Computer Networks 234 (2023) 109910

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Real time detection of malicious DoH traffic using statistical analysis

Marta Moure-Garrido [*], Celeste Campo, Carlos Garcia-Rubio

University Carlos III of Madrid, Department of Telematic Engineering, Avda. Universidad 30, Leganés (Madrid), E-28911, Spain

ARTICLE INFO

MSC:
68M10
68M12
90B18

Keywords:
Classification
DNS tunnels
DoH traffic
Intrusion Detection System (IDS)
Malicious DoH
Statistical analysis

ABSTRACT

The DNS protocol plays a fundamental role in the operation of ubiquitous networks. All devices connected to these networks need DNS to work, both for traditional domain name to IP address translation, and for more advanced services such as resource discovery. DNS over HTTPS (DoH) solves certain security problems present in the DNS protocol. However, malicious DNS tunnels, a covert way of encapsulating malicious traffic in a DNS connection, are difficult to detect because the encrypted data prevents performing an analysis of the content of the DNS traffic.

In this study, we introduce a real-time system for detecting malicious DoH tunnels, which is based on analyzing DoH traffic using statistical methods. Our research demonstrates that it is feasible to identify in real-time malicious traffic by analyzing specific parameters extracted from DoH traffic. In addition, we conducted statistical analysis to identify the most significant features that distinguish malicious traffic from benign traffic. Using the selected features, we achieved satisfactory results in classifying DoH traffic as either benign or malicious.

1. Introduction

The DNS (Domain Name System) is a ubiquitous protocol found in all networks due to its indispensability in supporting various functionalities ranging from service discovery to name resolution. However, its ubiquitous nature also renders it a potential security threat within these networks, as it can be leveraged by attackers to establish tunnels to extract sensitive information or transmit malicious commands surreptitiously, evading detection mechanisms. In light of this, the objective of this research is to analyze DNS traffic traces and extract metrics that characterize the traffic, with the goal of detecting such attacks.

When it was first defined, the DNS communication protocol, known as Do53 (DNS over port 53) [1], transmitted queries in clear, which posed certain security problems because DNS queries could reveal sensitive user information. The security problems were related to integrity, authenticity and confidentiality [2]. DNSSEC has provided security to the DNS, guaranteeing the integrity and authenticity of the responses received, but these responses travel unencrypted over the network, leaving the confidentiality uncovered. To solve the confidentiality issue, in 2016 DNS over TLS (DoT) [3] was defined, and DNS over HTTPS (DoH) [4], in 2018. DoH is the most widespread version, which was introduced in all major web browsers in 2020. In DoH, DNS messages travel over port 443 encrypted by TLS, and all transmitted content is hidden. DNS over QUIC (DoQ) [5] has been defined in 2022, the encryption properties provided by QUIC are similar to the properties of TLS. Encrypted DNS protocols (DoT, DoH, DoT and DoQ) encapsulate DNS at the encryption layer unlike Do53 as illustrated in Fig. 1.

DNS tunneling allows a DNS connection to be exploited as a hidden communication channel between client and server, a covert way of encapsulating data transmission. These tunnels can be detected by analyzing the content of DNS packets. The analysis of statistical characteristics is important for intrusion detection techniques against DNS. However, in DoH tunnels [6], since DNS traffic is encrypted and not perceptible to the client–server infrastructure, these detection methods are rendered obsolete [7].

Attackers leverage this vulnerability to conceal their malicious activities. According to a Netlab report, the Godlua Backdoor is the initial malware to employ DoH as a covert communication channel for concealing malicious traffic as DNS traffic, which was discovered in 2019 [8]. In 2020, the first known Advanced Persistent Threat (APT) incorporating DoH appeared [9]. APT34 performed data exfiltration through DoH using tunneling tools.

Tools are available that generate DoH tunnels, making it easier for malicious actors to send malicious traffic within DoH connections [10]. The objective of such tools is to establish covert data tunnels that enable traffic to be transmitted in DNS queries that are encapsulated and travel over HTTPS. In addition, certain tools allow an attacker to create a DoH tunnel by running a DoH proxy. Some of these tools are Iodine [11], dns2tcp [12] and dnscat2 [13].

[*] Corresponding author.
E-mail address: mamoureg@it.uc3m.es (M. Moure-Garrido).

https://doi.org/10.1016/j.comnet.2023.109910
Received 29 March 2023; Received in revised form 29 May 2023; Accepted 26 June 2023
Available online 30 June 2023

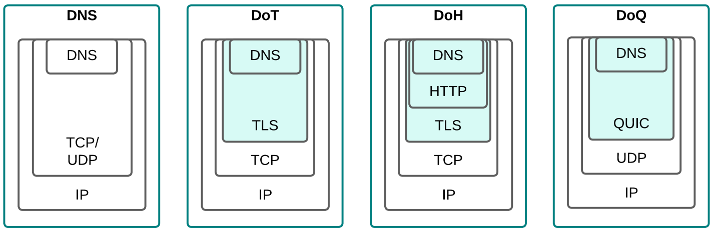


Fig. 1. Communication protocol stack (DNS, DoT, DoH and DoQ).

The present study involves an analysis of feature patterns extracted from DoH traffic to investigate whether there exists a statistical variance between benign and malicious DoH traffic. Through this analysis, we aim to identify the most significant features that enable us to differentiate between benign and malicious traffic. To evaluate the obtained results and to determine the adequacy of the selected features, we employ multiple machine learning classification techniques.

Furthermore, we propose a real-time malicious DoH tunnel detection system. Our aim is to detect any potential intrusion as early as possible, using a real-time Intrusion Detection System (IDS) classifier, in order to mitigate DoH tunneling attacks. To develop the real-time detection system, we rely on performing a statistical analysis of the different DoH traffic features that are available in real-time.

The rest of the paper is structured as follows: an overview of the state of the art related to the DNS tunnels and DoH tunnel detection is provided in Section 2. Section 3 describes the dataset used in this study and Section 4 includes the analysis performed and the results obtained. Then, in Section 5 we detail the process of implementing the real-time system and the subsequent evaluation that was conducted. Finally, the conclusions of this paper and future work are presented in Section 6.

2. State of the art

2.1. DNS tunnels

A DNS tunnel allows data to be encapsulated in a DNS packet for bidirectional communication between client and server. DNS tunneling is known as covert channels or data exfiltration. DNS tunnels are not always malicious, but in most cases, attackers use them for malicious intentions. DNS tunnel detection is the focus of many research papers [14,15]. One method to detect these tunnels versus benign DNS traffic is based on the analysis of the content of DNS packets. Different payload and packet connection features can be analyzed [16].

The characteristics extracted from the payload are, firstly, those related to the statistical analysis of the packet size and the ratio of sent and received data. Regarding the packet size, malicious DNS packets are larger in size due to the transmission of upstream encapsulated data in DNS tunnels in the domain name field. It should be considered that the attacker could conform the traffic by reducing the amount of data sent by increasing the number of requests made.

Secondly, there exist certain domain name characteristics that are relevant to the detection of tunnels. Attributes such as domain name length, number of subdomains, number of special characters in the domain name and character entropy (the character frequencies of a normal domain name are concentrated in a few high frequency characters) are crucial in detecting tunnels, as discrepancies between tunnel and legitimate domain names are unavoidable. Despite these discrepancies, evasion techniques may be employed to disguise them as normal domains.

Additionally, tunneling tools tend to use unusual record types, and certain tunneling tools leave a distinct footprint in the DNS packet, which can be an attribute in the DNS header or content in the payload. Hence, it is also advisable to analyze these features.

On the other hand, characteristics related to DNS traffic connection such as the volume of traffic to a given IP address or a given domain are analyzed. A long time between a query and a response can indicate the existence of a DNS tunnel because domain name resolution uses cached records and therefore takes less time than DNS tunneling. The domain history is also analyzed for whether the domain name has been involved in malicious activity or the geographic location of the authoritative domain name server.

Table 1
DNS traffic features and availability in encrypted traffic.

Feature	DNS	Encrypted DNS
Domain name	✓	X
Packet size	✓	✓
Record types	✓	X
Volume of traffic	✓	✓
Domain history	✓	X
Time between queries	✓	✓

The characteristics extracted from the DNS traffic discussed above are shown in Table 1. The table shows which DNS information is still visible in encrypted DNS (DNSSEC, DoT, DoH and DoQ). The domain name, the record types and the domain history are not visible when DNS traffic goes over TLS, HTTPS or QUIC because this data is encrypted [17]. Consequently, this renders the detection of DoH tunnels more challenging. However, some of these features are still visible, such as IP addresses and ports, packet length or timestamp [18]. This underscores the need to develop research on automatic techniques to detect this type of traffic, such as statistical studies and machine learning techniques.

2.2. DoH tunnel detection

The encryption of DoH traffic renders traditional methods of security analysis and tunnel detection ineffective. Hynek et al. [19] present research challenges for DoH abuse on the Web. Recently, research on DoH tunnel detection has been extended. Steadman and Scott-Hayward [20] propose an architecture that supports the analysis and detection of malicious DoH communications to mitigate data exfiltration. The results demonstrate that DoHxP accurately identifies 99.78% of the malicious traffic, while misclassifying only 0.22% as benign. In terms of the benign traffic, DoHxP correctly identifies 99.22% as benign, but misclassifies 0.78% as malicious.

The papers collected in Table 2 investigate the detection of DoH tunnels through various machine learning techniques. Several of these studies use a DoH traffic dataset generated by MontazeriShatoori et al. [21] called “CIRA-CIC-DoHBrw-2020”. Yusof et al. [22] examine and visually analyze the dataset.

Vekshin et al. [25] develop a classifier to differentiate between HTTPS and DoH traffic and another model to identify the DoH client model (Chrome, Cloudflare and Firefox) regardless of IP addresses and ports. The statistical features with the highest importance in the classification are the connection duration and average packet delay and the variance of the received packet size. The AdaBoost algorithm obtains the best accuracy.

Nguyen and Park [29] propose a DoH tunnel detection system using a semi-supervised learning technique based on a Transformer

Table 2
Related works of DoH tunnel detection.

Reference	Year	Dataset	Scope
MontazeriShatoori et al. [21]	2020	DoHBrw-2020	2 layers (HTTPS/DoH, DoH/malicious DoH)
Banadaki [23]	2020	DoHBrw-2020	2 layers (HTTPS/DoH, DoH/malicious DoH)
Singh and Roy [24]	2020	DoHBrw-2020	DoH/malicious DoH
Vekshin et al. [25]	2020	Custom	HTTPS/DoH, DoH client model
Behnke et al. [26]	2021	DoHBrw-2020	2 layers (HTTPS/DoH, DoH/malicious DoH)
Alenezi and Ludwig [27]	2021	DoHBrw-2020	Tunneling tools
Jha et al. [28]	2021	DoHBrw-2020	DoH/malicious DoH
Nguyen and Park [29]	2022	Custom	2 layers (HTTPS/DoH, DoH/malicious DoH)
Mitsuhashi et al. [30]	2022	DoHBrw-2020 DoH-HKD	HTTPS/DoH/ malicious DoH tunneling tools
Zebin et al. [31]	2022	DoHBrw-2020	HTTPS/DoH/ malicious DoH
Zhan et al. [32]	2022	Custom	DoH/malicious DoH

architecture. Although a semi-supervised learning technique, that does not require the data to be labeled, is used, the complexity of the proposed model is higher than the other models.

Zhan et al. [32] use a classifier that can detect DoH tunnels with 99% accuracy in a more realistic scenario. In addition, the experiment analyzes the influence of various characteristics like server location.

MontazeriShatoori et al. [21] studied DoH tunnel detection based on time-related features, and achieved an accuracy of approximately 100%. The time-based traffic features used are as follows: Source IP, source port, destination IP, destination port; time stamp; connection duration; number and rate of bytes sent and received; mean, median, mode, variance, standard deviation, coefficient of variation, median and mode skew of packet length, packet time and request/response time difference.

The following studies found similar results using the same dataset. The results obtained by Banadaki [23] confirm that LightGBM and XGBoost outperform the other algorithms with an accuracy of 100%. Singh and Roy [24] present several classifiers to detect malicious traffic, the Random Forest (RF) and Gradient Boosting (GB) classifiers are the best approaches obtaining an accuracy of 100%. Alenezi and Ludwig [27] study whether the proposed machine learning approaches are capable of classifying tunneling tools. The XGBoost and RF classifiers achieve better than 99% accuracy.

Behnke et al. [26] introduce a feature selection method that increases classification accuracy by decreasing overfitting. Considering the accuracy and training time, LightGBM obtains better performance. Jha et al. [28] perform feature analysis based on correlation coefficients to obtain the most significant features. The presented deep learning model achieves 99.5% accuracy. A balanced and stacked random forest that identifies malicious traffic with greater than 99% accuracy is proposed by Zebin et al. [31].

Mitsuhashi et al. [30] propose a tool to detect malicious DNS tunnels. The performance is evaluated on two datasets, CIRA-CIC-DoHBrw-2020 and DoH-Tunnel-Traffic-HKD. The results confirm that the system is able to detect the tools with an accuracy of 98.02% using the LightGBM classifier.

As mentioned above, DoH traffic enables attackers to evade existing DNS tunnel detection mechanisms, thereby introducing new research challenges in this area. One potential solution is to develop innovative detection systems using approaches such as machine learning. The majority of the papers reviewed apply machine learning techniques to analyze all the features extracted from the traffic. Table 3 presents the characteristics used in the classification of the previous studies.

Firstly, in this paper we investigate whether it is necessary to use all features to distinguish malicious from benign traffic using statistical feature analysis. The goal is to identify the most relevant features that allow us to distinguish malicious from benign traffic. We analyze whether it is possible to correctly classify traffic as benign or malicious from these features by reducing the overfitting of related works. Secondly, we present a real-time detection system for malicious DoH tunnels. The methodology of the system is based on a statistical analysis of the different features of DoH traffic.

3. Dataset

In our research, we used the “CIRA-CIC-DoHBrw-2020” [21] dataset. This dataset includes HTTPS and DoH traffic captured through different web browsers, along with malicious DoH traffic generated by DNS tunneling tools. As previously discussed, this research is focused on analyzing benign and malicious DoH traffic, hence our focus is solely on this particular type of traffic.

The authors defined a capture scenario and implemented the necessary infrastructure to generate the traffic. The traffic was generated by browsing the top 10,000 Alexa websites using two DoH-compliant browsers, Google Chrome and Mozilla Firefox. In contrast, the malicious traffic was created using tunneling tools such as dns2tcp, dnscat2, and Iodine. DoH traffic was captured between the DoH server and the DoH proxy. The DoH servers used are AdGuard, Cloudflare, Google and Quad9.

3.1. Remarks

The final dataset contains the statistical features extracted from the traffic captured using a tool developed by the authors. The tool produces a CSV file as output and the data is labeled according to network flow based on IP addresses. A DoH flow is understood as a sequence of one or more packets with the same source and destination with the singularity that there is a time limit that can elapse between two packets. Additionally, the dataset also contains the raw captured traffic files, Packet Capture (PCAP). We use the PCAP files from the dataset, which contain raw traffic and network trace information.

The dataset consists of 28 features extracted from DoH traffic, statistical features on packet length and time between packets are collected in Table 3. Although the flow information provided by the tool is extensive, it does not provide any information about sent/received packets. Our study is focused on TCP connections and the proposed tool groups DoH packets into flows. PCAP files allow a grouping of packets at TCP connection level. This is particularly relevant for the current study, which aims to develop a real-time detection system. Therefore, we are interested in processing the DoH traffic packets and not the traffic statistics. Finally, we decided to use the PCAP files directly, which contain the raw traffic.

3.2. Data processing

We use the PCAP files from the dataset containing the raw traffic, the information from network traces. First, we filter the traffic by IP to obtain the DoH traffic and extract its characteristics.

We obtain the IP addresses (source and destination), the ports (source and destination), the size (in bytes) and the timestamp of each packet from the PCAP files. We group the packets into TCP connections based on source IP, destination IP, source port and destination port. The number of data packets of the captured traffic contained in this dataset can be seen in Table 4.

When grouping the packets into TCP connections, it is observed that some connections do not contain data. Therefore, to distinguish between the number of connections with and without data, a differentiation is made. After processing, we obtain 20,063 benign data connections and a total of 126,380 malicious data connections. The

Table 3
Features used in related work.

Features	[21]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]	[31]	[32]
IP/ports	✓	✓	✓			✓					
Duration	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Bytes	✓	✓	✓	✓	✓	✓		✓	✓	✓	
Packet length	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Packet time	✓	✓	✓		✓	✓	✓	✓	✓	✓	
Time difference	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓

Table 4
Dataset details (number of connections and number of packets for each type of traffic).

Type	Web browser/ tool	Num. connections	Num. data connections	Num. packets
Benign	Chrome	2295	2293	673,991
	Firefox	39,911	17,770	2,719,007
Malicious	dns2tcp	115,437	115,228	4,200,633
	dnscat2	6095	6095	3,385,446
	Iodine	5058	5057	5,153,259

number of packets and the number of connections is higher in malicious traffic.

The following section will present the main results obtained in the traffic pattern analysis as well as a brief discussion of them.

4. Statistical analysis

In this section we present the statistical analysis of the DoH traffic performed. The aim of this analysis is to study the pattern of DoH traffic and to analyze whether there is a statistical difference between benign and malicious traffic. If there is a significant statistical difference, the malicious connections could be classified according to the statistical features.

The features obtained are analyzed and the features of the two types of traffic are then compared. In the statistical study performed, we did not consider outliers in order to visually analyze the distributions followed by the different characteristics. Although the distributions of all the characteristics mentioned above have been analyzed, only the histograms of the characteristics that are most relevant to this analysis, the distributions of the features that show the greatest difference, are shown. The histograms presented represent the distribution of malicious and benign traffic.

4.1. Connection

First, we study the statistical features extracted from each connection. The features used for the analysis of the DoH traffic pattern are extracted at the TCP connection level. We obtain 37 statistical characteristics related to duration, data packet size and time between two data packets, these characteristics are summarized in Table 5.

Fig. 2 shows the distribution of connection duration. The duration of the connection depends on the web browser or tunneling tool used, and the DoH server to which the requests are made. Most of the DNS tunneling tools that make malicious connections stay connected to the DoH server for a long time. Although other tools generate excessively short connections, as will be discussed later.

Fig. 3 shows the distribution of the ratio of received and sent direction packets, the number of packets received between the number of packets sent. The ratio of malicious traffic is concentrated in smaller values, while benign traffic has a wider distribution.

The number of packets sent per second can be seen in Fig. 3. This distribution shows the packet frequency, in which there is a difference between benign and malicious traffic. Tunneling attacks attempt to transmit large volumes of information, and this is reflected in the higher frequency values.

Table 5
Statistical features extracted from DoH traffic.

Category	Features
Duration	Connection duration
Packet size	Number of packets and bytes per connection Bytes per packet (min, max, mean, stdev) Number of packets and bytes sent Bytes per packet sent (min, max, mean, stdev) Number of packets and bytes received Bytes per packet received (min, max, mean, stdev) Down/up ratio Bytes per second Number of packets per second (total, sent and received)
Time between two packets	Time between consecutive packets (min, max, mean, stdev) Time between sent packets (min, max, mean, stdev) Time between received packets (min, max, mean, stdev)

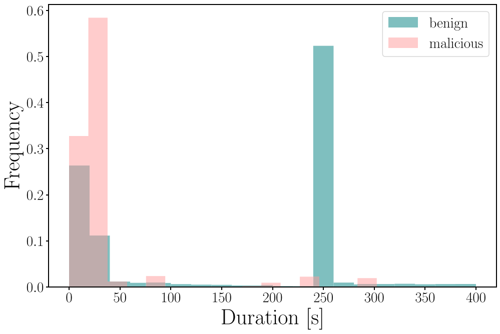


Fig. 2. Duration of benign and malicious connections in seconds.

If we look at certain statistical parameters extracted from the traffic such as connection duration, average bytes per packet sent or time between packets sent we could differentiate benign traffic from malicious traffic. The next step of this study is to apply more elaborate techniques based on the analysis performed, exploiting the statistical parameters that allow us to differentiate malicious from benign traffic. These techniques will determine whether malicious traffic can be differentiated from benign traffic based on the minimum number of features.

4.1.1. Evaluation

We apply machine learning techniques to classify benign and malicious traffic based on the features identified previously. The objective is to analyze whether the features selected by statistical analysis provide sufficient information to differentiate malicious from benign traffic. The classifiers used in this research are RF, DT (Decision Tree), GB and KNN (K-Nearest Neighbors). We have studied the application of these techniques depending on different input features extracted from the statistical analysis. The features selected as input are the duration of the connection and the number of packets per second in a connection. For classification, we used 80% of the data to train the models and the remaining 20% for testing.

In order to perform a classification, it is convenient to have balanced classes. However, as can be seen in Table 4, the number of malicious connections is 126,380 and the number of benign connections

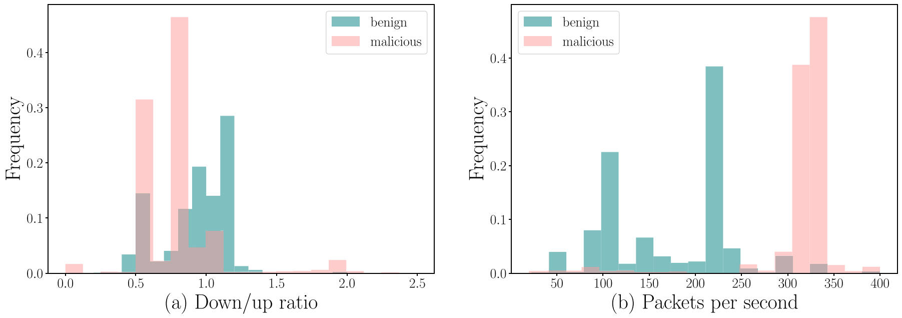


Fig. 3. Distribution of two features extracted from traffic.

Table 6
Results obtained in the classification of benign and malicious traffic.

Model	A	P	R	F1	TP	FN	FP	TN
RF	0.993	0.993	0.994	0.993	3978	26	27	3995
DT	0.992	0.992	0.993	0.992	3975	29	34	3988
GB	0.987	0.990	0.983	0.986	3936	68	40	3982
KNN	0.917	0.929	0.902	0.915	3611	393	274	3748

TP = Benign, FN = Blocked, FP = Permitted, TN = Malicious.

is 20,063, so the dataset is not balanced. To obtain a balanced distribution of both classes, we have selected the subsampling technique. We randomly select a subsample of malicious connections equal in length to the number of normal DoH traffic connections. After processing we obtain a dataset of 40,126 connections.

According to the performance metrics used in related work, the metrics used in this study are accuracy, precision, recall and F1-Score [33]. These metrics depends on the True Positive (TP) rate, False Positive (FP) rate True Negative (TN) rate and False Negative (FN) rate. Table 6 shows the comparison between the accuracy of the different machine learning models applied in this study and illustrate the number of samples that were correctly classified and those that were misclassified.

The results show that we obtain better results with the RF, DT and GB models with an accuracy of 99%. The RF and DT models perform better and achieve an accuracy higher than 99%. As can be seen, most of the connections are successfully detected by the RF model. Only 27 malicious samples are predicted as benign connections. Analogous to the previous model, the DT model has the similar performance and 34 malicious connections were not detected and permitted. GB model also achieves almost 99% accuracy and the number of malicious connections that are permitted amounts to 40. The number of undetected malicious connections is similar for the three models above. The KNN model is the worst performing model compared to the other models, although it reaches an accuracy of higher than 91%. In this model, the number of false positives increases, 393 connections are blocked and 274 connections are allowed.

The results obtained show that the selected features provide sufficient information to differentiate between malicious and normal traffic.

4.2. Packets

On the other hand, we study the distribution of different features extracted from all packets, although we only consider packets containing data for normal DoH traffic and malicious traffic.

If we examine the size of the packets, specifically the number of bytes sent that contain data in a packet, the traffic generated by the tools contains more bytes, which means that more data is sent. This difference can be seen in Fig. 4. Similarly, the size of the packets received is likely to be higher, Fig. 4, although in this case the difference is less obvious because there is more overlap.

As for the time between packets, we study the time between two consecutive packets, regardless of the direction of the message, the time between two packets sent and the time between two packets received. Fig. 5 shows the distribution of the time between two packets sent, received and consecutive, respectively, for both types of traffic. The time between two packets by tunneling tools is longer. This means that the packets of the traffic generated by the tunneling tools are sent with a lower frequency.

The distribution of features at packet level also shows a different pattern. Therefore, it is possible to distinguish malicious DoH traffic from normal DoH traffic. The next section explains the proposed system for detecting malicious connections in real time.

5. Real-time implementation

The objective of the real-time implementation is to try to detect an intrusion into the system as soon as possible by performing an IDS classifier in real time.

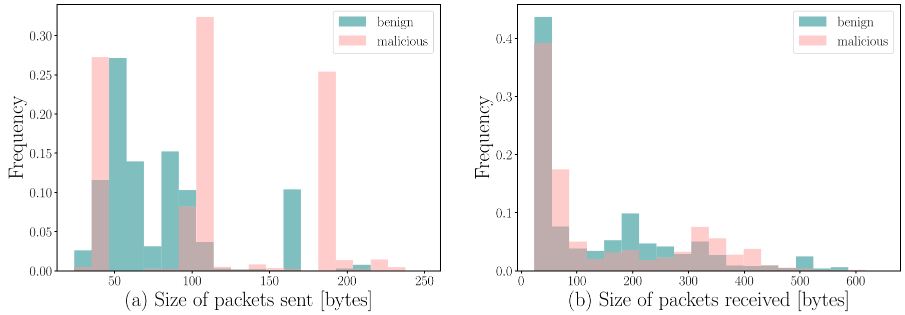
To implement the real-time detection system, we rely on a statistical analysis of the different features of DoH traffic. The goal is to get some rules to discriminate TCP connections as packets arrive. To do this, we define thresholds for certain traffic features that can be measured in real time (size of packets, time between packets, etc.). Points are then assigned if the packets meet the conditions to generate a traffic-light, once the score turns red the connection is marked as malicious. In the following we explain the feature selection process as well as the analysis performed to define the different thresholds. We also describe the methodology followed in the implemented system and the evaluation of the system.

5.1. Feature selection

Feature selection is conducted by analyzing the extracted features from the DoH traffic. These features can be at the packet level, including the number of packets, packet length, or time intervals between packets, as well as at the connection level, such as the duration of the connection. However, as mentioned in the previous section, DNS messages are transmitted in an encrypted form using TLS, which restricts the availability of certain traffic information.

Furthermore, when selecting features for real-time cyber-attack detection, it is important to consider that some statistical features may not be accessible. Statistical features like average packet size and global features like total connection duration are not readily accessible due to the limited information obtainable before a connection concludes. However, partial calculations of these values can be performed, and the values are updated as packets arrive.

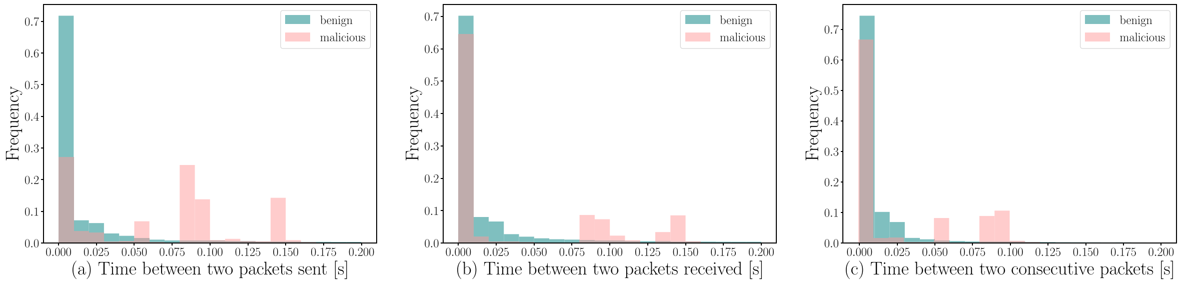
On the contrary, information at the packet or packet-by-packet level is available. The packet information encompasses packet length, distinguishing between sent and received packets, and time intervals between two packets, including both consecutive sent packets and consecutive received packets.



(a) Size of packets sent [bytes]

(b) Size of packets received [bytes]

Fig. 4. Packet size distribution of sent and received packets for benign and malicious traffic.



(a) Time between two packets sent [s]

(b) Time between two packets received [s]

(c) Time between two consecutive packets [s]

Fig. 5. Distribution of time between two packets.

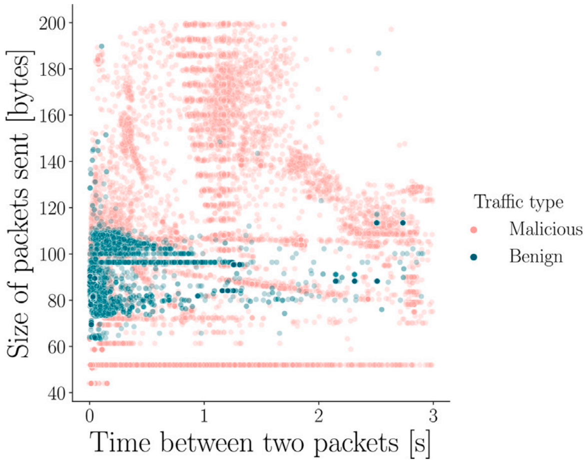


Fig. 6. Correlation between size of packets sent and time between two packets.

We analyze the distribution of the extracted features according to the type of traffic, whether benign or malicious. Although the distributions of all available features were examined, the distributions of the most significant features are shown below. These features are the size of the packets sent and the time between two consecutive packets. Fig. 6 shows the joint distribution of both features. Benign traffic is mainly concentrated in the lower left quadrant, with small values for size and time. Therefore, a threshold of size or time alone is not sufficient to adequately distinguish benign and malicious, we need both features to distinguish traffic.

5.2. Threshold definition

Threshold are established based on statistical analysis. To perform a quantitative analysis and set the necessary thresholds, we study the distributions of the features. The objective is to find a pattern in the data and fit all the data. From this fit we know where the data is concentrated and at what threshold the probability of having a packet decreases considerably.

Table 7
Defined thresholds to distinguish benign and malicious traffic.

Feature	Threshold
Length of packet sent	175 bytes
Time between two consecutive packets	0.05 s

Fig. 7 shows the distribution of the size of packets sent for both types of traffic, normal traffic and malicious traffic. From the distribution, we can see that the benign packets decrease significantly from 175 bytes, there is only a 1% probability of finding a larger packet. For malicious packets, 30% of the packets are larger.

The distribution of the time between two consecutive packets for both types of traffic are shown in Fig. 7, benign connections and malicious connections. From the distribution, we can see that the benign packets decrease significantly from 0.05 s, there is a 10% probability of finding a higher time. For malicious packets, 35% of packets are sent at a lower frequency. From the study of benign and malicious connection data we can determine threshold levels. These threshold levels are shown in Table 7.

These limits provide a method for distinguishing malicious traffic. The methodology used in the proposed system is described below.

5.3. System proposed

As mentioned above, the objective is to detect malicious connections by assessing whether the transmitted packets satisfy a predetermined set of criteria. After establishing the criteria, we proceed to analyze DoH traffic according to the methodology outlined in Fig. 8. As illustrated in the diagram, the proposed detection system consists of a collection of interconnected blocks: (1) traffic analysis, (2) feature extraction, (3) grouping packets by TCP connections, (4) feature verification and score updating, (5) score verification, and (6) traffic classification. The most important blocks are described below.

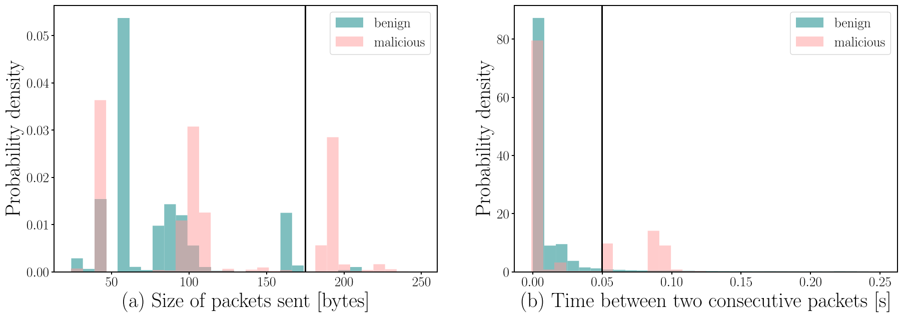


Fig. 7. Distribution of different features.

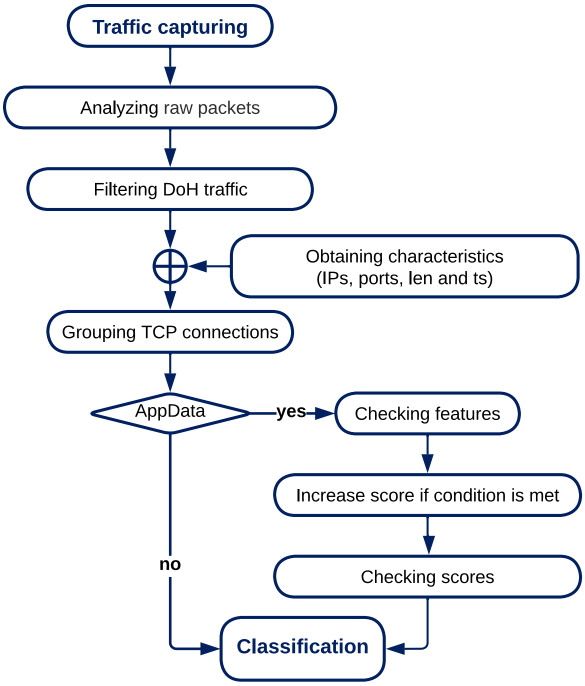


Fig. 8. Methodology of system proposed.

5.3.1. *Feature verification and score updating*

Fig. 9 details the verification process of the different characteristics values and the increase of the score if the defined limits are exceeded. Once the data packet has been added to the connection, the various selected features are checked. If the feature exceeds the pre-defined limit or threshold, the score is increased. For example, if a packet of 300 bytes arrives, one point is added because it is suspected to be from a malicious connection.

5.3.2. *Score verification*

We used a traffic-light based methodology to mark a connection as malicious as can be seen in Fig. 10. The checking of the score obtained in each connection has been described. Limits have been set at which the connection is classified as malicious. A traffic-light is created with three levels: green, orange and red. The limits have been established based on different tests performed. The upper limit, which is limit1 in the diagram shown, has been set at a value of 50 points. The second limit has been set at 25 points. If the connection exceeds the upper limit and the flag is marked as red, the connection is automatically suspicious and flagged as a malicious connection.

Table 8
Confusion matrix of the classification results.

True label		Predicted label	
		Benign DoH	Malicious DoH
Benign DoH	Chrome	89.20% 1998	10.80% 242
	Firefox	93.75% 11 328	6.25% 755
Malicious DoH	dns2tcp	96.97% 73 514	3.03% 2294
	dnscat2	2.15% 221	97.85% 10 036
	Iodine	4.40% 544	95.60% 11 807

5.4. *Evaluation*

We implemented the proposed IDS system and evaluated its performance using the data set “CIRA-CIC-DoHBrw-2020”. We use the PCAP files from the dataset containing the raw traffic. The system proposed analyzes the captured packets one by one and simulate a real-time analysis.

Table 8 shows the confusion matrix of the results obtained in the classification. High accuracy is observed in the classification of DoH traffic connections, but many malicious DoH traffic connections were misclassified as normal traffic. For benign traffic, 11% of Chrome browser connections would be blocked and 6% of Mozilla browser connections would be blocked.

In terms of malicious traffic, the performance of the system for connections from the dns2tcp tool stands out. Only 2% of connections from the dnscat2 tool are allowed and 4% of connections from the Iodine tool. On the contrary, 97% of connections from the dns2tcp tool would be allowed and not marked as suspicious.

Regarding how long it takes for the system to detect a malicious connection, it requires an average of 80 packets, which translates into an average time of 26 s, to detect a malicious connection.

The following section provides a detailed examination of the outcomes and analyzes the traffic generated by the dns2tcp tool. The goal is to gain a more comprehensive understanding of the observed deficiencies in identifying malicious network connections.

5.4.1. *Discussion of the results obtained*

We conducted an analysis of malicious connections, with a particular focus on the traffic generated by the dns2tcp tool. The goal was to explore the reasons why the proposed system presents limitations in identifying this category of connections.

Fig. 11 shows the temporal distribution of connections, illustrating how connections are distributed across the time. Benign traffic exhibits a degree of randomness that is absent in malicious traffic. In fact, malicious traffic is distinguished by a noticeable periodicity. Furthermore,

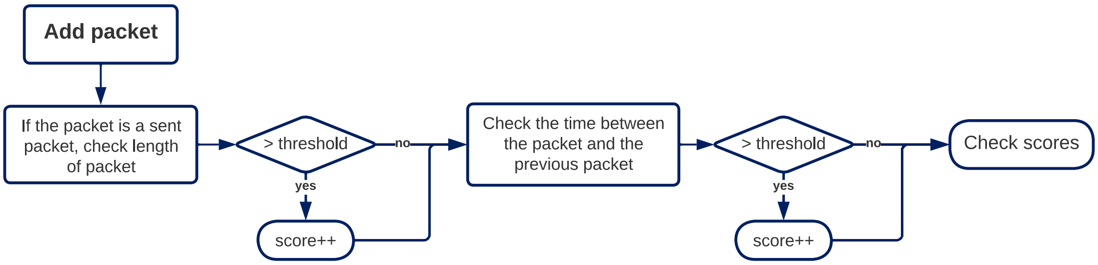

Add packet
If the packet is a sent packet, check length of packet
> threshold
no
yes
score++
Check the time between the packet and the previous packet
> threshold
no
yes
score++
Check scores

Fig. 9. Methodology of the feature verification.

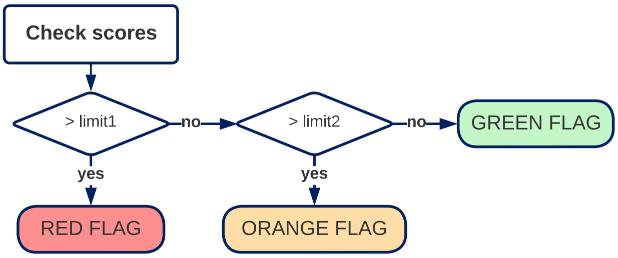


Fig. 10. Methodology of score checking. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 9
Confusion matrix.

True label		Predicted label	
		Benign DoH	Malicious DoH
Malicious DoH	tuns	0.00% 0	100.00% 134
	dnstt	50.00% 576	50.00% 576
	tcp-over-dns	3.75% 3	96.25% 77

despite the identical time period, the number of malicious connections is substantially greater.

When inspecting the traffic generated by the dns2tcp tool, we observed that the connections are short, leading to a small number of packets, as depicted in Fig. 12. This behavior complicates the identification of such malicious connections using the proposed system, as the connections are unable to surpass the predetermined thresholds.

This finding demonstrates that detecting malicious connections can prove challenging when individual connections are analyzed in isolation, without considering the frequency and volume of connections established within a short period of time.

5.4.2. Evaluation

Additionally, we evaluate the classifier by using another data set and compare the results obtained. The DoH-TunnelTraffic-HKD dataset [30] contains traffic data from three emerging tunneling tools: tuns [34,35], dnstt [36], and tcp-over-dns [37]. The traffic was captured for 48 h and then, scaled up assuming 20 clients using the tools.

Table 9 shows the confusion matrix of the results obtained in the detection of malicious DoH connections. Similar to the previous dataset, there are two types of tools that achieve almost 100% performance. There is one type that would allow 50% of malicious connections. The situation is analogous to that explained above, the connections are too limited to detect.

In this study we have implemented a real-time DoH tunnel detection system. This system analyzes DoH traffic and detects tunnels as packets arrive using an approach based on a threshold definition. The proposed system has been evaluated on dataset “CIRA-CIC-DoHBrw-2020”. This represents an initial step in our research, knowing that the insufficiency of analyzing connections individually, and thus, we aim to improve the proposed system by incorporating various additional parameters. In the next phase of our research, we intend to examine the number of requests initiated within a defined temporal interval, where a high frequency may indicate the presence of malicious traffic. Furthermore, if such traffic adheres to a regular pattern, it could potentially be classified as suspicious and subjected to quarantine for more in-depth analysis. These additional rules would improve detection by taking into account the long term behavior of tunneling tools.

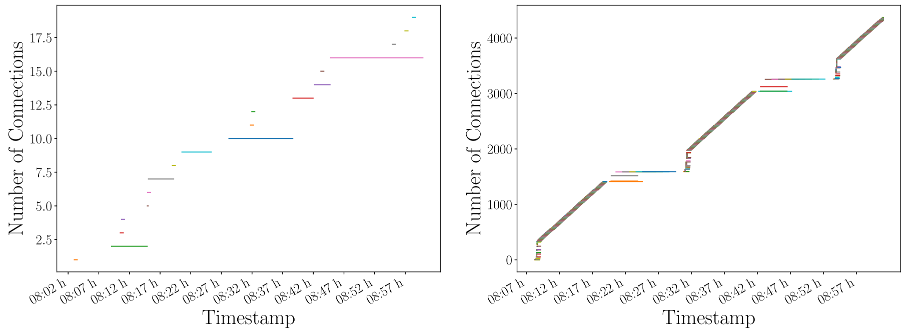
6. Conclusions

DoH was defined as a solution to the privacy problems associated with the DNS protocol. While this protocol encrypts DNS queries and mitigates attacks involving data manipulation, malicious actors exploit the fact that DNS traffic is encrypted to conduct data exfiltration attacks. In addition, tools that rely on data analysis to detect tunnels have certain limitations.

This article presents a real-time system for detecting malicious DoH tunnels. The system aims to identify potential security breaches as soon as possible using a real IDS classifier, thereby mitigating the impact of DoH tunnel attacks. In order to develop the detection system, we used a statistical analysis of the various DoH traffic features that are accessible in real-time.

We performed an analysis of benign and malicious traffic and obtained statistical parameters by partitioning the traffic into individual TCP connections. This facilitated the identification of patterns characteristic of benign and malicious traffic and enabled us to establish threshold values to distinguish malicious traffic. The proposed system performs a detection rate exceeding 95% for malicious connections. With respect to the dns2tcp tool, the findings deviated from the expected outcomes, as this tool generates a considerable volume of connections that are too brief to be identified by the detection system employing the proposed methodology.

The next phase of this study involves adapting the proposed system to account for a group of TCP connections and avoid treating them as separate and individual connections. On the other hand, the attacker can alter the malicious traffic to imitate legitimate traffic patterns. This can be achieved, for instance, by introducing packet padding or by reducing the payload size of each request and increasing the overall request frequency to mimic the distribution of DNS traffic. However, if the attacker is forced to mimic the benign DNS traffic pattern, then the attack will lose utility to the attacker by increasing the time it takes to send data and decreasing the amount of data it can send undetected. A future line of research is to study whether tunneling tools allow such traffic shaping.



(a) Benign traffic

(b) Malicious traffic

Fig. 11. Duration of connections.

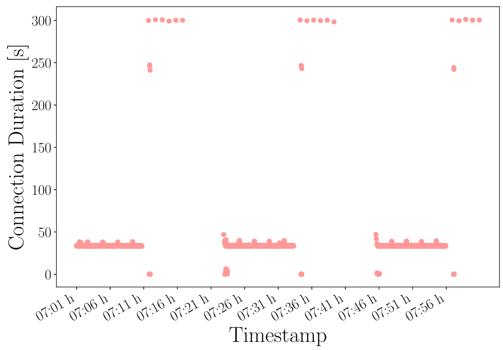


Fig. 12. Traffic generated by the dns2tcp tool.

CRediT authorship contribution statement

Marta Moure-Garrido: Methodology, Software, Validation, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Celeste Campo:** Conceptualization, Methodology, Validation, Formal analysis, Writing – review & editing, Supervision, Funding acquisition. **Carlos Garcia-Rubio:** Conceptualization, Methodology, Validation, Formal analysis, Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported by the Spanish Government under the research project "Enhancing Communication Protocols with Machine Learning while Protecting Sensitive Data (COMPROMISE)" PID2020-113795RB-C32, funded by MCIN/AEI/10.13039/501100011033, the research project "QUantum-based ReSistant Architectures and Techniques (QURSA)" TED 2021-130369B-C32 funded by MCIN/AEI/10.13039/501100011033 and by the "European Union NextGenerationEU/PRTR", and by the Comunidad de Madrid (Spain) under the project: CYNAMON (P2018/TCS-4566), co-financed by European Structural Funds (ESF and FEDER). And funding for APC: Universidad Carlos III de Madrid (Read & Publish Agreement CRUE-CSIC 2023).

References

[1] P. Mockapetris, Domain Names - Implementation and Specification, in: RFC 1035, 1987, URL https://www.rfc-editor.org/info/rfc1035.

[2] G. Schmid, Thirty years of DNS insecurity: Current issues and perspectives, IEEE Commun. Surv. Tutor. 23 (4) (2021).

[3] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, P.E. Hoffman, Specification for DNS over Transport Layer Security (TLS), in: RFC 7858, 2016, URL https://www.rfc-editor.org/info/rfc7858.

[4] P.E. Hoffman, P. McManus, DNS Queries over HTTPS (DoH), in: RFC 8484, 2018, URL https://www.rfc-editor.org/info/rfc8484.

[5] C. Huitema, S. Dickinson, A. Mankin, RFC 9250: DNS over Dedicated QUIC Connections, in: RFC 9250, 2022.

[6] D.A.E. Haddon, H. Alkhateeb, Investigating data exfiltration in DNS over HTTPS queries, in: 2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3), 2019, p. 212.

[7] N. Ishikura, D. Kondo, V. Vassiliades, I. Iordanov, H. Tode, DNS tunneling detection by cache-property-aware features, IEEE Trans. Netw. Serv. Manag. 18 (2) (2021).

[8] A. Turing, G. Ye, An analysis of godlua backdoor, 2019, 360 Netlab Blog.

[9] C. Cimpanu, Iranian hacker group becomes first known APT to weaponize DNS-over-HTTPS (DoH), 2020, URL https://zd.net/3EBD8OS.

[10] A. Merlo, G. Papaleo, S. Veneziano, M. Aiello, A comparative performance evaluation of DNS tunneling tools, in: Computational Intelligence in Security for Information Systems, Springer, 2011, pp. 84–91.

[11] E. Ekman, B. Andersson, iodine, 2014, URL https://github.com/yarrick/iodine.

[12] O. Dembour, N. Collignon, dns2tcp, 2017, URL https://github.com/alex-sector/dns2tcp.

[13] R. Bowes, dnscat2, 2017, URL https://github.com/iagox86/dnscat2.

[14] K. Xu, P. Butler, S. Saha, D. Yao, DNS for massive-scale command and control, IEEE Trans. Dependable Secure Comput. 10 (3) (2013).

[15] C.J. Dietrich, C. Rossow, F.C. Freiling, H. Bos, M.v. Steen, N. Pohlmann, On botnets that use DNS for command and control, in: 2011 Seventh European Conference on Computer Network Defense, 2011, pp. 9–16.

[16] Y. Wang, A. Zhou, S. Liao, R. Zheng, R. Hu, L. Zhang, A comprehensive survey on DNS tunnel detection, Comput. Netw. 197 (2021).

[17] M. Lyu, H.H. Gharakheili, V. Sivaraman, A survey on DNS encryption: Current development, malware misuse, and inference techniques, ACM Comput. Surv. 55 (8) (2022).

[18] K. Bumanglag, H. Kettani, On the impact of DNS over HTTPS paradigm on cyber systems, in: 2020 3rd International Conference on Information and Computer Technologies (ICICT), IEEE, 2020, pp. 494–499.

[19] K. Hynek, D. Vekshin, J. Luxemburk, T. Cejka, A. Wasicek, Summary of DNS over HTTPS abuse, IEEE Access 10 (2022).

[20] J. Steadman, S. Scott-Hayward, Detecting data exfiltration over encrypted DNS, in: 2022 IEEE 8th International Conference on Network Softwarization (NetSoft), 2022, pp. 429–437.

[21] M. MontazeriShatoori, L. Davidson, G. Kaur, A. Habibi Lashkari, Detection of DoH tunnels using time-series classification of encrypted traffic, in: 2020 IEEE Intl. Conf. on Dependable, Autonomic and Secure Computing, Intl. Conf. on Pervasive Intelligence and Computing, Intl. Conf. on Cloud and Big Data Computing, Intl. Conf. on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), 2020, pp. 63–70.

[22] M.H.M. Yusof, A.A. Almohammedi, V. Shepelev, O. Ahmed, Visualizing realistic benchmarked IDS dataset: CIRA-CIC-DoHBrw-2020, IEEE Access (2022).

[23] Y.M. Banadaki, Detecting malicious DNS over HTTPS traffic in domain name system using machine learning classifiers, J. Comput. Sci. Appl. 8 (2) (2020).

[24] S.K. Singh, P.K. Roy, Detecting malicious DNS over HTTPS traffic using machine learning, in: 2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT), 2020, pp. 1–6.

[25] D. Vekshin, K. Hynek, T. Cejka, Doh insight: Detecting DNS over HTTPS by machine learning, in: Proceedings of the 15th International Conference on Availability, Reliability and Security, ARES '20, Association for Computing Machinery, 2020, pp. 1–8.

[26] M. Behnke, N. Briner, D. Cullen, K. Schwerdtfeger, J. Warren, R. Basnet, T. Doleck, Feature engineering and machine learning model comparison for malicious activity detection in the DNS-over-HTTPS protocol, IEEE Access 9 (2021).

[27] R. Alenezi, S.A. Ludwig, Classifying DNS tunneling tools for malicious DoH traffic, in: 2021 IEEE Symposium Series on Computational Intelligence (SSCI), 2021, pp. 1–9.

[28] H. Jha, I. Patel, G. Li, A.K. Cherukuri, S. Thaseen, Detection of tunneling in DNS over HTTPS, in: 2021 7th International Conference on Signal Processing and Communication (ICSC), 2021, pp. 42–47.

[29] T.A. Nguyen, M. Park, DoH tunneling detection system for enterprise network using deep learning technique, Appl. Sci. 12 (5) (2022).

[30] R. Mitsuhashi, Y. Jin, K. Iida, T. Shinagawa, Y. Takai, Malicious DNS tunnel tool recognition using persistent DoH traffic analysis, IEEE Trans. Netw. Serv. Manag. (2022).

[31] T. Zebin, S. Rezvy, Y. Luo, An explainable AI-based intrusion detection system for DNS over HTTPS (DoH) attacks, IEEE Trans. Inf. Forensics Secur. (2022).

[32] M. Zhan, Y. Li, G. Yu, B. Li, W. Wang, Detecting DNS over HTTPS based data exfiltration, Comput. Netw. 209 (2022).

[33] M. Hossin, M.N. Sulaiman, A review on evaluation metrics for data classification evaluations, Int. J. Data Min. Knowl. Manag. Process 5 (2) (2015).

[34] L. Nussbaum, P. Neyron, O. Richard, On robust covert channels inside DNS, in: IFIP International Information Security Conference, Springer, 2009, pp. 51–62.

[35] L. Nussbaum, tuns, 2020, URL https://github.com/lnussbaum/tuns.

[36] D. Fifield, dnstt, 2020, URL https://bamsoftware.com/software/dnstt/.

[37] tcp-over-dns, 2008, URL https://analogbit.com/software/tcp-over-dns/.

Marta Moure-Garrido is a Ph.D. Student at the Department of Telematic Engineering of the UniversityCarlos III of Madrid. Her research interest is design and performance evaluation of communication protocols. She received her MS degree from the University Carlos III of Madrid in 2021. Contact her at mamoureg@it.uc3m.es.

Celeste Campo is an associate professor at the Department of Telematic Engineering of the University Carlos III of Madrid. Her research interests include design and performance evaluation of communication protocols for ad hoc networks, energy aware communications, and middleware technologies for pervasive computing. She received her Ph.D. degree from the University Carlos III of Madrid in 2004. Contact her at celeste@it.uc3m.es.

Carlos Garcia-Rubio is an associate professor at the Department of Telematic Engineering of the University Carlos III of Madrid. His research focus is centered in mobile and wireless networked computing systems, and in the design and performance evaluation of communication protocols, mainly at the transport and application layers. He received his Ph.D. degree from the Technical University of Madrid in 2000. Contact him at cgr@it.uc3m.es.