computer Communications. Paris: IEEE, 2019. 1171–1179. [doi: 10.1109/INFOCOM.2019.8737507]

[37] Gu GF, Zhang JJ, Lee WK. BotSniffer: Detecting botnet command and control channels in network traffic. In: Proc. of the 15th Annual Network and Distributed System Security Symp. San Diego: NDSS, 2008.

[38] Gu GF, Perdisci R, Zhang JJ, Lee WK. BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. In: Proc. of the 17th Conf. on Security Symp. San Jose: USENIX Association, 2008. 139–154.

[39] Boukhtouta A, Lakhdari NE, Mokhov SA, Debbabi M. Towards fingerprinting malicious traffic. Procedia Computer Science, 2013, 19: 548–555. [doi: 10.1016/j.procs.2013.06.073]

[40] Alshammari R, Zincir-Heywood AN. A flow based approach for SSH traffic detection. In: Proc. of the 2007 IEEE Int'l Conf. on Systems, Man and Cybernetics (ISIC). Montreal: IEEE, 2007. 296–301. [doi: 10.1109/ICSMC.2007.4414006]

[41] Alshammari R, Zincir-Heywood AN. Investigating two different approaches for encrypted traffic classification. In: Proc. of the 6th Annual Conf. on Privacy, Security and Trust. Fredericton: IEEE, 2008. 156–166. [doi: 10.1109/PST.2008.15]

[42] Alshammari R, Zincir-Heywood AN, Farrag A. Performance comparison of four rule sets: An example for encrypted traffic classification. In: Proc. of the 2009 World Congress on Privacy, Security, Trust and the Management of e-Business. St. John's: IEEE, 2009. 21–28. [doi: 10.1109/CONGRESS.2009.22]

[43] Alshammari R, Zincir-Heywood AN. Generalization of signatures for SSH encrypted traffic identification. In: Proc. of the 2009 IEEE Symp. on Computational Intelligence in Cyber Security. Nashville: IEEE, 2009. 167–174. [doi: 10.1109/CICYBS.2009.4925105]

[44] Alshammari R, Zincir-Heywood AN. Machine learning based encrypted traffic classification: Identifying SSH and Skype. In: Proc. of the 2009 IEEE Symp. on Computational Intelligence for Security and Defense Applications. Ottawa: IEEE, 2009. 1–8. [doi: 10.1109/CISDA.2009.5356534]

[45] Alshammari R, Zincir-Heywood AN. Can encrypted traffic be identified without port numbers, IP addresses and payload inspection? Computer Networks, 2011, 55(6): 1326–1350. [doi: 10.1016/j.comnet.2010.12.002]

[46] Boukhtouta A, Mokhov SA, Lakhdari NE, Debbabi M, Paquet J. Network malware classification comparison using DPI and flow packet headers. Journal of Computer Virology and Hacking Techniques, 2016, 12(2): 69–100. [doi: 10.1007/s11416-015-0247-x]

[47] Anderson B, McGrew D. Identifying encrypted malware traffic with contextual flow data. In: Proc. of the 2016 ACM Workshop on Artificial Intelligence and Security. Vienna: ACM, 2016. 35–46. [doi: 10.1145/2996758.2996768]

[48] McGrew D, Anderson B. Enhanced telemetry for encrypted threat analytics. In: Proc. of the 24th IEEE Int'l Conf. on Network Protocols. Singapore: IEEE, 2016. 1–6. [doi: 10.1109/ICNP.2016.7785325]

[49] Bilge L, Balzarotti D, Robertson W, Kirda E, Kruegel C. Disclosure: Detecting botnet command and control servers through large-scale NetFlow analysis. In: Proc. of the 28th Annual Computer Security Applications Conf. Orlando: ACM, 2012. 129–138. [doi: 10.1145/2420950.2420969]

[50] Williams N, Zander S, Armitage G. A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. ACM SIGCOMM Computer Communication Review, 2006, 36(5): 5–16. [doi: 10.1145/1163593.1163596]

[51] Anderson B, McGrew D. Machine learning for encrypted malware traffic classification: Accounting for noisy labels and non-stationarity. In: Proc. of the 23rd ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Halifax: ACM, 2017. 1723–1732. [doi: 10.1145/3097983.3098163]

[52] Shekhawat AS, Troia FD, Stamp M. Feature analysis of encrypted malicious traffic. Expert Systems With Applications, 2019, 125: 130–141. [doi: 10.1016/j.eswa.2019.01.064]

[53] Chen S, Wang R, Wang XF, Zhang KH. Side-channel leaks in web applications: A reality today, a challenge tomorrow. In: Proc. of the 2010 IEEE Symp. on Security and Privacy. Oakland: IEEE, 2010. 191–206. [doi: 10.1109/SP.2010.20]

[54] Stergiopoulos G, Talavari A, Bitsikas E, Gritzalis D. Automatic detection of various malicious traffic using side channel features on TCP packets. In: Proc. of the 23rd European Symp. on Research in Computer Security. Barcelona: Springer, 2018. 346–362. [doi: 10.1007/978-3-319-99073-6_17]

[55] Lokoč J, Kohout J, Čech P, Skopal T, Pevný T. k-NN classification of malware in HTTPS traffic using the metric space approach. In:

Proc. of the 11th Pacific Asia Workshop. Auckland: Springer, 2016. 131–145. [doi: 10.1007/978-3-319-31863-9_10]

[56] Kohout J, Komárek T, Čech P, Bodnár J, Lokoč J. Learning communication patterns for malware discovery in HTTPs data. Expert Systems with Applications, 2018, 101: 129–142. [doi: 10.1016/j.eswa.2018.02.010]

[57] Liu JY, Tian ZY, Zheng RF, Liu L. A distance-based method for building an encrypted malware traffic identification framework. IEEE Access, 2019(7): 100014–100028. [doi: 10.1109/ACCESS.2019.2930717]

[58] Ankerst M, Breunig MM, Kriegel HP, Sander J. OPTICS: Ordering points to identify the clustering structure. ACM SIGMOD Record, 1999, 28(2): 49–60. [doi: 10.1145/304181.304187]

[59] AlAhmadi BA, Martinovic I. MalClassifier: Malware family classification using network flow sequence behaviour. In: Proc. of the 2018 APWG Symp. on Electronic Crime Research (eCrime). San Diego: IEEE, 2018. 1–13. [doi: 10.1109/ECRIME.2018.8376209]

[60] Pastor A, Mozo A, Vakaruk S, Canavese D, López DR, Regano L, Gómez-Canaval S, Lioy A. Detection of encrypted cryptomining malware connections with machine and deep learning. IEEE Access, 2020(8): 158036–158055. [doi: 10.1109/ACCESS.2020.3019658]

[61] Finamore A, Mellia M, Meo M, Munafo MM, Torino PD, Rossi D. Experiences of internet traffic monitoring with tstat. IEEE Network, 2011, 25(3): 8–14. [doi: 10.1109/MNET.2011.5772055]

[62] Čech P, Kohout J, Lokoč J, Komárek T, Maroušek J, Pevný T. Feature extraction and malware detection on large HTTPS data using mapreduce. In: Proc. of the 9th Int'l Conf. on Similarity Search and Applications. Tokyo: Springer, 2016. 311–324. [doi: 10.1007/978-3-319-46759-7_24]

[63] Piskozub M, Spolaor R, Martinovic I. MalAlert: Detecting malware in large-scale network traffic using statistical features. ACM SIGMETRICS Performance Evaluation Review, 2018, 46(3): 151–154. [doi: 10.1145/3308897.3308961]

[64] Anderson B, Paul S, McGrew D. Deciphering malware's use of TLS (without decryption). Journal of Computer Virology and Hacking Techniques, 2018, 14(3): 195–211. [doi: 10.1007/s11416-017-0306-6]

[65] Schoinianakis D, Götze N, Lehmann G. MDiET: Malware detection in encrypted traffic. In: Proc. of the 6th Int'l Symp. for ICS & SCADA Cyber Security Research 2019 (ICS-CSR). 2019.

[66] Chao DC. A fingerprint enhancement and second-order Markov chain based malicious encrypted traffic identification scheme. In: Proc. of the 6th Int'l Conf. on Computing and Artificial Intelligence. Tianjin: ACM, 2020. 328–333. [doi: 10.1145/3404555.3404590]

[67] Chao DC. A mining policy based malicious encrypted traffic detection scheme. In: Proc. of the 9th Int'l Conf. on Computing and Pattern Recognition. Xiamen: ACM, 2020. 130–135. [doi: 10.1145/3436369.3436479]

[68] Zheng RF, Liu JY, Li K, Liao S, Liu L. Detecting malicious TLS network traffic based on communication channel features. In: Proc. of the 8th IEEE Int'l Conf. on Information, Communication and Networks (ICICN). Xi'an: IEEE, 2020. 14–19. [doi: 10.1109/ICICN51133.2020.9205087]

[69] Prasse P, Machlica L, Pevný T, Havelka J, Scheffer T. Malware detection by analysing encrypted network traffic with neural networks. In: Proc. of the 2017 European Conf. on Machine Learning and Knowledge Discovery in Databases. Skopje: Springer, 2017. 73–88. [doi: 10.1007/978-3-319-71246-8_5]

[70] Torroledo I, Camacho LD, Bahnsen AC. Hunting malicious TLS certificates with deep neural networks. In: Proc. of the 11th ACM Workshop on Artificial Intelligence and Security. Toronto: ACM, 2018. 64–73. [doi: 10.1145/3270101.3270105]

[71] Shah J. Detection of malicious encrypted Web traffic using machine learning [MS. Thesis]. Victoria: University of Victoria, 2018.

[72] Liu JY, Zeng YZ, Shi JY, Yang YX, Wang R, He LZ. MalDetect: A structure of encrypted malware traffic detection. Computers, Materials & Continua, 2019, 60(2): 721–739. [doi: 10.32604/cmc.2019.05610]

[73] Weng ZQ, Chen TM, Zhu TT, Dong H, Zhou D, Alfarraj O. TLSmell: Direct identification on malicious HTTPs encryption traffic with simple connection-specific indicators. Computer Systems Science and Engineering, 2021, 37(1): 105–119. [doi: 10.32604/csse.2021.015074]

[74] Zheng RF, Liu JY, Liu L, Liao S, Li K, Wei JH, Li L, Tian ZY. Two-layer detection framework with a high accuracy and efficiency for a malware family over the TLS protocol. PLoS ONE, 2020, 15(5): e0232696. [doi: 10.1371/journal.pone.0232696]

[75] Chen LC, Gao S, Liu BX, Lu ZG, Jiang ZW. THS-IDPC: A three-stage hierarchical sampling method based on improved density peaks clustering algorithm for encrypted malicious traffic detection. The Journal of Supercomputing, 2020, 76(9): 7489–7518. [doi: 10.1007/s11227-020-03372-1]

[76] Wang W, Zhu M, Zeng XW, Ye XZ, Sheng YQ. Malware traffic classification using convolutional neural network for representation learning. In: Proc. of the 2017 Int'l Conf. on Information Networking (ICOIN). Da Nang: IEEE, 2017. 712–717 [doi: 10.1109/ICOIN.2017.7899588]

[77] Hwang RH, Peng MC, Nguyen VL, Chang YL. An LSTM-based deep learning approach for classifying malicious traffic at the packet level. Applied Sciences, 2019, 9(16): 3414. [doi: 10.3390/app9163414]

[78] Goldberg Y, Levy O. Word2Vec explained: Deriving Mikolov *et al.*'s negative-sampling word-embedding method. arXiv:1402.3722, 2014.

[79] Marín G, Casas P, Capdehourat G. DeepMAL—Deep learning models for malware traffic detection and classification. arXiv:2003.04079, 2020.

[80] Wang B, Su Y, Zhang MS, Nie JK. A deep hierarchical network for packet-level malicious traffic detection. IEEE Access, 2020, 8: 201728–201740. [doi: 10.1109/ACCESS.2020.3035967]

[81] Thapa KNK, Duraipandian N. Malicious traffic classification using long short-term memory (LSTM) model. Wireless Personal Communications, 2021, 119(3): 2707–2724. [doi: 10.1007/s11277-021-08359-6]

[82] Yang J, Lim H. Deep learning approach for detecting malicious activities over encrypted secure channels. IEEE Access, 2021, 9: 39229–39244. [doi: 10.1109/ACCESS.2021.3064561]

[83] García S, Grill M, Stiborek J, Zunino A. An empirical comparison of botnet detection methods. Computers & Security, 2014, 45: 100–123. [doi: 10.1016/j.cose.2014.05.011]

[84] Erquiaga MJ, García S, Garino CG. Observer effect: How intercepting HTTPS traffic forces malware to change their behavior. In: Proc. of the 23rd Argentine Congress. La Plata: Springer, 2018. 272–281. [doi: 10.1007/978-3-319-75214-3_26]

[85] Shiravi A, Shiravi H, Tavallaee M, Ghorbani AA. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Computers & Security, 2012, 31(3): 357–374. [doi: 10.1016/j.cose.2011.12.012]

[86] Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proc. of the 4th Int'l Conf. on Information Systems Security & Privacy. 2018. 108–116. [doi: 10.5220/0006639801080116]

[87] Draper-Gil G, Lashkari AH, Mamun MSI, Ghorbani AA. Characterization of encrypted and VPN traffic using time-related features. In: Proc. of the 2nd Int'l Conf. on Information Systems Security and Privacy (ICISSP). 2016. 407–414. [doi: 10.5220/0005740704070414]

[88] McGrew D, Anderson B. JOY. 2016. https://github.com/davidmcgrew/joy

[89] Huang H, Deng HJ, Sheng YQ, Ye XZ. Accelerating convolutional neural network-based malware traffic detection through ant-colony clustering. Journal of Intelligent & Fuzzy Systems, 2019, 37(1): 409–423. [doi: 10.3233/JIFS-179096]

[90] Khraisat A, Gondal I, Vamplew P, Kamruzzaman J. Survey of intrusion detection systems: Techniques, datasets and challenges. Cybersecurity, 2019, 2(1): 20–22. [doi: 10.1186/s42400-019-0038-7]

[91] Li FF, Razaghpanah A, Kakhki AM, Niaki AA, Choffnes D, Gill P, Mislove A. lib·erate, (n): A library for exposing (traffic-classification) rules and avoiding them efficiently. In: Proc. of the 2017 Internet Measurement Conf. London: ACM, 2017. 128–141. [doi: 10.1145/3131365.3131376]

[92] Bock K, Hughey G, Qiang X, Levin D. Geneva: Evolving censorship evasion strategies. In: Proc. of the 2019 ACM SIGSAC Conf. on Computer and Communications Security. London: ACM, 2019. 2199–2214. [doi: 10.1145/3319535.3363189]

[93] Wang ZJ, Zhu ST, Cao Y, Qian ZY, Song CY, Krishnamurthy SV, Chan KS, Braun TD. SymTCP: Eluding stateful deep packet inspection with automated discrepancy discovery. In: Proc. of the 2020 Network and Distributed Systems Security (NDSS) Symp. San Diego: NDSS, 2020. [doi: 10.14722/ndss.2020.24083]

[94] Wright CV, Coull SE, Monrose F. Traffic morphing: An efficient defense against statistical traffic analysis. In: Proc. of the 2009 Network and Distributed Systems Security (NDSS) Symp. San Diego: NDSS, 2009.

[95] Dyer KP, Coull SE, Ristenpart T, Shrimpton T. Peek-a-boo, I still see you: Why efficient traffic analysis countermeasures fail. In: Proc. of the 2012 IEEE Symp. on Security and Privacy. San Francisco: IEEE, 2012. 332–346. [doi: 10.1109/SP.2012.28]

[96] Luo XP, Zhou P, Chan EWW, Lee W, Chang RKC, Perdisci R. HTTPOS: Sealing information leaks with browser-side obfuscation of encrypted flows. In: Proc. of the 2011 Network and Distributed Systems Security (NDSS) Symp. San Diego: NDSS, 2011.

[97] Cai X, Zhang XC, Joshi B, Johnson R. Touching from a distance: Website fingerprinting attacks and defenses. In: Proc. of the 2012 ACM Conf. on Computer and Communications Security. Raleigh: ACM, 2012. 605–616. [doi: 10.1145/2382196.2382260]

[98] Cai X, Nithyanand R, Johnson R. CS-BuFLO: A congestion sensitive website fingerprinting defense. In: Proc. of the 13th Workshop on Privacy in the Electronic Society. Scottsdale: ACM, 2014. 121–130. [doi: 10.1145/2665943.2665949]

[99] Cai X, Nithyanand R, Wang T, Johnson R, Goldberg I. A systematic approach to developing and evaluating website fingerprinting defenses. In: Proc. of the 2014 ACM SIGSAC Conf. on Computer and Communications Security. Scottsdale: ACM, 2014. 227–238. [doi: 10.1145/2660267.2660362]

[100] Nithyanand R, Cai X, Johnson R. GloVe: A bespoke website fingerprinting defense. In: Proc. of the 13th Workshop on Privacy in the Electronic Society. Scottsdale: ACM, 2014. 131–134. [doi: 10.1145/2665943.2665950]

[101] Wang T, Cai X, Nithyanand R, Johnson R, Goldberg I. Effective attacks and provable defenses for website fingerprinting. In: Proc. of the 23rd USENIX Conf. on Security Symp. San Diego: USENIX Association, 2014. 143–157.

[102] Juarez M, Imani M, Perry M, Diaz C, Wright M. Toward an efficient website fingerprinting defense. In: Proc. of the 21st European Symp. on Research in Computer Security. Heraklion: Springer, 2016. 27–46. [doi: 10.1007/978-3-319-45744-4_2]

[103] Wang T, Goldberg I. Walkie-talkie: An efficient defense against passive website fingerprinting attacks. In: Proc. of the 26th USENIX Conf. on Security Symp. Vancouver: USENIX Association, 2017. 1375–1390.

[104] Gong JJ, Wang T. Zero-delay lightweight defenses against website fingerprinting. In: Proc. of the 29th USENIX Conf. on Security Symp. Vancouver: USENIX Association, 2020. 717–734.

[105] Jahani H, Jalili S. Effective defense against fingerprinting attack based on autocorrelation property minimization approach. Journal of Intelligent Information Systems, 2020, 54(2): 341–362. [doi: 10.1007/s10844-019-00553-0]

[106] Feghhi S, Leith DJ. A Web traffic analysis attack using only timing information. IEEE Trans. on Information Forensics and Security, 2016, 11(8): 1747–1759. [doi: 10.1109/TIFS.2016.2551203]

[107] Feghhi S, Leith DJ. An efficient Web traffic defence against timing-analysis attacks. IEEE Trans. on Information Forensics and Security, 2019, 14(2): 525–540. [doi: 10.1109/TIFS.2018.2855655]

[108] Abusnaina A, Jang R, Khormali A, Nyang D, Mohaisen D. DFD: Adversarial learning-based approach to defend against website fingerprinting. In: Proc. of the 2020 IEEE Conf. on Computer Communications. Toronto: IEEE, 2020. 2459–2468. [doi: 10.1109/INFOCOM41043.2020.9155465]

[109] Al-Naami K, El-Ghamry A, Islam MS, Khan L, Thuraisingham B, Hamlen KW, Alrahmawy M, Rashad MZ. BiMorphing: A bi-directional bursting defense against website fingerprinting attacks. IEEE Trans. on Dependable and Secure Computing, 2021, 18(2): 505–517. [doi: 10.1109/TDSC.2019.2907240]

[110] De la Cadena W, Mitseva A, Hiller J, Pennekamp J, Reuter S, Filter J, Engel T, Wehrle K, Panchenko A. TrafficSliver: Fighting website fingerprinting attacks with traffic splitting. In: Proc. of the 2020 ACM SIGSAC Conf. on Computer and Communications Security. Virtual Event: ACM, 2020. 1971–1985. [doi: 10.1145/3372297.3423351]

[111] Chan-Tin E, Kim T, Kim J. Website fingerprinting attack mitigation using traffic morphing. In: Proc. of the 38th IEEE Int'l Conf. on Distributed Computing Systems (ICDCS). Vienna: IEEE, 2018. 1575–1578. [doi: 10.1109/ICDCS.2018.00174]

[112] Cui WQ, Yu JM, Gong YM, Chan-Tin E. Realistic cover traffic to mitigate website fingerprinting attacks. In: Proc. of the 38th IEEE Int'l Conf. on Distributed Computing Systems (ICDCS). Vienna: IEEE, 2018. 1579–1584. [doi: 10.1109/ICDCS.2018.00175]

[113] Rahman MS, Sirinam P, Mathews N, Gangadhara KG, Wright M. Tik-Tok: The utility of packet timing in website fingerprinting attacks. Proc. on Privacy Enhancing Technologies, 2020, 2020(3): 5–24. [doi: 10.2478/popets-2020-0043]

[114] Pinheiro AJ, Bezerra JM, Campelo DR. Packet padding for improving privacy in consumer IoT. In: Proc. of the 2018 IEEE Symp. on Computers and Communications (ISCC). Natal: IEEE, 2018. 925–929. [doi: 10.1109/ISCC.2018.8538744]

[115] Hafeez I, Antikainen M, Tarkoma S. Protecting IoT-environments against traffic analysis attacks with traffic morphing. In: Proc. of the 2019 IEEE Int'l Conf. on Pervasive Computing and Communications Workshops. Kyoto: IEEE, 2019. 196–201. [doi: 10.1109/PERCOMW.2019.8730787]

[116] Pinheiro AJ, De Araujo-Filho PF, De M. Bezerra J, Campelo DR. Adaptive packet padding approach for smart home networks: A tradeoff between privacy and performance. IEEE Internet of Things Journal, 2021, 8(5): 3930–3938. [doi: 10.1109/JIOT.2020.3025988]

[117] Acar A, Fereidooni H, Abera T, Sikder AK, Miettinen M, Aksu H, Conti M, Sadeghi AR, Uluagac S. Peek-a-boo: I see your smart home activities, even encrypted! In: Proc. of the 13th ACM Conf. on Security and Privacy in Wireless and Mobile Networks. Linz: ACM, 2020. 207–218. [doi: 10.1145/3395351.3399421]

[118] Dwork C, Roth A. The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science, 2014, 9(3–4): 211–407. [doi: 10.1561/0400000042]

[119] Xiong SJ, Sarwate AD, Mandayam NB. Defending against packet-size side-channel attacks in IoT networks. In: Proc. of the 2018 IEEE Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP). Calgary: IEEE, 2018. 2027–2031. [doi: 10.1109/ICASSP.2018.8461330]

[120] Liu JQ, Zhang C, Fang YG. EPIC: A differential privacy framework to defend smart homes against internet traffic analysis. IEEE Internet of Things Journal, 2018, 5(2): 1206–1217. [doi: 10.1109/JIOT.2018.2799820]

[121] Sirinam P, Imani M, Juarez M, Wright M. Deep fingerprinting: Undermining website fingerprinting defenses with deep learning. In: Proc. of the 2018 ACM SIGSAC Conf. on Computer and Communications Security. Toronto: ACM, 2018. 1928–1943. [doi: 10.1145/3243734.3243768]

[122] Nasr M, Bahramali A, Houmansadr A. Defeating DNN-based traffic analysis systems in real-time with blind adversarial perturbations. In: Proc. of the 30th USENIX Security Symp. USENIX Association, 2021. 2705–2722.

[123] Salman O, Elhajj IH, Kayssi A, Chehab A. Denoising adversarial autoencoder for obfuscated traffic detection and recovery. In: Proc. of

the 2020 Machine Learning for Networking. Paris: Springer, 2020. 99–116. [doi: 10.1007/978-3-030-45778-5_8]
[124] Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. arXiv:1412.6572, 2015.
[125] Kurakin A, Goodfellow I, Bengio S. Adversarial examples in the physical world. arXiv:1607.02533, 2017.
[126] Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. arXiv:1706.06083, 2019.
[127] Zhang XK, Hamm J, Reiter MK, Zhang YQ. Statistical privacy for streaming traffic. In: Proc. of the 26th ISOC Symp. on Network and Distributed System Security (NDSS) Symp. San Diego: NDSS, 2019. [doi: 10.14722/ndss.2019.23210]
[128] Li J, Zhou L, Li HX, Yan L, Zhu HJ. Dynamic traffic feature camouflaging via generative adversarial networks. In: Proc. of the 2019 IEEE Conf. on Communications and Network Security (CNS). Washington: IEEE, 2019. 268–276. [doi: 10.1109/CNS.2019.8802772]
[129] Fathi-Kazerooni S, Rojas-Cessa R. GAN Tunnel: Network traffic steganography by using GANs to counter Internet traffic classifiers. IEEE Access, 2020, 8: 125345–125359. [doi: 10.1109/ACCESS.2020.3007577]
[130] Hou CS, Gou GP, Shi JZ, Fu PP, Xiong G. WF-GAN: Fighting back against website fingerprinting attack using adversarial learning. In: Proc. of the 2020 IEEE Symp. on Computers and Communications (ISCC). Rennes: IEEE, 2020. 1–7. [doi: 10.1109/ISCC50000.2020.9219593]
[131] Rahman MS, Imani M, Mathews N, Wright M. Mockingbird: Defending against deep-learning-based website fingerprinting attacks with adversarial traces. IEEE Trans. on Information Forensics and Security, 2021, 16: 1594–1609. [doi: 10.1109/TIFS.2020.3039691]
[132] Tramèr F, Kurakin A, Papernot N, Goodfellow I, Boneh D, McDaniel P. Ensemble adversarial training: Attacks and defenses. arXiv:1705.07204, 2020.
[133] Cremers C, Horvat M, Hoyland J, Scott S, van der Merwe T. A comprehensive symbolic analysis of TLS 1.3. In: Proc. of the 2017 ACM SIGSAC Conf. on Computer and Communications Security. Dallas: ACM, 2017. 1773–1788. [doi: 10.1145/3133956.3134063]
[134] Langley A, Riddoch A, Wilk A, *et al*. The QUIC transport protocol: Design and internet-scale deployment. In: Proc. of the 2017 Conf. of the ACM Special Interest Group on Data Communication. Los Angeles: ACM, 2017. 183–196. [doi: 10.1145/3098822.3098842]
[135] Henri S, Garcia-Aviles G, Serrano P, Banchs A, Thiran P. Protecting against website fingerprinting with multihoming. Proc. on Privacy Enhancing Technologies, 2020, 2020(2): 89–110. [doi: 10.2478/popets-2020-0019]

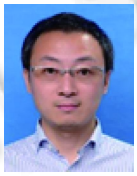
附中文参考文献:

[1] 2020年中国互联网网络安全报告. 2021. https://www.cert.org.cn/publish/main/upload/File/2020%20Annual%20Report.pdf
[23] 曾勇, 吴正远, 董丽华, 刘志宏, 马建峰, 李赞. 加密流量中的恶意流量识别技术. 西安电子科技大学学报, 2021, 48(3): 170–187. [doi: 10.19665/j.issn1001-2400.2021.03.022]

侯剑(1981－), 男, 高级实验师, CCF 专业会员, 主要研究领域为网络流量检测, 网络攻防对抗, 云计算.

王兴伟(1968－), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为未来互联网, 云计算, 网络安全.

鲁辉(1981－), 男, 博士, 教授, CCF 专业会员, 主要研究领域为网络攻防对抗, 智能化漏洞挖掘, 移动端脱壳, 反混淆技术.



田志宏(1978－), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为网络攻防对抗, APT 检测与溯源, 工控安全.

刘方爱(1962－), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为计算机网络, 并行处理, 推荐系统, 数据挖掘.