基于攻击流量自生成的 DNS 隐蔽信道检测方法

刁嘉文[1] 方滨兴[1),2)] 田志宏[2)] 王忠儒[3)] 宋首友[1),4),5)] 王 田[1)] 崔 翔[2)]

[1)](北京邮电大学网络空间安全学院可信分布式计算与服务教育部重点实验室 北京 100876)

[2)](广州大学网络空间先进技术研究院 广州 510006)

[3)](中国网络空间研究院 北京 100010)

[4)](北京丁牛科技有限公司 北京 100081)

[5)](丁牛信息安全科技(江苏)有限公司 江苏 南通 226014)

摘 要 高级持续性威胁(Advanced Persistent Threat,APT)是当前最为严重的网络安全威胁之一.DNS 隐蔽信道(DNS Covert Channel,DCC)由于其泛在性、隐蔽性成为攻击者手中理想的秘密信息传输通道,受到诸多 APT 组织的青睐.人工智能赋能的 DCC 检测方法逐步流行,但 APT 攻击相关恶意样本获取难、活性低等原因造成训练数据不平衡问题明显,严重影响了模型的检测性能.同时,已有检测工作使用 DCC 工具流量及少数恶意样本来评估系统,测试集覆盖范围有限,无法对系统进行全面、有效的评估.针对上述问题,本文基于攻击战术、技术和程序(Tactics, Techniques, and Procedures,TTPs)设计 DCC 流量生成框架并生成完备度可控的、覆盖大样本空间的、大数据量的恶意流量数据集.基于本研究生成的数据集,训练可解释性较强的机器学习模型,提出基于攻击流量自生成的 DCC 检测系统——DCCHunter.本研究收集了 8 个 DCC 恶意软件流量样本,复现了已被 APT 组织恶意运用的 3 个 DCC 工具产生流量,基于上述真实恶意样本评估系统对其未知的、真实的 DCC 攻击的检测能力.结果发现,系统对 DCC 的召回率可达 99.80%,对数以亿计流量的误报率为 0.29%.

关键词 DNS 隐蔽信道检测;数据不平衡;攻击流量自生成;恶意软件;未知样本

中图法分类号 TP393 **DOI 号** 10.11897/SP.J.1016.2022.02190

DNS Covert Channel Detection Based on Self-Generated Malicious Traffic

DIAO Jia-Wen[1] FANG Bin-Xing[1),2)] TIAN Zhi-Hong[2)] WANG Zhong-Ru[3)]

SONG Shou-You[1),4),5)] WANG Tian[1)] CUI Xiang[2)]

[1)](*Key Laboratory of Trustworthy Distributed Computing and Service (Beijing University of Posts and Telecommunications), Ministry of Education, Beijing* 100876)

[2)](*Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou* 510006)

[3)](*Chinese Academy of Cyberspace Studies, Beijing* 100010)

[4)](*Beijing DigApis Technology Co., Ltd, Beijing* 100081)

[5)](*DigApis Information Security Technology (Jiangsu) Co., Ltd, Nantong, Jiangsu* 226014)

Abstract The Advanced Persistent Threat (APT) is currently one of the most serious network security threats. Due to its ubiquity and concealment, DNS Covert Channel (DCC) has become an ideal secret channel in the hands of attackers, which remains active nowadays. With the development of DCC, from the exploration of attacks to organized attacks, the trend of organized attacks is becoming obvious. In response to this trend, researchers have also conducted in-depth studies on DCC detection. With the development of Artificial Intelligence (AI), AI-powered DCC detection

收稿日期:2021-09-28;在线发布日期:2022-04-16.本课题得到广东省重点领域研发计划项目(2019B010136003,2019B010137004)、国家自然科学基金(U20B2046)、国家重点研发计划(2019YFA0706404)资助.**刁嘉文**,博士研究生,主要研究方向为网络安全.E-mail:jarvand@bupt.edu.cn.**方滨兴**,博士,教授,中国工程院院士,主要研究领域为计算机体系结构、计算机网络、信息安全.**田志宏**(通信作者),博士,教授,主要研究领域为网络安全.E-mail:tianzhihong@gzhu.edu.cn.**王忠儒**(通信作者),博士,高级工程师,主要研究方向为人工智能、网络安全.E-mail:wangzhongru@bupt.edu.cn.**宋首友**,博士研究生,主要研究方向为网络安全.**王 田**,硕士研究生,主要研究方向为网络安全.**崔 翔**,博士,教授,主要研究领域为网络安全.

has made some progress, but it also suffers many problems, such as lacking of real malicious sample sets for training. Datasets of DNS-based APT suffer some problems, including the difficulty of obtaining, low activity, and small quantity. These problems have caused significant imbalance in training data, which severely limits the detection performance of the model. In the previous studies, the traffic generated by the DCC tools was used to replace the real malware traffic for training, but it could not describe the complete sample space of existing attacks. At the same time, the previous researchers generally used the traffic generated by DCC tools and few malware samples for testing. It is difficult to completely evaluate the detection capability of the system in this way. In this paper, we propose a new detection method, using self-generated malicious traffic to train the model and using real malware samples to evaluate the model. We summarize the TTPs based on a large number of APT reports, and design a framework for the traffic generation based on the TTPs. We generate the malicious data sets with characteristics of the controllable completeness, the large sample space and the large quantity for training. We can not only generate the attack traffic realistically and scientifically, but also predict the possible future attack traffic. This traffic can be customized and used for model training such as machine learning and deep learning. We extracted 19 features for training, 8 of which are newly proposed, and some features are optimized from previous work. We put forward three kinds of new features (totally 8), involving domain readability, domain structure, and IP discreteness. Experiments prove the superiority of the newly proposed and optimized features. Then, we use the machine learning algorithms with strong interpretability for experiments, and select the best model through the results of five-fold cross-validation. Through thoroughly reading and summarizing of the main APT reports, we define the real attack traffic into DCC malware traffic and traffic generated by DCC tools that have been maliciously used by APT organizations. We collect eight real malware traffic samples from various platforms, which cover all threat scenarios and common records. We reproduce traffic generated by three DCC tools that were maliciously used by APT groups. Based on the data sets mentioned above, the detection capability of the system against unknown DCC attacks can be evaluated. The results show that the recall rate of DCC can reach 99.80%, and the false positive rate of hundreds of millions of traffic is 0.29%.

Keywords　DNS Covert Channel detection; data imbalance; self-generated malicious traffic; malware; unknown sample

1　引　言

域名系统(Domain Name System,DNS)是一种将域名和 IP 地址相互映射的以层次结构分布的分布式数据库系统,广泛存在于互联网通信中.防火墙等基础防御设施一般不会对 DNS 进行过多过滤,泛在性、隐蔽性、穿透性使 DNS 隐蔽信道(DNS Covert Channel,DCC)成为攻击者手中理想的秘密信道.攻击者可以利用基于 DNS 协议的 DCC 进行命令控制(Command & Control,C&C)及数据泄露(Data Exfiltration)进而将其运用到恶意活动中.现如今,在高级持续性威胁(Advanced Persistent Threat,APT)攻击中,许多恶意软件利用 DCC 发起攻击,相应开源工具也逐步被攻击组织恶意利用,攻击组织化趋势明显.2020 年,影响较大的 SolarWinds 供应链攻击①,其涉及的恶意软件 SUNBURST 为追求隐蔽性利用 DCC 进行 C&C,导致包括美国政府部门在内的约 1.6 万 Orion 用户受到了影响;FBI 揭露了 Trickbot② 中的 Anchor_DNS 模块通过 DNS 泄露

① Highly Evasive Attacker Leverages SolarWinds Supply Chain to Compromise Multiple Global Victims with SUNBURST Backdoor. https://www.fireeye.com/blog/threat-research/2020/12/evasive-attacker-leverages-solarwinds-supply-chain-compromises-with-sunburst-backdoor.html

② Trickbot's Anchor_DNS Malware Allows for Data Exfiltration Over DNS to Evade Network Defense and Obfuscate Malicious Traffic. http://www.gnyha.org/wpcontent/uploads/2020/04/Trickbot_FLASH_FINAL.pdf

数据,针对高知名度组织发起攻击;WINNTI GROUP利用自定义开源工具 Iodine 对德国化工公司实施APT 攻击等.在 2021 年,美国国家安全局(National Security Agency,NSA)与网络安全和基础设施安全局(Cybersecurity and Infrastructure Security Agency,CISA)联合发布报告①指出,攻击者利用 DNS 查询将受感染主机上的数据密送至远程主机是网络安全防御的一个关键点.基于本团队之前对现存恶意软件的统计研究工作[1],我们发现命令控制、数据泄露以及两者融合运用这三种模式,构成了 DCC 应用的全部威胁场景,A、AAAA、TXT、NULL为DCC 常用的资源记录类型.

近年来,研究人员对 DCC 的检测方式也有所研究,有使用传统方式对隐蔽信道进行检测的方法,也有人工智能赋能的检测方法.传统的检测方法面临阈值固定带来的易被绕过问题.人工智能赋能的检测方法在一定程度上提高了检测模型的泛化性,但仍然面临训练样本不平衡问题.为解决这一问题,研究人员使用 DCC 工具流量进行训练.然而,使用配置单一且数量有限的工具产生的流量无法覆盖较大的样本空间.样本不平衡会导致模型训练效果不佳,这个问题一直没有得到解决.同时,我们发现现存检测研究主要使用隧道工具及少量真实恶意流量样本来评估系统.尽管已有检测方式对隐蔽信道工具的检测能力较好,但对真实攻击的检测效果并不明朗.

针对上述问题,本文提出了一种基于自生成攻击流量的 DCC 检测方法.具体地,我们生成了大量高质量的恶意流量并提取了 8 个新特征来训练模型,收集了 8 类恶意软件流量,复现了 3 类被恶意运用的 DCC 工具产生流量(如 5.3 节表 11 对比结果所示,为目前该领域最多、覆盖范围最广的真实攻击流量)来评估系统检测系统的有效性.本研究所生成的恶意流量具备数据量大、逼真度高(见图 3)、覆盖样本空间广(见图 3 及表 11)的特点.本研究基于构建机理提出贴合实际攻击的新特征并优化现有特征用于检测工作,所用于训练模型的特征为攻击者较难绕过的若干特征(欲绕过需要付出对应代价).基于上述两点进行检测模型建模,以提高系统对恶意流量的识别能力.同时,基于大量调查研究,我们收集了真实攻击流量,即 DCC 恶意软件流量及已被APT 组织恶意运用的 DCC 工具流量(见 4.2.2 节,据我们所知为被恶意运用的全部工具),来客观测试系统,对系统进行较为全面的评估.本研究使用自生成攻击流量及企业内网的良性流量对可解释性较强的机器学习模型进行训练,使用未用于训练的真实攻击流量来客观评估模型.基于覆盖大样本空间的恶意流量训练集及符合构建机理的特征,实现模型对其未知、真实 DCC 攻击的较精准辨识.分类后对恶意域进行阻止,防止其进行进一步的攻击活动.

本文的贡献包括以下 4 个方面:

(1)提出基于攻击流量自生成的 DNS 隐蔽信道检测系统——DCCHunter.使用自生成流量而非真实样本流量作为恶意流量进行训练,使用真实攻击样本流量评估系统性能;

(2)针对检测研究中一直面临的数据不平衡问题,基于攻击战术、技术和程序(Tactics, Techniques, and Procedures,TTPs)设计 DCC 流量生成框架并生成大量逼真的、有效的、完备度可控的 DCC 流量,构建覆盖大样本空间的恶意流量数据集.这种流量可按需定制且可用于机器学习、深度学习模型训练;

(3)提出符合构建机理的 8 个新特征并优化了 4 个现有特征用于检测.经理论分析及实验证明验证了所提部分特征的优越性;

(4)基于Hash及二级域名(Second Level Domain,SLD)在 HYBRID、ANY RUN 平台上检索信息近 100 条,收集到恶意软件流量样本 8 个(涵盖了全部威胁场景及常用资源记录类型),复现了被 APT 组织滥用的 DCC 工具 3 个.

2 问题定义及相关工作

2.1 概念定义

为准确对 DCCHunter 进行描述,本文总结了其对应的 6 个要素,并给出如下定义:

定义 1. DNS 隐蔽信道.反映的是攻击者利用 DNS 数据包中可定义字段(QNAME、RDATA、Raw-UDP)创建的隐蔽信息传输通道.

定义 2. 被控端.感染了 DCC 恶意软件进而可利用 DNS 数据包获取控制命令或外传数据的设备集合.

定义 3. 恶意权威名称服务器(Malicious Authoritative Name Server,MANS).攻击者搭建的权威名称服务器,用来托管恶意域名的名称解析,实现与被控端通信.

① Cybersecurity & Infrastructure security agency, selecting a protective DNS service. https://media.defense.gov/2021/Mar/03/2002593055/-1/-1/0/CSI_Selecting-Protective-DNS_UOO11765221.PDF

定义 4.　DCC 工具. 可以将其他协议/信息封装在 DNS 协议中传输的工具.

定义 5.　攻击流量自生成. 是指基于攻击 TTPs 自行定制生成的完备度可控的大量恶意流量. 该类流量可以包含已知攻击流量及科学预测未知攻击流量，可用作机器学习/深度学习模型的恶意流量训练集.

定义 6.　检测模式. 本文所提出的检测模式为使用本文自生成的攻击流量进行训练，使用收集到的真实流量样本对系统进行评估. 进一步地，本文所述训练集不包含收集到的真实恶意样本流量，仅为本文自生成的恶意流量. 本文收集到的真实恶意流量仅用于评估模型.

2.2　威胁模型

设备请求域名，经过本地 DNS 服务器或层层解析直到获得权威名称服务器的解析结果. 权威名称服务器负责解析其域名对应的子域，用户可以自行搭建权威名称服务器并配置解析记录，这是 DNS 协议可以被用于传输信息的关键，也是 DCC 得以构建的基础.

DCC 的威胁模型如图 1 所示，按照威胁场景划分，分为命令控制及数据泄露两种场景. 在数据泄露场景中，被控端利用 DNS 查询请求将待传送数据传送到 MANS 上，服务器返回对应记录应答. 在命令控制场景中，被控端向 MANS 请求命令，服务器利用 DNS 应答向被控端发出命令；按照连接方式划分，分为直连及中继两种连接方式. 被控端通过 IP 地址直接与 MANS 相连或者通过本地默认解析连接到 MANS 来传输信息，前者称为直连信道，后者称为中继信道.

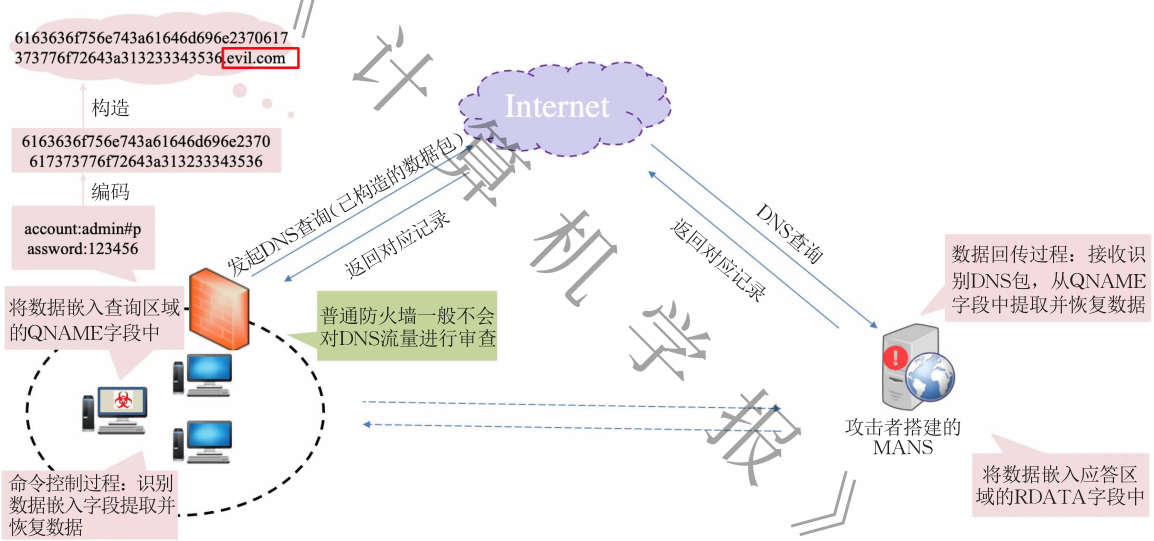

图 1　DCC 威胁模型

2.3　相关工作

通过全面阅读该领域文章，我们对具备代表性的检测文章进行了深入研究，旨在发现领域痛点，提出解决方案. 我们发现，近十年研究人员对 DCC 的检测方式进行了一系列的研究，包括从特征选取、检测算法选择等方面都进行了考虑，在识别 DCC 方面取得了一定的成果.

在传统异常检测方面，Born 等人[2]通过分析 DNS 查询和响应中域名单字母、双字母和三字母字符频率来检测 DCC. 该方法使用 n-gram 字符频率分析法对良性及隧道工具域名进行分析，利用自然语言与隐蔽信道流量在字符频率上的差别来识别 DCC. Karasaridis 等人[3]根据 DNS 数据包大小分布差异性和交叉熵等统计属性近实时检测异常. Ellens 等人[4]结合了流量信息和统计方法进行异常检测，结合阈值法、Brodsky-Darkhovsky 法、分布法实现了 5 个检测器，指出不同场景适用不同方法. Kara 等人[5]提出了一种基于资源记录分布情况近实时检测 DCC 的方法. 由于传统异常检测方式一般利用基于规则的静态阈值，其检测方法不灵活，误报率高，易被绕过. 随着机器学习的发展，使用其赋能安全检测的研究逐渐进入大众视野.

在机器学习赋能的检测方法中，已有研究涉及使用各类相关算法进行检测. 在监督学习赋能的检测方法中，Aiello 等人[6]将传统贝叶斯算法引入 DCC 检测中，使用查询、响应包大小及统计特征进行检测；章思宇等人[7]通过分析恶意流量特性提取区分特征 12 个，利用决策树、朴素贝叶斯和逻辑回

归 3 种机器学习方法训练模型，模型可以检测到已训练及未训练的隐蔽信道流量，但其训练样本数据量较少导致准确率偏低. Liu 等人[8]使用支持向量机、决策树和逻辑回归 3 种算法结合 18 种行为特征针对已知的隐蔽信道工具进行了二分类检测，获得了较高的准确率. 但已有工作均存在数据不平衡问题，恶意流量训练集通常使用 DCC 工具产生的流量，同时缺乏对模型未知/真实攻击效果的验证. 在无监督学习赋能的检测方法中，Dietrich 等人[9]提取 RDATA 差异性及通信行为差异性两方面特征使用 K-均值聚类的方式在网络流量中检测 C&C 信道. Nadler 等人[10]、Ahmed 等人[11]利用 iForest (isolation Forest)算法使用良性流量提取特征训练模型进行检测，达到了较高的准确率. 后述文章提取特征仅基于查询域名本身，若面临使用 UDP 额外携带信息及使用 DCC 应答接收命令的场景，这种检测方法几乎无效. 虽然上述工作涉及了对少数恶意软件流量的检测，但仅存在于数据泄露场景且数量较少. 同时，检测结果与良性流量选择关系过大，模型性能有待商榷.

随着深度学习的发展，使用深度学习赋能检测的方式也逐渐兴起. 近两年，Liu 等人[12]将流量以字节为单位转换成向量矩阵，通过独热(One-hot)编码将其转换为一个 257 维的向量，通过 CNN 模型对流量进行检测，并与 SVM 及逻辑回归等方式进行了对比. 张猛等人[13]，改进 CNN 形成 RDCC-CNN 并将特征向量转换为灰度图片表征 DNS 流量数据对隐蔽信道进行检测. Wu 等人[14]提出使用深度神经网络自学习特征进行检测，通过均方误差区别良性及恶意样本. Chen 等人[15]提出使用 LSTM 进行检测，并设置对照组验证其模型具备更强的泛化能力，表现更为优异. 研究工作均为针对模型已知数据集进行检测或针对 DCC 工具进行检测实验，对模型未知、真实恶意软件的检测情况并不明朗.

经过对上述重点文章及相关工作的对比分析，我们发现现存工作存在两个较为严重的问题：(1) 训练数据不平衡问题. 由于真实恶意流量不足，现存研究工作一般使用 DCC 工具产生的流量进行替代训练，但其结构单一，无法覆盖真实攻击样本空间；(2) 一般使用 DCC 工具产生的流量及少量真实恶意样本来评估模型，模型对真实攻击的检测情况并不明朗. 本研究拟针对上述两个问题，自生成攻击流量作为恶意流量进行训练(该部分内容将在第 3 节中详细阐述)，使用真实攻击流量样本来评估模型(该部分内容将在第 4 节 4.2 小节中详细阐述).

3 基于 TTPs 的攻击流量自生成技术

3.1 相关研究现状

人工智能赋能流量检测技术在安全研究中已广泛应用，恶意样本的质量严重影响着模型的性能，如何获得高质量的恶意流量样本是非常重要的研究课题. 然而，现阶段我们面临恶意流量样本大量获取困难问题. 原因有：(1) 及时发现并捕获流量较为困难，后期相关部门强制关停服务器等问题导致恶意软件活性降低难以获取流量；(2) 针对 DCC 的数据集一般不公开，难以找到大量公开可用的数据集. 通过对现有数据集(包括 Bot-DAD、CTU13 等数据集)的查找及统计，我们发现基本无专门针对 DCC 的数据集. 现存主要研究论文所使用的数据集一般不公开，各类研究工作一般使用 DCC 工具流量进行训练[7-8,12-17]，这类流量覆盖样本空间范围有限. 数据不平衡问题严重影响模型的精确度，急需科学地生成大量高质量的恶意数据集来解决这一问题.

当前，AI 模型训练的数据不平衡的问题也存在于其他领域，研究人员使用数据增强(Data Augmentation)或生成对抗网络(Generative Adversarial Networks，GAN)产生数据的方法来解决这一问题. 在数据增强技术方面，一般通过对图片进行旋转、缩放等操作来生成大量数据[18]. 然而，在 DCC 恶意流量的生成场景下，对于数据的自动化修改可能会导致语义信息发生改变，从而使得攻击失效. 在使用 GAN 生成数据方面[19]，Lee 等人[20]提出了使用 WGAN 来生成自相似的恶意流量，解决训练数据不平衡的问题. 但使用该方法需要使得生成器生成的数据可以欺骗判别器，还需事先积累一定规模的原始数据. 此外，设计一个能够完成攻击目的和数据有效性验证的判别器是困难的. 依据参数矩阵中各项数值完成自动化攻击目的和攻击原理验证是当前技术无法实现的.

故本文认为需要一种适用于 DCC 检测场景的独有数据生成方式来获取高质量的恶意数据集.

3.2 攻击流量自生成技术研究

目前，在国内外安全研究人员的共同努力下，DCC 攻击相关的 APT 报告已初具规模，内容详实的分析报告基本还原了攻击样本的技术实现情况. 本文基于 300 余篇 APT 分析报告及 10 余个 DCC 工具，梳理得到 DCC 攻击的 TTPs. 攻击 TTPs 作为攻击流量生成的重要理论基础，是流量数据科学性、完备性的保证. 由于目前真实恶意软件流量较少

无法有效用于训练，且每种 DCC 工具有其特定的数据传递方式，故我们的目标是突破现有 DCC 工具流量样本空间局限性的同时，生成大规模的流量用于训练，即广泛覆盖现有工具产生方式的同时拓展恶意样本空间.

第 2 节 2.2 小节中所述内容为我们高度概括的 DCC 威胁场景，分为命令控制及数据泄漏两个方面. 在数据泄漏方面，将待传输数据处理后嵌入 DNS 协议查询区域的 QNAME 字段中. 由于嵌入格式需要遵循字段规范且需要在服务端恢复，通常采用分片、编码（Base64、Base32 等）、加密（XOR、AES 等）及加入辅助字串等方式处理后嵌入对应字段来构造用于泄密 DNS 数据包，并通过正常 DNS 查询将其传送到攻击者搭建的 MANS，MANS 负责接收、提取并恢复敏感数据. 在命令控制方面，将待传输数据嵌入 DNS 协议应答区域的 RDATA 字段中，根据请求确定响应资源记录类型（A/TXT/AAAA 等）. 通常使用 A/AAAA 记录（使用 IP 地址预定义命令等）、TXT 记录（将命令编码或加密放入等）. 涉及的关键技术矩阵总结如表 1 所示（表中“*”为非必需实现的技术要点）.

表 1　DCC 关键技术矩阵

文件/命令处理	数据嵌入/恢复	传送方式
编解码转换	辅助字串	发包频率
*压缩	子域结构	记录类型
*加密	长度限制	包构建方法
*策略响应	*握手子域	

基于上述攻击的 TTPs，本研究设计了恶意流量生成框架，如图 2 所示. 具体地，在泄漏数据过程中，(1) 客户端按照配置文件设定的方式（编解码转换/压缩/加密）对目标内容进行加工处理，形成待泄漏数据；依据设定的子域名结构将待泄漏数据进行切片并加入辅助传输字串，将得到的数据嵌入到 DNS 数据包的 QNAME 字段中；最终按照设定的传送方式发出数据包；(2) 带有目标内容的 DNS 数据包一般的 DNS 解析流程到达服务端，服务端识别出该次泄密活动的数据包，从 QNAME 字段中提取信息暂存. 当泄密任务完成后，服务端依据辅助字符串重组数据分片，依照与客户端相逆的方式逐步恢复目标内容；(3) 服务端对客户端发来的 DNS 请求进行策略响应，避免异常. 依据服务端恢复出的目标内容与原内容做对比来确认文件是否泄漏成功；在命令控制过程中，(1) 客户端将带有请求命令标志的信息嵌入到子域中来构建用于请求命令的 DNS 数据包，并使用指定的传送方式发出数据包；(2) 该数据包经过常规的解析流程到达服务端，服务端识别请求命令的数据包，并下发对应的命令. 具体的下发方式为，将命令按照配置文件对应的方式对命令进行加工处理并嵌入到应答数据包的 RDATA 字段中；(3) 客户端收到后，按照对应的配置文件恢复目标命令并执行. 依据客户端恢复出的命令是否可以成功执行来判别命令是否下发成功.

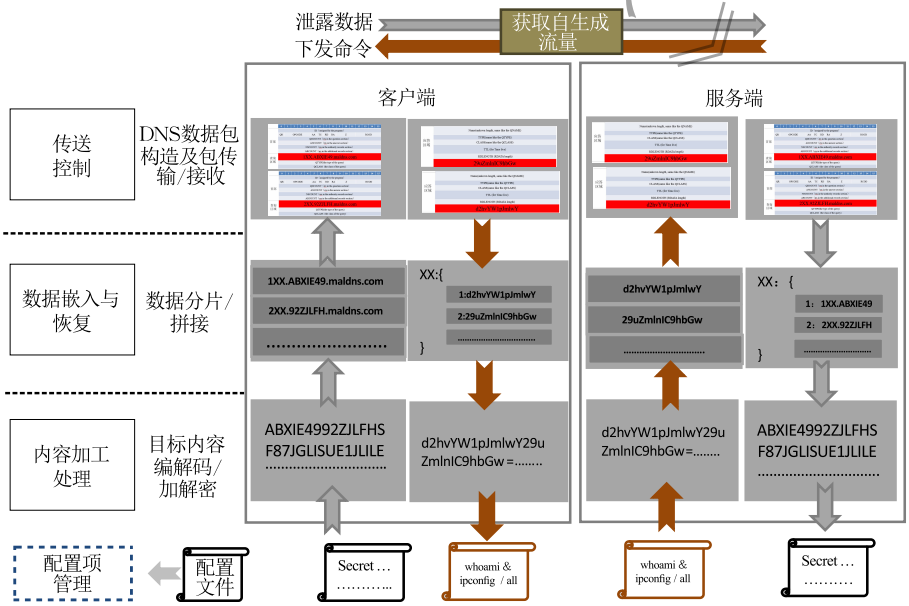

图 2　流量自生成框架设计

基于上述过程，我们使用配置文件（面向用户的json文件）将配置项（参照表1设计，共60余项）组合转化为系统参数，用户可以按照APT报告所述的关键点（例如，子域结构、编码方式等）进而高逼真的还原攻击流量. 同时，可以在攻击原理的范围内对各配置项进行定制化编辑，以达到预测未知攻击的效果. 我们通过更改配置文件的不同配置来执行攻击任务并检查攻击任务执行情况，在客户端捕获攻击流量，从而获得完成有效攻击的高质量恶意流量. 同时，我们也可以根据发现的真实恶意流量逐渐对恶意流量生成工具进行完善，从而使得恶意数据集更完备.

本文生成的流量具备两个重要优势：(1) 恶意流量训练集可以根据需求定制并大规模生成，解决了目前训练样本不平衡的问题；(2) 根据攻击TTPs科学地定制特征值，可以生成完备度可控的训练数据. 不仅可以高逼真还原真实攻击，还可以对未来攻击进行合理的前瞻性预测，解决了恶意流量训练集样本覆盖空间小的问题.

基于上述框架，本研究生成了服务于模型训练的恶意流量，并进行了两个方面的评估：(1) 对本研究生成数据与真实攻击数据的相似度进行评估. 使用ISOMAP降维算法，将我们生成的数据与APT34真实攻击数据降维至三维空间，如图3(a)所示，可以发现本研究生成的恶意流量（蓝色）与真实攻击流量（红色）高度拟合；(2) 对本研究生成恶意流量与现存工具流量就覆盖的样本空间进行评估. 本研究生成了有效执行DCC攻击的恶意流量共68.5 M. 同时，收集了企业日常DNS流量作为良性流量. 基于上述数据，共同绘制流量的T-SNE分布图（T-SNE算法是一种用于将高维数据降维到2/3维的可视化算法）. 本文使用T-SNE算法，以单条流量为单位将高维特征数据进行压缩，实现样本空间的可视化. 提取自生成恶意流量及现存DCC工具流量特征，将每条流量特征向量压缩到2维（即图中单点）. 如图3(b)所示，可以发现本文自生成的恶意流量比现存DCC工具流量覆盖更大的样本空间.

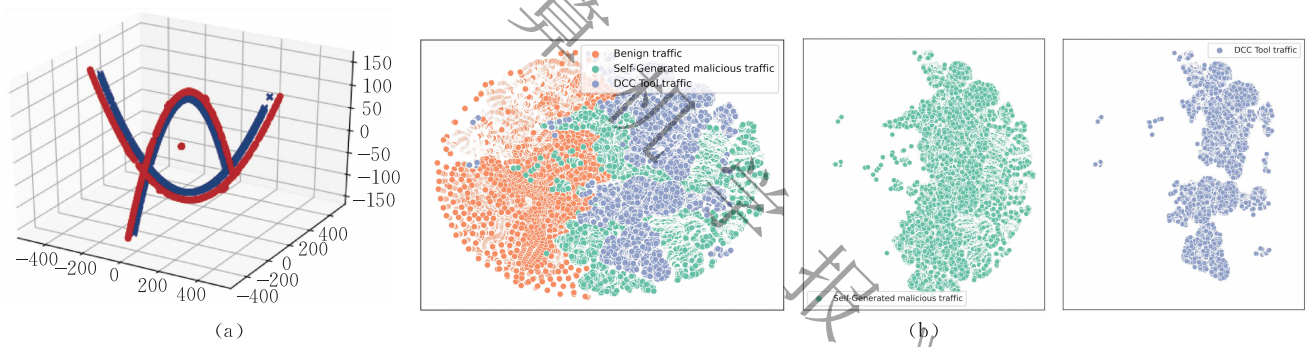

图3 良性/恶意流量的T-SNE分布

4 DCCHunter框架设计与实现

4.1 整体架构

DCCHunter整体架构如图4所示. 收集企业日常流量作为良性流量，基于攻击TTPs自生成完备度可控的攻击流量作为恶意流量，通过特征向量提取来训练可解释性较好的机器学习模型，根据K-折交叉验证结果选择表现最佳的模型. 将收集到的真实攻击流量样本输入系统进行测试，评估系统对真实攻击的检测能力.

与此同时，将该系统部署在企业内网中进行实时检测，以评估系统在大数据量下的检测能力.

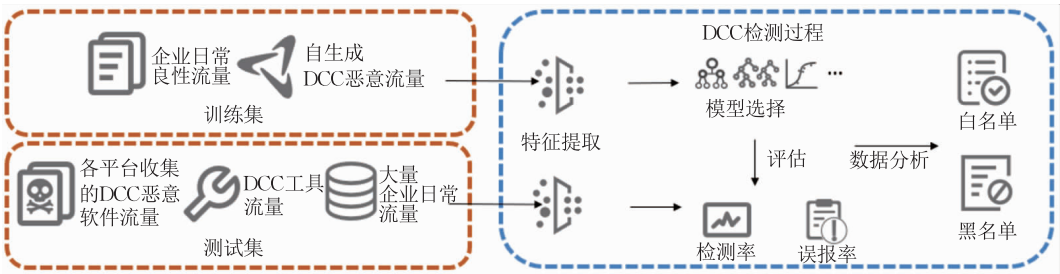

图4 数据集选择及DCC检测过程概览

4.2　数据集构建

4.2.1　训练集

在模型的训练过程中，本文使用企业内部良性流量及第 3 节中自生成的覆盖大样本空间的流量来训练模型. 具体的流量案例如表 2 所示，其中恶意流量覆盖了子域名较长、大写字符占比较高、数字占比较高、子域名较短、大写字符占比低、不含数字、数字占比很高及有公共子串且含数字等各类情况.

表 2　训练集分布案例

流量类型	流量样本案例
良性流量	www.baidu.com interface.music.163.com
自生成恶意流量	Hbrkknknkp.malicious.com(子域名较短)(大写字符占比较低)(不含数字) MzYw000af2ms7V0000E72C46T.HDNANN3IYOOONC3GQ7GSITV25S6FISTKJEPUC.MJJUZBZ5SEZ0-Z.malicious.com(子域名较长)(大写字符占比较高) 683439697c217c686f7374732e616c6c6fh49i.malicious.com(数字占比很高) FeQk0000JPSWCGO35MM3NLTQCC0-Z.malicious.com FeQk00014YLNNR55KGJXJXJRV70-Z.malicious.com(有公共子串)(大写字母占比较高) HUJCPn48554a43506e7c217c686f7374.malicious.com YEd28g5945643238677c217c74696d65.malicious.com(有公共子串)(小写字母、数字占比较高) ZMpE001EO5MF5NSTT2S2L7US6ML3GD2DDT7VQCTN42BLHXSZZPOAQ3NWNCTOIKC.YFPWWY4XD4MWFZYVYKUAHV2BUBTVDQDNT77QC3HDO6L2EMK5A7SYWOUUGQGNSH.J2NZLM4DYH3AG2NWKOCIU4TEC7WPO523NZVTBOK6HKYYG4AUDPUNOCFRSBOVFE.EB33WZH5PTMMHJ5KHYLERB4HN3IFY4ZVFMMJMVMW0-Z.malicious.com ZMpE002TLQRKRWHDSMR4D3X3KY5IJKG5LWISNMOF5XX2AFE6EB6ERE52S57QOSR.PTYQ4U63R3GDRGQ7PE3UY3COHNAL7XYDUTG2UHPUL2VWHAP6HPV5OGNEWGITVJ.GSHXUAKMRLLMNWCIZ6J2TX4JZLAYIUUKSZM5BJ6W45STCL7QWYF242V37USIKF.N3KY37K27G4D6RF3ZN5OQ642UMLOFRHBMEQSR3TJ0-Z.malicious.com(子域名很长)(有公共子串且含数字、大写字母)

4.2.2　测试集

根据对 DCC 发展以来相关威胁报告的通读总结，本文将真实攻击划分为 APT 攻击中使用的 DCC 恶意软件及已被 APT 组织恶意利用的 DCC 工具. 本节主要叙述根据 Hash 及 SLD 检索 DCC 恶意软件流量的部分.

真实 DCC 恶意软件流量. 由于真实恶意程序获取困难、现时复现困难，研究人员难以获取大量真实攻击流量. 本研究基于主要报告中的 Hash 及 SLD 在各大平台包括但不限于 VIRUSTOTAL、HYBRID、ANY RUN 等全面检索真实攻击样本流量. 区别于检索恶意样本程序本身，恶意样本流量一般为当时成功运行收集的真实攻击流量. 如今，这些流量由于服务器关停等原因并无法真实复现获得. 经过对近 100 条恶意样本 Hash 值及数十条 SLD 的检索结果进行排查，共获得 8 个(尽管不算大量，但为该领域内目前研究中的最大数字)DCC 恶意软件流量，囊括了全部威胁场景及常用资源记录类型. 包括 ISMDoor、Pisloader、RogueRobin、Denis、BONDUPDATER、DNSpionage、UDPPOS、DNSMessenger，具体分布情况如表 3 所示. 用其对系统进行评估，可以较为全面的了解系统对覆盖范围较全的真实恶意软件流量的检测情况.

表 3　真实恶意软件分布情况

攻击场景	应答记录类型	恶意软件名称	案例	域名长度
C&C	AAAA	ISMDoor	UjpNRT9hcHBJZD0tMSZtZXNzYWdlPWVz.0.dr.FF816E8D2B56458D80AB9065DCAE9AE4.basnevs.com	82
			n.n.fc.1998C3879EB146398102BCD988FC9721.basnevs.com	51
			www.0.s.FF816E8D2B56458D80AB9065DCAE9AE4.basnevs.com	52
	TXT	Pisloader	EWBDQOXDA0JVMVCWI.glb.it-desktop.com	36
		RogueRobin	aaa.stage.6388479.support.maccaffe.com	38
		DNSMessenger	CFCD208495.stage.0.ns0.website	16
Data Exfiltration	A	UDPPOS	e8cdf1ce69ec8ac.bin.de1c5732f0c8201f01ed8cc13f4005.bdd9e1fdf110dc5f741d58e289dea4.d104180cfb76dc7a8b39d074de8e8c.bcd7ca60892470c848172950c3d588.ns.service-logmeln.network	170
C2&Data Exfiltration	NULL	Denis	Igb-xAAAAAAAAAAAAAAAAAAAAAAAFQI.z.gl-appspot.org	49
	A	BONDUPDATER	180W758000aeb8000058136FAC73T.withyourface.com	46
	TXT		2180758ae023b800009F2B3C90T.22222222222222223246002224776 6E0E0E0E0E0E0E0A0EFDA000154F3.33333210100A.withyourface.com	118
	A	DNSpionage	jSiyGFDUWMD4JNBEUVSBKZFUGSZVJJIQ.microsoftonedrive.org	32

已被 APT 组织恶意利用的 DCC 工具流量. 本研究选取工具涵盖目前主要 APT 报告涵盖的全部用于定制化攻击/被 APT 组织纳入武器库的 DCC 工具. 申请域名并进行适当配置以复现,使用其秘密传递信息并抓取 5 个工具产生的流量各数万余 DNS 数据包. 包括 Cobalt Strike① 共 25.5 M、DNSExfiltrator② 共 8.18 M、Iodine③ 共 23.7 M、DET 共 3.18 M、Dns2tcp 共 9.54 M. 利用其对系统进行测试,可以较为全面的了解到模型对被用于真实攻击的 DCC 工具及一般 DCC 工具的检测能力.

4.3 特征提取

通过对 DCC 攻击构建机理的深入剖析,本文以大量真实攻击为依据,根据该攻击特点提取出 3 类(共 8 个)新特征,现分别阐述如下.

4.3.1 域名结构性

我们发现同一 DCC 攻击的众多域名中一般具备共同的字符串,如表 4 所示,列举了真实恶意软件域名构成的案例. 由于构建方法固定,因此一次攻击中会携带固定的令牌信息,如系统 ID、文件名等. 这些信息不仅相同且通常处于同一位置上. 在一次恶意活动中的不同域名共享着相同的结构,本文称这种现象所涉及的特征为域名结构性特征.

表 4 真实恶意软件域名构成案例

时间	恶意软件名称	子域名构成方法(编码)	字符长	二级域名
2016	Helminth	00〈SysI〉〈FN〉〈SeqN〉〈RN〉〈ED〉(Hex)	48	go0gIe.com
2017	ALMA Dot	〈RN〉.IDID.〈Vid〉.〈SeqN〉.〈TSeqN〉.〈ED〉.〈FN〉(Base16)	60	newuser.tk
2017	ISMAgent	〈ED〉〈SeqN〉.d.〈Vid〉(Base64)	13	ntpupdateserv.com
2017	BONUPDATER	〈RN〉4〈SeqN〉〈SysI〉B007(Base16)	50	poison-frog.club
2017	ALMA Dash	〈RN〉ID〈Vid〉-〈SeqN〉-〈TN〉-〈ED〉-〈FN〉(Base16)	20	prosalar.com
2018	QUADAGENT	〈ED〉.〈RN〉(Base64)	60	acrobatverify.com
2020	RDAT	〈ED〉.〈EM〉〈KEY〉(Base32 or Base64)	16	rsshay.com
2021	SUNBURST	〈Vid〉c00〈AD〉(Base64).appsync-api.eu/us-west/east-1/2	40	avsvmcloud.com
含义	系统标识符—〈SysI〉,文件名—〈FN〉,序列号—〈SeqN〉,序列总数—〈TSeqN〉,随机数—〈RN〉,编码后的待传送数据—〈ED〉,编码方法—〈EM〉,受害设备 ID—〈Vid〉,总包数—〈TN〉,加密密钥—〈KEY〉			

相同字符数、最大公共子串长度. 本文使用相同字符数及最大公共子串长度来描述同一 SLD 上不同子域结构之间的相似性. 相同字符数是指两个子域名中相对位置相同的字符数;最大公共子串长度指的是两个同域子域名中所有公共子串中最长的公共子串长度. 计算方式如式(1)、(2),$same_x$ 表示两个域名的子域名对应位置相同的字符数,x 表示第 x 组相同子串,共 i 组. $count$ 为该组的字符数,$allcommon\_length$ 表示两个子域名中全部相同的字符数. 其中第 i 组字符数最多,即为最大公共子串长度 $maxcommon\_length$.

$$allcommon\_length=\sum_{x=0}^{i} count[same_x(subdomain_1,subdomain_2)] \tag{1}$$

$$maxcommon\_length=\max_{0\le x\le i}\{count[same_x(subdomain_1,subdomain_2)]\} \tag{2}$$

最大公共子串是否包含数字、大写字母. 可以发现,令牌信息中含有编码信息及固定的大写字母、数字信息,而良性流量小写字母较多、数字较少且域名间一般无公共大写字母、数字的情况,这就造成了恶意流量与一般良性流量之间的差异. 本文使用最大公共子串是否包含数字、大写字母来描述此差异. 计算方式如式(3)、(4),$hasu$ 判断是否具备大写字母,$hasd$ 判断是否具备数字,具备则对应特征值为 1,不具备则为 0.

$$hasu\{\max_{0\le x\le i}[same_x(subdomain_1,subdomain_2)]\}=\begin{cases}0, & \text{最大公共子串不包含大写字母}\\ 1, & \text{最大公共子串包含大写字母}\end{cases} \tag{3}$$

$$hasd\{\max_{0\le x\le i}[same_x(subdomain_1,subdomain_2)]\}=\begin{cases}0, & \text{最大公共子串不包含数字}\\ 1, & \text{最大公共子串包含数字}\end{cases} \tag{4}$$

二级域名欺骗性. 攻击者可能会注册/接管一个更具欺骗性的域名(即 MANS 对应的域名)来进行攻击活动,从而使得攻击更不易被察觉. 例如表 4 中的 go0gIe.com,其与正常域名 google.com 极为类似. 为此,本研究使用 Levenshtein Distance 来表征域名与可信域名之间的相似性,即将一个字符串转换为另一个字符串所需要的最少编辑次数. 编辑方式一般可以包括替换、插入、删除等操作. 具体的计算方式如式(5),其中 $Deceptive(sld)$ 为 sld 的欺骗性,

① Cobalt Strike. https://www.cobaltstrike.com/

② Iranian hacker group becomes first known APT to weaponize DNS-over-HTTPS(DoH). https://www.zdnet.com/article/iranian-hacker-group-becomes-first-known-apt-to-weaponize-dns-over-https-doh/

③ Newly uncovered DNS tunnelling technique, and new campaign against South Korean gaming company. https://quointelligence.eu/2020/04/winnti-group-insights-from-the-past/

LD 为 Levenshtein Distance，top_list 为 Alexa Top 50 列表，最终计算为最小编辑距离来表示该 SLD 的欺骗性.

$$Deceptive(sld)=\min[Levenshtein\ Distance(sld,sld_{top\_list})] \quad (5)$$

良性查询较为随机，相邻子域间结构一般具备差异. 然而，同一攻击中的域名一般具备相似的结构. 如此使得良性与恶意域名间结构特性呈现不同.

本文选取上述特征进行作图，观测良性、恶意流量在所述特征上的差异，如图 5 所示.

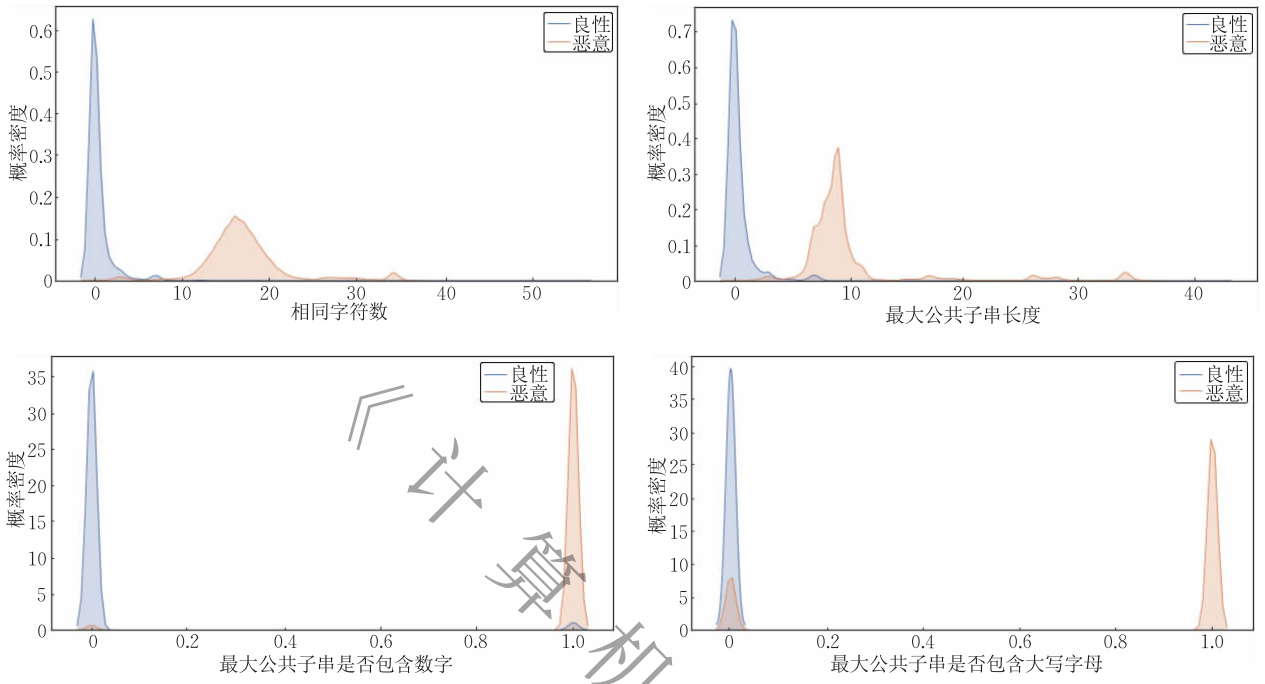

图 5　多域名结构性特征

4.3.2　域名可读性

如表 4 所示，恶意域名在构建过程中会对数据进行编码或加密操作，这种操作使得子域内容更为随机. 而对于一般的良性查询，为了使域名方便记忆，其更具备可读性.

子域名包含单词数、最大单词长度. 本文使用子域名中包含的单词数量及各单词中最长的单词长度来表征域名的可读性. 良性域名的子域名可读性好，一般由多个单词构成且最大单词长度合理（一般在 3～6 个字符范围内），恶意子域名中一般提炼不出单词或最大单词长度异常. 本研究调研了现存的分词工具，并对工具进行了逐一实验，最终选择效果较好的 wordninja 工具对域名进行分词操作. 若子域名由多个单词构成且最大单词长度合理，则认为是良性域名的可能性较大；若子域名中无法提炼出单词或最大单词长度异常，则认为是恶意域名的可能性较大. 具体表示为式（6）、（7），其中 $word_i$ 表示 $subdomain$ 分出的第 i 个单词，所有单词中最长单词所对应的字符数 max_len 为最大单词长度.

$$word_{1,2,\cdots,i}=wordninja(subdomain) \quad (6)$$

$$max\_len=\max[len(word_1),len(word_2),\cdots,len(word_i)] \quad (7)$$

同样地，本文使用这两个特征进行作图，发现了良性、恶意流量在这个两个特征上的差距，如图 6 所示. 尽管良性样本和恶意样本在图中显示出重叠，但

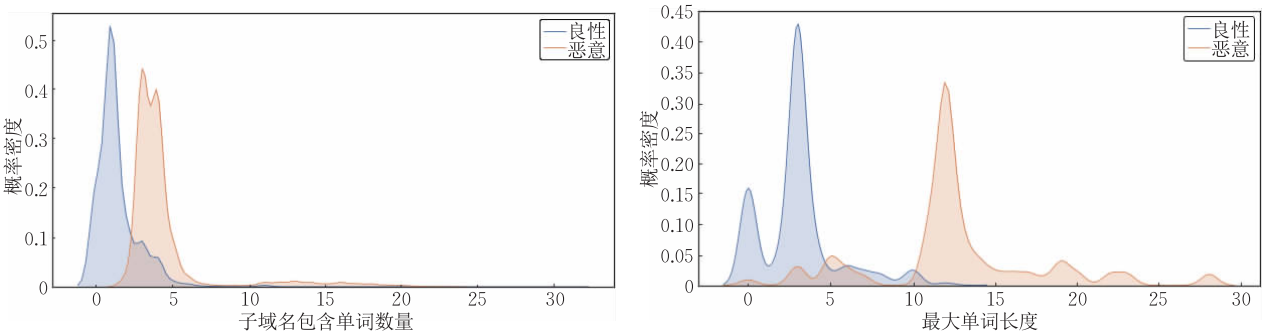

图 6　域名可读性特征

它们具有不同的集中区域和极值.因此,可以将域名可读性特征与其它特征结合使用,以提高模型的检测精度.

4.3.3 同域 IP 离散性

为了使得 DCC 流量与良性流量尽量相同,攻击者会定制 C2 服务器对恶意 DNS 查询的响应规则.本文根据收集到的真实恶意软件流量统计了其响应情况,如表 5 所示.攻击者一般使用固定 IP 响应或使用递增/递减 IP 响应,而良性流量同域对应 IP 响应为较为离散的 IP 地址.本文称前者为离散性差,后者为离散性好.具体计算方式如式(8),ips_sld 为单位时间内该 SLD 对应的IP地址数量,$subdomain\_sld$ 为该 SLD 对应的子域名数量,$\sum_{i=0,j=1}^{i,i+1} abs[digit\_i-digit\_j]$ 表示 SLD 的 IP 列表中对应位的差值和.$discrete\_score$ 表示了一个 SLD 列表中 IP 的平均增量,用其来表示 IP 的离散性.

$$discrete\_score=ips\_sld*\sum_{i=0,j=1}^{i,i+1} abs[digit\_i-digit\_j]/(subdomain\_sld*subdomain\_sld) \quad (8)$$

表 5 同域 IP 离散性统计

类型	名称(恶意/良性)	域名	响应 IP
离散	Baidu (良性)	www.google.com	202.160.128.16
		play.google.com	142.251.42.238
		maps.google.com	172.217.160.110
固定	UDPPOS (恶意)	e8cdf1ce69ec8ac.bin.b8753b5792ad47766fc0a6dc225d18.a0bfecf59117c1290b125cec83d5a8.dc342f11e66bc1679624cd69c39391.a1cad77d94396dd5550a344ddec895.ns.service-logmeln.network	185.73.240.207
		e8cdf1ce69ec8ac.bin.e3652d2a88ad2c147cd0b4cf304c0d.8deefce0ec0dc142692045ff94c3b9.cc190511e66bc1679624cd69c393ea.b1ddaa6794520fc645192358c9dcb8.ns.service-logmeln.network	185.73.240.207
		e8cdf1ce69ec8ac.bin.92753b5792ad47766fc0a6dc225d18.a0c4fce0ec0dc142692045ff94b8a9.d4641f118d09d2778136de79d6bebb.a1cad77d94396dd5550a344ddec895.ns.service-logmeln.network	185.73.240.207
递增	BONDUPDATER (恶意)	100380758ae2b80000C5643EC18T.46676766724677676766622246627 6D932F3F6403F20F2149FEE001CC029.33333210100A.withyourface.com	18.2.3.4
		180758a004e2b80000FB7AEC78T.667727676776620000435576775666784302535265 4EDADA3AC53523C14D.33333210100A.withyourface.com	18.2.3.5
		180758005ae2b80000FA891DC68T.665477467654666655667376666626 9EC1004141CCF31CC45D0E78F1D969.33333210100A.withyourface.com	18.2.3.6

通过对 DCC 构建机理的深入分析,本文共提取出两类 19 个特征用于检测.如表 6 所示,包括单域名特征:子域名长度、子域名数字占比、子域名大写字符占比、子域熵、二级域名欺骗性、子域名包含单词数、子域名包含最大单词长度、UDP 包内是否携带其他数据、资源记录类型、应答码、应答 IP 数;多域名特征:相同字符数、最大公共子串长度、最大公共子串是否包含数字、最大公共子串是否包含大写字母、DNS 请求应答比、DNS 请求频率、同域请求占比、同域 IP 离散性.

表 6 本文使用特征情况

特征类型	沿用特征	优化特征	新增特征
单域名特征	子域名长度、子域熵、UDP 包内是否携带其他数据、资源记录类型、应答码、应答 IP 数	子域名数字占比、子域名大写字符占比	二级域名欺骗性、子域名包含单词数、最大单词长度
多域名特征	同域请求占比	DNS 请求频率、DNS 请求应答比	相同字符数、最大公共子串长度、最大公共子串是否包含数字、最大公共子串是否包含大写字母、同域 IP 离散性

其中相同字符数、最大公共子串长度、最大公共子串是否包含数字、最大公共子串是否包含大写字母、二级域名欺骗性、子域名包含单词数、子域名中最大单词长度、同域 IP 离散性这 8 个特征为本文新提出特征.其中域名数字占比、子域名大写字符占比、DNS 请求频率、DNS 请求应答比这 4 个特征为本文优化特征(例如子域名数字占比从子域名数字数优化而来).其余为沿用原有研究特征.

4.4 攻击检测

检测工作基于两个前提假设:(1) DCC 流量与良性流量间存在时空差异.出于对嵌入字段格式、嵌入内容安全等问题的考虑需要对传输内容进行编码,必然会造成熵等特征相比于良性流量的异常.同时,由于 DNS 协议信息承载量较小,欲传输较多信

息势必会发送大量请求. 攻击者若想弱化异常，必定会损失攻击效率. 为保证此次攻击的攻击效果，异常是难以避免的；(2) 高可信度 SLD 不会被用于 DCC 攻击. 为提升检测效率，本文认为可以设置白名单对具备高可信度的 SLD 进行过滤. 这类域名可能会经常被访问且具备公司背书，若非权威名称服务器被攻陷，一般不会进行恶意活动，故认为其具备高可信度. 本文以 Alexa Top 为基准，认为其对应的域名具备高可信度. 同样地，也可以通过经验在此基础上合理扩充白名单范围.

攻击检测分为两个重要阶段，模型训练及使用模型进行检测. 检测的本质是根据已知的良性、恶意样本，判断最新输入样本的情况. 首先，对训练集数据进行预处理，使用 DNS 探针提取 DNS 流量. 良性流量选用过滤后的企业日常流量，恶意流量选用自生成的攻击流量. 从训练集中提取两类 19 个特征，将每一条流量构成一个 1×19 的特征向量作为训练输入，输入到可解释性较强的机器学习模型进行训练，选择表现较好的模型.

随后，使用模型对流量数据进行检测. 对需要检测的数据进行预处理，使用 DNS 探针提取 DNS 流量，设置白名单过滤器对常用域名进行过滤，提高任务的执行效率. 流量预处理工作完成后进行特征计算，与前述一致，提取待检测特征向量输入模型进行检测. 当一个新的样本到达时候，模型接收到并给它一个判断结果. 如式(9)所示，t 为时间窗口大小，经测试，此处设置为 30 min. L_t 为 t 时刻的 DNS 日志，$D_{t_{\text{now}}}$ 为该时间窗口内的 DNS 数据包.

$$D_{t_{\text{now}}}=\{L_t \mid t_{\text{now}}-\lambda\leq t<t_{\text{now}}\} \tag{9}$$

通过式(10)来判断数据包的良性/恶意情况. 其中 fe 为特征提取函数，通过提取待检测数据特征，将特征输入训练好的模型进行检测，结果为“0”则为良性流量，结果为“1”则为恶意流量，输出检测结果. 最后，可以通过检测结果对良性/恶意 SLD 进行分类，经分析设置白名单/黑名单对域名进行放行/阻止，以提高用户体验，阻止恶意域名的进一步活动.

$$MLModel(fe(D_{t_{\text{now}}}))=\begin{cases}0, & \text{良性}\\ 1, & \text{恶意}\end{cases} \tag{10}$$

5　实验结果与分析

5.1　实验设置

为了验证 DCCHunter 对于真实 DCC 攻击的检测情况，在如表 7 所示的环境中，部署了该检测框架并使用收集到的数据集对其进行测试.

表 7　实验环境

软/硬件	参数
操作系统	CentOS Linux release 7.9.2009 (Core)
CPU	Intel(R) Xeon(R) CPU E5-2620 0@2.00GHz
内核版本	Linux version 3.10.0-1160.15.2.el7.x86_64
Python 版本	Python 3.6.8

该评估主要侧重于系统对真实攻击的检测情况，集中在两个目标：对 DCC 恶意软件流量的检测情况及对被 APT 组织恶意运用的 DCC 工具产生流量的检测情况. 将所提出方法与近期发表的具备代表性的论文进行对比，以展示本研究情况.

5.2　评估结果

5.2.1　模型选择

本研究拟使用可解释性较强的机器学习算法进行检测. 使用表 8 中所示的数据集进行训练，良性流量选用企业日常 DNS 流量，恶意流量选用自生成流量. 基于前文(表 6)所述特征进行特征向量提取，将提取完毕的特征向量输入算法进行模型训练.

表 8　训练集情况

类型	数量(Pcap)
良性流量(企业日常 DNS 流量)	243 687
恶意流量(自生成攻击流量)	225 719

选用流行的机器学习算法进行训练，通过五折交叉验证均值对比模型各项指标的情况，结果如图 7 所示.

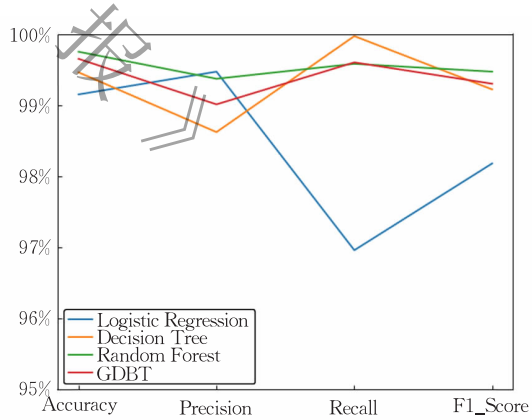

图 7　各机器学习模型检测指标对比图

通过观察，本研究拟采用各方面表现较好、较为稳定且 F1_Score(是统计学中用来衡量二分类模型精确度的一种指标)较大的随机森林进行训练. 通过其对应模型的进一步优化，兼顾各方面指标后，最终选择了一组表现较好的随机森林模型. 同时，为了解各个特征对模型的影响情况，特使用算法输出了对模型影响力排名前 9(共 19 个特征)的特征及其影响情况.

对模型影响较大的前 9 个特征依次为:DNS 请求应答比(q-a)、子域名数字占比(nump)、子域名大写字符占比(uppp)、子域长度(sublen)、最大公共子串是否含有大写字母(hasu)、同域请求占比(sldp)、最大公共子串长度(maxcommon)、最大公共子串是否含有数字(hasd)、子域熵(entropy).可以看出,以往的检测方法多为从子域长度等方面入手,未考虑 DNS 请求应答比、最大公共子串是否含有数字等,这些都是较为重要的特征.同时,虽然关注到子域中是否含有数字及字母特征,但本研究使用优化后的域名数字占比及大写字符占比作为特征更为科学合理.

5.2.2 检测结果

本研究使用该模型对其未知的、真实攻击流量进行检测,使用漏报率、误报率指标来衡量模型的检测情况.现分别介绍如下:

漏报率 FNR(False Negative Rate)反应了模型对恶意流量识别的能力,具体表示为未检测出的恶意流量占总恶意流量的比例.计算公式如下:

$$FNR=\frac{\text{未检测出的恶意流量记录数}}{\text{恶意流量记录总数}}=1-\frac{TP}{TP+FN}(\text{召回率})=\frac{FN}{TP+FN} \tag{11}$$

式中,TP 对应将恶意流量识别为恶意流量的数量,FN 对应将恶意流量识别为良性流量的数量,即为漏报.

误报率 FPR(False Positive Rate)反应了模型错误分类的良性流量占全部良性流量的比例.计算公式如下:

$$FPR=\frac{\text{错误分类的良性流量记录数}}{\text{良性流量记录总数}}=\frac{FP}{FP+TN} \tag{12}$$

式中,FP 对应将良性流量识别为恶意流量的数量,即为误报.TN 对应将良性流量识别为良性流量的数量.

将真实恶意软件流量输入模型进行检测,如表 9 所示,模型对其未知、真实恶意软件流量的漏报率均值约为 0.20%,误报率为 0.模型对覆盖全部威胁场景、常用资源记录的恶意流量的表现良好.

表 9 系统对真实恶意软件流量的检测结果

恶意软件名称	记录类型	数据包量	子域长度	漏报率/%	误报率	域名
BONDUPDATER	A	82 K	11～101	0	0	withyourface.com
	TXT	382 K	17	0.13	0	google4-ssl.com
Denis	NULL	259 K	49～143	0.43	0	gl-appspot.org
ISMDoor	AAAA	691 K	35～96	0	0	basnevs.com
Pisloader	TXT	57 K	20	0	0	it-desktop.com
RogueRobin	TXT	326 K	38	0.15	0	maccaffe.com
		268 K	34	0.18	0	cisc0.net
UDPPOS	A	65 K	23～189	0	0	service-logmeln.network
DNSMessenger	TXT	63 K	13～26	0	0	ns0.website
DNSpionage	A	1.6 M	11～63	0	0	microsoftonedrive.org

随后,使用模型对 DCC 工具流量进行测试(其中加粗项为已被 APT 组织恶意利用的全部 DCC 工具),得到结果如表 10 所示.模型对 DCC 工具流量的漏报率均低于 0.05%,可见,对 DCC 工具的检测也达到了较为理想的效果.综上,模型对其未知的、真实攻击流量具备较好的识别能力.

表 10 系统对 DCC 工具流量的检测结果

工具名称	记录类型	数据包量	子域长度	漏报率/%	误报率
DET	A	3.18 M	29～67	0.03	0
Dns2tcp	TXT	4.77 M	29～193	0	0
	KEY	4.77 M	29～193	0	0
DNSExfiltrator	TXT	8.18 M	39～248	0	0
Cobalt Strick	TXT	912 K	19～29	0.05	0
Iodine	A	3.89 M	26～198	0.04	0
	TXT	2.64 M	133～198	0	0
	CNAME	4.05 M	133～198	0	0
	NULL	3.38 M	133～198	0.02	0

在模型实际部署的过程中,会面向日常大数据量的 DNS 流量进行检测,误报率过高会严重影响用户体验,故系统误报情况是实际使用中需要考虑的一个重要问题.将系统部署在企业内网中,运行为时一个月,系统对近一个亿流量的误报率为 0.29%.可以发现,目前模型对良性流量的误报情况亦较为理想.

由上述评估结果可知,系统仍存在一些漏报、误报问题.漏报问题,通过对系统漏报的观察,存在问题如下:(1)类似于仅含 SLD 的灰色流量;(2)多为数据包中的第 1、2 条流量.对于问题(1),由于该流

量没有进行实质性的恶意行为且后续若进行可以检测出来，可忽略. 对于问题(2)，多域名分析在检测中起到了良好的作用，由于前两条数据没有足够的前文恶意流量可以进行关联分析，在训练数据不够完备的情况下，可能产生漏报问题. 造成漏报的原因可能是训练集不完备问题、特征选取问题、模型参数调整问题等. 实验中已通过遍历调参将模型参数调整到基本最佳；特征选取目前比较完善；在挑选训练集时发现目前自生成的攻击流量不够完备. 例如，观察到小写字母及数字组合中小写字母占比较少等问题，这类流量在漏报流量中也有存在. 后续会继续丰富自生成流量数据集，争取达到更好的检测结果. 对于系统误报问题，目前在可以接受的范围内. 同时，在实际应用中，可以根据误报情况进行进一步分析，通过设置白名单等避免合法服务被关停，进一步提升用户体验.

综上所述，模型的漏报率、误报率都处于一个较为理想的状态，在一定程度上实现了对未知、真实 DCC 攻击的精准检测.

5.3　与现存研究对比

在全面评估了模型的漏报率及误报率后，将本研究与近期重点研究就覆盖检测范围及检测效果两个层面进行了比较.

(1) 检测覆盖范围对比

对已有重点研究工作使用的方法、训练集、测试集详细情况进行统计，对比检测覆盖情况，如表 11 所示. 已有的研究工作主要针对 DCC 工具及少数数据泄露恶意软件进行检测. 本研究检测范围涵盖了 DCC 的全部威胁场景及常用资源记录类型，对系统进行了较为全面的评估工作.

表 11　与已有研究工作涉及检测范围对比

年份	文献	方法	训练集	测试集								
				KDT	UKDT	DCC 恶意软件						
						C&C	DE	C&C and DE	A	AAAA	TXT	NULL	总数
2013	[7]	DT，NB，LR	DCC Tool	√	√	—	—	—	—	—	—	—	—
2016	[16]	RF	DCC Tool	√	√	—	—	—	—	—	—	—	—
2017	[8]	SVM，DT，LR	DCC Tool	√	—	—	—	—	—	—	—	—	—
2018	[17]	Kernel SVM	DCC Tool	√	—	—	—	—	—	—	—	—	—
2019	[10]	iForest	Normal Traffic	—	√	—	√	√	√	—	—	√	2
2020	[11]	iForest	Normal Traffic	—	√	—	√	√	√	—	√	—	4
DCCHunter		RF	Self-Generated Malicious Traffic	—	√	√	√	√	√	√	√	√	8

备注：KDT，模型已知的 DCC 工具流量；UKDT，模型未知的 DCC 工具流量.

(2) 检测效果对比

以最新代表性成果[11]为例，进行了复现对比. 收集良性流量并根据论文提取了所需的 8 个特征(如域名字符数、大写字符数、数字字符数等)，将提取后的特征向量输入 iForest 模型进行训练，对照原文进行调整，使得模型达到最佳值. 输入本文的测试集对模型进行评估，检测结果如表 12 所示. 我们列出了详细的对比数据，以及对应部分特征的均值情况. 经实验发现，模型基本无法检测出本研究可精准检测的恶意软件 Denis 及 Pisloader，对 DNSpionage 的检测效果也不佳.

通过对恶意流量的对比发现(如表 12 中数字字符均值列所示)，Denis、Pisloder 及 DNSpionage 有一个共同的特征即几乎不含数字字符. 如图 8 所示，本文绘制了各恶意软件与良性流量在数字字符数特征上分布情况的箱型图. 可以发现在该特征上，这三个恶意软件流量与良性流量的表现趋同. 由于该特征权重占比过大，所以导致模型对它们的检测效果不佳. 在针对 DCC 工具的检测中，对 DET、DNSExfiltrator、Dns2tcp 的检测效果有所下降，其余工具基本持平.

表 12　对比工作的检测情况

恶意软件名称	标签个数	熵	数字字符均值	最大标签长	漏报率/%	误报率	变化量
BONDUPDATER	6	3.33	7	11	0	0	基本持平
Denis	4	1.51	1	32	99.57	0	漏报率：99.14% ↑
ISMDoor	6	3.80	22	36	0	0	基本持平
Pisloader	4	3.69	1	17	100	0	漏报率：100% ↑
RogueRobin	6	3.60	7	7	0	0	基本持平
UDPPOS	9	3.91	79	30	0	0	基本持平
DNSMessenger	5	3.62	7	10	0	0	基本持平
DNSpionage	3	2.98	1	18	94.88	0	漏报率：94.88% ↑

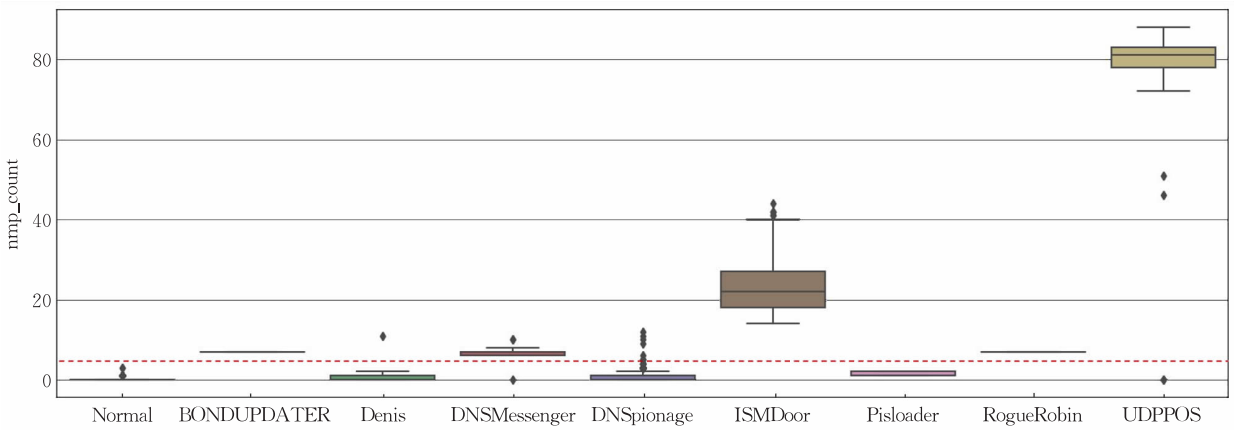

图 8 各恶意软件与良性流量在数字字符数上的分布情况

另外,该文章关注点仅在 QNAME 本身相关特征,这就可能造成对使用 UDP 额外携带信息的数据泄露场景及针对域名构造简单的命令控制场景是几乎无法检测的.后续可以增补特征,使其检测效果更为完善.

综上所述,DCCHunter 的检测效果较之前研究有所提升,无论从原理角度还是实验效果角度,都可以得出此结论.

6 总结与展望

在网络安全领域检测隐蔽信息传输通道是非常必要的,也具有一定的挑战性.尤其是 DNS 协议通常不被监控及限制,DCC 也经常被恶意软件用于命令控制及数据泄露场景中.虽然这个问题已经在近几年得到了广泛的研究,但主要集中在 DCC 工具及少数数据泄露恶意软件上,对真实攻击的检测尚缺乏全面研究,导致模型对未知、真实 DCC 攻击的检测能力并不明朗.

本研究基于攻击 TTPs 设计 DCC 攻击流量生成框架,自生成完备度可控的大量恶意流量数据集用于训练,解决了 DCC 检测研究中一直面临的数据不平衡的痛点问题,弥补了仅使用 DCC 工具生成恶意流量导致的样本空间的不足,使得模型更为精准.同时,全平台检索到目前该领域最多、覆盖范围最全的真实恶意流量样本,复现了在 APT 攻击中被恶意利用的 DCC 工具,基于上述数据集对系统进行测试.经实验,系统能够较为精准的检测真实、未知 DCC 攻击.

目前,使用深度学习检测 DCC 的研究还停留在初级阶段,评估工作一般使用 DCC 工具流量及模型已知的恶意流量作为测试集,模型对其未知的真实恶意软件流量检测情况尚不明朗.基于此,后续研究可以丰富自生成攻击流量数据集,利用深度学习进行建模检测与机器学习检测效果进行对比,也是可以继续深入思考、探索的方向.

参 考 文 献

[1] Diao Jia-Wen, Fang Bin-Xing, Cui Xiang, et al. Survey of DNS covert channel. Journal on Communications, 2021, 42(5): 164-178(in Chinese)
(刁嘉文, 方滨兴, 崔翔等. DNS 隐蔽信道综述. 通信学报, 2021, 42(5): 164-178)

[2] Born K, Gustafson D. Detecting DNS tunnels using character frequency analysis. arXiv preprint arXiv:1004.4358, 2010

[3] Karasaridis A, Meier-Hellstern K, Hoeflin D. NIS04-2: Detection of DNS anomalies using flow data analysis//Proceedings of the IEEE Globecom 2006. San Francisco, USA, 2006: 1-6

[4] Ellens W, Żuraniewski P, Sperotto A, et al. Flow-based detection of DNS tunnels//Proceedings of the IFIP International Conference on Autonomous Infrastructure, Management and Security. Berlin, Germany: Springer, 2013: 124-135

[5] Kara A M, Binsalleeh H, Mannan M, et al. Detection of malicious payload distribution channels in DNS//Proceedings of the 2014 IEEE International Conference on Communications (ICC). Sydney, Australia, 2014: 853-858

[6] Aiello M, Mongelli M, Papaleo G. Basic classifiers for DNS tunneling detection//Proceedings of the 2013 IEEE Symposium on Computers and Communications (ISCC). Split, Croatia, 2013: 000880-000885

[7] Zhang Si-Yu, Zou Fu-Tai, Wang Lu-Hua, et al. Detecting DNS-based covert channel on live traffic. Journal on Communications, 2013, 34(5): 143-151(in Chinese)
(章思宇, 邹福泰, 王鲁华等. 基于 DNS 的隐蔽通道流量检测. 通信学报, 2013, 34(5): 143-151)

[8] Liu J, Li S, Zhang Y, et al. Detecting DNS tunnel through binary-classification based on behavior features//Proceedings of the 2017 IEEE Trustcom/BigDataSE/ICESS. Sydney, Australia, 2017: 339-346

[9] Dietrich C J, Rossow C, Freiling F C, et al. On Botnets that use DNS for command and control//Proceedings of the IEEE 2011 Seventh European Conference on Computer Network Defense. Gothenburg, Sweden, 2011: 9-16

[10] Nadler A, Aminov A, Shabtai A. Detection of malicious and low throughput data exfiltration over the DNS protocol. Computers & Security, 2019, 80: 36-53

[11] Ahmed J, Gharakheili H H, Raza Q, et al. Monitoring enterprise DNS queries for detecting data exfiltration from internal hosts. IEEE Transactions on Network and Service Management, 2020, 17(1): 265-279

[12] Liu C, Dai L, Cui W J, et al. A byte-level CNN method to detect DNS tunnels//Proceedings of the 2019 IEEE 38th International Performance Computing and Communications Conference. London, UK, 2019: 1-8

[13] Zhang Meng, Sun Hao-Liang, Yang Peng. Identification of DNS covert channel based on improved convolutional neural network. Journal on Communications, 2020, 41(1): 169-179 (in Chinese)
(张猛, 孙昊良, 杨鹏. 基于改进卷积神经网络识别 DNS 隐蔽信道. 通信学报, 2020, 41(1): 169-179)

[14] Wu K M, Zhang Y Z, Yin T. TDAE: Autoencoder-based automatic feature learning method for the detection of DNS tunnel//Proceedings of the 2020 IEEE International Conference on Communications. Dublin, Ireland, 2020: 1-7

[15] Chen S, Lang B, Liu H, et al. DNS covert channel detection method using the LSTM model. Computers & Security, 2021, 104: 102095

[16] Buczak A L, Hanke P A, Cancro G J, et al. Detection of tunnels in PCAP data by random forests//Proceedings of the 11th Annual Cyber and Information Security Research Conference. New York, USA, 2016: 1-4

[17] Almusawi A, Amintoosi H. DNS Tunneling detection method based on multilabel support vector machine. Security and Communication Networks, 2018, 2018(1): 1939-0114

[18] Zhu J Y, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks//Proceedings of the IEEE International Conference on Computer Vision. Venice, Italy, 2017: 2223-2232

[19] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks//Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS). Montreal, Canada, 2014: 2672-2680

[20] Lee W, Noh B, Kim Y, Jeong K. Generation of network traffic using WGAN-GP and a DFT filter for resolving data imbalance//Proceedings of the International Conference on Internet and Distributed Computing Systems. Naples, Italy, 2019: 306-317

DIAO Jia-Wen, Ph. D. candidate. Her main research interest is network security.

FANG Bin-Xing, Ph. D., professor, academician of the Chinese Academy of Engineering. His main research interests include computer architecture, computer network and information security.

TIAN Zhi-Hong, Ph. D., professor. His main research interest is network security.

WANG Zhong-Ru, Ph. D., senior engineer. His main research interests include artificial intelligence and network security.

SONG Shou-You, Ph. D. candidate. His main research interest is network security.

WANG Tian, M. S. candidate. His main research interest is network security.

CUI Xiang, Ph. D., professor. His main research interest is network security.

Background

The Advanced Persistent Threat is currently one of the most serious network security threats. DNS Covert Channel has become an ideal secret channel in the hands of attackers due to its ubiquity and concealment, which remains active nowadays. In the past few years, researchers have also studied the detection methods of this threat. Some detection methods use traditional approaches to detect covert channels, while some methods are powered by Artificial Intelligence. AI-powered detection methods suffer several problems, such as the lack of malware samples for training, the use of DCC tools, and few malware samples for testing.

In this paper, according to APT reports, we generate complete controllable DCC traffic based on TTPs automatically, for the purpose of training. The traffic we generated

has the following advantages. (1) Traffic can be generated in large quantities, which solves the current limitation in malware samples for training. (2) Features that are difficult for attackers to bypass are extracted, and the values of features are scientifically customized on the basis of TTPs. We can generate not only the traffic of existing attack, but also the predicted traffic that may appear in the future attack. Forward-looking based on TTPs, solves the problem of the small sample space of training set. (3) Unknown DCC attacks can be accurately detected on the basis of the large sample space and the reasonable features. We extracted 19 features for training, 8 of which are newly proposed, and some features are optimized from previous work. We use the machine learning algorithms with strong interpretability for experiments, and select the best model through the results of five-fold cross-validation. We collect a lot more DCC malware traffic samples which cover all threat scenarios and common records and the traffic generated by the DCC tools which have been used in APT attacks to evaluate the system's ability to detect unknown samples. The results show that the system has good performance in DCC detection, both in terms of false negative rate and false positive rate.

This research is supported by the Key-Area Research and Development Program of Guangdong Province (Grant Nos. 2019B010136003, 2019B010137004), the National Natural Science Foundation of China (Grant No. U20B2046), the National Key Research and Development Program of China (Grant No. 2019YFA0706404).