DNS 隐蔽信道综述

刁嘉文[1]，方滨兴[1,2]，崔翔[2]，王忠儒[3]，甘蕊灵[1]，冯林[2]，姜海[4]

（1. 北京邮电大学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，广东 广州 510006；
3. 中国网络空间研究院信息化研究所，北京 100010；4. 北京丁牛科技有限公司，北京 100081）

摘 要：DNS 隐蔽信道是网络安全中不容忽视的重要安全问题。利用 DNS 访问服务器的操作广泛存在于传统 PC、智能手机及新型基础设施的联网通信中，防火墙等基础防御设施一般不会对 DNS 数据进行过多过滤。泛在性、隐蔽性使其成为攻击者手中较理想的秘密信道，因此关注已有研究成果及发展趋势都十分必要。首先，将 DNS 隐蔽信道的发展历程概括为 3 个发展阶段，并分析各个阶段的情况。然后，对其进行形式化定义，深入剖析构建机理，并对其存在的不可绕过的异常点进行分析归纳，总结检测方法并将其分为传统检测方式、人工智能赋能的检测方式，提出现存问题。最后，总结当前 DNS 隐蔽信道的主要研究方向，并对其未来的发展趋势进行展望。

关键词：DNS 隐蔽信道；命令控制；数据泄露；检测；高级持续性威胁

中图分类号：TP393

文献标识码：A

DOI: 10.11959/j.issn.1000−436x.2021090

Survey of DNS covert channel

DIAO Jiawen[1], FANG Binxing[1,2], CUI Xiang[2], WANG Zhongru[3], GAN Ruiling[1], FENG Lin[2], JIANG Hai[4]

1. Key Laboratory of Trustworthy Distributed Computing and Service (Beijing University of Posts and Telecommunications), Ministry of Education, Beijing 100876, China
2. Cyberspace Institute Advanced Technology, Guangzhou University, Guangzhou 510006, China
3. Chinese Academy of Cyberspace Studies, Institute of Information Technology, Beijing 100010, China
4. Beijing DigApis Technology Co., Ltd., Beijing 100081, China

Abstract: DNS covert channel is an important security issue that cannot be ignored in network security. The operation of using DNS to access the server is widely used in the network communication of traditional PC, smart phones and new infrastructure. Basic defense facilities such as firewalls generally do not filter DNS data too much. The ubiquity and concealment make it an ideal secret channel for attackers. It is necessary to pay attention to the existing research results and development trends. The development process was summarized into three stages, and the situation of each stage was analyzed. Formally it was defined and the construction mechanism was deeply analyzed. The existing abnormal points that cannot be bypassed were analyzed and summarized, the detection methods were summarized and divided into traditional detection methods and artificial intelligence-powered detection methods, the existing problems were raised. Based on the above classification, the construction and detection frontiers of DNS covert channel was reviewed, and an in-depth analysis was conducted from different perspectives such as development trends, technical mechanisms, and detection methods. Finally, the main research direction of the current was summarized, and its future development trend was prospected.

Keywords: DNS covert channel, C&C, data exfiltration, detection, APT

收稿日期：2020−11−23；**修回日期：**2021−03−24

通信作者：崔翔，cuixiang@gzhu.edu.cn

基金项目：广东省重点研发计划基金资助项目（No.2019B010136003，No.2019B010137004）；国家重点研发计划基金资助项目（No.2018YFB0803504, No.2019YFA0706404）

Foundation Items: The Key Research and Development Program of Guangdong Province (No.2019B010136003, No.2019B010137004), The National Key Research and Development Program of China (No.2018YFB0803504, No.2019YFA0706404)

1　引言

域名系统（DNS, domain name system）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[1]，也是互联网上普遍存在的基础解析服务。防火墙等基础防御设施为了保证用户体验一般不会对 DNS 数据进行过多过滤，使其成为攻击者手中较理想的秘密信道[2]。互联网名称与数字地址分配机构（ICANN, Internet Corporation for Assigned Names and Number）[3]将其命名为 DNS 隐蔽信道（DCC, DNS covert channel）。

DCC 是指利用 DNS 数据包中的可定义字段秘密传递信息的通道。其中，“DNS 协议”是目前网络上使用的标准域名解析协议[4]；“可定义字段”是 DNS 数据包中的 QNAME 字段、RDATA 字段及 RawUDP 字段。利用 DNS 数据包可以构建 2 种信道：存储信道及时间信道。由于时间信道可传输信息较少且对应的工具及恶意软件很少，故主要研究存储信道的利用情况。DCC 可以被用于数据泄露、命令控制（C&C, command & control）及绕过 Wi-Fi 连接注册等恶意行为，进而可以用于远控木马（RAT, remote access trojan）、僵尸网络（Botnet）、勒索软件（Ransomeware）、高级持续性威胁（APT, advanced persistent threat）等绝大多数网络攻击。

虽然 DCC 早在 1998 年就开始出现，但在 2020 年仍有许多恶意软件利用其发起攻击，相关开源工具也逐渐被攻击组织恶意利用，对各个领域都产生了一定程度的影响[5-20]。在金融领域，活跃在该领域中的销售点（POS, point of sale）恶意软件（如 AlinaPOS[5]）经常利用 DNS 查询请求来泄露信息[6]；在医疗领域，SentinelLabs[7] 及 Unit42[8] 报道称 Trickbot 开发人员在其恶意软件中新增了 Anchor_DNS 模块，在针对美国医疗系统的攻击中利用该模块使用 DCC 进行 C&C。大多数工具开源且日趋成熟，有理由相信，未来可能会有越来越多的 DCC 工具被恶意利用，DCC 恶意软件也时刻威胁着网络空间安全。

本文主要围绕 DCC 构建及 DCC 检测 2 个方面进行论述，具体贡献总结如下。

1) 对 DCC 的威胁模型进行了归类、总结；对 DCC 的发展历程进行了全面梳理，概括为 3 个发展阶段。

2) 对 DCC 概念进行了形式化定义；对构建机理进行了深入剖析，为检测研究提供有价值的参考。

3) 总结 DCC 难以绕过的异常点，并对其异常进行了分析；对传统检测方法及人工智能赋能的检测方法涉及的具有代表性的论文进行了总结梳理，指出现存问题。

4) 对未来的发展趋势进行了展望，旨在从整体角度发现重点问题，为研究人员提供进一步研究的方向。

2　DCC 威胁模型与发展历程

2.1　威胁模型

由于 DNS 协议具有泛在性、隐蔽性，因此可以将 DCC 运用于许多威胁活动中。DCC 分为 2 种类型，若受害设备通过 IP 地址直接与恶意权威名称服务器（MANS, malicious authoritative name server）相连，则称所构成的信道为直连信道，这种情况下一般使用 RawUDP 字段传输信息；若受害设备通过本地默认解析连接到 MANS，则称所构成的信道为中继信道，这种情况下一般使用 QNAME 字段及 RDATA 字段传输信息。由于前者较简单，故这里主要讨论后者所涉及的命令控制、数据泄露 2 种威胁场景，其过程有着细微的差别，下面将进行详细阐述。

1) 命令控制

普通防火墙等基础防御设施一般不对 DNS 进行过多过滤，使 DCC 成为具有一定隐蔽性、穿透性的 C&C 信道，可以实现受害设备与远程攻击者搭建的 MANS 的双向交互，如图 1 所示。受害设备向 MANS 请求控制命令，MANS 收到后，将欲下发的命令进行处理，利用 DNS 响应向受害设备发出命令，受害设备解码获得命令并执行。

2) 数据泄露

在这种场景下，攻击者会利用 DNS 查询请求将待传送数据如敏感信息、文件等传送到其搭建的 MANS。由于大部分安全基础设施都更加关注来自外部网络的攻击，因此这种攻击需要主动监测内部流出流量才能发现，使数据泄露的全过程更隐蔽。受害设备向攻击者搭建的 MANS 发出请求，利用请求来泄露信息，如图 2 所示。如果待泄露文件较大，则需对其进行分片，然后将分片后的信息进行编码压缩传送。服务器收到后，按照对应格式获取信息，并适当响应。

DCC 用于恶意活动一般存在于以上 2 种场景中。对 ATT&CK[21]上实际案例的利用情况统计发

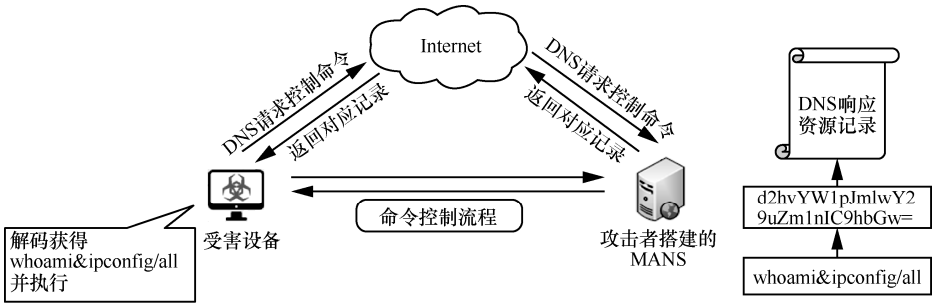

图 1　命令控制流程

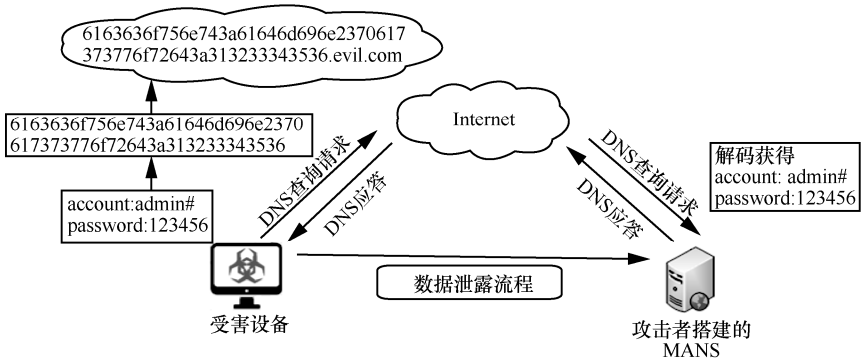

图 2　数据泄露流程

现，有些恶意软件仅使用 DCC 进行 C&C，如 Feederbot[22]、Pisloader[23]等；有些恶意软件将 DCC 仅用于数据泄露，如 FrameworkPOS[6]、Remsec[24]等；同时也有部分复用 C&C 信道进行数据泄露的案例，如 RDAT[25]、BONDUPDATER[26]等。

2.2　发展历程

DCC 的发展可以概括为 3 个阶段：第一阶段（1998 年—2010 年）是以 NSTX[27]、OzymanDNS[28]、Heyoka[29]、PSUDP[30]等工具为代表的攻击探索阶段；第二阶段（2011 年—2013 年）是以 Feederbot[22]、Morto[31]等恶意软件为代表的恶意利用阶段；第三阶段（2014 年—至今）是以 APT34、FIN6 等 APT 组织为代表的组织化攻击阶段。

第一阶段，攻击探索阶段。对 DCC 的最初运用由隧道协议、传输文件开始，其本质为利用 DNS 数据包进行数据的秘密传输。1998 年，Pearson[32]首次描述了 DNS 隧道的基本情况，实现了利用 DNS 数据包进行的客户端与服务器间的简单通信；2002 年，由 Szerb[27]完成的第一个较流行的隧道工具 NSTX 使 IP over DNS 成为可能，但其正常运行需要创建一个虚拟网络设备（VDN, virtual network device），并且只能运行在 Linux 系统上；2004 年，在 Black Hat 大会上，Kaminsky[28]演示了其编写的 OzymanDNS 工具，该工具可以利用 DNS 传输文件，也可以封装 SSH 隧道（SSH over DNS）；2008 年，Miller[16]提出了旨在利用 DNS 做 C&C 的 Reverse DNS，其将 shellcode 代码放入 DNS 响应的 TXT 资源记录中，受害设备收到响应并执行后可以主动与服务器建立连接；2009 年，Revelli[29,33]在信道容量上进行了进一步思考提出了 Heyoka，指出许多 DNS 接受域名标签中的二进制数据，利用二进制编码可以将带宽从每字符 5 位增加到每字符 8 位。同时提出可以使用 EDNS0，TXT 响应最多可存储 1 024 B；2010 年，Born[30,34]在 Black Hat 上提出 PSUDP，使用 UDP 增加信道带宽，指出可以在 DNS 消息末尾注入数据而不会影响 DNS 服务器解析，从而增加了信道带宽。这一阶段初步实现了利用 DNS 数据包对数据的传输功能，扩宽了信道容量，但也为恶意活动提供了良好的技术基础。

第二阶段，恶意利用阶段。2011 年，Dietrich 等[22]分析了第一个基于 DCC 的僵尸网络 Feederbot，发现其 TXT 应答数据消息块中使用 RC4 流密码加密（利用 DNS 查询来传输密钥派生参数），同时使用循环冗余校验保证数据完整性；卡巴斯基实验室发现利用 DNS 进行 C&C 的蠕虫 W32.Morto[31]，其只请求 TXT 记录，解密后得到 IP 地址进而下载文

件执行；美国能源局[35]发布白皮书表示可以通过 DNS 查询请求泄露机密信息且难以对其进行检测。2013 年，Xu 等[36]论证了使用 DCC 作为僵尸网络 C&C 通道的可行性，描述并定量分析了包搭载查询、指数分布等查询策略，这些策略可用于在网络级别有效隐藏恶意 DNS 活动，并指出 DNS 是一个极有效的 C&C 信道。这一阶段的技术被恶意运用到僵尸网络等网络恶意行为中，对网络安全造成了较严重的威胁。

第三阶段，组织化攻击阶段。2014 年，APT 组织 FIN6 的 FrameworkPOS[37]使用 DNS 请求泄露了 5 600 万张借记卡/信用卡信息，在 DNS 查询中编码了 IP 地址、主机名、进程名等字段；2016 年，Wekby（APT18）的 Pisloader[23]使用 DNS 协议做 C&C 信道发起攻击；2017 年，OilRig（APT34）开发人员对 Helminth[38]不同变体的子域首字符进行更改，进而躲避检测，其依赖 DNS 请求获得具有指令含义的 IP 地址（A 记录）应答，将欲传送的窃密文件使用 DNS 查询请求分块传送；2018 年，OilRig 将 QUADAGENT 用于定向攻击，初次握手时受害设备会得到 C&C 服务器提供的会话标识符和预共享密钥，并将其保存到注册表，不必每次通信都进行握手；2019 年，Lab Dookhtegan Telegram Chanel 中泄露了关于 APT34 的攻击工具，其中 Glimpse[39-40]可以利用 DNS 传输指定目录下的文件；2020 年，QuoIntelligence 发现 WINNTI（APT41）组织自定义 Iodine 针对德国化工企业，其 DNS 查询使用 Base128 编码。OilRig 利用 RDAT[10]针对中东电信组织，与其之前样本相比，该样本仅使用 DNS 进行 C&C 通信而无 HTTP 备用信道。Black Lotus Labs 发现 Alina POS 编码信用卡信息使用 DNS 查询泄露，指出 DNS 经常不受监控，同时，为了确保在搜索设备的 RAM 时准确找到信用卡数据，该恶意软件引入了 Luhn 校验和算法。APT 组织自制恶意软件或集成工具发起攻击活动，逐渐将这一技术纳入攻击中，利用其传输敏感信息及进行 C&C，攻击组织化更明显。图 3 列出了 DCC 的发展历程。

3　DCC 构建机理

3.1　DCC 定义

定义 1　DCC = (VictimMachine, DDP, CMD, POLICY,MANS,ConnectionType,δ) 由七元组构成，反映的是攻击者利用 DNS 数据包中可定义字段创建的隐蔽信息传输通道。

1) VictimMachine 指的是感染了 DCC 恶意软件进而可利用 DNS 数据包传输数据的受害设备集合，受害设备种类可以是 PC、服务器、智能手机、物联网（IoT, Internet of things）设备等一切当前已存在的、未来可能出现的、具备计算能力和通信能力并可发起 DNS 查询请求的设备，记为 VictimMachine={DCCMalware, S, ACTIVITY}。

DCCMalware 表示运行在受害设备上的恶意程序集合，记为 DCCMalware={dccmalware$_1$, dccmalware$_2$,⋯,dccmalware$_n$}，其中 dccmalware$_i$ 表示运行在 VictimMachine$_i$ 上的恶意程序，n 表示受害设备的规模。

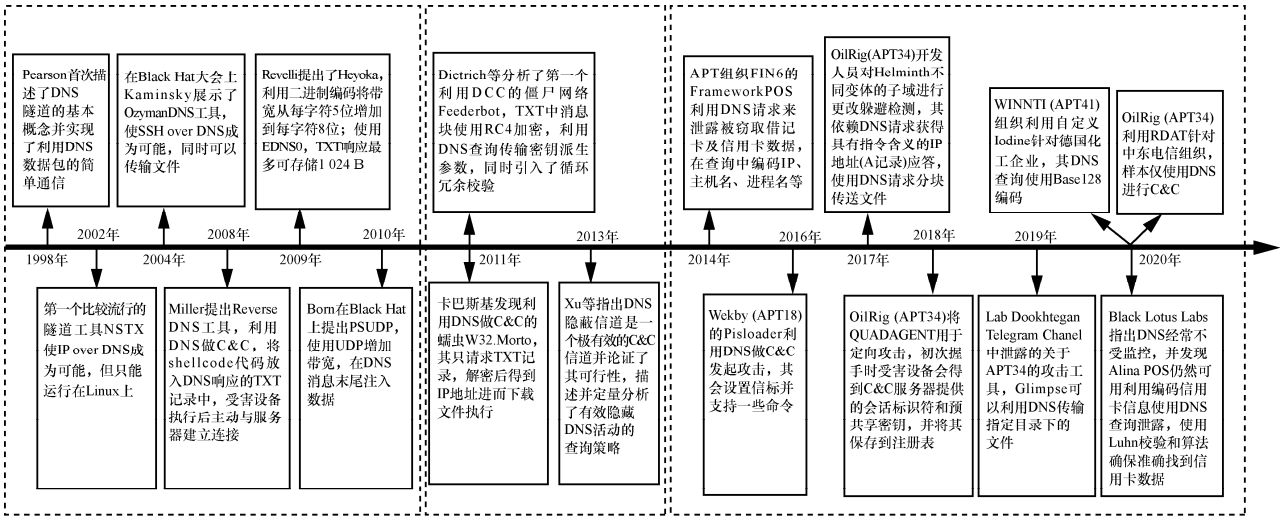

图 3　DCC 发展历程

S 表示 DCCMalware 的状态集合，记为 $S=\{s_1,s_2,\cdots,s_n\}$。

ACTIVITY 表示 DCCMalware 的动作集合，记为 ACTIVITY={SendData,ReceiveData, ReadData, ProcessData,…}。

2) DDP（definable DNS packet）指的是可定义的 DNS 数据包，记为 DDP={QNAME, RDATA, RawUDP}。

QNAME 表示将待传送信息进行处理后嵌入 DNS 查询区域的 QNAME 字段中，可以包含编码后的待传送数据、编码方法、序列号等及其各种组合方式。

RDATA 表示将命令/信息进行处理后嵌入 DNS 响应区域的 RDATA 字段中。

RawUDP 表示将数据处理后嵌入 DNS 分组载荷结束与 UDP 分组载荷结束为止的空间中（需构造 DNS 查询包）。

3) CMD 表示 DCCMalware 可执行的控制命令集合（DCCMalware 可以从 RDATA 字段中提取控制命令或其本身硬编码控制命令），记为 CMD = {rdata-derived cmd,hard-coded cmd}。

4) POLICY 表示保证数据高效可靠传输的策略，记为 POLICY={encode,encryption,CRC, query structure, time interval,…}。

5) MANS 表示恶意权威名称服务器。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，实现与受害设备集群的通信。

6) ConnectionType 表示连接类型，指的是受害设备与 MANS 的连接类型。中继信道是指受害设备通过本地默认解析连接到 MANS；直连信道是指受害设备与 MANS 通过 IP 地址直接连接。

7) δ 表示转换函数，反映了恶意程序收到命令后产生的相应动作及状态变迁，记为 δ：DCCMalware$\times S\times$CMD$->$DCCMalware$\times S\times$ ACTIVITY, 并满足 $\delta(\text{dccmalware}\times s_i\times\text{cmd})=(\text{dccmalware}\times s_j\times\text{activity})$, $i\neq j$。

3.2 构建机理

深入了解恶意软件的构建机理对于防御而言十分必要，图 4 显示了 DCC 的构建方式及通信过程。攻击者通过一定手段使 VictimMachine 感染 DCCMalware 后与其控制的 MANS 进行信息交互，通过 POLICY 定制的传输策略，将信息嵌入 DDP 传输，建立起攻击者与受控设备可靠、隐蔽的通信桥梁。VictimMachine 可以通过该信道向 MANS 泄露信息，MANS 可以通过该信道向 VictimMachine 发送控制命令，进而使攻击者获取敏感信息/控制设备的活动状态。

通过对 DNS 协议的分析发现，查询区域中除 QNAME 字段外，其他字段内容特定或可变动字符极少/一般不被恶意软件利用，故 QNAME 字段为该区域待传送信息嵌入的最佳位置；应答区域中 RDATA 字段为该区域信息嵌入的最佳位置。对 DNS 数据包进行分析发现，DNS 头中不包含资源记录或包总长度信息，解析器依赖报头中指定的资源记录数量确定解析数据，最后一条解析完成就认为到达了末尾。因此，可在 DNS 数据包末尾添加任意数量的数据，RawUDP 为信息嵌入的最佳位置。这样，就构成了对 DNS 数据包的利用。本文

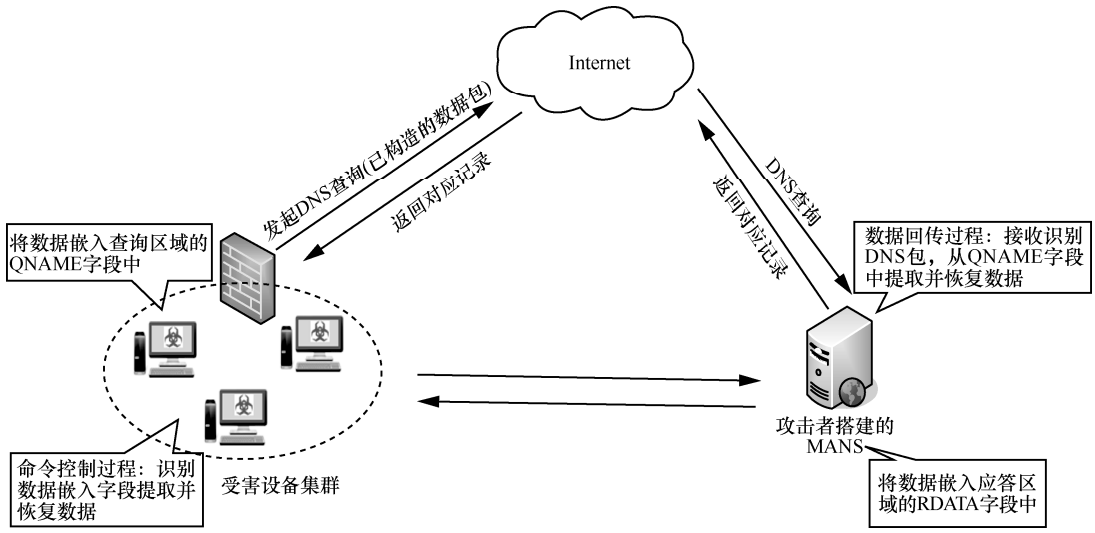

图 4 DCC 的构建方式及通信过程

利用这些字段构建信道进行隐蔽通信，并对其中涉及的内容进行了详细的说明（主要分析利用 QNAME、RDATA 字段进行构造的模式，利用 RawUDP 构造的模式较简单而不做赘述）。

3.2.1　基于 QNAME 嵌入的数据回传

将待传输数据嵌入 DNS 协议查询区域的 QNAME 字段中，可以将数据根据 POLICY 处理后嵌入。嵌入格式需要遵循 QNAME 字段规范，具有标签，标签间由句点间隔。每个标签的最大长度为 63 个字符，由字母、数字、连字符组成，且必须以字母或数字开头、结尾。QNAME 的最大长度为 253 个字符。受害设备将数据编码后嵌入，可以使用 Base64、Base32、Base16 及二进制、十六进制等方式进行编码，也可以使用 XOR、AES 等方式进行加密。将 DNS 查询包构造完成后，利用 DNS 查询请求将其传输到攻击者搭建的 MANS；MANS 接收识别 DNS 查询请求包，从 QNAME 字段中按照对应的 POLICY 提取并恢复数据，得到欲泄露的敏感信息或窃取的重要文件等。

以 APT34 中的 Helminth[38]为例，其 DNS 查询格式为 00 <系统标识符> <文件名字符> <序列号> <Base36 编码的、小于 46 655 的随机数> <十六进制数据>.<MANS 域名>，真实案例查询请求情况如图 5 所示。其使用特制域名进行信息传输，域名中包含系统标识符、文件名字符、分块序号、随机数、编码数据内容，利用请求将其引至 MANS 进行解析，从而将数据传输至 MANS，最终到达攻击者手中。

再如，对 APT34 核心组件 Glimpse[39]进行复现，该组件中的一种 DNS 查询格式为<GUID><数据包标识><操作类型字符><随机字符>C<数据包标识偏移量><操作类型偏移量>T.<传输内容>.<文件名称>.<MANS 域名>，使用 wireshark 抓包如图 6 所示。查询域名中第一个标签为查询结构字串，用来确定数据包分片序号及 C&C 服务器操作类型；第二个标签主要用来嵌入处理后的数据；第三个标签主要用来指示所属文件名称。

对该组织涉及的其他恶意软件 DNS 查询情况进行统计得到表 1，可以发现每种恶意软件具备独有的子域名构成方法，子域名中不仅包括编码后的数据，还包括一些序列号、随机数等辅助字串。同时，域名“go0gie.com”与“google.com”较类似。对 ATT&CK[21]上涉及的全部恶意软件进行统计发现，编码方式主要集中在 Base64、Base32 及 Hex 编码 3 种；每种恶意软件采用一种或几种固定的子域名结构来传送数据，一般包含辅助传输字串；可以一次传递一位字串[41]，也可以传输较多信息。当

图 5　真实案例查询请求情况

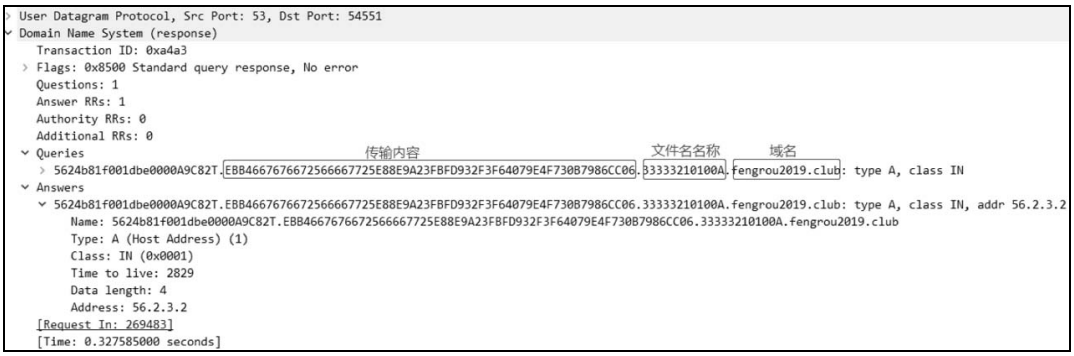

图 6　使用 wireshark 抓包观察情况

表 1 **APT34 恶意软件域名构成情况举例**

时间	恶意代码	子域名构成方法（编码）	字符长度/个	域名
2016 年	Helminth	00<SysI><FN><SeqN><RN><ED>（Hex）	48	go0gie.com
2017 年	ALMA Dot	<RN>.IDID.<Vid>.<SeqN>.<TSeqN>.<ED>.<FN>（Base16）	60	newusers.tk
2017 年	ISMAgent	<ED><SeqN>.d.<Vid>（Base64）	13	ntpupdateserver.com
2017 年	BONUPDATER	<RN>4<SeqN><SysI>B007（Base16）	50	poison-frog.club
2017 年	ALMA Dash	<RN>ID<Vid>-<SeqN>-<TN>-<ED>-<FN>（Base16）	20	prosalar.com
2018 年	QUADAGENT	<ED>.<RN>（Base64）	60	acrobatverify.com
2020 年	RDAT	<ED>.<EM><KEY>（Base32 or 64）	16	rsshay.com

注：<SysI>为系统标识符, <FN>为文件名, <SeqN>为序列号, <TSeqN>为序列总数, <RN>为随机数, <ED>为编码后的待传送数据, <EM>为编码方法, <Vid>为受害设备 ID, <TN>为总包数, <KEY>为加密密钥。

数据较大时，可分块传送。

3.2.2 基于 RDATA 嵌入的命令获取

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）。嵌入数据需要符合对应资源记录的格式规范，可以将数据根据 POLICY 处理后嵌入。例如，A 记录用来指定主机名对应的 IP 地址，只能使用 IP 地址；TXT 记录为域名设置说明，可使用文本信息，单个 TXT 记录不超过 255 B（不同服务商的实际限制会有所区别）。此外，还有 CNAME 记录、NULL 记录、MX 记录等，这里不做赘述。构造完成 DNS 响应包后，将其发送至受害设备。受害设备接收识别 DNS 响应包，从 RDATA 字段中按照对应的 POLICY 提取并恢复控制命令，得到 rdata-derived cmd。DCCMalware 根据 rdata-derived cmd 通过δ产生相应动作及状态变迁。

同样以 APT34 的 Helminth[38]为例，由于该样本请求 A 记录，DNS 应答格式为 IP 地址。其响应 33.33.x.x 表示为提供脚本文件名，指示恶意软件开始下载数据以保存到批处理脚本；33.33.33.33 指示恶意软件停止下载数据并执行下载的批处理脚本。真实案例响应情况如图 7 所示，33.33.97.97 中 97.97 指的是创建一个名为 aa.bat 的文件，数字 97 表示“a”的 ASCII 字符，可以利用这样的字符转换来传递信息及指令。

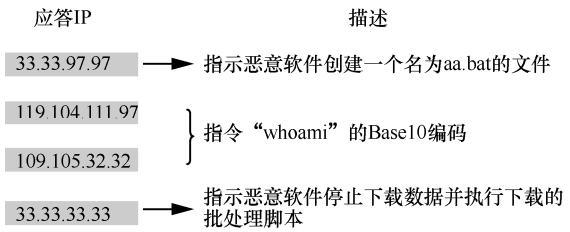

图 7 真实案例响应情况

再如，以复现的 Glimpse[39]为例。如图 8 所示，通过查询子域名 000564b81fdbDe0000A2C09T.fengrou2019.club 的 TXT 记录发现，Glimpse 使用 Base64 将命令编码在应答 TXT 记录中，TXT 记录中 d2hvYW1pJmlwY29uZmlnIC9hbGw=解码后为“whoami&ipconfig /all”。

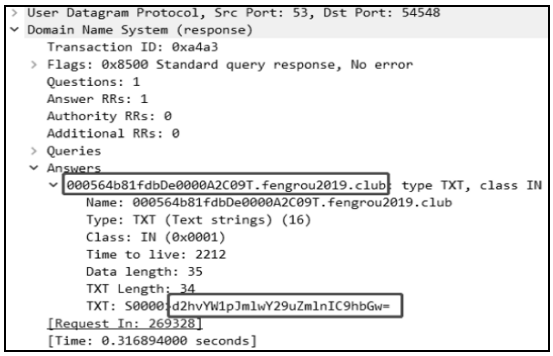

图 8 Glimpse 查询响应情况

为了解实际的资源记录利用情况，本文对一些主要的恶意软件及工具情况进行了统计，结果如表 2 和表 3 所示。从表 2 和表 3 可以发现，恶意软件应答字段记录类型一般为 A、TXT、AAAA 记录等；工具应答字段记录类型一般为 TXT、CNAME、NULL 记录，以获得更大的带宽，传递更多信息，可运行在 Linux 及 Windows 等主流操作系统上。

3.2.3 两者关系

基于 QNAME 嵌入的数据回传与基于 RDATA 嵌入的命令获取两者有一定关系但并不相同，前者主要利用查询泄露数据，后者主要利用应答获取命令，具体说明如下。

基于 QNAME 嵌入的数据回传主要为将处理后的数据利用 DNS 查询的 QNAME 字段进行泄露，

表 2　　**恶意软件利用应答字段记录类型情况**

时间	恶意软件名称	应答	连接方式
2011 年	Feederbot	TXT	中继
2011 年	Morto	TXT	中继
2014 年	FrameworkPOS	A	中继
2015 年	HTTPBrowser	TXT	中继
2016 年	Pisloader	TXT、Base32	中继
2016 年	C3PRO-RACCOON	CNAME、Base64	中继
2016 年	Helminth	A	中继
2017 年	Denis	NULL、Base64	中继
2017 年	Goopy	TXT、Base64	直连
2017 年	Matroyshka	A	中继
2017 年	POWERSOURCE	TXT	中继
2017 年	Ebury	A、TXT	中继
2017 年	ALMA Communicator	A	中继
2017 年	ISMAGENT	AAAA	中继
2018 年	BONDUPDATER	A、TXT	中继
2018 年	QUADAGENT	AAAA	中继
2018 年	RogueRobin	A、AAAA、TXT、CNAME、MX 等	中继
2019 年	Glimpse	A、TXT	中继
2020 年	RDAT	A、AAAA、TXT	中继

表 3　　**工具利用应答字段记录类型情况**

时间	工具	使用记录	平台
2004 年	OzymanDNS	TXT	Linux, Windows
2004 年	Dnscat-P/Dnscat2	A、CNAME	Unix
2006 年	Iodine	A、CNAME、TXT、MX、SRV、NULL、PRIVATE 等	Linux, Mac OS X, Windows
2007 年	TUNS	CNAME	Linux
2008 年	Dns2tcp	TXT、KEY	Linux, Windows
2008 年	tcp-over-dns	TXT	Windows, Linux and Solaris
2009 年	Heyoka	TXT、NULL	Windows
2011 年	DNScapy	CNAME、TXT	Linux
2015 年	Your Freedom	CNAME、TXT、MX、NULL、WKX	Windows, Mac OSX, Linux and Android
2015 年	ReverseDns Shell	A、TXT	Windows, Mac OS X, Linux
2018 年	DNSExfiltrator	TXT	Linux, Windows
2019 年	DNSlivery	TXT	Linux, Windows

对应的 DNS 响应一般为固定或连续的 IP 地址应答。数据泄露场景对应应答情况如表 4 所示，其响应并不嵌入命令，而是使用相同或连续的 IP 地址。例如，xHunt 的应答 IP 相同，Glimpse 的应答 IP 呈现递增趋势，避免产生大量 NXDOMAIN 过于异常的情况。

表 4 数据泄露场景对应应答情况

名称	域名	应答 IP 地址	规律
Wekby-xHunt	xxx7303d.windows64x.com	1.2.3.4	相同
	xxx73d3d.windows64x.com	1.2.3.4	
	xxx13033.windows64x.com	1.2.3.4	
APT34-Glimpse	xxx5E41A.malicious.com	41.2.3.1	递增
	xxx5E41A.malicious.com	41.2.3.2	
	xxx5E41A.malicious.com	41.2.3.3	

注：由于域名过长，xxx 表示省略子域名后五位之前的元素写法。

基于 RDATA 嵌入的命令获取主要为将处理后的数据利用 DNS 响应的 RDATA 字段进行传输，进而使受害设备可以通过响应获取命令，与响应对应的查询本身不用来泄露过多数据，而是一种获取命令的请求方式。如图 8 所示，000564b81fdbDe0000A2C09T.fengrou2019.club，其中“000564b81fdbDe”为 GUID，“0000A2”为随机字符，“C”为固定位，“0”为数据包表示偏移量，“9”为操作类型偏移量，“T”为固定位。可见，在获取命令的 DNS 响应所对应的 DNS 查询中，并没有泄露过多数据。

3.3 DCC 的高效可靠通信策略

为了保证数据可以从受害设备完整、隐秘地传输到 MANS 上，保证命令成功下达到受害设备上，成功窃取需要的数据/获得有效的命令，攻击者在传输过程中使用 POLICY 来保障信息的可靠传输。

1) 加入辅助传输字串/使用循环冗余校验（CRC, cyclic redundancy check）机制，保证传输数据完整性。由于 DNS 协议一般封装在 UDP 数据包中传输，UDP 只提供数据的不可靠传输[42]。服务器可能收到其他 DNS 查询请求或网络时延造成数据无法按正常顺序传输等情况，导致数据丢失/乱序，进而恢复失败。为避免此现象发生和信息失效，攻击者通常会在子域名中引入辅助传输字串。如表 1 所示，子域名构成时可以加入一些如序列号、序列总数、文件名等辅助传输字串来保证可靠通信。在接收端，识别构造结构，按照对应的方式进行处理，得到最终的传输信息；同时，攻击者也会在应答中引入 CRC 机制等来保证命令/信息传递的完整性。例如，最初的 Feederbot 就在其 DNS 应答 TXT 记录中引入了 CRC32，以此来保证僵尸网络命令的完整传输。

2) 在子域名/应答资源记录嵌入过程中，引入编码/加密机制，保护数据机密性。从受害设备中获取并利用 DNS 查询传递的信息可能为设备本身信息或敏感信息等，为了使数据隐秘传输/不被感知，这类信息一般不在网络上明文传输。攻击者引入编码技术，对待传送的信息进行编码后嵌入，再逐一传送。同时，攻击者欲窃取内容，通常不能完全符合 DNS 域名的语法规定，需要对目标内容执行编码转换，使转换后的内容基本符合 DNS 协议规范。在数据保密性要求较高的场景下，可以引入 AES 等加密算法，对传输的数据进行加密。但在传送之初，双方要协商好通信密钥，以便在接收端可以顺利解密出信息。

4 DCC 的检测方法

4.1 异常点

基于对构建机理的深入剖析可以发现，DCCMalware 构建的 DNS 查询及响应数据包与正常的 DNS 数据包有所差别。由于 DNS 协议信息承载量较小，QNAME 及 RDATA 的长度都有限制，例如请求中 QNAME 长度不能超过 253 个字符。若要传输成百上千的 MB 级别的文件，势必要发出数十万甚至上百万规模的 DNS 请求。同时，需要对传输内容进行编码，从而不可避免地导致若干异常。从攻击者角度看，弱化异常势必会降低攻击效率。为了保证攻击效果，异常难以避免。防御方需要提取其中难以绕过的异常点进行检测，从而达到较良好的检测效果。通过对构建机理的深入剖析及对检测论文的通读，目前主要的检测方向是针对 DDP 及请求应答情况的异常进行分析。将异常特征概括归为单域名异常及多域名统计异常 2 类，并分别对其进行阐述。

4.1.1 单域名异常

异常点可以反映在基础特征、可读性特征、结构性特征 3 个方面，下面进行详细介绍。

1) 基础特征

DNS 数据包长度/UDP 长度。DNS 协议一般封装在 UDP 数据包中传输且 DNS 分组载荷结束与 UDP 分组载荷结束的位置一致。攻击者为了传输信息，可能会在 DNS 分组载荷结束与 UDP 分组载荷结束之间嵌入数据，造成与正常 DNS 数据包的差异。例如 Iodine 具备直连模式，当发现可以与 MANS 通过 IP 地址直接连接时，就会切换到该模式，在 DNS 分组载荷结束与 UDP 分组载荷结束之间嵌入

数据，使用 RawUDP 进行通信。

子域名长度。攻击者将待泄露数据处理后嵌入查询中的 QNAME 字段传输，会尝试在子域名中放入尽可能多的数据来获得更高的带宽，导致与正常子域名在长度上的差异性。如表 1 所示，DCCMalware 的子域名一般较长，而正常的域名如“scholar.google.com”，其中子域名“scholar”一般较短。

子域名数字数量/占比、子域名大写字符数量/占比。将数据加密/编码后，会产生更多的大写字符及数字字符，导致与正常子域名的差异性。如图 5 所示，子域名 005aa003P3T647071636F6E626C316E385C61646D696E7369747261746F 中大写字符和数字字符的比例较高，而在正常域名如“store.google.com”中，“store”不含大写字符及数字字符。同时，部分情况下子域名允许使用二进制数据，攻击者可能用此编码方式来扩展信道带宽，从而使其具备较高的数字占比。

子域熵。正常子域名通常是一些看起来有意义的字符串，而恶意子域名通常是被加密/编码后看起来无意义的数据。一般地，信息被编码后会呈现更大的熵，此差距可以成为衡量正常子域名与恶意子域名的一个指标。

C2 域名欺骗性。一般地，默认使用 Alexa top 列表中的域名发出的请求为正常请求。攻击者为使 C2 域名看起来更正常，可能会为其 MANS 注册具备欺骗性的域名。如表 1 所示，恶意软件使用了类似 google.com 的 go0gie.com 域名，从而呈现欺骗性。因此可以通过计算恶意域名与常用域名的相似性来进行判断。

资源记录分布及长度。DNS 一般用于域名与 IP 地址间的映射，所以请求记录通常为 A/AAAA 记录。对主要的 DCCMalware 使用的资源记录进行统计，如表 2 和表 3 所示，DCCMalware 也可能会使用 TXT、CNAME 等记录进行信息传递，造成与正常请求的差异。同时，由于要将待传递信息嵌入资源记录中，如图 8 所示，所以资源记录的长度也是一个重要的衡量指标。

2) 可读性特征

子域名包含单词数、子域名中最大单词长度。正常子域名一般由多个单词构成且最大单词长度合理，恶意域名中提炼不出单词或最大单词长度异常。例如“zhidao.baidu.com”为正常域名，5624b81f001dbe0000A9C82T.EBB466767667256666677725E88E9A23FBFD932F3F64079E4F730B7986CC06.33333210100A.fengrou2019.club 为恶意域名。可以看出，正常域名、恶意域名在这 2 个方面的差异性。

子域名单/双/三字母频率。正常子域名一般属自然语言遵循 Zipf 定律，而恶意子域名字符频率分布更均匀。

3) 结构性特征

子域名标签数量。正常子域名一般含有较少标签，而恶意域名可能具有较多标签，所以标签数量也是一个可以参考的指标。例如，正常域名“map.baidu.com”，子域名具有一个标签；恶意子域名如表 1 所示，ALMA Dot 子域名具有 7 个标签。

平均/最大标签长度。正常子域名标签一般较短，恶意子域名可能含有较长的标签长度。正常域名如“tieba.baidu.com”，其中标签“tieba”仅含 5 个字符。恶意子域名如图 6 所示，Glimpse 查询请求中的第二个标签表示传输内容，其标签长度为 60 个字符，明显较长。

4.1.2 多域名统计异常

同位相同字符数/最大公共子串长度。同位相同字符数是指 2 个同域子域名中相对位置相同的字符数，最大公共字串长度指的是 2 个同域子域名中所有公共子串中最长的公共子串长度。正常子域名没有固定结构，而同一次恶意活动中的子域名一般具有相同结构，如表 1 所示。相应地，正常流量几乎不会有大量字符相同的情况，也不会具备较长的公共子串。而恶意子域名所携带的标志信息会成为它们之间的相同字符，且可能存在较长的公共子串。以 Glimpse 为例，图 9 给出了部分查询流量，它们具备相同的公共子串。

最大公共子串是否包含数字/字母。正常域名间

图 9　恶意子域名间相同字符情况

一般不具备相似的结构特征，而同一次恶意活动恶意域名间一般具备相似结构。如表 1 中的 BONUPDATER 所示，同一次攻击在特定位置上具有“4”“B007”这样相同的字母及数字，而几个正常域名间一般不具备这样的特征，造成正常域名与恶意域名间的差异。

同域 IP 离散性。正常域名中同域子域名对应 IP 地址较离散，如正常域名 www.baidu.com 对应应答 IP 地址为 39.156.66.14，域名 map.baidu.com 对应应答 IP 地址为 111.206.208.32，域名 tieba.baidu.com 对应应答 IP 地址为 112.34.111.194。对于恶意域名请求，攻击者会制定 MANS 对查询的应答规则，一般使用固定 IP 或递增 IP 应答。如表 4 所示，恶意软件 Wekby-xHunt 同域应答 IP 相同，APT34-Glimpse 同域应答 IP 递增，从而造成正常域名与恶意域名间同域 IP 应答差异。

DNS 数据包总数。在正常情况下，例如在企业局域网中，每天的 DNS 数据包总数一般在一个固定区间范围内。当 DCCMalware 运行时，一般会产生较大量的 DNS 查询，DNS 数据包总数则可能超出正常情况下的统计区间。

DNS 请求频率。在正常情况下，一般为手动输入 DNS 发起请求，请求频率有限；在恶意情况下，由于 DNS 协议可传输字符有限，恶意软件欲传输大文件时，需要大量 DNS 请求才能完成传送。为提高攻击效率，可能会使用较高频率的请求来泄露数据。所以，DNS 请求频率也是一个比较重要的指标。

DNS 请求应答比。在正常情况下，DNS 请求应答比大概为 1:1；在恶意情况下，可能只利用 DNS 请求来泄露数据而不设置应答，大量的无应答构成异常。

应答码。应答码 RCODE 表示对应响应的状态。在正常 DNS 请求与应答中，应答码一般为 0，表示 DNS 请求应答过程成功完成。当应答码为 3 时，表示此域名没有任何类型的解析记录。在恶意软件如 Heyoka 中，其将响应设为应答码为 3 的简单 NXDOMAIN 应答。当利用其传输数据产生大量请求时，会伴随着大量 NXDOMAIN 应答。所以，应答码的情况也可以作为一个参考指标。

同域请求数量/占比。在正常情况下，一般良性的域不太可能被同一设备经常反复查询；在恶意情况下，恶意软件需要对同一域名反复查询来泄露信息。所以，同域请求数量/占比是一个可以用来衡量异常的指标。

由于 DCC 在进行恶意活动过程中产生的流量与正常流量有所差别，因此可以利用这些异常点对恶意活动进行检测。研究人员对检测方式进行了一系列研究，利用基于匹配的方法及机器学习、深度学习等方法进行检测。接下来，将对这些检测方法涉及的具有代表性的论文进行逐一说明并进行对比，提出相关问题。

4.2 传统检测方式

2010 年，Born 等[43]通过分析 DNS 查询和响应中域名单字母、双字母和三字母的字符频率检测 DCC。可以使用该方法检测的原因是自然语言遵循 Zipf 定律，而隐蔽信道流量的字符频率分布更均匀。该文使用 n-gram 字符频率分析法，对前 100 万个顶级域名及 Iodine、Dns2tcp 等隐蔽信道工具进行分类，清楚地显示了合法流量和隐蔽信道传输数据的区别，但仅使用字符频率的差别来进行二分类的方法并不灵活，容易绕过。Karasaridis 等[44]提出了一种基于流的检测方法，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。2013 年，Ellens 等[45]结合了流量信息和统计方法进行异常检测，特别在流方面使用了每个流的字节数、每个流的数据包数、每个数据包的字节数和流持续时间等特征，通过阈值法、Brodsky-Darkhovsky 法、分布法并进行结合，实现了 5 个检测器，并对其进行比较，指出不同场景适合不同方法。2014 年，Kara 等[46]全面分析了恶意载荷分布位置，提出了一种基于资源记录分布情况来检测 DCC 的方法，并使用近实时数据评估了其有效性。

4.3 经典机器学习方法

传统检测方式一般利用基于规则的静态阈值[47]，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。有针对恶意域名检测的论文[48-49]，也有专门针对 DCC 检测的论文。在 DCC 检测方面，研究人员做了不少工作，但也存在一些较棘手的问题。恶意活动一旦被发现，攻击者控制的 DNS 服务器很容易被关停，造成活性样本较少，真实攻击数据匮乏，对 DCC 的检测面临更严峻的数据集问题。对利用经典机器学习检测发展以来的代表性论文进行总结归纳，厘清发展趋势，提出存在问题，更好地服务于防御工作。

在非监督学习检测方面，2011 年，Dietrich 等[22]提出利用 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，实验结果显示无误报，同时也可以实现对未知样本的检测。

在监督学习检测方面，2013 年，Aiello 等[50]将传统贝叶斯引入 DCC 检测中，使用查询、响应包大小及统计特征共 12 个特征进行检测，并评估了其检测方法的可靠性；同年，章思宇等[51]通过分析 DCC 流量特性，提取可区分特征 12 个，利用 J48 决策树、朴素贝叶斯和逻辑回归 3 种方法分别进行训练检测，可以检测已经训练的及未训练的隐蔽信道，但其训练集样本数量较少，导致准确率不高。2016 年，Buczak 等[52]使用随机森林算法进行检测，对未知工具的检测率仅有 95.89%。2017 年，Liu 等[53]使用支持向量机、决策树和逻辑回归 3 种算法综合了 4 种特征（18 种行为特征）针对已知隐蔽隧道工具集进行二分类检测，获得了较高的准确率，并指出使用 SVM 算法的检测效果最佳。2018 年，Almusawi 等[54]提出了一种多标签支持向量机方法来检测和分类隐蔽信道，并与多标签贝叶斯分类器比较，不仅区分了正常流量及恶意流量还对 FTP、HTTP 及 POP3 等协议进行了分类[55-56]，但是只使用了隐蔽信道工具进行检测，未涉及对恶意软件的检测情况。2019 年，Nadler 等[57]提取了 7 种特征训练正常流量，通过 iforest 构建异常检测模型准确识别现有工具及恶意软件流量。2020 年，Ahmed 等[58]描绘了科研机构网络与校园网的 DNS 流量各项特征的密度图，基于密度图提出检测特征并使用 iForest 算法进行检测。实验主要针对利用 DNS 查询进行数据泄露的检测，使用窃密工具 DET 进行训练，对 DET 及未训练的恶意软件进行检测，达到了较高的准确率。

使用机器学习进行检测不可避免地会提到使用方法及数据集问题，对于实验正确性及效果都有一定程度的影响。本文对使用经典机器学习进行检测的论文进行通读，对重点论文利用的方法、训练集、测试集及结果进行统计对比，把握基于机器学习的检测发展情况。

从表 5 可以发现，目前论文的检测方法涵盖了监督学习及非监督学习方法。决策树及支持向量机等是研究者较为热衷的检测算法；训练集多使用隐蔽信道工具进行训练，测试集方面涵盖针对已知数据集[53-54]及已知、未知数据集[51-52,57-58]的检测；部分论文使用隐蔽信道工具进行实验，针对真实攻击的检测较为匮乏[51,53-54]。

目前，许多论文中研究人员使用 DCC 工具生成的恶意流量进行检测工作，缺乏对真实攻击流量的检测。对工具产生良好的检测效果，并不意味着该检测方式可以很好地应对真实攻击及未知攻击，研究者应当着重关注对真实 DCC 攻击的检测情况。

表 5　　利用经典机器学习进行检测的各主要论文情况

时间	文献	方法	训练集	测试集	目的	结果
2013 年	文献[51]	J48 决策树、朴素贝叶斯和逻辑回归	Iodine 、 Dns2tcp 、 DNSCat 、 tcp-over-dns、PSUDP	Iodine、Dns2tcp、DNSCat、tcp- over-dns、PSUDP 及 OzyManDNS、Heyoka	针对已知及未知数据集的二分类检测问题	J48 决策树 AUC 最大（平均性能最优），正检率为 95.6%，误报率为 0.15%
2016 年	文献[52]	随机森林	Iodine 、 DNSCat2 、 Cobalt Strike	Iodine、DNSCat2、Cobalt Strike、Pick Pocket	针对已知及未知数据集的二分类检测问题	对已知数据集：99.92%，对未知数据集：95.89%
2017 年	文献[53]	支持向量机、决策树和逻辑回归	Dnscat2 、 Iodine 、 Dns2tcp、OzymanDNS	Dnscat2、Iodine、Dns2tcp、OzymanDNS	针对已知数据集的二分类检测问题	使用 SVM 效果最佳，准确率为 99.96%，精度为 99.98%，召回率为 99.93%
2018 年	文献[54]	多标签支持向量机（Kernel SVM）	Iodine、Dns2tcp	Iodine、Dns2tcp	针对已知数据集的多分类检测问题	Kernel SVM 效果更佳，平均精度为 0.795，召回率为 0.805 6，F-measure 为 0.800 028
2019 年	文献[57]	iForest	良性流量	Iodine、Dns2tcp、FrameworkPOS、Backdoor.Win32.Denis	异常检测模型	阈值为 0.653，检测率为 100%
2020 年	文献[58]	iForest	良性流量	DET 、 Iodine 、 BernhardPOS 、 DNSMessenger、FrameworkPOS、DNSpoinage	针对未知数据的二分类检测问题	准确率为 99.50%，误报率为 0.55%

4.4 深度学习检测方法

由于经典机器学习检测方法面临特征选取问题，特征选择依赖于专家知识只能提取有限的特征，但实际流量中可能还含有一些隐含的统计特征未被人发现，故有学者提出使用深度学习进行 DCC 检测。早在 2009 年，Hind[59]就首次提出使用人工神经网络（ANN, artificial neural network）构建分类器对 DCC 工具进行检测的想法。随着深度神经网络近几年逐步发展，研究者开始使用深度学习对 DCC 进行检测。2019 年，Liu 等[60]将流量以字节为单位转换成向量矩阵，每一个字节通过独热编码（One-Hot）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对比，得到了较好的准确率及召回率。2020 年，张猛等[61]对 CNN 进行了改进形成 RDCC-CNN 方法，提取了 48 个表征元素，将其转换成灰度图片表征 DNS 流量数据，对隐蔽信道进行检测，达到了很好的准确率及误报率。同年，Wu 等[62]提出利用深度神经网络自动学习特征进行检测，学习正常 DNS 流量的特征，通过计算正常样本与恶意样本之间的均方误差来检测 DCC。本文对深度学习检测方面的 2 篇主要论文进行分析，如表 6 所示，主要针对使用方法、训练集、测试集实验结果等进行对比。

从表 6 可以发现，2 篇论文都使用 CNN 及其改进算法进行检测，检测方法较单一；2 篇论文都是针对已知的数据集进行检测，未体现针对未知样本的表现情况；完全针对隐蔽信道工具进行实验，缺乏对真实攻击的检测。同时，针对真实攻击、未知攻击的检测并不明朗，还有较广泛的思考空间及待挖掘价值。

5 结束语

DCC 作为一种有效的攻击手段，已经被用于大量的 APT 攻击中，对众多领域都造成了较严重的危害，是网络安全领域关注的热点。随着万物互联时代开启和 5G 技术的发展，联网设备数量将持续攀升，DCC 可能带来更严重的数据泄露及各类安全问题。

本文回顾 DCC 的发展历程，将其发展历程分为 3 个阶段，描绘了其演进过程，指出攻击组织化及隐蔽信道工具恶意化趋势；界定范围，形式化定义了 DCC，提出七元组并对构建机理进行了深入的剖析；提取 DCC 构建机理中难以改变的异常特征，进而更有针对地应对 DCC，更好地服务于检测工作。从 3 个方面（包括传统检测方法、经典机器学习检测方法及深度学习检测方法）对已有的检测工作进行归纳分析发现，传统检测方法面临阈值设置易绕过、重要特征提取不全面等问题，可能导致对未知攻击检测效果不理想；经典机器学习检测方法面临真实攻击数据匮乏、部分论文使用特征及数据集不全面的问题；深度学习检测方法起步较晚，除面临数据集匮乏问题，其涉及重点论文使用相同数据集进行训练、检测，对真实及未知攻击的检测较为匮乏，有较广阔的研究空间。未来研究可能会将传统方式与人工智能方式相结合进行检测，取长补短，达到更好的检测效果。

考虑到当前互联网发展及 DCC 发展的新趋势，研究人员务必对其进行深入研究，完善防御体系，协同安全研究团队及数据分析团队共同应对日趋严重的网络威胁。

参考文献：

[1] MOCKAPETRIS P V. Domain names-implementation and specification[R]. RFC Editor, 1987.

[2] HINCHLIFFE A. DNS tunneling: how DNS can be (ab)used by malicious actors[R]. Unit42, 2019.

[3] PISCITELLO D. What is a DNS covert channel?[R]. ICANN, 2016.

[4] ARENDS R. Domain name system (DNS) parameters[R]. IANA, 2020.

[5] Black Lotus Labs. Alina point of sale malware still lurking in DNS[R]. LUMEN, 2020.

[6] KREMEZ V. FIN6 “FrameworkPOS”: point-of-sale malware analysis & internals[R]. Sentinel LABS, 2019.

[7] REAVES J. Anchor project for Trickbot adds ICMP[R]. Sentinel LABS, 2020.

[8] BARBEHENN B. Threat assessment: Ryuk ransomware and Trickbot targeting U.S. healthcare and public health sector[R]. Unit42, 2020.

[9] FALCONE R. xHunt campaign: newly discovered backdoors using

表 6 利用深度学习检测方法进行检测的各主要论文情况

时间	文献	方法	训练集	测试集	目的	结果
2019 年	文献[60]	CNN	Iodine、Dns2tcp、Dnscat2、OzymanDNS、Reverse_DNS_hell	Iodine、Dns2tcp、Dnscat2、OzymanDNS、Reverse_DNS_ hell	针对已知数据集的二分类检测问题	准确率为 99.98%，精度为 1.00，召回率为 99.96%，F1-Score 为 0.999 8
2020 年	文献[61]	改进 CNN（RDCC-CNN）	DNSCat、Iodine、PSUDP、Dns2tcp、tcp-over-dns	DNSCat、Iodine、PSUDP、Dns2tcp、tcp-over-dns	针对已知数据集的二分类检测问题	准确率为 99.50%，误报率为 0.55%

deleted email drafts and DNS tunneling for command and control[R]. Unit42, 2020.

[10] FALCON R. OilRig targets middle eastern telecommunications organization and adds novel C2 channel with steganography to its inventory[R]. Unit42, 2020.

[11] EKMAN E. Iodine[R]. GitHub, 2021.

[12] ANDERSSON B. Iodine[R]. kryo.se, 2014.

[13] ARNO0X. DNSExfiltrator[R]. GitHub, 2017.

[14] RON. Dnscat2[R]. Skullsecurity, 2019.

[15] BORGES D. Reverse_DNS_shell[R]. GitHub, 2015.

[16] MILLER T. Reverse DNS tunneling staged loading shellcode[R]. Black Hat, 2008.

[17] DEMBOUR O. Dns2tcp[R]. GitHub, 2017.

[18] CIMPANU C. Iranian hacker group becomes first known APT to weaponize DNS-over-HTTPS(DoH)[R]. ZDNet, 2020.

[19] WINNTI GROUP: Insights from the past[R]. Quointelligence, 2020.

[20] PAGANINI P. China-linked Winnti APT targets south korean gaming firm[R]. Securityaffairs, 2020.

[21] Application Layer Protocol: DNS[R]. MITRE ATT&CK, 2020.

[22] DIETRICH C J, ROSSOW C, FREILING F C, et al. On botnets that use DNS for command and control[C]//2011 Seventh European Conference on Computer Network Defense. Piscataway: IEEE Press, 2011: 9-16.

[23] GRUNZWEIG J. New Wekby attacks use DNS requests as command and control mechanism[R]. Unit42, 2016.

[24] Global Research, Analysis Team. The projectsauron APT[R]. Kaspersky, 2016.

[25] SEALS T. OilRig APT drills into malware innovation with unique backdoor[R]. Unit42, 2020.

[26] WILHOIT K. OilRig uses updated BONDUPDATER to target middle eastern government[R]. Unit42, 2018.

[27] SZERB T. NSTX[R]. Nongnu, 2002.

[28] KAMINSKY D. Black ops of DNS[R]. Black Hat USA, 2004.

[29] REVELLI A. Introducing Heyoka: DNS tunneling 2.0[R]. SOURCE Boston, 2009.

[30] BORN K. PSUDP: a passive approach to network-wide covert communication[J]. Black Hat USA, 2010.

[31] Morto worm sets a (DNS) record[R]. Symantec, 2011.

[32] PEARSON O. DNS tunnel-through bastion hosts[R]. Gray-world.net, 1998.

[33] REVELLI A. Playing with Heyoka: spoofed tunnels, undetectable data exfiltration and more fun with DNS packets[R]. Shakacon, 2009.

[34] BORN K. PSUDP: passive network covert communication slides[R]. Black Hat USA, 2010.

[35] BROMBERGER S. DNS as a covert channel within protected networks[R]. NESCO, 2011.

[36] XU K, BUTLER P, SAHA S, et al. DNS for massive-scale command and control[J]. IEEE Transactions on Dependable and Secure Computing, 2013, 10(3): 143-153.

[37] New Framework POS variant exfiltrates data via DNS requests[R]. GDATA, 2014.

[38] FALCONE R. DNS tunneling in the wild: overview of OilRig's DNS tunneling[R]. Unit42, 2019.

[39] PAGANINI P. APT34: Glimpse project[R]. Securityaffairs, 2019.

[40] LEE B. Behind the scenes with OilRig[R]. Unit42, 2019.

[41] PAXSON V, CHRISTODORESCU M, JAVED M, et al. Practical comprehensive bounds on surreptitious communication over DNS[C]//22nd USENIX Security Symposium. Berkeley: USENIX Association, 2013: 17-32.

[42] BARR D. Common DNS operational and configuration errors[R]. RFC Editor, 1996.

[43] BORN K, GUSTAFSON D. Detecting DNS tunnels using character frequency analysis[J]. arXiv Preprint, arXiv:1004.4358, 2010.

[44] KARASARIDIS A, MEIER-HELLSTERN K, HOEFLIN D. NIS04-2: detection of DNS anomalies using flow data analysis[C]//IEEE Globecom. Piscataway: IEEE Press, 2006: 1-6.

[45] ELLENS W, ŻURANIEWSKI P, SPEROTTO A, et al. Flow-based detection of DNS tunnels[C]//IFIP International Conference on Autonomous Infrastructure, Management and Security. Berlin: Springer, 2013: 124-135.

[46] KARA A M, BINSALLEEH H, MANNAN M, et al. Detection of malicious payload distribution channels in DNS[C]//2014 IEEE International Conference on Communications. Piscataway: IEEE Press, 2014: 853-858.

[47] FARNHAM G. Detecting DNS tunneling[R]. SANS, 2013.

[48] BILGE L, KIRDA E, KRUEGEL C, et al. EXPOSURE: finding malicious domains using passive DNS analysis[C]//Proceedings of the Network and Distributed System Security Symposium. [S.n.:s.l.], 2011: 1-17.

[49] BILGE L, SEN S, BALZAROTTI D, et al. Exposure[J]. ACM Transactions on Information and System Security, 2014, 16(4): 1-28.

[50] AIELLO M, MONGELLI M, PAPALEO G. Basic classifiers for DNS tunneling detection[C]//2013 IEEE Symposium on Computers and Communications. Piscataway: IEEE Press, 2013: 880-885.

[51] 章思宇, 邹福泰, 王鲁华, 等. 基于 DNS 的隐蔽通道流量检测[J]. 通信学报, 2013, 34(5): 143-151.
ZHANG S Y, ZOU F T, WANG L H, et al. Detecting DNS-based covert channel on live traffic[J]. Journal on Communications, 2013, 34(5): 143-151.

[52] BUCZAK A L, HANKE P A, CANCRO G J, et al. Detection of tunnels in PCAP data by random forests[C]//Proceedings of the 11th Annual Cyber and Information Security Research Conference. New York: ACM Press, 2016: 1-4.

[53] LIU J K, LI S H, ZHANG Y Z, et al. Detecting DNS tunnel through binary-classification based on behavior features[C]//2017 IEEE Trustcom/BigDataSE/ICESS. Piscataway: IEEE Press, 2017: 339-346.

[54] ALMUSAWI A, AMINTOOSI H. DNS tunneling detection method based on multilabel support vector machine[J]. Security and Communication Networks, 2018: 1-9.

[55] HOMEM I, PAPAPETROU P, DOSIS S. Information-entropy-based DNS tunnel prediction[C]//IFIP International Conference on Digital Forensics. Geneva: IFIP Newsletter, 2018: 127-140.

[56] 单康康, 郭晔, 陈文智, 等. 基于混合分类算法模型的 DNS 隧道检测[J]. 通信学报, 2018, 39(S1): 53-57.

SHAN K K, GUO Y, CHEN W Z, et al. Detection of DNS tunneling based on combined classification algorithm model[J]. Journal on Communications, 2018, 39(S1): 53-57.

[57] NADLER A, AMINOV A, SHABTAI A. Detection of malicious and low throughput data exfiltration over the DNS protocol[J]. Computers & Security, 2019, 80: 36-53.

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

[59] HIND J. Catching DNS tunnels with A.I[R]. Defcon, 2009.

[60] LIU C, DAI L, CUI W J, et al. A byte-level CNN method to detect DNS tunnels[C]//2019 IEEE 38th International Performance Computing and Communications Conference. Piscataway: IEEE Press, 2019: 1-8.

[61] 张猛, 孙昊良, 杨鹏. 基于改进卷积神经网络识别 DNS 隐蔽信道[J]. 通信学报, 2020, 41(1): 169-179.
ZHANG M, SUN H L, YANG P. Identification of DNS covert channel based on improved convolutional neural network[J]. Journal on Communications, 2020, 41(1): 169-179.

[62] WU K M, ZHANG Y Z, YIN T. TDAE: autoencoder-based automatic feature learning method for the detection of DNS tunnel[C]//2020 IEEE International Conference on Communications. Piscataway: IEEE Press, 2020: 1-7.

[作者简介]

刁嘉文（1995− ），女，黑龙江林口人，北京邮电大学博士生，主要研究方向为网络安全。

方滨兴（1960− ），男，江西万年人，博士，中国工程院院士，主要研究方向为计算机体系结构、计算机网络、信息安全。

崔翔（1978− ），男，黑龙江讷河人，博士，广州大学教授，主要研究方向为网络安全。

王忠儒（1986− ），男，山东烟台人，博士，中国网络空间研究院高级工程师，主要研究方向为人工智能、网络安全。

甘蕊灵（1996− ），女，广西贵港人，北京邮电大学硕士生，主要研究方向为网络安全。

冯林（1995− ），男，重庆巫溪人，广州大学硕士生，主要研究方向为网络安全。

姜海（1976− ），男，陕西富平人，北京丁牛科技有限公司工程师，主要研究方向为网络安全、大数据、云计算。