Computer Science and Application 计算机科学与应用, 2021, 11(6), 1823-1833
Published Online June 2021 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2021.116187

Hans汉斯

基于子域上下文关系的DNS隐蔽信道检测方法

王杉杉[1]，杜　飞[2]

[1]中国联合网络通信有限公司河南省分公司，河南　郑州
[2]北京锐驰信安技术有限公司，北京
Email: move170@163.com

收稿日期：2021年5月28日；录用日期：2021年6月21日；发布日期：2021年6月28日

摘　要

目前，攻击者进行私密信息传输、信息泄露、恶意信息传播等活动的主要手段之一是使用DNS协议作为隐蔽信道，特别是在僵尸网络和匿名通信网络中。为此，提出一种基于子域上下文关系的DNS隐蔽信道检测方法，该方法不仅提取了请求应答时间间隔、请求/应答报文大小、子域熵值以及资源记录类型频率等基础异常流量统计特征，同时对子域内容本身及其上下文关系进行了特征学习和提取。实验结果表明，该方法获得了99%以上的精度和召回率，具有很好的检测性能。

关键词

DNS隐蔽信道，流量特征分析

Toward DNS-Based Covert Channel Detection Using Subdomain Context Relation

Shanshan Wang[1], Fei Du[2]

[1]Henan Branch, China United Network Communications Co., Ltd., Zhengzhou Henan
[2]Beijing Ruichixinan Technology Company Limited, Beijing
Email: move170@163.com

Received: May 28th, 2021; accepted: Jun. 21st, 2021; published: Jun. 28th, 2021

Abstract

At present, one of the main means for attackers to conduct private information transmission, information leakage, malicious information dissemination and other activities is to use the DNS protocol as a covert channel, especially in botnets and anonymous communication networks. To this end, a DNS covert channel detection method based on sub-domain context is proposed. This

文章引用: 王杉杉, 杜飞. 基于子域上下文关系的 DNS 隐蔽信道检测方法[J]. 计算机科学与应用, 2021, 11(6): 1823-1833.
DOI: 10.12677/csa.2021.116187

method not only extracts the basic anomaly traffic statistics such as request response time interval, request/response message size, sub-domain entropy value and resource record type frequency. At the same time, the sub-domain content itself and its context relationship are characterized and extracted. The experimental results show that the method achieves more than 99% accuracy and recall rate, and has good detection performance.

Keywords

DNS Covert Channel, Traffic Feature Analysis

1. 引言

当前，恶意软件利用 DNS 协议作为隐蔽信道进行信息窃取、恶意信息传播等活动日益猖獗。因此，在构建完善的安全监测防御技术体系中，研究 DNS 隐蔽信道检测技术是一项重要的研究内容。隐蔽信道最初是作为计算机系统内部安全威胁被提出，随着计算机网络技术的快速发展，隐蔽信道的概念逐步迁移到网络环境，称之为网络隐蔽信道。网络提供了非常丰富的协议，这些协议通常具有一些弱点，使得攻击者可以利用其作为隐蔽信道的良好载体。由于 DNS 协议是互联网基础协议，被广泛采用，DNS 协议几乎不会被安全设备策略阻拦，即使在一个企业内部网络中，也需要架设 DNS 服务器进行主机名解析。因此，基于 DNS 协议实现应用层隐蔽信道成为攻击者绕过网络安全策略进行数据传输的主要手段。现有的 DNS 隐蔽信道检测技术主要有基于负载分析的方法和基于异常流量分析的方法两种。基于负载分析方法的基本原理是对 DNS 数据报文中的有效负载进行分析以提取字符串特征或者统计特征[1]，该方法健壮性较差，不能有效应对复杂多变的隐蔽信道实现手段[2] [3] [4]。Kenton 等[5]提出使用字符频率分析进行 DNS 隐蔽信道检测。Qi 等[6]提出使用二元语法的词频分析对 DNS 报文中的域名进行分析，发现隐蔽信道流量的词频并不遵循 zipf 分布而是随机分布。基于异常流量分析的方法通常需要观察一定时间内一定数据量的流量，因此其数据依赖性较高。Ellens 等[7]提出通过检测单位时间内 DNS 报文速率来检测是否存在 DNS 隧道。章思宇等[4]提出通过检测单位时间内的回答数据的总字节数与合法请求具有明显的差异。Singh 等[8]采用 15 种流量行为统计特征进行受感染主机的单点检测。

综上，针对当前 DNS 隐蔽信道健壮性差和数据依赖度高的问题，本文提出一种基于数据流的 DNS 隐蔽信道检测方法，该方法的新颖性主要包括如下三个方面：1) 对 DNS 数据流中的子域序列进行了上下文关系特征建模和学习，进一步使用卷积操作对子域上下文特征进行了进一步高阶特征提取；2) 结合使用了请求应答报文时间间隔、请求应答报文大小、子域熵值、资源记录类型频率四种类别等 26 种启发式基础统计特征；3) 在检测模型上，在子域上下文特征和基础统计特征的基础上，使用全连接神经网络实现 DNS 隐蔽信道的分类器模型。

2. 相关工作

术语隐蔽信道(Covert channel)最早由 Lampson [2]于 1973 年提出，其给出的定义是“如果一种通信信道的设计目的不是用来传输信息，那么称信道为隐蔽信道。”之后，更多的研究人员在此定义的基础上不断丰富和完善。其中，Tsai 等[3]于 1990 年将安全访问策略的概念引入隐蔽信道，对隐蔽信道的本质进行了深入的阐述，其给出的定义为“给定一个强制安全策略模型 M 以及其在一个操作系统中的解释 I (M)，

I(M)中两个主体 I (Alice)和 I(Bob)之间的潜在通信是隐蔽的，当且仅当模型 M 中主体 Alice 和 Bob 之间的通信是非法的。”从上述定义中，可以发现，隐蔽信道是与公开信道相对应的一个概念，公开信道传输合法信息，而隐蔽信道是在公开信道的掩护下，采用一定的技术手段实现私密或非法信息的传输，具有极强的隐蔽性。但是要注意到，上述这些研究工作给出的定义都是在研究操作系统安全的背景下提出的，因此所给出的隐蔽信道定义主要是针对单个系统内部的，而不是面向网络环境的。直到 2004 年，Serdar 等[3]提出网络隐蔽信道的概念，其给出的定义为“网络隐蔽信道违反了系统的安全策略，它是通过网络传播将隐蔽信息泄露出去的技术手段，该方式具有很强的隐蔽性，可以逃脱安全设备的检测”。对比单个系统内部的隐蔽信道的定义和网络隐蔽信道的定义，可以看出网络隐蔽信道沿用了单个系统内部隐蔽信道的定义，其区别仅仅在于结合了网络环境特征研究隐蔽信道。根据隐蔽信道的位置，可以将隐蔽信道分为系统隐蔽信道和网络隐蔽信道。根据对信息采用的编码方式(或信息载体)，隐蔽信道可分为时间型信道和存储型信道[9] [10] [11]。本文所研究的是网络存储隐蔽信道，具体来讲是基于 DNS 协议的存储型网络隐蔽信道，通常简称为 DNS 隐蔽信道/DNS 隧道。

2.1. 网络隐蔽信道的基本构建原理

网络隐蔽信道主要利用 TCP/IP 模型中的协议来构建。TCP/IP 模型提供了 IP、TCP、UDP、ICMP 等协议。这些协议一个显著的缺陷是报文格式中存在不常使用的字段可以作为载体用于构建隐蔽信道。图 1 给出了 IP 协议和 TCP 协议报文格式中可能用于隐蔽信道通信的字段。网络隐蔽信道构建的基本思路如下：1) 选取常见的公开协议构建隐蔽信道，所选取的协议应该满足两点要求：第一，该协议不是网络安全检测软件或设备(防火墙、入侵检测系统等)的关注对象；第二，该协议构造的隐蔽信道能够提供一定的带宽能力；2) 通信双方事先协商好封装和提取隐蔽信息的规则(包括选用的字段、隐蔽信息的编码/加密方式)，发送方执行隐蔽信息的封装，接收方执行隐蔽信息的解封装。早期网络隐蔽信道主要关注点在网络层和传输层的协议，例如 IP 协议、ICMP 协议、TCP 协议、UDP 协议等。近年来，隐蔽信道的构建主要采用应用层协议，如 HTTP、DNS、FTP、SSH、TELNET 等。这是因为基于网络层和传输层协议所构建的隐蔽信道容易被安全产品检测出来[12] [13] [14]。

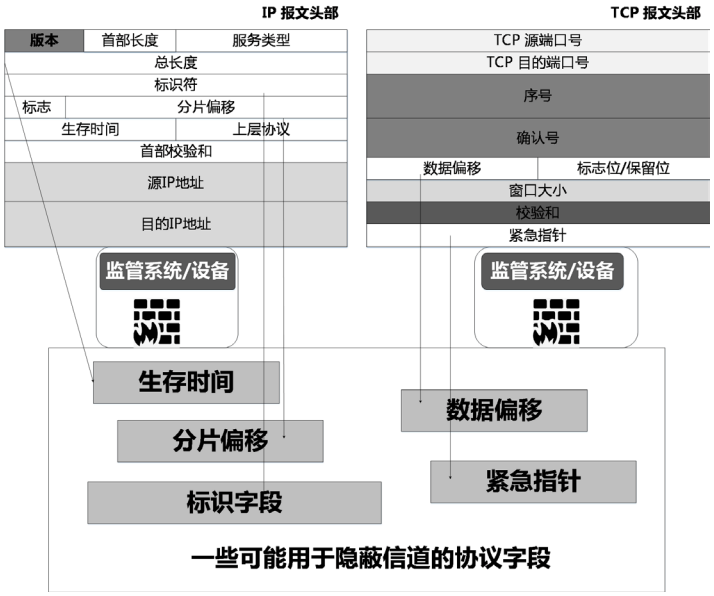

Figure 1. Protocol fields that may be used by network covert channels
图 1. 网络隐蔽信道可能使用的协议字段

2.2. DNS 隐蔽信道的构建

根据前述网络隐蔽信道的构建原理，可分为存储型隐蔽信道和时间型隐蔽信道。因此实现任何一种基于特定协议的网络隐蔽信道，也主要有两种构建手段。当前，主流的 DNS 隐蔽信道工具(OzymanDNS、DNS2TCP、Iodine 等[15] [16] [17])主要采用存储型构建技术。为构建一个 DNS 隐蔽信道，用户首先需要设置一个隐蔽信道客户端，该客户端在端口 53 上工作，向 DNS 隐蔽信道服务器发送请求。DNS 隐蔽信道服务器可以使用此创建的隧道通过 DNS 响应消息的 TXT 字段发出 C2 回调。这些数据包的有效负载可能会引入各种安全隐患。但是由于 DNS 流量的性质，这些安全隐患通常无法检测到。图 2 给出了 DNS 隐蔽信道的工作原理。

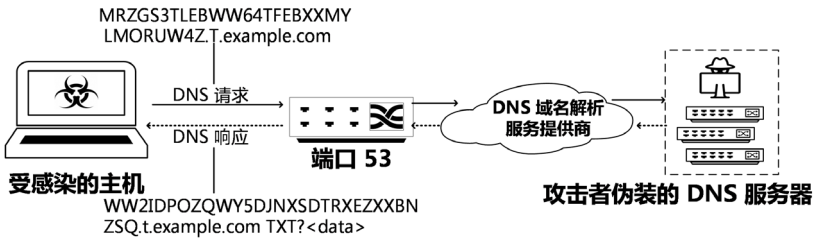

Figure 2. Schematic diagram of DNS covert channel principle
图 2. DNS 隐蔽信道原理示意图

受感染的主机通过向攻击者伪装的 DNS 服务器发送 DNS 请求，击者伪装的 DNS 服务器向受感染的主机响应 DNS 请求来构建隐蔽通道。

3. scHunter 检测模型

本文提出一种基于数据流的 DNS 隐蔽信道检测方法，称之为 scHunter (Subdomain-Context Hunter)。具体而言，首先对 DNS 隐蔽信道流量和正常 DNS 流量结合领域专家知识进行统计特征分析，提取出区分性强的、启发式的基础统计特征；进一步，在数据流子域序列进行上下文特征学习和卷积特征提取。从模型的训练和测试的角度，该方法的工作流程如图 3 所示。该方法分为两个阶段，第一个是离线训练阶段，负责根据有标签的数据进行模型的训练，第二个是在线测试阶段，负责使用已经训练好的分类器模型对新采集的未知 DNS 流量(即没有标签)进行检测，区分出其是合法 DNS 流量还是异常 DNS 隐蔽信道流量。其中的特征向量为基础统计特征和子域上下文特征的组合。

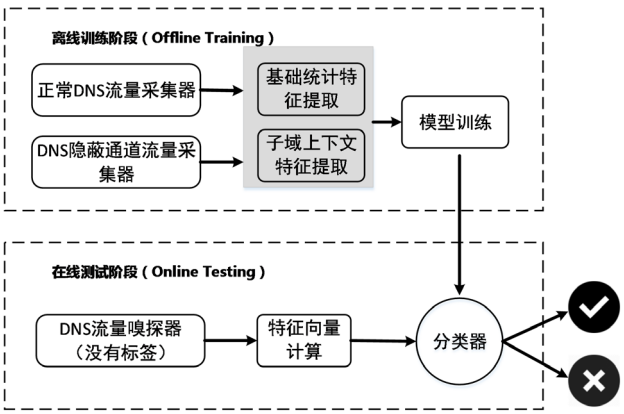

Figure 3. scHunter model work flow chart
图 3. scHunter 模型工作流程图

3.1. DNS 数据流定义

scHunter 检测模型是面向数据流的，一条 UDP 数据流是指一个具有相同五元组<源 IP 地址、源端口号、目的 IP 地址、目的端口号，UDP 协议>的数据包序列集合，不考虑方向性，即源 IP 地址、源端口号可以与目的 IP 地址、目的端口号互换位置，这里 UDP 协议为 DNS。如图 4 所示，表示一条源 IP 为 10.0.2.15，源端口号为 40197，目的 IP 地址为 45.77.39.243，目的端口号的 UDP 数据流，攻击 28 个数据包，14 个请求/响应对(req./resp.pairs)。

Figure 4. A UDP data stream generated by the DNSCAT tool
图 4. 一条 DNSCAT 工具产生的 UDP 数据流

3.2. 基础统计特征提取

在基础统计特征提取部分，本文主要分析提取了四种类别的特征：请求应答报文时间间隔(Request/Response Packet Time-Interval)、请求应答报文大小(Request/Response Packet Size)、子域熵值(Subdomain Entropy)、资源记录类型频率(Resource Record Type Frequency)。

1) 请求/应答报文时间间隔

请求应答报文时间间隔是指客户端发送一个请求报文到接受到对应的应答报文之间的时间间隔。通常，在 RDNS 服务器会缓存最近的查询和响应记录。在 DNS 应答报文中的每一条应答资源记录都有一个 TTL 字段，RDNS 会对这条资源记录缓存 TTL 给定的时间(通常为几秒)。DNS 隐蔽信道的每次请求报文的域名会有变动，不会命中本地缓存。因此，请求应答报文时间间隔可作为一种区分 DNS 隐蔽信道流量与正常 DNS 流量的特征。

2) 请求/响应报文大小

在树状层级结构的域名系统中，子域(subdomain)是指属于更高一层域的域，是一个相对于其父域而言的概念。例如，mail.example.com 是 example.com 的一个子域，而 example.com 则是顶级域.com 的子域。当前，对于每一级域名长度的限制是 63 个字符，域名总长度则不能超过 253 个字符。通常 DNS 隐蔽信道客户端将要发送的信息通过采用一定的编码方法(如 base32 或 base64 等)封装在 DNS 查询报文中的查询问题部分，具体是每个查询问题的主机名(Name)字段中。而 DNS 隐蔽信道服务器端将要传输的信息通过采用一定的编码方法后封装在 DNS 应答报文中资源记录部分，具体是每个资源记录的主机名字段。为

最大效率使用带宽以传输更多信息，DNS 隐蔽信道所产生的请求和响应报文较长，正常 DNS 请求报文中所提交查询域名长度适中。因此 DNS 请求报文大小可以作为一个区分特征。进一步，由于 DNS 隐蔽信道的服务端是控制端，其通常向客户端发送的控制命令比较短小精悍，比正常 DNS 响应报文大小相对较小，因此 DNS 响应报文大小也可作为区分特征。

3) 子域熵值

DNS 隐蔽信道中，在将要传输的信息存放在子域部分之前，通常会进行加密处理(如 base64、base32 等)，并且会大量使用 63 种字符集以外的字符。因此，可以通过检测子域的编写规范性作为一种识别特征。本文采用熵值进行子域规范性的量化度量。记 F 表示 DNS 报文中的子域，f_k 表示该子域有 k 个连续字符的集合，h_k 表示其对应的熵值，具体计算公式如下：

$$h_k = -\sum_{i=1}^{|f_k|} \frac{m_{ik}}{m-k+1} \log\left(\frac{m_{ik}}{m-k+1}\right) \tag{1}$$

根据公式(1)，对于包含 m 个字节子域的 F，可以得到它的熵值特征集合，使用 hb 代表子域 F 前 b 个字节的熵值特征集合。以 m 表示子域 F 包含的全部字节，以 b 表示 F 包含的前 b 个字节，以 ngram 代表连续字节的个数。本文对每一个子域所提取的熵值特征如表 1 所示。

Table 1. Subdomain entropy feature
表 1. 子域熵值特征

特征名称	特征描述
F1m	全部 m 字节的 1 连续字节(unigram)的熵值
F2m	全部 m 字节的 2 连续字节(bigram)的熵值
F3m	全部 m 字节的 3 连续字节(trigram)的熵值
F1b	前 b 字节的 1 连续字节(unigram)的熵值
F2b	前 b 字节的 2 连续字节(bigram)的熵值
F3b	前 b 字节的 3 连续字节(trigram)的熵值

4) 资源记录类型频率

DNS 是一个域名和 IP 地址相互映射的一个分布式数据库。因此在域名服务器上需要对每一个 IP 地址与域名之间的映射关系维持一个记录，这个记录称之为资源记录。另外，DNS 提供了多种丰富的资源记录类型，不同的记录类型有着不同的用途，常见的资源记录类可参见文献[17]。DNS 隐蔽信道工具更倾向于使用不常见的记录类型来封装信息，因为这些不常见的记录类型能够提供更大的带宽。例如，Iodine 的作者建议用户使用 NULL、PRIVATE、EDNS0 等记录类型。RFC 1035 中规定 NULL 记录允许相应数据报文的长度最高可达 65,535 字节，而 EDNS0 允许 DNS 数据报文长度可以超过 512 字节。

5) 基础统计特征总结

表 2 列出了本文所提取的 26 个用于区别正常 DNS 流量与 DNS 隐蔽信道流量的基础统计特征。

Table 2. Basic statistical characterization
表 2. 基础统计特征描述

特征类别	特征描述	特征个数
请求应答报文时间间隔	一个请求报文和一个应答报之间时间间隔的均值、方差	2
请求报文大小	请求报文大小的均值和方差	2
应答报文大小	应答报文大小的均值和方差	2

Continued

子域熵值	*F1m* 的均值和方差 *F2m* 的均值和方差 *F3m* 的均值和方差 *F1b* 的均值和方差 *F2b* 的均值和方差 *F3b* 的均值和方差	12
子域记录类型频率	A 记录的频率 AAAA 记录的频率 TXT 记录的频率 NULL 记录的频率 MX 记录的频率 CNAME 记录的频率 记录的频率 PTR 记录的频率	8

3.3. 子域上下文特征提取

1) 子域上下文特征向量学习

在基础统计特征提取部分，子域熵值特征主要对一个 DNS 报文中子域的内部结构进行了刻画，并没有捕获多个报文子域之间的上下文关系。这也是之前研究工作[18] [19]在子域特征提取方面的局限性。因此，本文提出对子域内容本身和多个子域之间的上下文关系进行特征提取。记 $F=\{F_1,F_2,\cdots,F_i,\cdots,F_n\}$ 为 DNS 数据流中子域序列集合，其中 $F_i=\{s_1,s_2,\cdots,s_j,\cdots,s_{n_i}\}$ 是由 n_i 个子域构成的序列。子域上下文特征提取的目标是在保留子域彼此之间的时间序列关系的同时，将每一个子域映射成低维空间的特征向量，该映射操作可数学形式化为似然概率最大化问题，即最大化子域序列集合 F_i 的概率，公式如下：

$$\max P(F)=\max\prod_{i=1}^{n}P(F_i) \tag{2}$$

其中，$P(F_i)$ 为 n_i 个子域的联合概率。为了降低最大化上述目标函数的复杂度，受 Skip-Gram 的启发，本文基于滑动窗口的思想将长度为 n_i 的序列 F_i 划分多个短小的子序列，也就是只保留子域和滑动窗口内子域的上下文关系，而忽略与较远位置子域的上下文关系。进一步，采用随机梯度下降算法对目标函数进行优化，便可获得每个子域 s_j 在实数空间的上下文特征向量。

2) 子域上下文卷积特征提取

基于自然语言模型的思想，通过对子域序列进行建模，便可获得每一个子域的向量化表示。然而该特征向量存在两方面不足：一方面该特征的学习并不是基于最终 DNS 隐蔽信道检测任务目标函数所获得，仅仅是一个初步学习的子域上下文特征向量；另一方面该特征在对一个 DNS 数据流中的所有子域进行向量拼接后，会形成一个维度较高特征向量，需要进一步降维。

因此，为提取子域上下文更高阶的特征，本文进一步对子域上下文向量进行卷积操作和池化操作，形成子域上下文卷积特征。这是受卷积神经网络(CNN, Convolutional Neural Network)在图像处理、语音识别、自然语言理解等研究领域的启发[13]。其主要特点是能够在初级特征提取基础之上，进一步提取更高阶的特征，这种高阶特征是算法根据目标任务自动调整。

3.4. 检测分类器构建

在对每一条样本进行基础特征提取和卷积特征提取之后，进行特征合并形成最终的特征向量，进一步，在此特征向量的基础上使用全连接神经网络进行二值分类，即最终输出有两个神经元。本文使用全连接的神经网络实现正常 DNS 流量和 DNS 隐蔽信道流量两种类别的二值分类任务，图 5 给出了分类器模型的神经网络结构示意图。

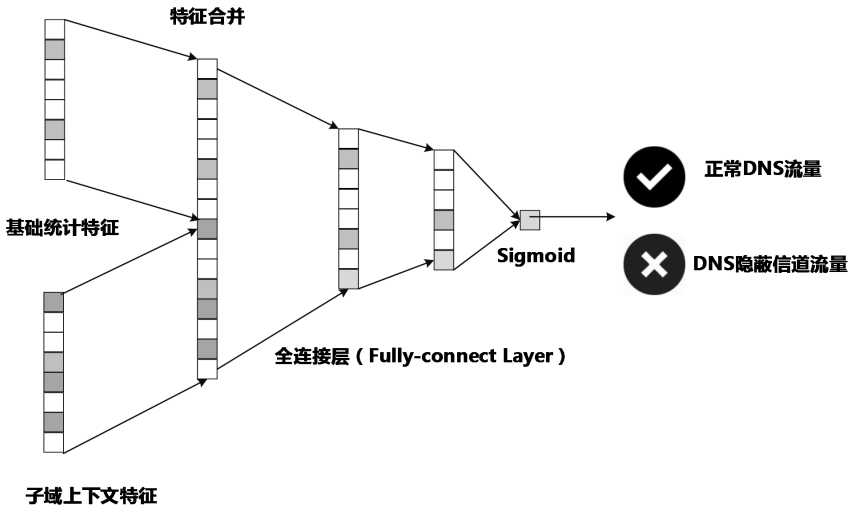

Figure 5. Schematic diagram of scHunter model neural network structure
图 5. scHunter 模型神经网络结构示意图

4. 实验结果与分析

4.1. 实验数据集

为了训练和评估 scHunter 模型，需要采集两部分数据：正常的 DNS 流量(白样本流量)和 DNS 隐蔽信道流量(黑样本流量)。本文从一个 ISP DNS 服务器上收集了正常的 DNS 样本流量。为收集异常的 DNS 隐蔽信道流量，本文搭建了一个专用网，在该网的多个端系统中部署运行了四种流行的 DNS 隐蔽信道工具(包括 DNSCAT2，DNS2TCP，Iodine 和 OzymanDNS)以产生流量，并在该专用网的路由关口处捕获。在收集好正负样本流量集后，将数据集分为两部分：训练集和测试集，训练集用于训练模型，包含 20 万 req/res 对正常 DNS 流量和 10 万 req/res 对隐蔽信道流量。测试集用于评估所学习好的模型，包含 10 万 req/res 对正常 DNS 流量和 5 万 req/res 对隐蔽信道流量。

4.2. 评估指标

为量化评估 scHunter 模型在未知 DNS 流量样本上的隐蔽信道检测能力。本文采用如下 2 个评估指标：

1) 精度(Precision)：在测试集数据中，scHunter 判定为 DNS 隧道流量的样本中确实是隧道流量的比例。

2) 召回率(Recall)：在测试数据集所有的隧道流量域名样本中，scHunter 能成功识别出其是隧道流量样本的比例。

4.3. 结果分析

1) 不同模型的对比分析

为评估本文所提出 scHunter 检测方法的有效性，本文将“基础统计特征 + 传统机器学习算法”的检测方法作为基线方法进行了实验对比分析，基线方法分别采用了决策树(Decision Tree)和支持向量机(SVM, Support Vector Machine)作为分类器模型。scHunter 在进行子域卷积特征学习时，有一个关键参数 N，表示滑动窗口的大小。表 3 给出了 N 取不同值的情况下的检测结果。从表中可以看出 scHunter 在精度和召回率均高于传统机器学习检测模型。另外，通过对漏报的隐蔽信道流量进行深入分析后，发现其中有一部分是隐蔽信道控制流量。隧道信道控制流量有两个目的：一是在不使用隐蔽信道的情况下防止服务器超时；二是在服务器有数据要发送给客户端时，确保 DNS 隐蔽信道服务器有可用于响应的来自客户端的请求报文。因此，DNS 隐蔽信道客户端每隔几秒就会发送一次请求。但是，这些隐蔽信道控制流量与正常 DNS 流量具有相同的行为特征。

Table 3. Test results of different models under different N values
表 3. 不同 N 值下，不同模型的检测结果

参数 N	scHunter		决策树		支持向量机	
	精度(%)	召回率(%)	精度(%)	召回率(%)	精度(%)	召回率(%)
1	100	99.7	100	95.4	100	97.5
3	100	99.5	99.7	95.7	100	98.1
5	100	99.6	100	96.3	100	98.6
10	100	99.3	99.5	96.9	100	98.7
15	100	99.5	100	97.1	100	98.5

2) 不同数据集的对比分析

为消除 DNS 隐蔽信道控制流量导致的召回率较低的问题。本实验构造了两个数据集：一个称为数据集 1，其中的隐蔽信道流量是由传输文件生成的；另一个是数据集 2，其是在数据集 1 的基础上用隐蔽信道控制流量替换了其中的一半的隐蔽信道流量。图 6 和图 7 分别给出了在数据集 1 和数据集 2 上精度和召回率的对比结果。可以发现，在加入隐蔽信道控制流量的数据集 2 上，召回率明显提升，召回率的提升意味着算法的适用性在提升。

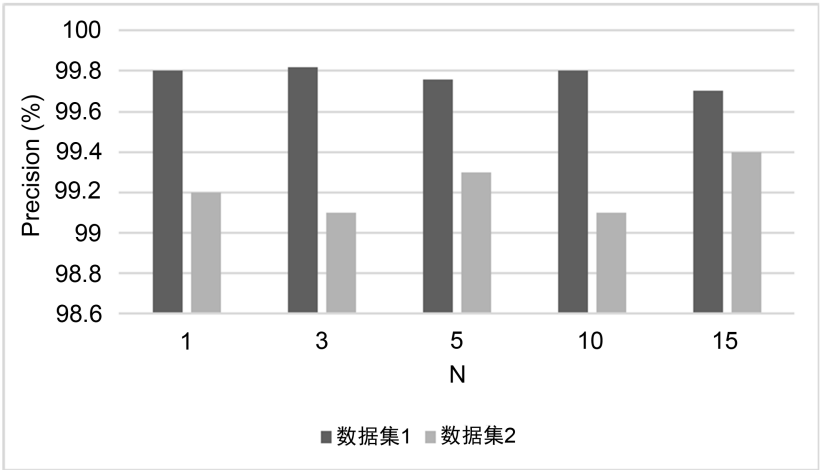

Figure 6. Comparative analysis of accuracy on data set 1 and data set 2
图 6. 数据集 1 和数据集 2 上的精度对比分析

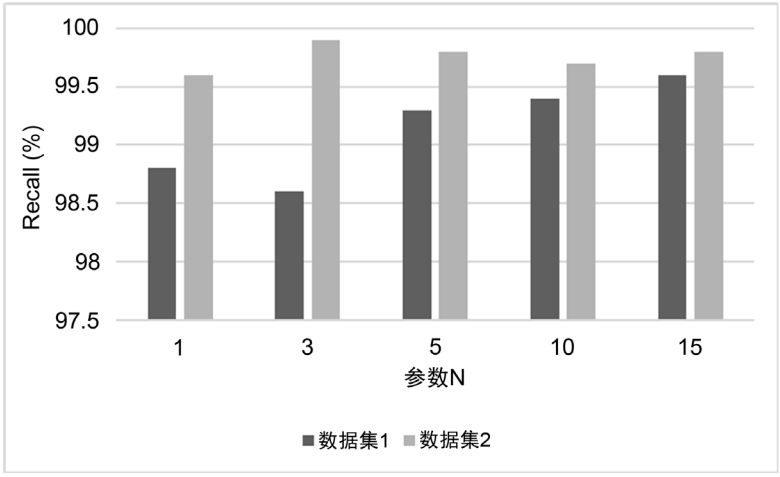

Figure 7. Comparative analysis of recall rates on data set 1 and data set 2
图 7. 数据集 1 和数据集 2 上的召回率对比分析

5. 结论

当前，DNS 是互联网基础设施服务，对用户使用其它互连网应用服务至关重要。近年来，DNS 隐蔽通道是恶意软件进行信息传输的重要手段之一。基于 DNS 协议的隐蔽信道技术一直是网络安全领域的重要研究课题。本文在深入分析 DNS 隐蔽信道构建技术的基础上，提出一种基于子域上下文关系的 DNS 隐蔽信道检测方法。实验评估结果表明，该方法获得了可观的检测精度和召回率，并能够在一定程度上有效缓解传统 DNS 隐蔽信道流量特征依赖人工分析的弊端。

下一步的工作中，隐蔽信道检测的研究方向可分为两个：一是提升检测算法的检测效果，二是提升检测算法的适用性。对隐蔽信道的检测算法研究是一项需要不断推进的工作，才能在复杂的网络环境下应对隐蔽信道带来的威胁。

参考文献

[1] Aiello, M., Mongelli, M. and Papaleo, G. (2015) DNS Tunneling Detection through Statistical Fingerprints of Protocol Messages and Machine Learning. *International Journal of Communication Systems*, **28**, 1987-2002. https://doi.org/10.1002/dac.2836

[2] 王永吉, 吴敬征, 曾海涛, 等. 隐蔽信道研究[J]. 软件学报, 2010, 21(9): 2262-2288.

[3] 谷传征. DNS 协议隐蔽信道的构建和检测技术研究[D]: [硕士学位论文]. 上海: 上海交通大学, 2012.

[4] 章思宇, 邹福泰, 王鲁华, 等. 基于 DNS 的隐蔽信道流量检测[J]. 通信学报, 2017, 34(5): 143-151.

[5] Born, K. and Gustafson, D. (2010) Detecting DNS Tunnels Using Character Frequency Analysis.

[6] Qi, C., Chen, X., Xu, C., *et al*. (2013) A Bigram Based Real Time DNS Tunnel Detection Approach. *Procedia Computer Science*, **17**, 852-860. https://doi.org/10.1016/j.procs.2013.05.109

[7] Romana, D.A.L. and Musashi, Y. (2008) Entropy Based Analysis of DNS Query Traffic in the Campus Network. *Journal of Systemics*, **6**, 42-44.

[8] Homem, I., Papapetrou, P. and Dosis, S. (2017) Entropy-Based Prediction of Network Protocols in the Forensic Analysis of DNS Tunnels.

[9] Ellens, W., Żuraniewski, P., Sperotto, A., *et al*. (2013) Flow-Based Detection of DNS Tunnels. In: *IFIP International Conference on Autonomous Infrastructure, Management and Security*, Springer, Berlin, 124-135. https://doi.org/10.1007/978-3-642-38998-6_16

[10] Singh, M., Singh, M. and Kaur, S. (2018) Detecting Bot-Infected Machines Using DNS Fingerprinting. *Digital Investigation*, **28**, 14-33. https://doi.org/10.1016/j.diin.2018.12.005

[11] Dietrich, C.J., Rossow, C., Freiling, F.C., *et al*. (2011) On Botnets That Use DNS for Command and Control. 2011 *Se-*

venth European Conference on Computer Network Defense IEEE, Gothenburg, 6-7 September 2011, 9-16. https://doi.org/10.1109/EC2ND.2011.16

[12] Zander, S., Armitage, G. and Branch, P. (2007) A Survey of Covert Channels and Countermeasures in Computer Network Protocols. *IEEE Communications Surveys & Tutorials*, **9**, 44-57. https://doi.org/10.1109/COMST.2007.4317620

[13] 李彦冬，郝宗波，雷航. 卷积神经网络研究综述[J]. 计算机应用, 2016, 36(9): 2508-2515.

[14] Kara, A.M., Binsalleeh, H., Mannan, M., *et al*. (2014) Detection of Malicious Payload Distribution Channels in DNS. 2014 *IEEE International Conference on Communications* (*ICC*), Sydney, 10-14 June 2014, 853-858. https://doi.org/10.1109/ICC.2014.6883426

[15] Almusawi, A. and Amintoosi, H. (2018) DNS Tunneling Detection Method Based on Multilabel Support Vector Machine. *Security and Communication Networks*, **2018**, Article ID: 6137098. https://doi.org/10.1155/2018/6137098

[16] Homem, I., Papapetrou, P. and Dosis, S. (2018) Information-Entropy-Based DNS Tunnel Prediction. In: *IFIP International Conference on Digital Forensics*, Springer, Cham, 127-140. https://doi.org/10.1007/978-3-319-99277-8_8

[17] List of DNS Record Types. https://en.wikipedia.org/wiki/List_of_DNS_record_types

[18] Shafieian, S., Smith, D. and Zulkernine, M. (2017) Detecting DNS Tunneling Using Ensemble Learning. In: *International Conference on Network and System Security*, Springer, Cham, 112-127. https://doi.org/10.1007/978-3-319-64701-2_9

[19] Nadler, A., Aminov, A. and Shabtai, A. (2017) Detection of Malicious and Low Throughput Data Exfiltration over the DNS Protocol.