基于信息熵与服务器识别的 DoH 流量检测

徐　魁[1],海　洋[1],李晓辉[2],陶　军[3]
(1. 宝鸡市公安局通信处,陕西 宝鸡 721014;
2. 宝鸡创天清航科技发展有限责任公司,陕西 宝鸡 721000;
3. 东南大学 网络空间安全学院,江苏 南京 211189)

摘　要:DNS over HTTPS(DoH)协议是一种针对域名系统(DNS)的最新改进方案,然而用户可使用第三方 DoH 服务规避内网原有的监管,所以异常流量检测方法不再适用于检测 DoH 流量。针对该问题提出了一种 DTESI 算法。首先,基于信息熵将 DoH 流量作为异常流量从全部网络流量中筛选出来;然后,利用 DoH 服务器与同一客户端建立 TLS 连接时响应方式总是相同的特性,用指纹识别检测客户端与 DoH 服务器之间的 TLS 协商,确定 DoH 服务器身份;最后,使用 Top-K 抽样算法选出一定时段内网络中前 K 台活跃主机着重进行流量检测,使算法能应用于中大型组织的网络。实验结果表明,针对发现的异常流量,DTESI 算法检测出的 DoH 服务提供商准确率超过 94%。在此基础上比较了在不同 K 值下的算法检测时间和对网络中全部 DoH 流量的检测覆盖率,结果表明合理选择 K 值可以提升算法的整体效能。

关键词:DNS over HTTPS;网络流量检测;信息熵;指纹识别;TLS 协议
中图分类号:TP393　　**文献标识码**:A　　**文章编号**:1673-629X(2024)04-0132-07
doi:10.20165/j.cnki.ISSN1673-629X.2024.0020

DoH Traffic Detection Based on Entropy and Server Identification

XU Kui[1],HAI Yang[1],LI Xiao-hui[2],TAO Jun[3]
(1. Communication Office of Baoji Public Security Bureau,Baoji 721014,China;
2. Baoji Chuangtian Qinghang Technology Development Co.,Ltd.,Baoji 721000,China;
3. School of Cyber Science and Engineering,Southeast University,Nanjing 211189,China)

Abstract:The DNS over HTTPS (DoH) protocol is the latest improved solution for the Domain Name System (DNS). However,users can use third-party DoH services to avoid the original supervision of the intranet,so the abnormal traffic detection method is no longer suitable for detecting DoH traffic. Aiming at this problem,a DTESI algorithm is proposed. Firstly,DoH traffic is screened as abnormal traffic from all network traffic based on information entropy. Then,according to the characteristic that the response mode is always the same when the DoH server establishes a TLS connection with the same client,the TLS negotiation between the client and the DoH server is detected by fingerprint identification to determine the identity of the DoH server. Finally,the Top-K sampling algorithm is used to select the top K active hosts in the network within a certain period of time to focus on traffic detection,so that the proposed algorithm can be applied to the network of medium and large organizations. The experimental results show that the accuracy rate of DoH service providers detected by DTESI algorithm exceeds 94% for the abnormal traffic found. On this basis,the detection time and the detection coverage of all DoH traffic in the network are compared under different K values,and it is showed that a reasonable choice of K value can improve the overall performance of the algorithm.

Key words:DNS over HTTPS;network traffic detection;information entropy;fingerprint identification;Transport Layer Security Protocol

0　引　言

域名系统(Domain Name System,DNS)是互联网重要的基础设施,然而随着互联网服务与技术的发展,DNS 协议的一些问题逐渐暴露出来。DNS over HTTPS(DoH)协议是一种针对 DNS 的最新改进方案,但关于其对用户隐私的保护能力、对组织内部网络带来的监管挑战等方面仍有一定争议。

对用户使用互联网产生的网络流量进行统计分

收稿日期:2023-06-07　　**修回日期**:2023-10-09
基金项目:中国高校产学研创新基金-阿里云高校数字化创新专项(2021ALA03006)
作者简介:徐　魁(1973-),男,高级工程师,研究方向为大数据分析;通信作者:陶　军(1975-),男,教授,博导,博士,研究方向为网络安全、物联网技术。

析,可以研究用户的行为规律,进一步为网络资源优化和异常行为检测提供依据。关于用户网络行为分析所用的方法比较多且技术成熟,目前主流的有基于机器学习的统计分析法[1-3]、关联规则分析法[4-5]、时序数据分析法[6-7]。所以在研究面向 DoH 协议流量的用户网络行为时,关键是如何从原始网络流量中检测出 DoH 流量。

在异常流量检测方面,目前主要的研究方法包括以下三种:基于端口的检测方法、基于 DPI/DFI 的检测方法和基于机器学习的检测方法。在基于端口的检测方法中,文献[8]提出了一种基于模糊逻辑控制器的 TCP 端口扫描检测框架,可以实时跟踪网络流量行为并快速有效地识别恶意端口扫描活动。文献[9]受人类免疫系统的启发,使用树状细胞算法(DCA)检测恶意 TCP 端口扫描。文献[10]指出特定异常用户主机的异常端口号具有关联性,结合机器学习算法可以对用户异常行为进行有效的识别。深度包检测(Deep Packet Inspection, DPI)和深度流检测(Deep Flow Inspection,DFI)中,文献[11]将 DPI 技术结合双向长期短期记忆的递归神经网络(BLSTM-RNN)应用于检测物联网中的僵尸网络活动。文献[12]使用了 Tsallis 熵用于检测流异常并提出了一种基于深度 IP 流检测的网络异常检测系统。基于机器学习的异常流量检测方法中,文献[13]使用马尔可夫过程思想,利用自回归综合滑动平均模型(ARIMA)进行网络流量预测与异常检测。文献[14]中讨论了一种结合遗传算法和模糊逻辑的网络异常检测方案,该方案具有较高的准确率。文献[15]中提出了一种在云计算环境下的流量异常检测方法,该方法将信息熵和支持向量机(SVM)模型结合来判断异常。文献[16]使用 Hadoop 的 MapReduce 分布式并行计算模型,基于网络流量相关性的模糊 C-均值聚类算法,完成网络异常流量分析。

对于大型机构而言,用户可能通过使用第三方 DoH 服务规避内网原有的基于 DNS 的监管策略,网络管理者迫切需要从其他流量特性入手寻找从原始网络流量中检测 DoH 流量的方法。文献[17]研究了使用机器学习可以从 HTTPS 扩展数据中获取的信息并评估了五种机器学习方法,找到的最佳的 DoH 分类器对 DoH 识别的准确率超过 99.9%,但由于 DoH 协议尚未在全球得到大范围使用,领域内尚未有公认比较理想的测试用 DoH 流量数据集。文献[18]将 DoH 流量包的大小与 n-grams 相结合作为特征进行研究,表明使用 DoH 协议后仍然能够获取用户的访问记录。文献[19]提出了一种完全基于填充并加密的 DNS 流量的分析方法,可以结合流量大小和时间信息来推断用户访问的网站。文献[20]研究了利用 DoH 发送的 C&C 攻击消息,讨论了其对网络安全的影响,但其主要根据服务器的 IP 地址进行检测,当 DoH 服务器大范围部署后,该方法的检测效果可能降低。

该文筛选了可用于区分 DoH 流量与常规 HTTPS 流量的部分特性,并基于信息熵理论与服务器 TLS 指纹识别技术提出一种 DoH 流量检测算法。为了检测使用了不受信第三方 DoH 服务的用户,基于信息熵理论提出一种主机粒度的 DoH 流量检测方案,进一步提出活跃主机抽样的方法,检测最活跃的 K 台主机,在保证较高 DoH 流量检测覆盖率的前提下提升局域网内异常主机的定位检测速度,并通过服务器指纹识别算法验证嫌疑 DoH 服务器的身份。

1　基于信息熵与服务器识别的 DoH 流量检测方法

1.1　基于信息熵的异常流量检测

作为一种常用的检测度量,信息熵(information entropy)常用于异常流量检测。由于无法仅凭端口号将 DoH 流量同常规 HTTPS 流量区分开,该文首先基于信息熵将其作为异常流量从全部网络流量中筛选出来,以便于进一步确认 DoH 服务器的身份。

1.1.1　信息熵理论

在信息论中,熵是指每条消息中包含的信息的平均量,又称为信息熵。当取自有限的样本时,熵值可用公式 1 表示:

$$H(X) = -\sum_{i=1}^{n} P(x_i)\log_b P(x_i) \tag{1}$$

其中,$H(X)$ 是随机变量 X 的信息熵,$X = \{x_i, i = 1, 2, \cdots, n\}$,表示变量 X 的 n 个随机状态的集合,$P(x_i)$ 表示第 i 个随机状态发生的概率,对数的底 b 通常取 2,e 或 10,取 2 时熵的单位是 bit。网络流量具有很多可以量化的统计学特征,可以作为离散的信息源,因而信息熵经常作为一种有效的量度用于异常流量检测。

1.1.2　基于信息熵的异常流量检测方案

由于 DoH 协议内容承载于 HTTPS 协议报文中不能直接通过协议类型加以区分,该文采集用户主机上网时使用 DoH 协议解析域名后单位时间内的聚合流量,对比访问不同目标网站和选择不同 DoH 服务器或不选择 DoH 协议上网的情况,对聚合流的熵值进行分析。正常情况下,一定时间窗口内源地址、源端口、目的地址、目的端口以及使用协议这几个特征的熵值序列维持在较为稳定的水平。使用 DoH 协议解析域名时由于用户主机会频繁地通过 HTTPS 协议访问特定的 DoH 服务器,同时基于 UDP 协议 53 端口的 DNS 流量显著减少,特征熵值会随之出现明显变化。则根据

公式 1 可得到流量中某个特征信息熵的计算方法,如公式 2 所示:

$$H(X) = -\sum_{i=1}^{n}\left(\frac{n_i}{S}\right)\log_2\left(\frac{n_i}{S}\right) \tag{2}$$

其中,$H(X)$ 是该特征的信息熵,X 表示该特征的 N 个状态,$X=\{x_i, i=1,2,\cdots,N\}$,$n_i$ 是第 i 个状态 x_i 出现的次数,$S=\sum_{i=1}^{N} n_i$ 表示该特征 N 个状态总的出现次数,因此 n_i/S 代表第 i 个状态 x_i 发生的概率。

在得到多个相同时间窗口下的特征熵之后,可以根据正态分布的基本检测原理来判断某一特征熵值是否超过了异常阈值。正态分布是一种连续型随机变量的概率分布,服从正态分布的随机变量 X 取值越接近正态分布均值 μ 概率越大,正态分布的标准差 σ 越小变量 X 的分布越集中。任意一正态分布均服从公式 3 所示的分布规律:

$$\begin{cases} P(\mu-3*\sigma<X<\mu+3*\sigma)=99.74\% \\ P(\mu-4*\sigma<X<\mu+4*\sigma)=99.993\,6\% \\ P(\mu-4.8*\sigma<X<\mu+4.8*\sigma)=99.999\,9\% \end{cases} \tag{3}$$

根据该分布规律,只需要通过最大似然估计求出已有样本的均值 μ 与标准差 σ 即可确定该正态分布的取值概率范围,并根据该范围设置熵值的异常阈值 λ_t,进一步帮助判断某用户主机是否出现了异常流量。

1.2 DoH 服务器身份识别

在定位异常流量之后,需要进一步确认其 DoH 流量的性质。DoH 服务器本质上是一种特殊的 HTTP 服务器,由于反向代理、负载均衡等机制这些服务器往往由多台物理机器组成并且 IP 可能产生变动。因此,直接通过 IP 地址来检测客户端是否与某 DoH 服务器进行了通信的方法效果不佳。

由于使用 DoH 协议的客户端在访问多种 WEB 资源前首先会访问选定 DoH 服务提供商的一台或几台物理服务器,而这些服务器与同一客户端建立 TLS 连接所采用的响应方式总是相同的,所以可以通过指纹识别检测客户端与 DoH 服务器之间的 TLS 协商。互联网中 DoH 服务器的数量远不及常规的 HTTP 服务器的数量,因而该文建立使用 DoH 服务器指纹数据库为 DoH 服务器的身份识别提供参考。

1.2.1 服务器 TLS 指纹验证

TLS 是 TCP 之上的安全协议,采用了主从式的架构,用于为两个程序之间的通信提供机密性和完整性保护,而 TLS 握手协议用于在两台主机建立起 TLS 通信之前协商安全参数,例如双方都支持的加密算法,生成本次会话的密钥所需的随机数、公钥、证书以及签名等。

如图 1 所示,为了启动 TLS 会话,客户端将在 TCP 三次握手后进行 TLS 握手,整个握手过程主要分如下:

(1)客户端向服务器发送一个包含客户端支持的 TLS 版本、加密算法列表以及客户端生成的随机数等内容的 Client Hello 消息。

(2)服务器收到后向客户端发送一个包含服务器支持的 TLS 版本、生成的会话 ID、支持的加密算法列表、服务器选择使用的加密算法以及服务器生成的随机数 Server Hello 消息;服务器发送包含公钥、签名以及证书签发机构等信息的证书给客户端;如果使用 Diffie-Hellman key exchange 等相关加密算法,服务器还会向客户端发送包含算法相关参数的 Server Key Exchange 消息;最后,服务器发送 Server Hello Done 消息,表示自身的握手消息发送完毕。

(3)客户端确认收到的证书可信后,向服务器发送包含 Diffie-Hellman key exchange 算法相关参数的 Client Key Exchange 消息,双方可根据自身与对方的参数计算出 Premaster Secret 用于生成对称加密传输所用的会话密钥;客户端发送 Change Cipher Spec 消息,告知服务器将使用加密方式发送消息;客户端使用会话密钥加密之前收发握手消息的哈希值、MAC 值,并发送给服务器。

(4)服务器同样以计算出的会话密钥解密消息,验证哈希值和 MAC 值,验证无误后向客户端发送 Change Cipher Spec 消息,告知客户端将使用加密方式发送消息;服务器使用会话密钥加密之前收发握手消息的哈希值、MAC 值,并发送给客户端校验。校验无误后握手阶段完成,双方将使用加密方式进行通信。

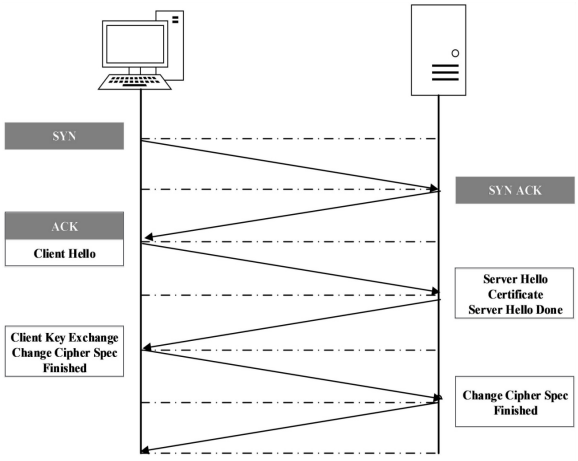

图 1 TLS 握手流程

由于整个握手阶段都是无法加密、使用明文传输的,这些信息足以用于为任意服务器生成指纹,进而查询服务器指纹数据库验证其身份。服务器发送的握手

消息中的特定字段只与客户端程序兼容的 TLS 版本、加密算法、扩展有关,不会受到 IP 地址、端口的影响,同一 DoH 服务提供商的服务器发送给固定客户端的握手消息总是相同的。

表 1　TLS 握手消息中用于生成指纹的字段

字段 ID	名称	说明
1	TLS Version	支持的 TLS 协议版本
2	Cipher Suites	可接受的加密算法
3	Compression Methods	压缩方法
4	Rare ALPN	TLS 扩展 ALPN 协议信息
5	Extension List	扩展列表

因此基于可用的 TLS 握手消息,该文计算异常流量中服务器的指纹用于识别 DoH 服务器身份。为此,需要进行如下步骤:

(1)解析保存原始流量信息的 pcap 文件,从每个 TLS 会话中读取表 1 中所有字段的值;

(2)对这些值进行连接操作,使用“;”分隔不同的字段,使用“,”分隔同一字段内不同的值;

(3)使用模糊哈希算法对连接后的文本进行哈希,得到服务器的 TLS 指纹。

模糊哈希算法又称为基于内容分割的分片哈希算法(Context Triggered Piecewise Hashing,CTPH),主要原理是根据特定条件对数据进行分片,根据数据局部内容的特点选择使用弱哈希或强哈希算法,最后将这些部分的哈希值加起来就得到了模糊哈希结果。CTPH 算法对输入数据进行分片,将大块的数据切分成多个小块,有助于在处理大量数据时,将数据划分为更小的单元进行处理,提高处理效率。且 CTPH 算法能够检测数据之间的相似性,即使数据存在细微的变化,也能生成具有相似性的哈希值。

在生成 TLS 指纹时为了尽量减少冲突的概率,使得哈希冲突对结果产生的影响最小化,需要生成的哈希值进行唯一性验证。如果发现哈希冲突,通过增加哈希值的长度和选择更强大和更具分散性的哈希算法减少哈希冲突的概率。因为支持的 TLS 协议版本、可接受的加密算法、压缩方法这几个字段取值范围比较有限,因而分片后采用编写的弱哈希算法,而 TLS 扩展 ALPN 协议信息和 TLS 扩展列表可能出现较多组合,因而分片后采用哈希空间更大的 sha256 算法进行哈希。

在将此方法应用于服务器身份识别前,还需要计算常见的 DoH 服务器指纹并存储至数据库中定期维护。上文中提到,同一 DoH 服务提供商的服务器发送给固定客户端的握手消息总是相同的,总是与 Client Hello 消息有关。为此,该文选择了一组可代表大部分客户端 Client Hello 数据包的报文字段组合,如表 2 所示。按照这些组合构造数据包,并向 curl wiki 列表中推荐的多个 DoH 服务器发送,最终解析服务器响应的 Server Hello 数据包计算得到多组服务器指纹存储在数据库中。

表 2　构造不同 Client Hello 数据包所需字段组合

VERSION	CYPHER_LIST	CIPHER_ORDER	GREASE	RARE_ALPN	1.3_SUPPORT	EXTENSION_ORDERS
TLS_1.2	ALL	FORWARD	NO_GREASE	APLN	1.2_SUPPORT	REVERSE
TLS_1.2	ALL	REVERSE	NO_GREASE	APLN	1.2_SUPPORT	FORWARD
TLS_1.2	ALL	TOP_HALF	NO_GREASE	APLN	NO_SUPPORT	FORWARD
TLS_1.2	ALL	BOTTOM_HALF	NO_GREASE	RARE_APLN	NO_SUPPORT	FORWARD
TLS_1.2	ALL	MIDDLE_OUT	GREASE	RARE_APLN	NO_SUPPORT	REVERSE
TLS_1.1	ALL	FORWARD	NO_GREASE	APLN	NO_SUPPORT	FORWARD
TLS_1.3	ALL	FORWARD	NO_GREASE	APLN	1.3_SUPPORT	REVERSE
TLS_1.3	ALL	REVERSE	NO_GREASE	APLN	1.3_SUPPORT	FORWARD
TLS_1.3	NO1.3	FORWARD	NO_GREASE	APLN	1.3_SUPPORT	FORWARD
TLS_1.3	ALL	MIDDLE_OUT	GREASE	APLN	1.3_SUPPORT	REVERSE

1.2.2　基于访问模式的 DoH 服务器识别

对于利用信息熵检测出的异常网络流,若在 DoH 服务器指纹数据库中未能查询到其对应的指纹,需要结合 DoH 流量特性进一步给出判断策略。虽然 DoH 协议通过加密直接保护了下层 DNS 流量,但是并没有改变用户访问 WEB 资源的模式,这也为该文从 WEB 访问模式的角度检测 DoH 流量提供了启发。

用户在使用 DoH 协议时,仍需要先向递归域名解析服务器发送解析请求,等待递归域名解析服务器从上游服务器处查询并返回解析响应后,再通过得到的 IP 地址与目标 WEB 服务器建立通信。对应地,以流为单位观测一段时间窗口 W 内与用户主机进行加密 HTTPS 通信的服务器指纹,可以发现一部分指纹呈现在单个短时间周期 T 内出现较为密集且流传输字节

数较大的特点,而另一部分指纹则是呈在时间窗口 W 下的多个短时间周期 T 内多次出现且流传输字节数相对较小的特点,两者分别可判定为常规 HTTPS 流量和 DoH 流量。因为在客户端 WEB 访问模式中,用户访问的一个 WEB 页面可能会使浏览器加载多个其他 WEB 服务器的资源(例如字体、外链图片等),这就导致一次 WEB 访问行为产生的流量中可能包含多次域名解析流量。尽管如此,域名解析流量的规模远小于常规 WEB 流量,上述特点不会因此受到影响。

1.3 DoH 流量检测策略

1.3.1 局域网内 Top-K 活跃主机抽样

在中大型组织的网络中,如果直接针对网关边界采集到的全部流量进行分析,那么少数用户使用 DoH 协议的情况对测得的特征熵值影响很小,导致难以检测到异常。此外,出于性能考虑,难以同时对网络中所有的主机流量进行细粒度流量检测,因此还需要使用抽样算法选择出一定时段内网络中的前 K 台活跃主机着重进行流量检测。

在衡量特定主机活跃程度的标准方面,使用单位时间内流量大小可能是最直观的,然而在涉及用户访问网站的行为时,用户主机在单位时间内产生的连接数(即访问不同服务器的数量)更为重要。因为连接数越多往往意味着用户产生的域名解析请求越多;而流量大不一定意味着主机的网站访问行为更为活跃,例如在单一站点长时间观看网络视频、传输文件等,这种情况下获取的涉及域名解析的流量可能不足。因此,该文使用了用户主机在单位时间内产生的连接数来衡量活跃程度。

Top-K 活跃主机抽样方法需要创建一个 Map 结构,用于存储每个用户 IP(usrIP)对应的服务器 IP(svrIP)的集合,还需要维持一个容量为 K 的 Heap 结构,存储用户主机的 IP。将单位时间窗口内产生的用户主机与服务器的 IP 对维护至 Map 中,使用维护的 usrIP 的连接数来排序前 K 个活跃主机,最终可以得到连接数最多的 K 个用户的 IP。

1.3.2 目标主机 DoH 流量检测算法

在得到当前时间窗口下网络中前 K 个活跃主机的 IP 地址的前提下,该文提出基于信息熵与服务器身份识别的 DoH 流量检测算法(DoH Traffic detection algorithm based on Entropy and Server Identification, DTESI),可针对特定主机的网络流量检测其中潜在的 DoH 流量。算法描述如下所示:

算法 1 基于信息熵与服务器身份识别的 DoH 流量检测算法(DTESI)

输入:NF_{all}, $\vec{F}$, Seq^{F_i}, DB

输出:doh traffic/regular traffic

1. FOR i←1 to 10 DO:
2. 按式 2 计算 NF_{all} 中流特征 F_i 的熵值 $H(F_i)$;
3. 按照式 3 计算熵序列 Seq^{Fi} 的均值 μ,标准差 σ,设定阈值 λ_t;
4. IF $|H(F_i)-\mu|<\lambda_t$ THEN /* $H(F_i)$ 在分布阈值内 */
5. Update Seq^{F_i};
6. ELSE /* $H(F_i)$ 分布超出阈值,存在异常流量 */
7. 计算该事件窗口内全部流的 TLS 指纹集 FP;
8. IF DB∩FP≠ØTHEN /* 数据库中查询到已知指纹 */
9. Predict as doh traffic;
10. ELSE /* 判断该指纹集是否为可能的 DoH 服务器指纹 */
11. FOREACH j in FP DO:
12. 统计 NF_{all} 中指纹 j 的 A_j,S_j 数值;
13. IF $\ln(A_j/S_j)$ < t_mode THEN:
14. Predict as doh traffic;
15. Update DB;
16. END IF
17. ELSE /* 所有指纹都不能判断为 Doh 指纹 */
18. Predict as other abnormal traffic;
19. END FOREACH
20. END IF
21. END IF
22. END FOR
23. Predict as regular traffic;

其中,NF_{all} 表示时间窗口 W 内的目标主机全部原始流量数据;$\vec{F}=\{F_1,F_2,\cdots,F_{10}\}$ 表示算法使用的特征向量,包含上述 10 种流量特征;$Seq^{F_i}=\{H(F_i)^1, H(F_i)^2,\cdots,H(F_i)^t\}$ 表示特征 F_i 的历史熵值序列;DB 表示已知的 DoH 服务器 TLS 指纹数据表;A_i 为指纹 i 共在多少个检测周期中出现,S_i 表示指纹 i 在整个时间窗口内总共出现的次数,t_mode 表示判断某个网络流的服务器是否为 DoH 服务器的阈值。

由历史熵值序列 Seq^{F_i} 计算得到的均值 μ、标准差 σ、阈值 λ_t 判断 NF_{all} 中是否存在异常流量。为此,需要提前将其在一段时间内使用固定的时间滑动窗口测得的特征熵值序列得到正常流量的特征熵值序列,并提前维护测得的特征熵值序列与已知 DoH 服务器指纹数据库。选择目的 IP 地址、使用协议、持续时长、发送流字节数、发送流字节速率、接收流字节数、接收流字节速率、包长度、包传输时间以及响应时间这 10 个特征并对其中连续的特征进行离散化,以降低计算特征熵值所需的空间消耗。

在检测过程中,短期保留一段时间内的原始流量数据,若检测出异常则会保存对应时间段的数据用于分析,结合使用信息熵与 DoH 服务器访问模式的识别方法也有助于在保证检测准确率的前提下提升检测效

率。该算法主要的优势在于能够通过维护特征熵值序列实现近实时的异常流量检测与告警,并可以更新未知的服务器信息。

2　实验仿真及结果分析

目前,由于 DoH 协议尚未在全球得到大范围使用,虽然已有学者公开了其有关研究中采集的 DoH 流量数据,但领域内尚未有公认比较理想的测试用 DoH 流量数据集。为了收集能代表真实 DoH 流量的数据,在服务器方面,该文基于开源解决方案“m13253/DNS-over-HTTPS”自建了用于测试的 DoH 服务器,并选择了 curl 官方 wiki 推荐列表中国内的阿里云(Alidns)、红鱼(Rubyfish)以及国际知名的两家厂商 Google 和 Cloudflare 的服务器作为上游递归解析服务器。

在实验网络环境中,共有 34 台用户设备,选择 10 台进行 WEB 自动化测试以模拟活跃主机,其中有 4 台主机配置了 DoH 服务并分别使用了上述 4 家 DoH 服务提供商。控制测试主机以不同的网页访问频率批量访问 Alexa Top Sites 中的前 10 000 个站点,选择 1 分钟的时间窗口,先以未使用常规 DNS 协议客户端的出口流量作为历史记录计算多个测度的特征熵值,在 100 分钟后依次在 4 台使用 DoH 服务的虚拟机中启动新的浏览器进程访问 WEB 内容。如图 2 所示,以 H(Protocol)这一特征熵测度为例,检测到在使用 DoH 访问 WEB 内容前后,熵值序列的均值 μ 由 1.697 降至 1.240,标准差从 0.441 降至 0.365。这说明使用 DoH 协议造成的客户端流协议种类与数量的改变对特定的特征熵值的分布情况有明显的影响。

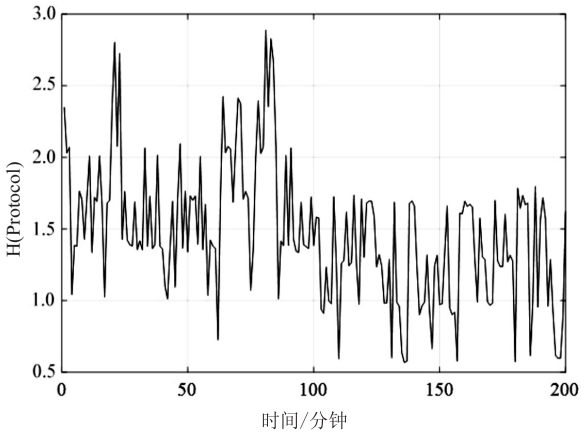

图 2　测试流量的 H(Protocol)熵值变化

在服务器指纹识别方面,在 DTESI 算法中使用包含 DoH 流量的实验数据计算得出的 4 个 DoH 服务商均具有独特的指纹特征,实验中保留了每一次预测为 DoH 服务器的指纹与对应网络流的五元组,之后对照实验主机的仿真配置计算检测的准确率。如表 3 所示,结果表明针对发现的异常流量,算法检测出的 DoH 服务提供商准确率均达到 94% 以上。在此基础上,该文比较了在不同 K 值下算法的检测时间和对网络中全部 DoH 流量的检测覆盖率。如图 3 所示,可以看到当 $K>4$ 后随着检测时间的增加,DoH 流量检测覆盖率已经趋于饱和,这说明合理选择 K 值对提升算法的整体效能有重要作用。

表 3　服务器指纹验证准确率

序号	DoH 服务商	准确率/%
1	Alidns	95.27
2	Rubyfish	94.12
3	Cloudflare	97.23
4	Google	96.34

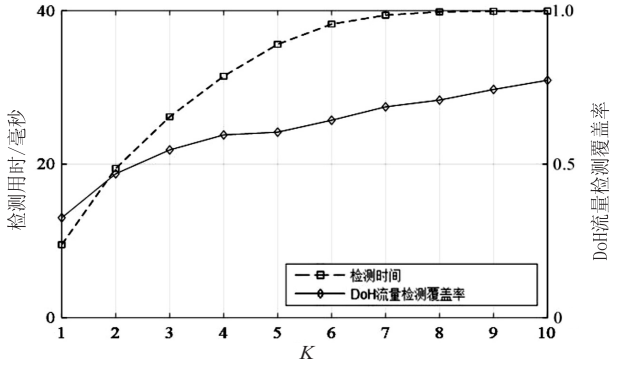

图 3　不同 K 值下的检测时间与 DoH 流量覆盖率

3　结束语

DoH 协议内容承载于 HTTPS 协议报文中,不能直接通过协议类型或者端口号将 DoH 流量通常规 HTTPS 流量区分开。在基于信息熵与服务器识别的 DoH 流量检测方法中,首先针对 DoH 流量研究了基于信息熵的异常流量监测方案,接着使用 TLS 握手消息计算服务器指纹,并基于 WEB 访问模式对包含 DoH 流量的网络流进行了检测。实验结果表明,DoH 流量虽然不能通过协议类型与常规的加密 HTTPS 流量区分开但并非不可被识别,提出的 DTESI 算法可以在保证较高检测效率的前提下取得比较理想的准确率。在此基础上,该文比较了在不同 K 值下算法的检测时间和对网络中全部 DoH 流量的检测覆盖率,结果表明合理选择 K 值对提升算法的整体效能有重要作用。涵盖更广泛的网络环境和使用场景是进一步改进和拓展的重点,这将包括考虑大规模网络的情况,以及不同网络拓扑结构对于识别准确性的影响。

参考文献:

[1] REN W, ZHANG J, DI X, et al. Anomaly detection algorithm based on CFSFDP[J]. Journal of Advanced Computational Intelligence and Intelligent Informatics, 2020, 24(4): 453-

460.
[2] SARKER I H, COLMAN A, HAN J, et al. BehavDT: a behavioral decision tree learning to build UserCentric context-aware predictive model[J]. Mobile Networks and Applications, 2020, 25(3): 1151-1161.
[3] TANG X, LI S, YU W. WEB user preferences and behavior clustering based on BP neural network[J]. Journal of Intelligent and Fuzzy Systems, 2020, 38(2): 1189-1196.
[4] AZEEZ N A, AYEMOBOLA T J, MISRA S, et al. Network intrusion detection with a hashing based apriori algorithm using hadoop MapReduce[J]. Computers, 2019, 8(4): 86.
[5] ELKABANI I, DAHER L A, ZANTOUT R N. Use of FP-growth algorithm in identifying influential users on twitter hashtags[C]//4th international conference on computer and data analysis. New York: ACM, 2020: 113-117.
[6] HUSIN H S, CUI L, HAMID H R M H, et al. Time series analysis of web server logs for an online newspaper[C]//7th international conference on ubiquitous information management and communication. New York: ACM, 2013: 1-4.
[7] 张永锋. 个性化推荐的可解释性研究[D]. 北京: 清华大学, 2016.
[8] SAIDI F, TRABELSI Z, GHAZELA H B. Fuzzy logic based intrusion detection system as a service for malicious port scanning traffic detection[C]//16th international conference on computer systems and applications (AICCSA). Abu Dhabi: IEEE, 2019: 1-9.
[9] ALMASALMEH N, SAIDI F, TRABELSI Z. A dendritic cell algorithm based approach for malicious TCP port scanning detection[C]//15th international wireless communications & mobile computing conference. Tangier: IEEE, 2019: 877-882.
[10] 周颖杰. 基于行为分析的通信网络流量异常检测与关联分析[D]. 成都: 电子科技大学, 2013.
[11] MCDERMOTT C D, HAYNES W, PETROVSKI A V. Threat detection and analysis in the internet of things using deep packet inspection[J]. International Journal on Cyber Situational Awareness, 2018, 3(1): 61-83.
[12] ALEXANDRE A A, LEONARDO S M, BRUNO B Z, et al. Deep IP flow inspection to detect beyond network anomalies[J]. Computer Communications, 2017, 98: 80-96.
[13] MOAYEDI H Z, MASNADI-SHIRAZI M A. Arima model for network traffic prediction and anomaly detection[C]// International symposium on information technology. Kuala Lumpur: IEEE, 2008: 1-6.
[14] HAMAMOTO A H. Network anomaly detection system using genetic algorithm and fuzzy logic[J]. Expert Systems with Applications, 2017, 92: 390-402.
[15] CHEN Y. Anomaly network traffic detection algorithm based on information entropy measurement under the cloud computing environment[J]. Cluster Computing, 2018(7): 1-9.
[16] 马晓亮. 基于 Hadoop 的网络异常流量分布式检测研究[D]. 重庆: 西南大学, 2019.
[17] VEKSHIN D, HYNEK K, CEJKA T. DoH insight: detecting DNS over HTTPS by machine learning[C]//15th international conference on availability, reliability and security (ARES '20). New York: ACM, 2020: 1-8.
[18] SIBY S, JUÁREZ M, DÍAZ C, et al. Encrypted DNS -> privacy? a traffic analysis perspective[C]//The network and distributed system security symposium (NDSS '20). San Diego: Internet Society, 2020.
[19] BUSHART J, ROSSOW C. Padding ain't enough: assessing the privacy guarantees of encrypted DNS[C]//USENIX security symposium 2020. [s. l.]: USENIX, 2020.
[20] HJELM D. A new needle and haystack: detecting dns over https usage[EB/OL]. 2019-01-28. https://www.sans.org/white-papers/39160/.