基于改进卷积神经网络识别 DNS 隐蔽信道

张猛[1]，孙昊良[2]，杨鹏[2]

（1. 中国电子信息产业发展研究院网络安全研究所，北京 100846；2. 国家计算机网络与信息安全管理中心，北京 100029）

摘 要：为了全面有效地识别 DNS 隐蔽信道，对多种 DNS 隐蔽信道软件的实现方式进行了研究，提出了一种基于改进的卷积神经网络的 DNS 隐蔽信道识别方法。基于真实的校园网流量进行了实验，结果表明，所提方法可检测出全部 22 种数据交互模式的 DNS 隐蔽信道，并且具有识别未知的 DNS 隐蔽信道流量的能力。其识别性能的全面性和准确率相较于现有方法有显著提高。

关键词：隐蔽信道；域名系统；卷积神经网络

中图分类号：TP393

文献标识码：A

doi: 10.11959/j.issn.1000−436x.2020017

Identification of DNS covert channel based on improved convolutional neural network

ZHANG Meng[1], SUN Haoliang[2], YANG Peng[2]

1. Institute of Cyberspace, China Center for Information Industry Development, Beijing 100846, China

2. National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China

Abstract: In order to effectively identify the multiple types of DNS covert channels, the implementation of different sorts of DNS covert channel software was studied, and a detection based on the improved convolutional neural network was proposed. The experimental results, grounded upon the campus network traffic, show that the detection can identify twenty-two kinds of data interaction modes of DNS covert channels and is able to identify the unknown DNS covert channel traffic. The proposed method outperforms the existing methods.

Key words: covert channel, domain name system, convolutional neural network

1 引言

域名系统（DNS, domain name system）是现代互联网的基础设施和重要资源，其基本功能是实现域名和 IP 双向映射。借助于 DNS，应用程序可以使用字符串 example.com 等名称而不必使用复杂难记的数字 IP 地址，对互联网的普及发挥重要作用。由于 DNS 协议和 DNS 分组的普遍应用和合法性，大多数用户忽略了 DNS 可能存在数据泄露的隐患，一般单位更注重对经常受到攻击的流量进行安全监控，忽略了利用 DNS 协议传送信息的隐蔽行为。攻击者基于这点脆弱性，常利用 DNS 协议传送敏感信息。

当前很多应用借助 DNS 协议的普遍性达到隐蔽通信的目的。通过 DNS 进行信道传输的软件工具中，大多数的目的是借助 DNS 信道实现恶意代码远程控制的功能，或者传送敏感信息，窃取重要情报或数据。网络安全监测设备无论对网络通信行为做出多么严格的访问控制，通常都至少要允许 DNS 流量通过，这就为恶意通信行为提供了条件。

收稿日期：2019–04–11；**修回日期**：2019–12–11

通信作者：杨鹏，yp@cert.org.cn

基金项目：国家自然科学基金资助项目（No.61672495）

Foundation Item: The National Nature Science Foundation of China (No.61672495)

现在许多攻击者借助 DNS 隐蔽信道对信息系统进行攻击，窃取关键信息，破坏系统数据完整性和机密性。针对这一问题，许多不同类型的 DNS 隧道检测方法被提出。目前，DNS 隐蔽通信行为的检测主要包括通信流量与网络分组分析和基于域名字符串分析。通信流量与网络分组分析一般是分析多个域名请求，从不同角度计算这些 DNS 请求的总体属性，挖掘分析隐蔽信道。Crotti 等[1]和 Dusi 等[2]提出基于网络数据分组到达的先后顺序和分组尺寸等基本统计特征的分类机制，用于检测隐藏在 DNS、HTTP 和 SSH 信道中的信息传送流量。Casas 等[3]通过计算 DNS 通信过程中传送的数据量和单网络分组尺寸来分析是否存在 DNS 隐蔽通信的行为。Marchal 等[4]在分析被动 DNS 流量数据过程中，将机器学习方法应用在 DNS 信道检测，提取分组长度和字节数等属性，有针对性地挖掘分析了几种 DNS 信道。Karasaridis 等[5]基于被动 DNS 流量研究了 DNS 相关的隐蔽信道和缓存投毒等恶意行为，计算 DNS 分组尺寸的分布情况，通过分布差异性和交叉分布熵等统计属性标识出用于 DNS 隐蔽通信的分组，经真实网络数据实验验证，这个方法计算出了僵尸网络 Sinit 在爆发期间交叉分布熵的变化情况。Sheridan 等[6]搭建了隐蔽通信实验环境，收集了当前较为活跃的典型 DNS 隐蔽信道通信流量，提取其通信指纹，计算相应特征，在检测阶段，实施进一步匹配计算。Shafieian 等[7]提出使用集成学习的技术结合多种机器学习算法提高分类器的准确率和稳健性。以上方法主要适用于传送大量数据的隐蔽信道，对于传送少量信息、DNS 分组尺寸较小的隐蔽通信行为无法有效发现，Nussbaum 等[8]和 Aiello 等[9]在 DNS 隐蔽信道的信息传送能力方面开展了相关研究工作，分析了域名的主机名数、位置和域历史记录等信息，发现目前很多恶意代码通过 DNS 信道接收控制命令和传递敏感信息的分组尺寸与正常域名通信相近，而这些通信行为不具有以上可识别的属性。Nadler 等[10]专注于针对基于 DNS 协议的低吞吐量的数据泄露的恶意行为的检测，提出利用孤立森林（isolated forest）技术来检测基于 DNS 信道的数据泄露行为。在大规模数据集上能成功识别高吞吐量的 DNS 信道和低吞吐量的数据泄露行为。

在基于域名字符串分析方面，提取 DNS 分组中的有效负载内容部分，分析隐蔽信道域名的某些特征属性来确定 DNS 的请求和应答分组中是否被用于隐蔽通信。Farnham 等[11]利用正则表达式对网络流量中的域名字符串进行分析，发现网络中的隐蔽通信行为，并基于这个方法建立了可实用的商业化隐蔽信道检测系统，提取 DNS 分组中的特征字段，并与事先设定的阈值进行比对。Bilge 等[12]分析了正常的真实域名和用于 DNS 隐蔽通信域名的特点，计算最长有意义的字符串（LMS, longest meaning string）所占比例，同时应用累计和（CUSUM, cumulative sum）算法，分段统计域名字符分布情况。这两项工作是从字符串的角度分析用于隐蔽通信的域名表现出的特性，认为实际应用的真实域名通常由常见的有意义单词或其缩写组成，主要针对具有特殊域名字符串的 DNS 隐蔽通信，没有考虑模仿真实域名通信这一类 DNS 信道软件的隐蔽行为。

基于以上分析，目前传统的 DNS 隐蔽信道识别方法具有一定的倾向性，一般主要针对某几种 DNS 信道通信软件有较高的识别率，无法做到对绝大部分的 DNS 隐蔽信道的进行有效检测。近期，深度学习中的卷积神经网络在计算机视觉领域有良好的性能，充分展示了其特征提取的能力。本文认为同一类型的 DNS 信道在网络行为方面有着相似的行为模式，而网络行为模式是 DNS 隐蔽信道在需要实现特殊目的的前提下难以改变的特点。因此，本文提出了一种基于改进卷积神经网络的 DNS 隐蔽信道识别方法（RDCC-CNN, recognition of the DNS covert channel based on convolutional neural network）。这是一种识别整体 DNS 隐蔽信道的方法，通过分析 DNS 隐蔽通信流量特性，提取区别于合法流量的 DNS 隐蔽信道的表征元素数据，并将其转换成灰度图片，利用灰度图来表征 DNS 流量数据，对 DNS 隐蔽通信行为进行检测，并在校园网出入口的真实流量环境中开展实验，达到了理想的效果。

2 DNS 隐蔽信道表征元素分析

表征元素是指根据 DNS 流量来分析提取的某一特征值，经过数据转化成 0～255 之间的一个整数，作为 DNS 通信流量灰度图中的一个元素。

2.1 DNS 隐蔽信道

用 DNS 协议进行的隐蔽通信行为一般通过 2 种方法实现：一种是在域名解析的过程中，与某一

特定的服务器建立连接，这里称为标准 DNS 隐蔽信道；另一类通过客户端与隐蔽信道的服务器直接建立连接，这里称为非标准 DNS 隐蔽信道。对于标准 DNS 隐蔽信道，攻击者需要一台完全被控制的 DNS 解析服务器和一个已注册的域名，并将解析服务器设置为该域名的域名服务器，即可作为隐蔽信道的服务器。当隐蔽信道的客户端向任意的 DNS 递归服务器发送包含该域名下的子域名请求，需要发送的信息通过互联网域名解析系统的标准域名解析过程，传送到被控制的解析服务器。非标准 DNS 隐蔽通道能成功实施的前提是 DNS 隐蔽信道的客户端能与任一 DNS 服务器通信，攻击者将利用 UDP 封装的隧道服务绑定在服务器的 53 端口，即可从客户端直接建立连接。

2.2　表征元素选取

DNS 隐蔽信道的分组与合法的 DNS 分组相比存在许多不同点。一是从网络分组分析来提取相关的表征元素；二是考虑数据流的表征元素，数据流的表征元素有别于深度分组分析，主要通过计算网络分组的统计量得到。利用这些特征能有效地表征 DNS 隐蔽信道。

2.2.1　分组深度分析的表征元素

本节分析了 DNS 信道实现方法和过程，以及 DNS 网络分组的结构，考虑隐蔽信道 DNS 请求分组和响应分组的表征元素、深度分组检查中存在的问题和网络分组解析过程的表征元素，构造特征集来识别 DNS 隐蔽信道。

标签指针是在 DNS 协议中用于指向分组中每个存储标签长度的位置。Casas 等[3]提出在 DNS 分组中使用标签指针。标签指针的解析功能通常在 DNS 递归服务器和域名服务器中不会被实现，因此，加大了 DNS 分组的数据注入行为的检测难度。显然，这可以作为表征元素之一。在应答 DNS 请求的 A 类型记录请求时，通常会在 CNAME 记录中存放许多的服务器返回的应答数据，因此这是一个有辨识度的表征元素。

标准 DNS 隐蔽信道，在应答部分的资源记录中包含了绝大部分的服务器应答数据，使应答部分资源记录的数据长度之和与正常的应答分组的相同统计量相差较大，同时也影响了整个分组全部资源记录之和的计算结果。

此外，DNS 隐蔽信道会使用一些不常用的记录类型（例如 TXT 记录）来进行数据传输。因此，在检测的隐蔽信道的过程中，可以检查网络分组中使用的不常见的记录类型的数目。

考虑发起域名请求过程中隐含的特征，发现 DNS 隐蔽信道网络分组和正常的 DNS 网络分组的 QNAME 字段有较大的区别。因为根据 DNS 协议的规定，DNS 查询段除 QNAME 以外的字段仅含有限的数据空间，所以 QNAME 字段中包含了绝大部分的隐蔽信道的客户端发送的数据。其中，本文计算了 QNAME 的标签数量和 QNAME 二级域名部分的标签长度（域名的二级域名去掉顶级域名后剩下的字符串长度）。

DNS 隐蔽信道借助 DNS 协议进行通信，因此提高传输效率尤其重要。而 Marchal 等[4]的工作表明一种有效地提高传输效率方法是在域名数据中使用二进制数据。经过本文的实验证明，确实大多数现有工具建立的 DNS 隐蔽通道使用了非常规的字符。二进制串经过 Base32 编码后所表达的信息量与二进制串本身所能表达的信息量之比约为 3:5。而 DNS 协议中规定域名需经过 Base32 编码，这有别于 DNS 隐蔽信道所采取的编码方式。因此，QNAME 子域名承载的二进制数据量和 QNAME 子域名存储的数据量是 2 个重要的表征元素。例如，在实验环境中，DNSCat2 与 C&C 服务器通信的过程中的域名为 3f290169550l8bd5b7.malwareserver.com，则其有 3 个域名标签，二级域名标签的长度为 13 个字符，子域名二进制的数据量为 9 B。

正常的域名通常使用规范的单词或其他含有实际意义或便于记忆的字符串，而 DNS 信道编码的域名通常是杂乱无章，且人们所不能理解的。此外，DNS 隐蔽信道的域名各个字符出现的概率比较接近。杂乱无章的字符串具有较高的熵。因此，域名字符串的熵也是识别 DNS 信道的重要因素之一。

当 DNS 协议对合法的网络分组解析时，不会出现异常且正好完全解析（即不存在数据注入）的情况。而 DNS 隐蔽信道的解析情况往往与之相反，即解析异常且无法完全解析的可能性较大。所以，本文除了检查分组解析情况之外，还需要计算注入数据的数据量，即 DNS 分组载荷结束位置与 UDP 分组载荷结束位置的距离差。

DNS 信道实则是借助 DNS 协议的伪装而建立的隐蔽通道。显然，数据传输的过程需要借助 DNS 请求分组和 DNS 响应分组。因此，为了提高传输效率，要尽可能占用 DNS 分组或 Raw UDP 分组中所提供的所有的剩余空间。这导致 DNS 隐蔽信道

的请求分组和响应分组的长度区别于正常的 DNS 分组。但是因为无法将 Raw UDP 隧道的分组看作 DNS 分组解析，所以额外考虑 UDP 载荷长度。

如果组织内部的网络安全策略禁止所有未经过内部 DNS 服务器的 DNS 查询，那么有些 DNS 信道工具可能会违反该策略。但是仍有大多数 DNS 信道工具能够绕过该特征的过滤（即 DNS 请求经过内部 DNS 服务器转发后，依然能正常工作）。

本文对网络分组深度分析和考虑后，提取了 16 个网络分组的表征元素用于鉴别合法的 DNS 请求与 DNS 隐蔽信道流量，如表 1 所示。

表 1　　深度分组分析的表征元素

序号	表征元素名称	含义
1	Label_pointer	网络分组含标签指针
2	Exist_cname	应答分组含 CNAME 记录
3	Response_rdlength	应答部分资源记录数据长度总和
4	All_rdlength	全部资源记录数据长度总和
5	Num_unusual_record_type	不常用的记录类型的数目
6	Num_qname_label	QNAME 的标签数量
7	Qname_2ld_length	QNAME 二级域名部分的字符串长度
8	Qname_subdomain_2_data_amount	QNAME 子域名承载的二进制数据量
9	Qname_subdomain_data_amount	QNAME 子域名存储的数据量
10	Domain_str_entropy	域名字符串的熵
11	Parse_exception	网络分组解析是否异常
12	Injected_data_amount	注入数据的数据量
13	DNS_request_length	DNS 请求分组的长度
14	DNS_response_length	DNS 响应分组的长度
15	UDP_payload_length	UDP 载荷长度
16	Violate_strategy	是否违反策略

2.2.2 非标准 DNS 通信行为的表征元素

1) 相关定义

为了方便下文阐述，对相关概念进行简单定义。

定义 1 <source_ip, sd_name>。针对合法的且能完全解析的 DNS 分组，定义 DNS 通信双方为<source_ip, sd_name>。其中，source_ip 为 DNS 查询分组的源 IP 地址，或 DNS 响应分组的目的 IP 地址；sd_name 为 source_ip 请求的全域名(QNAME)去除相同域的域名标签后剩余的子域名字符串。

定义 2 <source_ip, destination_ip>。针对非法的 DNS 分组，即解析过程中出现解析异常或者不能完全解析，定义 DNS 通信双方为<source_ip, destination_ip>。其中，source_ip 为发起 DNS 请求的源 IP 地址，destination_ip 为 DNS 查询分组的目的 IP 地址，即提供域名解析服务的 IP 地址。如果是在网关处捕获的 DNS 分组，这里将内网主机地址视为 source_ip。

2) DNS 通信行为的表征元素

DNS 流量数据传输是一个持续的过程，仅有数据分组的静态表征元素无法满足检测任务的需求，因此，本文统计 DNS 通信行为的表征元素。算法首先将属于同一数据流的通信双方的网络分组依据分组顺序拼接；然后，统计选取的数据流内的表征元素；最后，综合所有表征元素，形成 DNS 通信流量灰度图对分类器进行训练。

除了上述的表征元素之外，本文根据分组捕获的实际情况、DNS 信道工具实验观测结果以及 DNS 信道的域名与域名生成算法（DGA, domain generate algorithm）生成的域名的相似性，选择了以下 3 个表征元素。

1) 通常情况下，正常的 DNS 请求和应答通信行为与其他类型的流量数据存在一定前后关联。例如，应用程序发起 HTTP 请求之前，一般会先发起相关的 DNS 查询，而 DNS 信道中只存在 DNS 请求的流量。因此，将是否存在独立的 DNS 请求和应答流量作为判断 DNS 信道的指标之一。对于存在的例外情况，设立白名单进行过滤。

2) DNS 信道工具在每次发起 DNS 请求时只会针对一个特定的域名，这造成了相同时间内对特定域名发起请求的主机名个数远多于对合法域名发起请求的个数。因此，主机名数量较多的域名很可能是 DNS 信道的域名[12]。

3) DGA 生成的域名（AGD, algorithm generated domain）大多是不存在的，因此识别 AGD 一个较有效的方法是寻找产生过多的不存在域名（NXDomain, non-existent domain）的响应[13]。考虑到 DNS 信道的域名与算法生成的域名在字符串的熵和域名长度等方面都非常相似，同样也会产生大量的 NXDomain 响应（如 Heyoka）。所以，可以借鉴该方法检测 DNS 信道。

如表 2 所示，非标准 DNS 通信行为的统计特征分为 4 个集合。Feature_Set_1 统计的特征值描述了客户端与服务器之间通信的流量大小。Feature_Set_2

则是从单个网络分组的角度，详细地统计了数据流中符合 DNS 隐蔽信道流量的表征元素。$Feature\_Set_3$ 是对网络分组中具体的参数进行统计，同时计算同一数据流中所有相应参数的均值、最大值和最小值等统计量，从而进一步描述流量。其中，注入数据的数据量、DNS 请求分组的长度、DNS 响应分组的长度和 UDP 载荷的长度，需要对发送和接收 2 个方向分别统计；QNAME 的标签数量、QNAME 二级域名部分的字符串长度和 QNAME 子域名存储的数据量，只对 DNS 请求分组统计；应答部分资源记录数据长度总和以及全部资源记录数据长度总和，只需要对 DNS 回答分组进行统计。$Feature\_Set_3$ 共 36 个特征，从双向 DNS 分组的特征、双向的流量大小、所包含的域名信息以及应答分组中资源记录的统计量等角度，计算了 DNS 隐蔽信道的表征元素。$Feature\_Set_4$ 包括独立的 DNS 请求个数、每个域名的主机名个数、NXDomain 响应的个数。例如在观察时间窗口内，本文实验中设置为 1 h，DNSCat2 通信记录如图 1 所示。统计该观察窗口内的 DNS 请求分组共 6 231 个，请求 malwareserver.com 下子域名的主机个数 5 个，NXDomain 响应次数为 5 989 次。

表 2　非标准 DNS 通信表征元素

特征集	特征集说明
$Feature\_Set_1$	客户端发送 DNS 分组的总数 客户端接收 DNS 分组的总数
$Feature\_Set_2$	异常解析 违反策略 标签指针 CNAME 记录 使用了不常见的记录类型 QNAME 中具有二进制的分组总数
$Feature\_Set_3$	网络分组特征 1、2、6、7、9、10、12、13、14、15 的统计值
$Feature\_Set_4$	独立的 DNS 请求个数 每个域名的主机名个数 NXDomain 响应的个数

```
src_ip1.1679 > dst_ip.53: 59036 [1au] TXT
35bc006955018b0021636f6d6d616e642073657373696f6e00.malwareserver.com.
dst_ip.53 > src_ip1.1679: 59036*- q: TXT
35bc006955018b0021636f6d6d616e642073657373696f6e00.malwareserver.com.
1/0/0 [1d] TXT "6c29006955d5b70000"
src_ip2.2584 > dst_ip.53: 41672 [1au] TXT
3f29016955018bd5b7.malwareserver.com.
dst_ip.53 > src_ip2.2584: 41672*- q: TXT
3f29016955018bd5b7.malwareserver.com. 1/0/0 [1d] TXT "11b8016955d5b7018b"
```

图 1　DNSCat2 与服务器通信记录

3　DNS 隐蔽信道识别方法

根据第 3 节分析的各类 DNS 信道软件的若干表征元素，针对每一条通信流量，提取并计算这些表征元素，通过数据规范化和预处理，将数据转换为 0～255 之间的整数，并构造二维矩阵，将网络流量特征数据转化为灰度图片，二维矩阵中的每一位数据对应灰度图片的一个元素；然后将灰度图片作为卷积神经网络的输入数据进行学习，训练并建立 RDCC-CNN 分类器，用训练好的分类器对 DNS 隐蔽信道进行识别。其整体架构如图 2 所示。

3.1　数据规范化

数据规范化是为了使数据的各个维度能够在同一量纲上，减少对模型训练时对不同维度的依赖程度，同时需要将数据转换为 0～255 之间的整数。

假设每条 DNS 流量的 m 个表征元素形成一个行向量，一共 n 条 DNS 流量所构成的数据矩阵为

$$\boldsymbol{T}=\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ A_{11} & A_{12} & \cdots & A_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{11} & A_{12} & \cdots & A_{1m} \end{bmatrix}$$

将所有样本中相同的表征元素，提取后形成一个列向量，那么该矩阵可以表示为

$$\boldsymbol{B}_i=[A_{1i},A_{2i},A_{3i},\cdots,A_{ni}]^{\mathrm{T}} \tag{1}$$

$$\boldsymbol{T}=[\boldsymbol{B}_1,\boldsymbol{B}_2,\boldsymbol{B}_3,\cdots,\boldsymbol{B}_m] \tag{2}$$

对每个表征元素列向量进行转换后得到 $B_i^{'}$，即

$$B_i^{'}=\left\lfloor\frac{\boldsymbol{B}_i}{\max(\boldsymbol{B}_i)-\min(\boldsymbol{B}_i)}\right\rfloor(255-0) \tag{3}$$

其中，$\max(\boldsymbol{B}_i)$表示在向量 $\boldsymbol{B}_i$ 中取最大元素，$\min(\boldsymbol{B}_i)$表示在向量 $\boldsymbol{B}_i$ 中取最小元素。转换后的 $B_i^{'}$ 为 0~255 之间的整数。

处理后的矩阵可以表示为

$$\boldsymbol{T}'=[B_1',B_2',B_3',\cdots,B_m'] \tag{4}$$

式(4)具体为上文所描述的 DNS 通信流量数据，根据每行表征元素的个数，构造合适的灰度图尺寸。DNS 通信流量数据的数据集具有 48 个表征元素，可以构建一个 8×8 的图片尺寸，由于表征元素的数量少于图片像素点个数，因此需要在图像矩阵的最后多余部分用 0 进行填充。例如，根据某一条

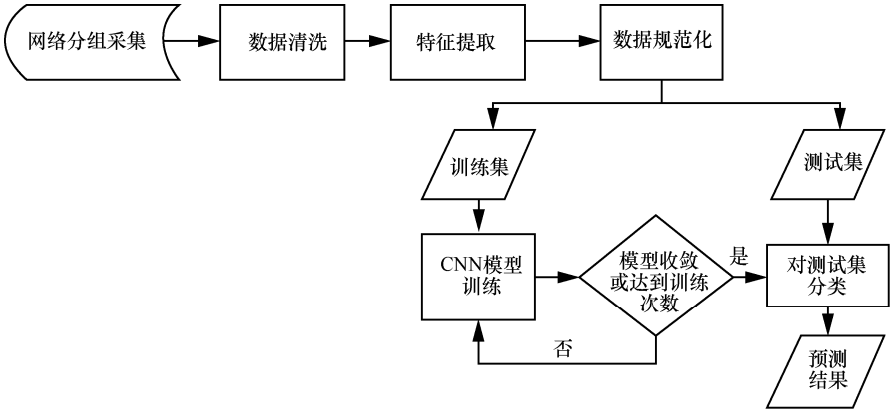

图 2 整体架构

DNS 通信流量数据的所计算得到表征向量为 $\left[A_{i1},A_{i2},\cdots,A_{im}\right]'=[5,1,\cdots,0]$。通过对每列的数据进行规范化表示得到

$$\left[A_{i1},A_{i2},\cdots,A_{im}\right]'=[68,32,\cdots,0]$$

将 $\left[A_{i1},A_{i2},\cdots,A_{im}\right]'$ 转换成 8×8 的矩阵，即可得到如图 3 所示的能够表征某一条 DNS 通信流量数据的灰度图。

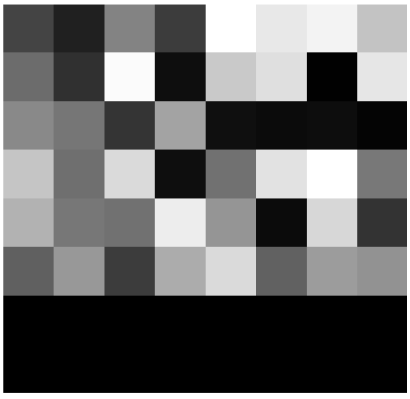

图 3 $\left[A_{i1},A_{i2},\cdots,A_{im}\right]'$ 所计算的灰度图

将每一个 DNS 隐蔽信道的网络分组数据转变成相应的灰度图片，作为卷积神经网络的输入。

3.2 卷积神经网络结构

1) 卷积层

每个卷积层由多个卷积单元（即滤波器）组成。卷积层中每一个节点的输入是上一层卷积核的局部区域的计算结果。卷积核的长、宽以及个数均可自定义。卷积层作用是对上一层输入的特征图片中局部区域进行进一步更高层次的特征抽取。卷积层的形式为

$$x_j^l=f\left(\sum_{i\in M_j}x_j^{l-1}k_{ij}^l+b_j^l\right) \tag{5}$$

其中，l 为当前层，b 为当前层的偏置，k 为卷积核，M_j 为第 j 个卷积核对应的卷积窗口，激活函数通常采用 sigmoid、tanh 或 ReLU 等函数。

2) 池化层

池化层也称为采样层，它是将输入的图像划分为若干个矩形区域，对每个子区域进行计算，能够不断地减小数据的空间大小。利用池化层，参数的数量和模型的计算量也会大大减少。其中，最常用的是最大池化和均值池化。池化层的形式为

$$x_j^l=f\left(\beta_j^l\mathrm{down}(x_j^{l-1})+b_j^l\right) \tag{6}$$

其中，down(·)为下采样函数。若下采样函数是最大池化，则对输入图像中每一个 $n\times n$ 大小的区域取最大值，用于代表这个区域的信息，因此经过下采样的图像尺寸缩小为原来尺寸的 $\frac{1}{n}$，输出的特征图像有对应的权重参数 β 和偏置 b。

3) 全连接层

全连接层是连接在模型最后的分类器，依据前面所提取的特征训练分类器。最后将从灰度图片中抽取的特征输入到全连接层中，并输出分类的标签。

3.3 改进卷积神经网络

卷积神经网络模拟了人脑在图像数据中提取特征的原理，并在计算机视觉领域取得了显著的效果。在本文提出的方法中，首先将 DNS 数据连接特征转换为数值表示，再把数值视为灰度

值，进而将数据转化为图像。最后将 DNS 隐蔽信道的识别任务转化成能采用卷积神经网络解决的分类任务。

LeNet-5 模型是卷积神经网络的经典结构，主要应用在手写数字的识别[13]。针对 DNS 通信流量数据特点，本文根据实际应用场景更改了模型结构。

1) 根据 DNS 通信数据特性，转化出适当灰度图片，输入层设计为 8×8 矩阵。

2) 识别 DNS 隐蔽信道是一个二分类任务。因此，LeNet-5 模型的输出层 10 个神经元改为 2 个神经元。

3) 根据 DNS 数据连接特征，设计了 6 种具有不同网络结构的卷积神经网络。如表 3 所示。

3.4 改进后 CNN 的训练过程

卷积神经网络模型的训练过程主要采用反向传播算法来传递误差信息，运用梯度下降的优化算法来更新神经元之间连接的权重。

1) 前向传播。其计算方法可以表示为

$$x_j^l = f\left(\sum_{i\in M_j} x_i^{l-1}k_{ij}^l + b_j^l\right) \tag{7}$$

式(7)使用的激活函数为 ReLU 函数。

2)反向传播。为了传递误差信息的计算方法，需要结合优化算法一起更新神经网络中的参数值。这里使用的目标函数定义为

$$E = -\frac{1}{n}\sum_{x_j}\left[y_j \ln a_j^l + (1-y_j)\ln(1-a_j^l)\right] + \frac{\lambda}{2n}\sum_{\omega}\omega^2 \tag{8}$$

其中，等号右边第一项多项式为常规交叉熵表达式；第二项为正则化项，这里使用正则化项是为了避免模型的过拟合，同时设定惩罚因子λ=0.01。

为了能最优化目标函数的结果，需要对参数进行调整。

权重的更新为

$$\omega \to \omega - \eta\frac{\partial E^n}{\partial \omega}$$

偏置的更新为

$$b \to b - \eta\frac{\partial E^n}{\partial b}$$

其中，η为学习速率，设$\eta = 0.55$。

具体实现过程如下。

1) 对输入的图像进行 Padding 操作，能够保持边缘信息，且使卷积前后尺寸不变。同时卷积核移动步长设为 1。

2) 设计卷积核的数量不大于 64，避免 5 层卷积神经网络陷入局部最优，无法达到全局最优。

对于每一种设计的网络，均采用最大池化和平均池化这 2 种池化层进行实验，并进行性能对比。实验结果分别如表 4 和表 5 所示。基于改进卷积神经网络 DNS 隐蔽信道检测方法，由于其应用场景对于实时性的要求较高，因此，本文只针对各种算法的测试时间进行对比分析。

表 4　采用最大池化方法时的检测结果

编号	总体正确率	测试时间/s
模型 1	88.98%	50.21
模型 2	96.96%	48.32
模型 3	97.78%	42.15
模型 4	98.68%	41.23
模型 5	98.92%	36.57
模型 6	99.65%	35.56

表 3　6 种不同网络结构的卷积神经网络

编号	C1 卷积层		S2 池化层		C3 卷积层		S4 池化层		F5 全连接层	
	卷积核	输出	采样窗口	输出	卷积核	输出	采样窗口	输出	卷积核	输出
模型 1	48×(3×3)	48×(8×8)	2×2	48×(4×4)	96×(3×3)	96×(4×4)	2×2	96×(2×2)	96×(2×2)	96×1
模型 2	32×(3×3)	32×(8×8)	2×2	32×(4×4)	64×(3×3)	64×(4×4)	2×2	64×(2×2)	32×(2×2)	32×1
模型 3	16×(3×3)	16×(8×8)	2×2	16×(4×4)	32×(3×3)	32×(4×4)	2×2	32×(2×2)	64×(2×2)	64×1
模型 4	16×(3×3)	16×(8×8)	2×2	16×(4×4)	32×(3×3)	32×(4×4)	2×2	32×(2×2)	32×(2×2)	32×1
模型 5	8×(3×3)	8×(8×8)	2×2	8×(4×4)	16×(3×3)	16×(4×4)	2×2	16×(2×2)	64×(2×2)	64×1
模型 6	8×(3×3)	8×(8×8)	2×2	8×(4×4)	16×(3×3)	16×(4×4)	2×2	16×(2×2)	32×(2×2)	32×1

表 5 采用平均池化方法时的检测结果

编号	总体正确率	测试时间/s
模型 1	97.98%	50.57
模型 2	98.10%	48.59
模型 3	98.32%	41.59
模型 4	98.33%	40.35
模型 5	97.99%	36.39
模型 6	98.67%	35.19

从表 4 和表 5 可以看出，随着卷积核数的增加，总体上，分类准确率基本保持不变，但运行时间差距比较明显。在应用最大池化的池化层运行时，模型 1 在实验过程中发生了局部最优点的问题，但是从模型 2~模型 6 的实验结果中可以看出，适当地减少卷积核的数量，能够提高模型分类性能。对比表 4 和表 5 中的实验结果，模型 6 在取得较高分类准确率的前提下，实现较低的计算复杂度。因此，应用了最大池化的模型 6 被选为最终的分类器模型。

参照如表 3 所示模型 6 的网络结构，在 C1 卷积层对输入图像进行 Padding 操作后，使用权重随机生成的 8 个 3×3 的卷积核对输入的矩阵进行卷积运算，输出 8 张 16×16 的特征图像。在 S2 池化层，采用 2×2 尺寸的窗口对上一层输出的 8 张特征图像进行下采样，输出 8 张 8×8 的特征图像。在 C3 卷积层，同样也是 Padding 后使用预训练得到的 16 个 5×5 的卷积核对输入数据进行卷积，得到 16 张 8×8 的特征图像。在 S4 池化层使用 2×2 尺寸的窗口对 C3 层的 16 张特征图像进行下采样，得到 16 张 4×4 的特征图像。在 F5 全连接层，将 S4 层所得到的表征 DNS 通信行为的图像矩阵拉伸成单一的 128×1 向量。输出层利用 soft-max 分类器，输出结果为 2 类。

4 实验测试与结果分析

4.1 实验设置

1)数据集

本文实验中采集的数据集是混合流量，包含背景流量的正常 DNS 流量样本和包含背景流量的 DNS 隐蔽通道流量样本。其中，正常 DNS 流量样本采集自某高校校园网出入口，在 20 天内，提取 480 h 单固定 IP 对单一全域名请求和应答数据 558 400 条，人工确认这些连接均不属于 DNS 隐蔽信道流量，取 300 000 条作为训练数据，其余作为测试数据。

DNS 隐蔽通道流量样本通过几个典型 DNS 信道软件生成，这些 DNS 信道软件在实际应用中出现较为频繁，主要包括 DNSCat、Iodine、PSUDP、Dns2tcp 和 tcp-over-dns。其中，DNSCat、Dns2tcp 和 Iodine 能实现多类型资源记录通信，可分别产生 CNAME、KEY、SRV、MX、NULL、TXT 等类型资源记录，同时生成 Raw UDP 模式下的流量。在可控的网络环境中运行这些软件，每种 DNS 信道软件在 5 台主机上运行，分别获取激活状态下有信息传输时的流量和未激活状态下没有信息传输时的流量，使训练产生的模型在传输数据时和无数据传输空闲时都可有效识别。另外，PSUDP 通过向现有的 DNS 网络分组中注入数据来传送信息，软件本身不发送任何 DNS 请求，因此以随机时间发送 DNS 请求，模拟真实网络环境中的 DNS 分组，以承载 PSDU 的数据。这些 DNS 隧道软件主要存在 22 种数据交互模式。以 1 h 为时间段，这里对每种模式分别按照一天 24 个时段获取 20 天交互信息，共产生 10 560 个 DNS 隐蔽通道流量样本，采用其中 10 天 1 472 000 条数据作为训练数据，另 10 天的 1 472 000 条数据用来测试，如表 6 所示。

表 6 被动 DNS 数据和信道样本集

样本类型	训练样本数	测试样本数
校园网 DNS 流量	300 000	279 200
Iodine（6 类资源记录）	312 000	312 000
Dns2tcp（2 类资源记录）	320 000	320 000
DNSCat（2 类资源记录）	320 000	320 000
tcp-over-dns	280 000	280 000
PSUDP	240 000	240 000

测试集中的背景流量直接在校园网出入口处通过镜像获取，捕获 10 天的出入口流量作为背景流量。

2) 评估指标

为了验证 DNS 信道的识别效果，这里设立了 3 个评测指标，分别为准确率 ACC、误报率 FPR 和整体准确率，包括对每一类 DNS 信道的识别，具体为

$$\mathrm{ACC}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i} \tag{9}$$

$$\mathrm{FPR}_i = \frac{\mathrm{FP}_i}{\mathrm{TP}_i + \mathrm{FN}_i} \tag{10}$$

$$\mathrm{OAcc}_i = \frac{\sum_{i=1}^{n} \mathrm{TP}_i}{\sum_{i=1}^{n} \left(\mathrm{TP}_i + \mathrm{FN}_i\right)} \tag{11}$$

其中，TP_i（true positive）表示隐蔽信道 i 被正确识别出的样本数量，FN_i（false negative）表示实际类型为 i 的信道样本被误判为其他类型的样本数量，FP_i（false positive）表示实际类型非 i 的信道样本被误判为类型 i 的样本数量。整体准确率主要用来评测识别方法对利用 DNS 分组进行通信行为的识别效果。

3) 实验环境

本文原型系统的开发及训练测试工作在 2 台服务器上进行，每台服务器使用深度学习框架 PyTorch，卷积神经网络改进过程的训练与测试均在 PyTorch 环境下完成。试验的硬件环境应用 128 GB 内存，中央处理器 Intel Xeon E5-2620，每处理器含八核，主频 2.00 GHz，采用 8 块 GeForceGTX TITAN X 系列 GPU，软件环境的操作系统为 64 位的 Ubuntu 16.04，cuda8.0，模型的实现采用 Python 语言。

4.2　结果分析

1) 测试集样本识别

传统的 DNS 隐蔽信道检测方法一般包括 DNS 有效分组分析和流量分析 2 个方面。DNS 分组分析典型方法是 Farnham 等[11]提出的使用正则表达式对域名进行匹配的 DNS 检测模型；而 Karasaridis 等[5]提出使用网络流量的统计量来检测 DNS 异常机制是流量分析工作的代表之一。其通过在一定时间窗口内计算数据分组尺寸的分布情况，然后分析比较实验中得到的分布和标准分布估算出一个交叉分布的熵值，利用该熵值进行对比检测。近两年，Shafieian 等[7]认为传统的 DNS 隐蔽信道检测方法存在不足，如内容分发网络（CDN, content delivery network）的广泛使用增加了传统检测方法的误报，并提出使用集成学习技术对 DNS 隐蔽信道进行检测；此外，根据 Nadler 等[10]调查研究显示，低吞吐量的 DNS 隐蔽信道常用于信用卡信息和密码等隐私信息的泄露，因此提出利用孤立森林的方法针对低吞吐量的基于 DNS 协议的数据泄露行为进行检测，并在大规模的数据上测试模型的性能。为了方便讨论，将其分别称为 Farnham 模型、Karasaridis 模型、Shafieian 模型和 Nadler 模型。本文还原了这 4 个模型，并与本文提出的方法进行了对比，以验证本文提出的方法在 DNS 信道识别问题上的优势。

针对建立的数据样本集，利用选取的网络特征来表征每一条 DNS 流量，选取网络流量数据的特征并生成表征矩阵，经过标准化计算转化为灰度图片，作为 RDCC-CNN 的输入。根据第 3 节选取的最优卷积神经网络模型，将提取的 DNS 流量特征合并，并按照文中的归一化操作生成灰度图片的表征矩阵作为输入。先通过训练数据集建立检测模型，然后将测试数据生成的特征矩阵依次输入模型进行实验分析。模型训练了 50 000 步长（Step），针对不同类型的 DNS 信道目标函数的损失（Loss）收敛情况如图 4 所示。根据实验数据计算评价指标，实验结果分别如表 7 和表 8 所示。

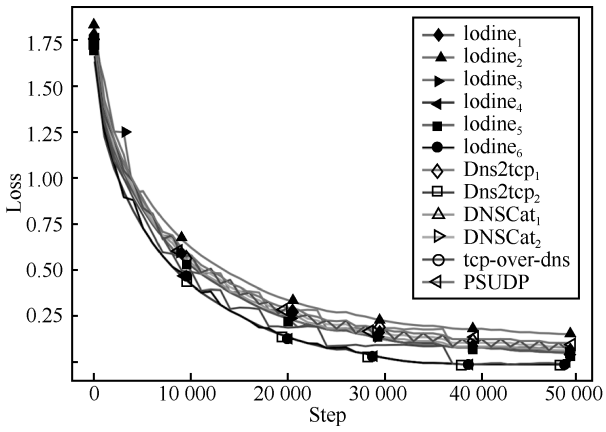

图 4　目标函数损失（Loss）收敛情况

表 7　采用不同算法的总体准确率和测试时间

方法	总体准确率	测试时间/s
Farnham 模型	93.96%	5.11
Karasaridis 模型	96.86%	4.66
Shafieian 模型	97.23%	5.00
Nadler 模型	98.37%	5.01
本文方法	99.50%	5.03

从表 7 中可以看出，本文提出的改进卷积神经网络的方法的总体准确率达到 99.50%，相较于其他 4 种方法有一定的提升。Farnham 模型针对全量域名，提取相关的表征特征，并进行复杂的正则表达式匹配计算，其消耗的时间相对较长，相对于本文提出的方法多出 0.08 s，而且仅计算域名的相关信息，因此识别的总体准确率表现较低；Karasaridis 模型计算时间较快，其使用网络流量的统计量来识别 DNS 隐蔽信道，在一段时间内，计算网络分组尺寸分布的熵，相对来说选取的特征类型少，计算复杂度低，比本文提出的方法快 0.37 s，但总体识别准确率低于本文提出的方法。

表 8 各类型 DNS 信道识别准确率

类型	Farnham 模型	Karasaridis 模型	Shafieian 模型	Nadler 模型	本文方法
$Iodine_1$	98.6%	98.1%	99.1%	99.3%	99.8%
$Iodine_2$	91.9%	93.6%	97.9%	97.5%	99.3%
$Iodine_3$	93.8%	95.6%	97.3%	97.8%	99.1%
$Iodine_4$	96.9%	97.2%	98.6%	98.5%	99.9%
$Iodine_5$	78.6%	87.9%	96.2%	96.8%	99.8%
$Iodine_6$	75.9%	88.9%	97.2%	97.6%	99.3%
$Dns2tcp_1$	61.8%	68.7%	97.4%	97.1%	99.0%
$Dns2tcp_2$	96.2%	93.3%	98.2%	98.1%	99.9%
$DNSCat_1$	91.2%	90.9%	95.9%	96.1%	99.9%
$DNSCat_2$	91.7%	92.6%	96.9%	97.0%	99.6%
tcp-over-dns	77.1%	79.1%	92.1%	93.3%	98.3%
PSUDP	78.6%	78.9%	91.2%	92.1%	96.9%

本文方法相对于前 2 种传统检测方法，从 DNS 通信行为的不同角度分析了相关特征，丰富了 DNS 信道的表征元素，就方法的内在属性而言，分析的元素更全面，显然具有更高的识别准确性，同时对卷积神经网络模型的计算过程进行了优化改进，降低了计算复杂度，提升了运算效率，在运算的网络 DNS 数据量较大、表征属性更全面的场景下，仍然保持较快的计算速度，尽管相对于 Karasaridis 模型执行时间延长了 0.37 s，但在总体准确率有较大提升的前提下，这个时延的消耗量可以接受。而且，通过少量增加运算设施，在可接受范围内，轻度提高算力成本，可以修补检测效率的时间差。在保证计算速度的前提下，提高检测准确率使其有较好的实用性。

Shafieian 模型和 Nadler 模型的检测速率与本文方法接近。但是，针对多种不同版本和类型的 DNS 隐蔽信道的检测的总体准确率要稍低于本文方法。在文献[7]中，用于测试 Shafieian 模型的 DNS 信道工具有 Iodine、DNSCat 和 Ozyman，且 Shafieian 模型中所选择的特征数量和随机森林所使用的树的数量较少。因此，当该方法应用于检测多种不同版本和类型的 DNS 隐蔽信道时，准确率有所降低。Nadler 模型[10]在兼顾高吞吐量的 DNS 隐蔽信道检测效果的前提下，能有效地针对低吞吐量的 DNS 信道进行检测。但是，该方法所采用的孤立森林算法是无监督学习算法，需要在正常的 DNS 流量上建立基准，而不同的网络环境所建立的基准不同，从而导致检测性能的波动。与这 2 种模型相比，本文所采用多种不同版本和类型的 DNS 信道作为训练数据，采用改进的卷积神经网络作为检测模型，具有稳定的检测性能，能有效地检测不同的 DNS 隐蔽信道。

表 8 中的实验结果表明，针对不同类型的 DNS 隐蔽信道，流量的识别准确率都有大幅度提升。其中，PSUDP 是通过数据注入方式隐藏在 UDP/53 网络分组数据中的非标准 DNS 隐蔽信道。本文提出的改进深度神经网络的识别方法依然能进行有效应用，并达到 96.9%的识别准确率。因此，无论是标准的 DNS 信道还是非标准的 DNS 信道，其流量的表征元素都可以转换为灰度图像，相较于其他方法，本文适用范围更加广泛，且性能优良。本文认为，Farnham 模型的域名匹配仅能匹配已知的 DNS 隐蔽信道域名，而对于更改域名和新的 DNS 信道，则失去了检测能力。Karasaridis 熵值对比的方式，需要事先设定熵的阈值，而 DNS 隐蔽信道只需修改数据分组尺寸即可躲避检测。Shafieian 模型利用集成学习技术结合了多个不同类型的分类器，能够得到较好的检测效果。但需要多次实验调整不同分类器之间的权重，且未在大规模网络环境下测试。Nadler 模型利用无监督的孤立森林算法，需要在正常的流量样本上训练建立基准，而在一个大规模网络环境下，确保不存在任何 DNS 隐蔽信道的流量是困难的，且基准容易受到其他恶意 DNS 流量的影响。而改进深度神经网络模型能够基于大量的样本数据学习 DNS 隐蔽信道的网络行为模式，这是本文认为 DNS 隐蔽信道难于改变的特点。因此，有较好的检测效果。

误报率上的实验结果如表 9 所示。每个 DNS 信道都存在识别上的误差，相比较于其他 4 种模型，RDCC-CNN 具有更低的误报情况，由于协同应用深度分组分析和网络流量特征做综合分析，相比于传统方法，RDCC-CNN 具有更全面的识别能力；相比于 Shafieian 和 Nadler 模型，RDCC-CNN 能捕捉更多的 DNS 信道的特征信息。通过这些特征表征 DNS 流量，作为卷积神经网络的输入，通过深度学习训练，对 DNS 的正常流量和信道流量具有更深刻的识别能力，标识更准确，因此误报率较低。

同时，从表 8 中可以看出，tcp-over-dns 和 PSUDP 的识别准确率没有达到较理想的水平，主要原因在于这 2 种 DNS 隐蔽信道的训练样本采集不足，因此数据不平衡，导致模型针对这 2 类信道的分类性能并不非常准确。但随着互联网通信应用的快速普及，这一问题在以后的网络流量数据分析中会逐步解决。

表 9　　各类型 DNS 信道识别误报率

类型	Farnham 模型	Karasaridis 模型	Shafieian 模型	Nadler 模型	本文方法
$Iodine_1$	1.6%	2.17%	0.93%	0.91%	0.11%
$Iodine_2$	1.7%	1.81%	0.98%	0.97%	0.13%
$Iodine_3$	1.32%	1.66%	1.22%	1.11%	0.91%
$Iodine_4$	1.91%	1.32%	1.14%	0.99%	0.31%
$Iodine_5$	1.86%	1.87%	0.70%	0.72%	0.18%
$Iodine_6$	1.93%	1.91%	0.68%	0.66%	0.25%
$Dns2tcp_1$	2.58%	1.97%	1.32%	1.29%	0.71%
$Dns2tcp_2$	2.82%	3.13%	1.33%	1.30%	1.19%
$DNSCat_1$	2.19%	2.89%	1.24%	1.21%	0.9%
$DNSCat_2$	3.17%	2.11%	1.08%	1.03%	0.61%
tcp-over-dns	2.16%	1.91%	0.89%	0.91%	0.23%
PSUDP	3.68%	3.93%	1.45%	1.50%	1.13%

2) 未知隐蔽信道流量识别

测试集由样本流量和背景流量组成。在实验过程中，除了存在误报 DNS 信道的情况，同时发现了未在测试集中的 DNS 信道流量，经过人工核实为软件 DeNise 和 Heyoka 的流量，这些信道主要出现在未作标注的原始背景流量中。不同 DNS 信道软件在程序编制、实现细节和目标场景应用方面存在一定差异，但其核心原理基本相似，在通信分组和通信流量方面具有相似行为特性。因此，这里提取的深度分组和通信流量特征可有效用于其他未在测试集中 DNS 信道软件的标识，并且通过卷积神经网络的学习分析，相比于其他方法，可深度分析出相同表征元素的流量。因此，可有效识别出原始背景中未做标注的 DNS 信道流量。

5　结束语

本文分析了目前可见的 2 种 DNS 隐蔽信道的通信形式，研究了 DNS 流量转化为灰度图片的表征元素，提出了基于改进卷积神经网络的 DNS 隐蔽信道识别方法，实现了对标准和非标准的 DNS 隐蔽信道的有效检测，体现了卷积神经网络在该任务上的优良性能，突破了现有解决方案能够有效检测的 DNS 隐蔽信道类型上的局限性。通过对采集的 Passive DNS 数据和典型的 DNS 隐蔽通信信道样本进行全面的分析，利用表征元素来表示原始数据，将其转化成灰度图片输入改进的深度卷积神经网络中，提高了识别效果。

参考文献：

[1] CROTTI M, DUSI M, GRINGOLI F, et al. Detecting HTTP tunnels with statistical mechanisms[C]//IEEE International Conference on Communications. IEEE, 2007: 6162-6168.

[2] DUSI M, CROTTI M, GRINGOLI F, et al. Tunnel hunter: detecting application-layer tunnels with statistical fingerprinting[J]. Computer Networks, 2009, 53(1): 81-97.

[3] CASAS P, MAZEL J, OWEZARSKI P. MINETRAC: mining flows for unsupervised analysis & semi-supervised classification[C]//The 23rd International Teletraffic Congress. 2011: 87-94.

[4] MARCHAL S, FRANCIS J, WAGNER C, et al. DNSSM: a large scale passive DNS security monitering framework[J]. Network Operations & Management Symposium IEEE, 2012, 131(5): 988-993.

[5] KARASARIDIS A, MEIER-HELLSTEM K, HOEFLIN D. NIS04-2: detection of DNS anomalies using flow data analysis[C]//Global Telecommunications Conference. IEEE, 2006: 1-6.

[6] SHERIDAN S, KEANE A. Detection of DNS based covert channels[C]//The 14th European Conference on Cyber Warfare and Security (ECCWS). 2015: 66-77.

[7] SHAFIEIAN S, SMITH D, ZULKERNINE M. Detecting DNS tunneling using ensemble learning[C]//International Conference on Network and System Security. 2017: 112-127.

[8] NUSSBAUM L, NEYRON P，RICHARD O. On robust covert channels inside DNS[J]. IFIP Advances in Information＆Communication Technology, 2009, 297(7): 51-62.

[9] AIELLO M, MERLO A, PAPALEO G. Performance assessment and analysis of DNS tunneling tools[J]. Logic Journal of IGPL, 2013, 21(4): 592-602.

[10] NADLER A, AMINOV A, SHABTAI A. Detection of malicious and low throughput data exfiltration over the DNS protocol[J]. Computer & Security, 2019, 80(10): 36-53.

[11] FARNHAM G, ATLASIS A. Detecting DNS tunneling[C]//SANS Institute InfoSec Reading Room. 2013: 1-32.

[12] BILGE L, KIRDA E, KRUEGEL C, et al. EXPOSURE: finding malicious domains using passive DNS analysis[C]//The Network and Distributed System Security Symposium. 2011: 68-82.

[13] LENCUN Y, BOTTOU L, BENGIO Y. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 862(10): 2278-2324.

[作者简介]

张猛（1988– ），男，山东兖州人，博士，中国电子信息产业发展研究院助理研究员，主要研究方向为网络空间安全、域名安全、区块链技术。

孙昊良（1983– ），男，辽宁朝阳人，博士，国家计算机网络与信息安全管理中心高级工程师，主要研究方向为威胁检测与信息对抗、安全态势感知。

杨鹏（1982– ），男，内蒙古集宁人，博士，国家计算机网络与信息安全管理中心高级工程师，主要研究方向为信息安全、人工智能。