Privacy Illusion: Beware of Unpadded DoH

1st Karel Hynek
DDD & Department 707
CTU in Prague & CESNET z.s.p.o.
Prague, Czech Republic
Karel.Hynek@fit.cvut.cz

2nd Tomas Cejka
Department 707
CESNET z.s.p.o.
Prague, Czech Republic
cejkat@cesnet.cz

***Abstract*—DNS over HTTPS (DoH) has been created with ambitions to improve the privacy of users on the internet. Domain names that are being resolved by DoH are transferred via an encrypted channel, ensures nobody should be able to read the content. However, even though the communication is encrypted, we show that it still leaks some private information, which can be misused. Therefore, this paper studies the behavior of the DoH protocol implementation in Firefox and Chrome web-browsers, and the level of detail that can be revealed by observing and analyzing packet-level information. The aim of this paper is to evaluate and highlight discovered privacy weaknesses hidden in DoH. By the trained machine learning classifier, it is possible to infer individual domain names only from the captured encrypted DoH connection. The resulting trained classifier can infer domain name from encrypted DNS traffic with surprisingly high accuracy up to 90 % on HTTP 1.1, and up to 70 % on HTTP 2 protocol.**

***Index Terms*—DNS over HTTPS, DoH, DNS, Fingerprinting, Privacy, Classification, Machine Learning, Datasets**

I. Introduction

The development of modern technologies and protocols starts to pay more attention to the privacy of users. Currently used protocols are continually evolving to novel versions that help to avoid information leakage, and also some bright new protocols focused on security and privacy by design. The motivation is clear. Lack of privacy represents a severe threat for a user that may lead to information gathering by potential enemies, user profiling based on the content of activities, possible extortion.

According to findings presented in the 2018 Internet Organised Crime Threat Assessment report [1], the attacks are becoming tailored to target small businesses or larger targets. The reason is simple – targeted attacks are more lucrative. However, these types of attacks are more challenging to deploy because it requires information about the victims, their users, or employees.

Spear phishing is an example of a critical security threat that requires information about the user's activity. Let us imagine that an attacker can observe the web traffic of a user. In case an attacker gains information about visited websites and the date of access, it is much easier to plan a well-targeted social engineering attack (e.g., tailored phishing e-mail) that looks trustful. Luckily, HTTP connections are currently encrypted by TLS mechanisms, so the attacker can observe only encrypted data. However, many published research papers related to website fingerprinting present the feasibility of such estimation even on the encrypted traffic.

978-1-7281-8416-6/20/$31.00 ©2020 IEEE

Fingerprinting and activity tracking was even easier in the past due to unencrypted DNS [2], [3] mechanism to translate domain names of the websites. In practice, every web page access produces several domain queries. Security and privacy engineers proposed an enhanced mechanism of such translation that uses standard HTTPS — DNS over HTTPS (DoH) [4]. Therefore, this traffic is assumed to be resistant to eavesdropping. This paper provides an experimental evaluation of encrypted DoH from the privacy perspective that shows that DoH is still not a perfect solution in some cases.

The main goal of this paper is to highlight potential privacy weaknesses hidden in DoH, that can be exploited to disclose users' information. The greatest danger of privacy-based solutions is the users' illusion of invulnerability and potential change in his usual behavior.

We focused on behavior patterns at the network packet level. Usually, this packet level is totally out of users' control (assuming that only a web-browser is used without VPN etc.). On the other hand, it is a natural expectation that modern web-browsers should handle this scope of privacy themselves. However, our results show that some of them do not handle it sufficiently. Based on the thorough analysis of packet traces, we propose a feature vector to fingerprint encrypted DoH responses and recognize domain names by machine learning models. The accuracy of the DoH response fingerprinting is studied across different HTTP protocol versions and multiple browsers. Last but not least, we summarize possible defenses and even propose new one.

II. Related Work

DoH is a novel technology, which is starting to be massively supported by industry leaders. Currently, all most popular web-browsers, such as Firefox, Chrome, and Edge, already support DoH (the list of web-browsers that currently support DoH can be found in ZDNET article [5]). Microsoft also published an announcement [6] about DoH support in the Windows operating system for testing. Therefore, it is highly expected that the use of DoH will increase rapidly in the near future. Hence we believe, with the increasing insemination of this technology, the privacy aspects must become an essential focus of the current research. The privacy benefits and also pitfalls of DoH, in general, are mentioned in [7].

Borgolte et al. [8] provides general discussion about DoH and several areas such as performance, security, and privacy. However, the paper does not analyze the encrypted communication of DoH at the network level.

It is worth noting that our scope of interest for this paper is to examine various DoH clients, i.e., web-browsers, even though DoH providers can also be a significant privacy risk. However, our observations show that known and popular DoH providers support the modern enhancements of protocols, so it is mainly the client's responsibility that a secure connection is in use.

There has been a general skepticism that encryption of DNS alone is sufficient for preserving users' privacy (for example, [9], [10]). Therefore, the engineering community developed a DNS protocol privacy enhancement feature called EDNS padding [11]. Supporting clients sends DNS requests padded with random content to equalize the sizes of all packets, which reduces the possibility of side-channel information leakage.

According to our knowledge, the EDNS padding feature is currently supported in majority of web-browsers. However, Mozzila Firefox still does not support this feature. During the time of writing this paper, there is a one-year-old request[1] in the Mozilla bug report platform for implementing EDNS padding but, the support is still not confirmed.

The lack of encrypted DNS padding was already exploited in [12], [13]. Both papers study identification of encrypted traffic on Alexa's top websites list[2]. As a component of their feature vector, the authors use sequences of message bursts and gaps, and they were able to fingerprint webpages based on DoH traffic with very high accuracy. Additionally, both papers also studied traffic with EDNS padding feature enabled, and they were successful with more than 70 % accuracy.

The inconsistency of DNS padding usage was also mentioned by Vekshin et al. [14]. They trained ML-Based model capable of DoH recognition and even DoH client classification. They stated that padding usage is an essential feature for the client classification problem.

The unpadded website fingerprinting was also studied by Chen et al. [15]. The authors analyzed and fingerprinted all pages in web application. By observing only packet sizes, they were able to determine users' actions in an incredible amount of detail, such as the content of input forms.

Some website fingerprinting approaches also deals with padded communication. For instance, Hayes et al. [16] proposes a *k*-fingerprinting method that can be used to recognize the visited website even in the case where the packets are padded by Tor browser.

Our paper is very close to the topic of fingerprinting; however, we aim at the recognition of independent domain names just according to DoH communication. This is the main difference of other papers related to fingerprinting, which is targeted on recognition of, e.g., the whole web site.

[1]https://bugzilla.mozilla.org/show_bug.cgi?id=1543811

[2]https://s3.amazonaws.com/alexa-static/top-1m.csv.zip

Paper [17] is the only study that mentions the inferring of DoH content. However, they do not present any classification results and rather introduce a new challenge.

To our best knowledge, we are not aware of any published paper, which presents a packet-level based DoH response fingerprinting method and evaluates its accuracy.

III. Datasets

A proper dataset is an essential prerequisite for an excellent ML model. The quality of the model is directly linked with the heterogeneity of information contained in the dataset. However, to the best of our knowledge, there are no publicly available annotated datasets targeted for fingerprinting of domains in DoH. Therefore, we created our own and made it publicly available on the Zenodo platform [18].

DoH is currently supported in almost all commonly used web-browsers [5]. However, in the dataset creation and further analysis, we decided to use just Mozilla Firefox and Google Chrome browser since they are used by the majority of users [19]. Unfortunately, another mostly used browser — Safari — does not support DoH at all. We also evaluated several Chrome-based browsers such as Microsoft Edge and Opera and did not spot any difference in DoH connection from Chrome. Therefore, Firefox and Chrome are the two main representatives of DoH implementations in web-browsers.

Additionally, we evaluated the traffic from multiple DNS providers (Google, Cloudflare, and NextDNS). However, we did not observe any statistically significant differences. Therefore, the further text explains our experiments and analysis using only one resolver — Cloudflare (which is the default for multiple browsers) — for the sake of simplicity of the description.

To create the DoH communication datasets, we used several virtual machines with Windows and GNU/Linux operating systems. A simplified scheme is shown in Fig. 1. We captured the traffic from the DoH enabled web-browsers using tcpdump [20]. To automate the process of traffic generation, we installed Google Chrome and Mozilla Firefox into separate virtual machines and controlled them with the Selenium framework [21] (Tab. I shows detailed information about used browsers and environments). Selenium simulates a user's browsing according to the predefined script and a list of domain names (i.e., URLs from Alexa's top websites list in our case). The selenium was configured to visit pages in random order multiple times. For capturing the traffic, we used the default settings of each browser. We did not disable the DNS cache of the browser, and the random order of visiting webpages secures that the dataset contains traces influenced by DNS caching mechanisms.

Each virtual machine was configured to export TLS cryptographic keys, that was used for decrypting the traffic using WireShark application. Python scripts consequently processed the decrypted PCAPs and extracted the feature vectors for the datasets. The encrypted content of DoH responses was used only as a ground truth for labels at the end of the dataset preparation process.

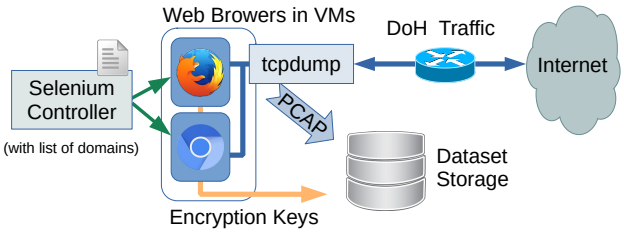

Fig. 1. Scheme of capturing datasets using Selenium, several virtual machines with web-browsers, and tcpdump. The web-browsers were forced to dump cypher keys, so the captured PCAP files can be decrypted.

TABLE I
VERSIONS OF BROWSERS AND OS USED FOR DATASET GENERATION.

Browser Name	Browser Version	OS
Mozilla Firefox	74	Fedora 31
	77.1	Windows 10
Google Chrome	83.0.4103.97	Windows 10

We captured multiple datasets with different amount of visited pages to evaluate the accuracy of the classifier with varying quantities of labels. We also used both common versions of HTTP. Detailed information about captured datasets is shown in the Tab. II. Each dataset is composed of testing and training parts of the approximately same size. The capturing of training and testing part was made during different days for even more realistic results.

Unfortunately, we were not able to enforce the Windows version of Firefox to use with HTTP 1.1 only. After disabling the HTTP 2 in the settings, the browser established the TLS connection to DoH resolver, but it used traditional unencrypted DNS, and the DoH connection remained silent. We submitted a question to the Firefox support forum, but we have received no response until the paper submission. We also could not capture the traffic of the Linux version of Chrome, because the DoH is currently supported in Windows version only.

TABLE II
OVERALL INFORMATION ABOUT CREATED DATASET CONTAINING THE NUMBER OF DOH IP FLOWS AND THE TOTAL NUMBER OF IP FLOWS. THE ABBREVIATIONS IN COLUMNS LABEL STAND FOR: **OS** – OPERATING SYSTEM, DOH RESP. – TOTAL NUMBER OF DOH RESPONSES INCLUDED IN THE DATASET, **B** – BROWSER (**F** – FIREFOX, **C** – CHROME), **HV** – HTTP VERSION, **UP** – UNIQUE WEBPAGES, **TV** – TOTAL VISITED WEBPAGES, **UD** – UNIQUE DOMAINS (NUMBER OF LABELS)

Dataset Name	OS	B	HV	UP	DoH rsp	TV	UD
Lin-Fir-H2-30	Lin	F	2	30	162,078	1,200	409
Lin-Fir-H2-50	Lin	F	2	50	230,025	2,000	455
Lin-Fir-H2-70	Lin	F	2	70	356,311	2,800	627
Win-Fir-H2-50	Win	F	2	50	147,839	2,000	445
Win-Chr-H2-50	Win	C	2	50	37,125	2,000	389
Lin-Fir-H1-30	Lin	F	1	30	110,949	1,200	308
Lin-Fir-H1-50	Lin	F	1	50	186,070	2,000	421
Lin-Fir-H1-70	Lin	F	1	70	272,470	2,800	572
Win-Chr-H1-50	Win	C	1	50	22,787	2,000	382

IV. DOH COMMUNICATION ANALYSIS

The essential part of DoH fingerprinting is a deep understanding of the traffic. Therefore, we manually analyzed decrypted raw PCAP data with the DoH communication, which we captured for our datasets.

Since DoH is quite a new protocol, there are still two coexisting significantly different implementations. The RFC [4] compliant definition uses classic DNS Wireformat [3] encapsulated in the HTTPS protocol using the GET, or POST methods. The other approach introduced by Google uses JSON based messages transferred via HTTPS GET. The majority of the DNS providers support both implementations. However, all the DoH enabled browsers, including the Chrome-based ones, and most of the other DoH clients are currently using RFC compliant Wireformat messages with the HTTPS POST method.

A. Traffic shape of DoH

The DoH traffic follows the HTTP request-response scheme, with the expected differences across browsers, e.g., in HTTP headers. On the other hand, we did not observe any differences between the Linux and Windows versions of Firefox. The most significant difference was in the use of EDNS padding by Google Chrome. All requests and responses coming from Chrome had the same size.

B. DNS over HTTP 2

The DNS over HTTP 2 communication pattern is shown in Fig. 2. The browser sends multiple DNS requests when loading the page. However, the resolver does not maintain the sequence order of queries and sends responses in an arbitrary order. This behavior makes the association of particular requests with corresponding encrypted responses impossible.

Another DNS over HTTP 2 characteristics originate from the stream management. Each request creates a new stream, which is then closed by the response. The queries and also the responses are split into exactly two datagrams. The first packet is always larger, with at least 100 bytes (total length in the IP header field). The second packet contains only HTTP2 stream control information such as End of stream flag and therefore has a fixed size of 71 bytes.

However, there are some exceptions. The *Lin-Fir-H2-30* dataset contains 162,078 responses in total, only 78 of them were received as a single packet. Those larger packets contain multiple HTTPS streams (DoH data stream & control streams), which effectively obfuscates the size od DoH communication and precludes fingerprinting. However, the number of such anomalous responses is negligible.

HTTP 2 header regarding the header compression (HPACK [22]) was also identified as an important characteristic affecting the fingerprinting. The header fields with nonpersistent content across all packets (such as timestamps) result in the different compressed header sizes. Thus packets with the same data inside the data stream might have different sizes. The data size inconsistency in

HTTP 2 is the most significant complication for DNS traffic fingerprinting, except for the EDNS padding.

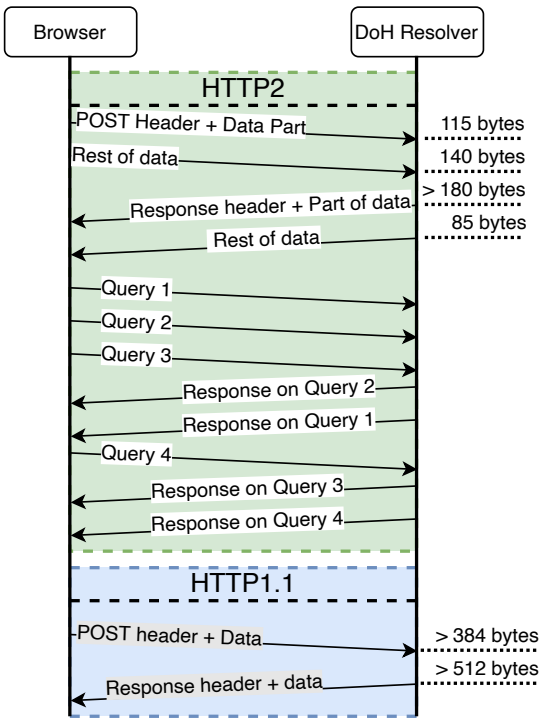

Fig. 2. DoH Communication pattern difference between HTTP 2 and HTTP 1.1. The abbreviations in legends stand for: **H1** – HTTP 1.1, **H2** – HTTP 2

C. DNS over HTTP 1.1

The HTTP 1.1 is not officially recommended by [4] due to performance reasons. The biggest performance bottleneck is that HTTP 1.1 does not support multiple concurrent requests in a single connection; therefore, it always has to wait for the response before sending the next query. According to our observations, the browsers reduce performance difficulties by creating multiple parallel connections (usually 2). By switching between connections, they are able to perform concurrent requests.

Performing DoH response fingerprinting is more feasible in case of HTTP 1.1. By observing a single TCP connection, we are able to pair each request with an appropriate response. Also, the DNS requests and responses are always placed in individual packets. The figure Fig. 3 depicts a histogram of DoH response sizes in our dataset. We can notice that the packet sizes of Chrome DoH are larger due to the applied EDNS padding. The padding effect is more noticeable in HTTP 1.1, where we observed only two packet sizes.

The differences among the analyzed DoH communication is clearly summarized in Tab. III.

V. Our Approach

This paper studies the possibility to retrieve details from the DoH connection. Previously published related works showed us a possibility of inferring the visited websites based on the

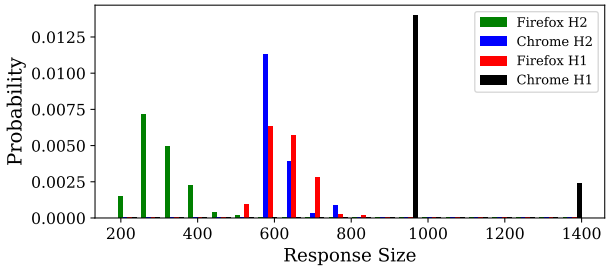

Fig. 3. The histogram of DoH packet sizes

TABLE III
Summary of observed differences across analyzed browsers' DoH communication., The abbreviations in columns label stand for: **F** – Firefox, **C** – Chrome

	HTTP 2		**HTTP 1.1**	
Browsers:	**F**	**C**	**F**	**C**
padding (EDNS)	no	yes	no	yes
split packets	yes	yes	no	no
average packet size	lower	lower	higher	higher
Multiple parallel conn.	no	no	yes	yes
order of responses	arbitrary	arbitrary	ordered	ordered
prefered DoH format	RFC	RFC	RFC	RFC
pairable req. & resp.	no	no	yes	yes

DoH traffic fingerprinting; However, we study whether it is possible to infer individual DNS queries and gain even more comprehensive insight into the traffic.

During our first experiments, we observed a considerable number of DNS queries targeting and obviously generated subdomain name[3] or subdomain name with number of particular server[4]. Those domains were often misclassified because of their similarity. We also noticed a similar problem with domains that differs only in top-level domain[5]. Therefore we reduced the problem only on inferring the second-level domain (a domain name before the top-level domain), as they provide us with most of the important information.

A. Recognition of DoH Requests&Responses in Encrypted Traffic

The essential prerequisite for inferring the encrypted domain names is the identification of DoH itself. Assuming, that the attacker can intercept the communication, the DoH connection from a web-browser can be recognized by a particular IP address of DoH resolver, or by trained machine learning classifier [14].

The shuffled order of DNS over HTTP 2 responses prevents from pairing DNS requests with a corresponding response (see Sec. IV). After further analysis, we decided to use only responses even with HTTP 1.1, where the pairing is possible. The DNS queries are often smaller and have very similar size, as it is shown in the Fig. 4; Thus, the query size in feature vector confused the classifier resulting in worse results.

[3]such as `i7gjqlci(...)0836525.nuid.imrworldwide.com`
[4]such as `i0.sinaimg.cn` and `i1.sinaimg.cn`
[5]such as `google.com` and `google.fr`

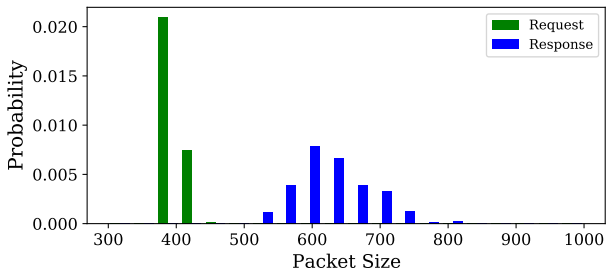

Fig. 4. The histogram of DoH packet sizes on Firefox Linux version with HTTPS 1.1.

The request-response shape of the DoH traffic is easily recognizible. Anybody, who observes the DoH, can straightforwardly recognize the DoH responses even though they are sent through the encrypted channel with the EDNS padding feature. In the case of HTTP 1.1, the attacker can distinguish the responses by filtering the communication coming from the DoH and stripping out the initial TLS handshake. The response detection in HTTP 2 is a bit more complicated, because it also contains control stream packets [23], which might be mistakenly considered as DoH responses. However, the number of control stream packets is negligible (around 1 %). Also, every response is always followed by the "end of stream" packet with precisely 71 bytes (Sec. IV), which reduces the falsely classified responses even more.

B. Feature engineering

The website fingerprinting focused papers use a large number of features obtained from the traffic. However, the field of DNS content fingerprinting is entirely different. The DoH traffic is one long TCP connection, with requests and responses. Thus the only feature we can extract from the communication is the length of the individual packets, their timestamps, and direction.

Similarly to the website fingerprinting, the size of transmitted packets would play an essential role in our feature vector. However, only the packet size feature is insufficient for DNS query fingerprinting because of the large number of collisions and packet size variation. The only other feature we can directly extract from the network is timing characteristics.

The browsers usually send batches of DNS queries in a short time period during the website loading. After the main HTML content is loaded, it usually asks for multiple sources, such as CDN, advertising server, or JavaScript libraries. For each website load, we can observe multiple DNS bursts, because each loaded asset might have other dependencies. Our analysis revealed that even though the order of responses is shuffled, the unordered set of packet sizes remains almost unchanged in one web page load. These observations are consistent with the previous website fingerprinting approaches based on DNS presented in [12], [13]. The batches of DNS queries and responses are observable at the traffic level as bursts of packets in both directions.

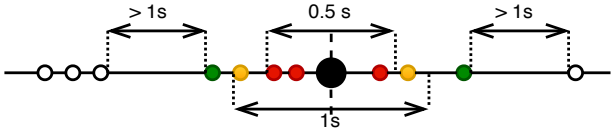

Fig. 5. Neighborhoods of a DoH response. The black dot represents the fingerprinted DoH response. Red packets belongs to *Close* neighborhood, yellow and red packets belong to *Medium* neighborhood and the green, yellow and red packets belong to *Webpage* neighborhood.

TABLE IV
CALCULATED MUTUAL INFORMATION VALUE (MI) FOR EACH EXTRACTED FEATURE. THE FEATURES WITH MI HIGHLIGHTED BY GRAY BACKGROUND WERE INCLUDED IN OUR FEATURE VECTOR. THE ABBREVIATIONS IN COLUMNS LABEL STAND FOR: **CN** – CLOSE NEIGHBORHOOD, **MN** – MEDIUM NEIGHBORHOOD AND **WN** – WEBPAGE NEIGHBORHOOD

Feature name	**Mutual Information value**		
Packet Size			**2.085**
	CN	*MN*	*WN*
Mean Size	0.866	0.848	0.892
Median size	1.133	1.097	1.053
Var. of Sizes	**1.276**	**1.297**	**1.841**
Num. packets in Neigh.	0.751	**0.824**	1.429
Max size in Neigh	**1.390**	**1.374**	**1.330**
Min size in Neigh	**1.376**	**1.268**	**1.015**
Num. of larger packets in Neigh.	0.413	0.448	0.606
Num. of smaller packet in Neigh.	0.468	0.509	0.639

For each DoH response, we consider three neighborhoods — *Close*, *Medium*, and *Webpage*. The *Close* neighborhood includes only the responses that belong to a single burst of communication. The *Webpage* neighborhood includes all responses that are related to the whole page load. The *Medium* neighborhood was added to the feature vector as a trade of between a burst and webpage. The sizes of each neighborhood were determined experimentally and are depicted in Fig. 5. Assuming the fingerprinted packet is in the middle of the interval. The *Close* neighborhood includes all packets within the half-second interval, the *Medium* part includes all packets within the second interval, and the *Webpage* neighborhood includes all packets, that are bounded at least one-second-long communication gap with zero responses.

Together from all three time intervals, we extract 29 features based on packet sizes. After calculating the feature score with Mutual Information, we reduced the feature list to the final 11. All identified features are clearly outlined in Tab. IV.

C. Algorithm selection

We experimented with multiple supervised learning algorithms and evaluated their precision. The models such as C4.5 decision tree [24] or K-Nearest Neighbours [25] performed poorly; thus, we decided to focus on ensemble algorithms.

At first, we experimented with various stacked [26] model architectures, especially the state-of-the-art k-fingerprinting approach based architecture. However, in our initial testing, the k-fingerprinting [16] based ensemble performed with around 40 % precision. We achieved the best results using the combi-

TABLE V
EXPERIMENTALLY SELECTED VALUES OF MODEL HYPERPARAMETERS.

Algorithm	Hyperparameter name	Value
C4.5 Desicision tree	Max Depth	30
	Min. number of samples in leaf node	1
Adaboost	Number of estimators	3
Bagging	Max. ratio of features	0.4
	Max. ratio of data	0.4
	Number of estimators	55

nation of the AdaBoosted decision tree [27] and the Bagging meta-learning algorithm [28].

The AdaBoost ML algorithm is sequentially learning multiple decision trees, one tree in each iteration. Each consecutive iteration attempts to correct the errors from the models trained in the previous iteration.

The Bagging meta-learning algorithm then trains multiple AdaBoosted decision trees, each on a subset of training data and subset of features. The Bagging approach is designed to reduce variance in classification accuracy and train a robust and stable model. The training on feature and data subsets also effectively prevents the dataset overfitting.

The Hyperparameters of our classifier was set experimentally and the most important of them are written in the Tab. V

D. Classification Output

To make the classification more reliable, we used a multi-label output approach. The output of our classification algorithm might be multiple most probable domains (output label vector). We also added an extra label – *None* – for difficult cases, where the classifier is uncertain.

When the confidence of the classifier is larger than the probability threshold value, the domain is added into the output label vector. Our aim was to keep the length of the output label vector under two possible resulting domains. After the experimental evaluation with multiple datasets, we found a threshold value of 10 %, which results in an average output vector length of 1.6 – 1.7 domains.

VI. RESULTS

This section evaluates the possibility of DoH response fingerprinting based on the described feature vector. We measured the performance of the classifier according to its accuracy and the number of unassigned labels (i.e., *None* label). We trained the classifier on the training part of each dataset, and then we performed the classification in the test parts. The results of the classifier were divided into three groups. *None* – The classifier was not able to assign any label. *True* – One of the domains contained in the output label vector was indeed queried. *False* – The classifier did not recognize the queried domain correctly and assigned a wrong label. The accuracy is then calculated only from the class with assigned labels.

A. DNS Content Fingerprinting Accuracy with HTTP 2 Datasets

The detailed results of the classifier used with HTTP 2 are written in the Tab. VI. We can notice that the accuracy of our classifier on the Firefox traffic varies around 70 %, which is surprisingly high. As it can be seen, the classifier does not perform significantly worse with a larger number of unique webpages. The 70 % accuracy and only 10 % of unclassified responses might suggest, that the unpadded DoH is a serious privacy threat for Firefox users. Compared to the performance of Google Chrome, where the classifier performs poorly.

TABLE VI
THE PRECISION OF TRAINED CLASSIFIER WITH HTTP 2 DATASETS. THE VALUES IN BRACKETS SHOW THE RATIO OF CLASS IN THE TEST PART OF THE DATASET. THE ABBREVIATIONS IN COLUMNS LABEL STAND FOR: **AL** – AVERAGE LENGTH, **ACC.** – ACCURACY

Dataset name	None	True	False	AL	Acc.
Lin-Fir-H2-30	9.5 %	64.5 %	26 %	1.7	71.33 %
Lin-Fir-H2-50	14.2 %	56.7 %	29.1 %	1.6	66.16 %
Lin-Fir-H2-70	10.1 %	62.5 %	27.4 %	1.6	69,52 %
Win-Chr-H2-50	28.8 %	12.2 %	58.9 %	1.7	17.23 %
Win-Fir-H2-50	11.7 %	64.8 %	23.4 %	1.7	73.46 %

B. Classifier Precision with HTTP 1.1

According to our evaluation, the DoH connections without padding that use HTTP 1.1 are even worse for the users' privacy. In Tab. VII, we can see that the accuracy of our classifier is around 90 %, which is higher than in the previous case. Additionally, the amount of *None* labels is almost negligible. Contrary to the HTTP 2 cases, we observe a slightly decreasing inaccuracy with higher number of web pages. However, this decrease is not linear, so it would not be substantial with larger datasets.

C. Open-world evaluation

In the previous experiments, we knew which domain names were resolved by the user, and these domains were also included in our datasets (this way of experiment is usually referred as a closed-world environment). However, as we know from the website fingerprinting area, the classifier is also usually tested with "unknown" webpages that were not seen during the testing phase as well. This open-world evaluation approach is more realistic because in practice, it is expected that a possible attacker has observed only a limited number of websites. Inspired by the website fingerprinting, we applied

TABLE VII
DNS CONTENT FINGERPRINTING ACCURACY WITH HTTP 1.1 DATASETS. THE VALUES IN BRACKETS SHOW THE RATIO OF CLASS IN THE TEST PART OF THE DATASET. THE ABBREVIATIONS IN COLUMNS LABEL STAND FOR: **AL** – AVERAGE LENGTH, **ACC.** – ACCURACY

Dataset name	None	True	False	AL	Acc.
Lin-Fir-H1-30	1 %	89.2 %	9.8 %	1.7 %	90.14 %
Lin-Fir-H1-50	3 %	85 %	12 %	1.7 %	87.5 %
Lin-Fir-H1-70	4.3 %	82.7 %	13 %	1.6 %	86.34 %
Win-Chr-H1-50	56.8 %	4.6 %	38.6 %	1.6 %	10.73 %

TABLE VIII
FIVE MOST AND LEAST SUCCESSFULLY CLASSIFIED DOMAINS. THE RATIO IS CALCULATED AS $size(True)/size(False)$

	Best		**Worst**	
	Name	Ratio	Name	Ratio
1	imrworldwide	189.0	bdstatic	0.016
2	smartadserver	35.6	lxsvc	0.030
3	pubmatic	25.4	scimedia	0.045
4	mozilla	20.7	zgjx	0.045
5	netflix	15.6	bdydns	0.051

this open-world evaluation approach on the DoH responses fingerprinting, where the classifier must recognize uknown domain names.

We simulated the open-world environment by training our classifier with the *Lin-Fir-H2-30* dataset, and then we evaluated it by the *Lin-Fir-H2-70*. The precision strongly depends on the probability threshold value, and we achieved 50 % accuracy with a threshold value set to 20 %. However, the *None* label was assigned to 80 % of answers. Therefore, the classifier determines the correct label only in 10 % of DoH responses. The similar results were achieved also with HTTP 1.1 datasets. The poor performance in the open-world environment can be improved by combining the classifier with website fingerprinting methods to recognize known webpages that are included in the training dataset. Naturally, increasing the size of the training set of the "known" domain names also works well to improve the accuracy for a potential attacker.

D. Discussion of Evaluation

Our classifier performed with a very high accuracy in the closed-world environment using the DoH traffic without enhanced privacy protection. The accuracy, especially for HTTP 1.1 connections, is surprisingly high and proves the importance of fingerprinting defenses such as EDNS padding.

However, the classifier has also some weak points. We observed that the accuracy of the classification of the DoH responses depends on the referring web page, which creates a context of the DoH communication. Tab. VIII contains the list of domain names with the highest, resp. lowest classification accuracy, and the *True*/*False* ratio. The results show that the classifier accuracy is not uniform for all domain names, i.e., some domain names are much easier recognized than others.

The second weak point is the performance in an open-world environment, which reduces the possibility of attack deployment. Therefore, an attacker must create large training dataset with as many domain names as possible to prepare an efficient classifier. However, the results show the attack works and the performance can be improved with some additional information, e.g., results of some existing website fingerprinting approaches, or knowledge about destination IP addresses. Our main finding is the possibility to retrieve an incredible level of detail (particular domain names) from the encrypted DoH connections without padding.

VII. DEFENSE AGAINST DOH RESPONSE FINGERPRINTING

The biggest weakness of DoH is the observable request-response pattern. According to our evaluation (in Sec. VI), the content padding is an efficient defense against response fingerprinting.

However, the padding also has several disadvantages. The attacker can easily observe the number of resolved domains and timing characteristics, even in EDNS padding enabled connections. This is a dangerous amount of information that can be misused. Moreover, according to to [13], the padding does not completely prevent website fingerprinting using DoH traffic. Last but not least, the padding also requires additional communication resources, so it necessarily reduces the available network throughput and affects performance.

The other possible defense is sending a confusion query to unrelated domains during webpage load, which is similar to camouflage website defense presented in [29] for website fingerprinting. The goal is to mix multiple unrelated traffic and make any DNS based fingerprinting impossible. The browser can issue a resolve request for the random domain or periodically update DNS cache for popular websites. Therefore the confusion packets might be useful and by preloading even speed up the user experience. However, by using this approach, we can still observe the request-response scheme and infer a number of queries and responses.

The visibility into DoH traffic can be reduced even more by supporting multiple queries and multiple responses inside one packet. According to our observation, browsers usually send a large number of queries simultaneously. This behavior creates bursts of messages. For example, 98.67 % of DoH responses were sent within a burst (i.e., their *Close* neighborhood was not empty) in the *Lin-Fir-H2-30* dataset. The responses (as well as requests) within the same burst might be merged into a joint HTTP message. This approach would hide the number of resolved domain names, and it would certainly prevent DoH response fingerprinting.

Multiple DNS queries inside a single HTTP message are not currently supported by DoH specification [4]. However, the same effect can also be reached in HTTP 2, by a proper handling of data streams and sending multiple of them in one TLS record.

The multi-DNS message approach can be combined with data padding, and confusion query approach, which even more strengthens the defense against the fingerprinting with a lower impact on the network throughput (the amount of useful transmitted data would be larger). Last but not least, merged DNS queries would disrupt the feasibility of website fingerprinting methods based on observing DoH traffic.

VIII. CONCLUSION

DNS over HTTPS (DoH) is a natural reaction of the engineering community related to IETF to deal with privacy issues of the currently used DNS protocol. The main principle of DoH is to encapsulate standard DNS queries and answers into encrypted communication of HTTPS. It is clear that the

encryption hides the content of the users' queries. Relying on the encryption mechanism, users expect that their resolved domain names remain private from network operators and potential attackers who would like to track users' activities.

To challenge this expectation of the increased level of privacy using DoH, we have focused on a comprehensive analysis of DoH traffic at the packet level. The aim was to check the possibility to reveal more in-depth information about resolved domain names by an attacker.

The research task required creating specific DoH traffic datasets, that would contain packet-level information and annotation with ground truth labels. The created datasets of traffic generated by Firefox and Chrome web-browsers in GNU/Linux and Microsoft Windows environments were used for evaluation and made publicly available to the research community.

Surprisingly, our experiments show that it is possible to recognize particular domain names even though DoH uses a TLS connection. According to our results, the best accuracy of our classifier was in the traffic of DoH that uses old HTTP 1.1 without EDNS padding extension. That means that this case is the worst option for users' privacy because the accuracy of the classifier reached about 90 %. The currently existing HTTP 2 performed much better and protected the privacy more efficiently. However, the classifier was able to reach about 70 %, which is still incredibly high. Our results prove the necessity of using defense techniques against fingerprinting such as EDNS padding, which reduces the classifier accuracy to 17.23 % (HTTP 2), and 10.73 % (HTTP 1.1).

Naturally, the highest accuracy was achieved in the so-called closed-world setup of the experiment, which means classifiers could learn all domain names from the training dataset. However, we evaluated the open-world environment experiment, as well. Accuracy of the classifier was lower as expected; however, the more domain names an attacker has in the training dataset, the better accuracy the classifier can achieve.

At the end of our evaluation, we proposed some precautions that can be used as defense mechanisms to ensure better privacy. It is worth noting that our experiments showed that it is able to train a classifier based on machine learning techniques that can reveal the activity of a user even from the encrypted traffic of DoH. Whereas, the main difference between our research and other related works is the level of details. We focused on the identification of particular domain names contrary to whole websites fingerprinting.

Acknowledgment

This work was supported by the European Union's Horizon 2020 research and innovation program under grant agreement No. 833418 and also by the Grant Agency of the CTU in Prague, grant No. SGS20/210/OHK3/3T/18 funded by the MEYS of the Czech Republic and the project Reg. No. CZ.02.1.01/0.0/0.0/16_013/0001797 co-funded by the MEYS and ERDF

References

[1] Europol, *Iocta, Internet Organised Crime Threat Assessment : 2018*. European Union Agency for Law Enforcement Cooperation 2018., 2018. [Online]. Available: https://doi.org/10.2813/858843

[2] P. Mockapetris, "Domain names - concepts and facilities," RFC 1034 (Internet Standard), RFC Editor, Fremont, CA, USA, Nov. 1987.

[3] P. Mockapetris, "Domain names - implementation and specification," RFC 1035, Tech. Rep. 1035, Nov. 1987.

[4] P. E. Hoffman and P. McManus, "DNS Queries over HTTPS (DoH)," RFC 8484, Tech. Rep. 8484, Oct. 2018.

[5] C. Cimpanu, "Here's how to enable DoH in each browser, ISPs be damned," Feb 2020, https://www.zdnet.com/article/dns-over-https-will-eventually-roll-out-in-all-major-browsers-despite-isp-opposition/.

[6] T. Jensen, "Windows insiders can now test dns over https," May 2020. [Online]. Available: https://techcommunity.microsoft.com/t5/networking-blog/windows-insiders-can-now-test-dns-over-https/ba-p/1381282

[7] B. Dickson, "Does google chrome's dns-over-https (doh) feature enhance your privacy?" Dec 2019. [Online]. Available: https://bdtechtalks.com/2019/12/11/google-chrome-dns-over-https-privacy/

[8] K. Borgolte, T. Chattopadhyay, N. Feamster, M. Kshirsagar, J. Holland, A. Hounsel, and P. Schmitt, "How dns over https is reshaping privacy, performance, and policy in the internet ecosystem," *Performance, and Policy in the Internet Ecosystem (July 27, 2019)*, 2019.

[9] S. Bortzmeyer, "Dns privacy considerations," Internet Requests for Comments, RFC Editor, RFC 7626, August 2015.

[10] H. Shulman, "Pretty bad privacy: Pitfalls of dns encryption," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, 2014, pp. 191–200.

[11] A. Mayrhofer, "The EDNS(0) Padding Option," RFC 7830, May 2016.

[12] J. Bushart and C. Rossow, "Padding ain't enough: Assessing the privacy guarantees of encrypted dns," *arXiv preprint arXiv:1907.01317*, 2019.

[13] S. Siby, M. Juarez, C. Diaz, N. Vallina-Rodriguez, and C. Troncoso, "Encrypted DNS –> Privacy? A Traffic Analysis Perspective," 2019.

[14] D. Vekshin, K. Hynek, and T. Cejka, "Doh insight: Detecting dns over https by machine learning," ser. ARES '20. New York, NY, USA: ACM, 2020. [Online]. Available: https://doi.org/10.1145/3407023.3409192

[15] S. Chen, R. Wang, X. Wang, and K. Zhang, "Side-channel leaks in web applications: A reality today, a challenge tomorrow," in *2010 IEEE Symposium on Security and Privacy*, 2010, pp. 191–206.

[16] J. Hayes and G. Danezis, "*k*-fingerprinting: A robust scalable website fingerprinting technique," in *25th USENIX Security Symposium*, 2016, pp. 1187–1203.

[17] S. Siby, M. Juarez, N. Vallina-Rodriguez, and C. Troncoso, "Dns privacy not so private: the traffic analysis perspective," 2018.

[18] K. Hynek and T. Cejka, "Dataset used for fingerprinting of DNS over HTTPS responses." Sep 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4039587

[19] "Browser market share worldwide," May 2020. [Online]. Available: https://gs.statcounter.com/browser-market-share

[20] Tcpdump Group, "Tcpdump/libpcap public repository," 2020. [Online]. Available: https://www.tcpdump.org/index.html

[21] "Selenium automates browsers. that's it!" 2020. [Online]. Available: https://www.selenium.dev/

[22] R. Peon and H. Ruellan, "HPACK: Header Compression for HTTP/2," RFC 7541, May 2015.

[23] M. Belshe, R. Peon, and M. Thomson, "Hypertext Transfer Protocol Version 2 (HTTP/2)," RFC 7540, May 2015.

[24] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[25] C. Stanfill and D. Waltz, "Toward memory-based reasoning," *Commun. ACM*, vol. 29, no. 12, p. 1213–1228, Dec. 1986.

[26] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, Jan. 1992. [Online]. Available: https://doi.org/10.1016/s0893-6080(05)80023-1

[27] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *ICML*, 1996.

[28] L. Breiman, "Bagging predictors," *Mach. Learn.*, Aug. 1996. [Online]. Available: https://doi.org/10.1023/A:1018054314350

[29] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, "Website fingerprinting in onion routing based anonymization networks." New York, NY, USA: ACM, 2011. [Online]. Available: https://doi.org/10.1145/2046556.2046570