Data in Brief (2022)

Contents lists available at ScienceDirect

Data in Brief

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

Data Article

Collection of Datasets with DNS over HTTPS Traffic

Kamil Jeřábek[a], Karel Hynek[b,c,1], Tomáš Čejka[c], Ondřej Ryšavý[a]

[a] *Faculty of Information Technology BUT, Bozetechova 1/2 612 00 Brno, Czech Republic*
[b] *Faculty of Information Technology CTU, Thakurova 9, 160 00 Prague, Czech Republic*
[c] *CESNET a.l.e, Zikova 4, 160 00 Prague, Czech Republic*

ARTICLE INFO

Keywords:

DNS over HTTPS
DNS
HTTPS
Computer
Network
Monitoring
Network Traffic

ABSTRACT

The DNS over HTTPS (DoH) is becoming a default option for domain resolution in modern privacy-aware software. Therefore, researchers have already focused on various aspects; however, a comprehensive dataset from an actual production network is still missing. This paper presents a collection of novel datasets comprising multiple PCAP files of DoH and HTTPS traffic. The captured traffic is generated towards multiple DoH providers to cover differences of various DoH server implementations and configurations. In addition to generated traffic, we also provide real network traffic captured on high-speed backbone lines of a large Internet Service Provider with around half a million users. Even though the network identifiers (excluding network identifiers of DoH resolvers) in the real network traffic (e.g., IP addresses and transmitted content) were anonymized, the essential characteristics of the traffic can still be obtained from the data. Therefore, the dataset can be used in whole network traffic analysis areas such as traffic classification research.

*Corresponding author: Karel.Hynek@fit.cvut.cz, Tel.: +420 224 359 725,
e-mail: ijerabek@fit.vutbr.cz (Kamil Jeřábek), Karel.Hynek@fit.cvut.cz (Karel Hynek), cejkat@cesnet.cz (Tomáš Čejka), rysavy@fit.vut.cz (Ondřej Ryšavý)

http://dx.doi.org/10.1016/j.dib.xxxx.xx.xxx

Specifications Table

Subject	Computer Networks and Communications
Specific subject area	Monitoring of encrypted network traffic, its analysis and classification
Type of data	Packet captures in the form of PCAP files and CSV files with flow data enriched for TLS metadata such as Server Names (from TLS SNI extensions), JA3 fingerprint, and used application protocol (from TLS ALPN extension)
How data were acquired	There are two types of datasets – *(i)* Generated, and *(ii)* Real-world. The *generated* datasets was obtained by generating traffic by DoH enabled web browsers towards multiple DoH resolvers. The *real-world dataset* was obtained by capturing at the perimeter CESNET2 network, a large internet service provider network type with around half a million users.
Data format	Raw (PCAP files) and Analyzed (CSV files)
Description of data collection	Datasets were collected using a tcpdump packet capture program locally on computer hosts or on the monitoring points located at the perimeter of the CESNET2 network. The CESNET2 capturing was performed with packet filtering based on 246 IP addresses of known DNS over HTTPS resolvers and port 443. The non-DoH traffic was captured based on IP filtering of addresses space assigned to university campus located in the Czech Republic. Packet capturing was performed on each monitoring point separately. The partial captures were then merged together and anonymized. The anonymization hides actual MAC addresses, transferred payload, and IP addresses of the DoH Clients. The IP addresses of DoH resolvers were left intact.
Data source location	Brno University of Technology, Brno, Czech Republic. CESNET z.s.p.o, Prague, Czech Republic.
Data accessibility	Repository name: Zenodo Data identification numbers: [10.5281/zenodo.5957277, 10.5281/zenodo.5957121, 10.5281/zenodo.5957420, 10.5281/zenodo.5957465, 10.5281/zenodo.5957676, 10.5281/zenodo.5957659, 10.5281/zenodo.5956043, 10.5281/zenodo.6024913] Direct links to dataset parts: **DoH-Gen-F-AABBC** — https://doi.org/10.5281/zenodo.5957277 **DoH-Gen-F-FGHOQS** — https://doi.org/10.5281/zenodo.5957121 **DoH-Gen-F-CCDDD** — https://doi.org/10.5281/zenodo.5957420 **DoH-Gen-C-AABBCC** — https://doi.org/10.5281/zenodo.5957465 **DoH-Gen-C-DDD** — https://doi.org/10.5281/zenodo.5957676 **DoH-Gen-C-CFGHOQS** — https://doi.org/10.5281/zenodo.5957659 **DoH-Real-world** — https://doi.org/10.5281/zenodo.5956043 **Supplementary files** — https://doi.org/10.5281/zenodo.6024913

Value of the Data

- Presented datasets collection represents unique captures of DNS over HTTPS (DoH) traffic and other HTTPS traffic. The real-world dataset comes from a real Internet Service Provider (ISP) backbone lines servicing half-million users. Additionally, the remaining datasets consist of traffic generated by several techniques and clients towards 16 selected DoH servers with different configurations and supporting different features. The generated part represents 64,000 loads of web pages and related DoH communication. The aim of the datasets is to provide a comprehensive sample of DoH traffic as observed in real networks and for various implementation and configuration specifics.
- Researchers can use the provided datasets collection as i) a benchmark for the DNS over HTTPS protocol classification and detection algorithms [1, 2], ii) for studying differences between the behavior and performance of various DNS resolution methods [3, 4], and because of the presence of both of DoH and HTTPS communication also for other iii) encrypted traffic analysis research [5, 6].

- The datasets collection enables researchers to experiment with DNS over HTTPS traffic recognition and pattern analysis. However, since the data are provided in raw packet captures, it can be suitable for a number of various network traffic analysis tasks, e.g., it can be used as a comprehensive benign traffic sample in malware identification challenges [7, 8].
- The datasets collection provides a unique combination of generated and labeled real-world traffic. Hence, it provides necessary ground truth data for training network classifiers and evaluating their performance. Moreover, packet captures can be used for IP Flow-based traffic analysis, detection, and classification to experiment with novel traffic features.
- The datasets collection can also improve the understanding of DoH behavior and other HTTPS traffic phenomena in a large-scale network environment. Researcher can observe and analyze performance and other relevant metrics to improve the specification and implementations.
- We are not aware of any other dataset of the comparable size and variety that provides real HTTPS traffic captured on a large-scale ISP in the form of raw packet captures that enable unbounded extraction of arbitrary available features.

1. Data Description

The collection of datasets contains DoH and HTTPS raw packets, together it contains 430 GB of traffic. The datasets were captured in two enviroments — the (i) *generated* DoH and HTTPS traffic from a controlled experimental system and (ii) *real-world* DoH and HTTPS traffic from a real large ISP network.

The aim of generated data is to provide heterogeneous DoH traffic for various existing DoH implementations. Real-world captures aim to capture the effects of the real network environment on communication, including packet timing and connections errors. The captured traffic was anonymized to protect the privacy of real users (packets have no payload, clients IP addresses were hashed); however, the anonymization does not pose a limitation for data usability and traffic research. The dataset consists of full-packet captures provided in standardized PCAP format [9], which is a default format for libpcap library. Therefore PCAP format is broadly supported by the network analysis software such as Wireshark[1], tcpdump[2] or Intrusion Detection Systems such as Suricata[3]. Full-packet captures can be considered the primary source of raw network data that can be used for further analysis or feature extraction.

In addition to raw packet captures, we also provide CSV files of flow data enriched with TLS information to compensate for the payload strip-off in real network data. The flow data were generated from all captures (even the generated ones for easy to use and dataset consistency). It contains server names (TLS SNI extension), used application protocol (TLS ALPN extension), and TLS JA3 fingerprints[4]. All of these values are commonly used in encrypted traffic analysis [10, 6, 11]. The description of provided flow data fields is written in Tab. 2.

1.1. Generated Data

The datasets with generated traffic was created in an isolated, controlled environment to provide samples of DoH communication for different existing implementations as mainly found in

[1] wireshark.com
[2] www.tcpdump.org
[3] suricata.io
[4] https://github.com/salesforce/ja3

Table 2: The description of column headers in CSV files with extended flow data.

Column Name	Column Description
DST_IP	Destination IP address
SRC_IP	Source IP address
BYTES	The number of transmitted bytes from Source to Destination
BYTES_REV	The number of transmitted bytes from Destination to Source
TIME_FIRST	Timestamp of the first packet in the flow in format YYYY-MM-DDTHH-MM-SS
TIME_LAST	Timestamp of the last packet in the flow in format YYYY-MM-DDTHH-MM-SS
PACKETS	The number of packets transmitted from Source to Destination
PACKETS_REV	The number of packets transmitted from Destination to Source
DST_PORT	Destination port
SRC_PORT	Source port
PROTOCOL	The number of transport protocol
TCP_FLAGS	Logic OR across all TCP flags in the packets transmitted from Source to Destination
TCP_FLAGS_REV	Logic OR across all TCP flags in the packets transmitted from Destination to Source
TLS_ALPN	The Value of Application Protocol Negotiation Extension sent from Server
TLS_JA3	The JA3 fingerprint
TLS_SNI	The value of Server Name Indication Extension sent by Client

operating systems and web browsers. Browsers are capable of DoH resolution and also generate HTTPS traffic by fetching websites. Hence, they create ideal applications for our traffic generation. The collection of datasets with generated data contains captures from both Firefox and Chrome browser.Tab. 3 shows the properties of published datasets of generated data.

The captured files contain a mix of non-DoH HTTPS and DoH traffic. To discriminate the DoH traffic, we provide with each dataset a list of IP addresses of known DoH resolvers used in our experiments. This list can be used to identify the DoH flows in the PCAP files.

A sample of 16 different DoH servers was chosen to support a higher diversity of traffic characteristics in the process of domain name resolution. For each DoH server, we generated traffic by loading 2000 websites on each considered web browser. Totally, 64,000 websites loads were generated and captured in the dataset. As Firefox web browser supports both GET and POST methods for resolving DoH, 1000 websites were visited by using the DoH GET method, and the other 1000 websites visit were performed with the DoH-POST method. Moreover, traffic generated with Firefox browser contain larger packets (reaching up to 64 KB), which can occur in virtualized infrastructures with TCP offloading enabled [12]. The inclusion of packets with large MTU allows dataset usage for training models targeting virtualized infrastructures, thus increasing the comprehensiveness of the network data.

Datasets with Firefox data have captures located at `/data/generated/pcap/firefox/` directory. Each filename consists of the HTTP method involved in resolution (GET, POST) and the name of the DoH server used for resolution. Tab. 4a shows summary metrics of all files created by Firefox across all generated datasets in the collection.

The datasets with google Chrome web browser traffic has captures located at `/data/`

Table 3: Properties of Generated datasets. Abbrevation stands for: **Brws** — Used Browser, **F** — Firefox, **C** — Chrome

Name	DoI	Brws	Resolvers
DoH-Gen-F-AABBC	10.5281/zenodo.5957277	F	AdGuard, AhaDNS, BlahDNS, BraveDNS, CloudFlare
DoH-Gen-F-CCDDD	10.5281/zenodo.5957420	F	Comcast, CZNIC, DNSForge, DNSSB, DOHli
DoH-Gen-F-FGHOQS	10.5281/zenodo.5957121	F	FFMuc, Google, Hostux, OpenDNS, Quad9, Switch
DoH-Gen-C-AABBCC	10.5281/zenodo.5957465	C	AdGuard, AhaDNS, BlahDNS, BraveDNS, Comcast, CZNIC
DoH-Gen-C-DDD	10.5281/zenodo.5957676	C	DNSForge, DSNSB, DOHli
DoH-Gen-C-CFGHOQS	10.5281/zenodo.5957659	C	CloudFlare, FFMuc, Google, Hostux, OpenDNS, Quad9, Switch

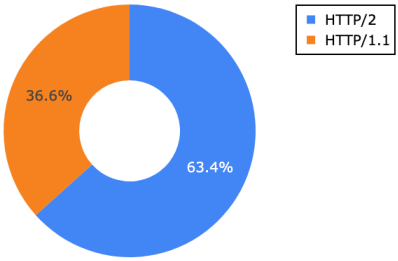


Fig. 1: Ratios of application layer protocols extracted from APLN extension in generated TLS traffic.

`generated/pcap/chrome/`. Each subdirectory name determines used DoH server. Tab. 4b depicts summary information about all files created with Chrome browser across all generated datasets in the collection. The difference in the number of connections can be caused by the generating process discussed later and browser-specific implementation. The ratio between used HTTP protocol version in generated datasets is shown in Fig. 1.

1.2. Real-world data

The *real-world* data are provided in the `DoH-Real-world` dataset (10.5281/zenodo.5956043) and contains HTTPS traffic, which consists of DoH communication and web-based HTTPS flows. The DoH captures contain only DNS over HTTPS communication, obtained by filter-

Table 4: Total stats of Firefox and Chrome generated data.

(a) Total stats of Firefox generated data.

Name	Value
Total Data Size	131 GB
Total files	32
DoH connections	~255 K
Non-DoH connections	~995 K

(b) Total stats of Chrome generated data.

Name	Value
Total Data Size	119.5 GB
Total files	32
DoH connections	~91 K
Non-DoH connections	~634 K

ing connections using the IP addresses of known DoH resolvers. The list of known DoH resolvers as appears in the dataset is included within the dataset, and also in the supplementary files, see Sec. 1.3.

The HTTPS traffic captures contain all communication transmitted over standard HTTPS port TCP/443, including DoH since DNS over HTTPS also runs on standard HTTPS port. Nevertheless, the provided list of identified DoH addresses can be used to remove DoH traffic from the dataset if necessary. The complete list of packet capture files is written in Tab. 5. The packet capturing was performed between June and October 2021 for several days. Packet capture files do not overlap.

Table 5: Real-world captures description.

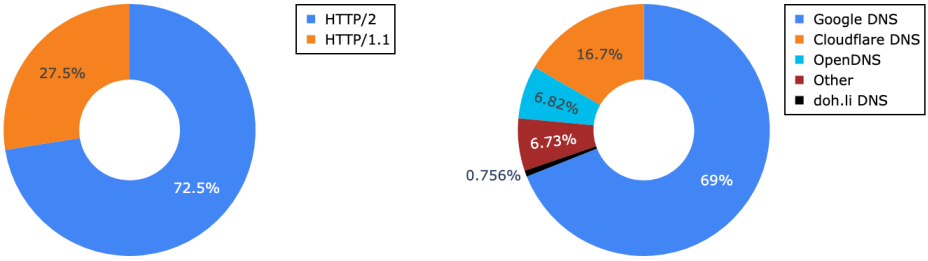
File Name (pcap)	Size	Date of capture	Dur.	Conn.	Packets
DoH-01082021-48h	6 GB	2021-08-01	48h	435 313	29 M
DoH-03082021-48h	5 GB	2021-08-03	48h	329 974	22 M
DoH-06102021-48h	19 GB	2021-10-06	48h	1 376 067	88 M
DoH-08102021-48h	8 GB	2021-10-08	48h	467 328	41 M
DoH-13072021-48h	5 GB	2021-07-13	48h	414 085	21 M
DoH-15072021-48h	4 GB	2021-07-15	48h	367 761	18 M
DoH-17072021-48h	3 GB	2021-07-17	48h	265 145	12 M
DoH-19072021-48h	5 GB	2021-07-19	48h	520 766	24 M
DoH-27072021-48h	6 GB	2021-07-27	48h	483 405	30 M
DoH-28062021-24h	3 GB	2021-06-28	24h	359 275	17 M
DoH-30072021-48h	3 GB	2021-07-30	48h	228 895	14 M
HTTPS-20102021-10h	20 GB	2021-10-20	10h	43 616	17 M
HTTPS-20102021-12h	20 GB	2021-10-20	12h	33 864	18 M
HTTPS-21102021-12h	20 GB	2021-10-21	12h	33 993	17 M
HTTPS-04102021-01h-1	20 GB	2021-10-04	0.1h	15 028	17 M
HTTPS-04102021-01h-2	20 GB	2021-10-04	0.1h	16 124	17 M
HTTPS-04102021-02h	20 GB	2021-10-04	0.2h	13 410	18 M

Contrary to *generated* traffic, the DoH communication was captured in the real network in this case. As can be seen in Fig. 2 most DoH connections involve Google DoH resolvers, and the five most popular resolvers represent more than 93 % of all DoH traffic.

The main value of the *real-world* dataset is in its real nature because it contains the traffic created by thousands of genuine users. The Tab. 6 shows overall statistics of captured dataset. It should be noticed that there is 116,263 unique client addresses in the network. Since the capture was created on Internet Service Provider (ISP) backbone, each IP address may correspond to multiple devices due to Network Address Translation (NAT). The actual number of clients can thus be even an order of magnitude larger.

1.3. Supplementary Files

In addition to the packet captures that stand for the main part of the dataset, we also provide Supplementary files (10.5281/zenodo.6024913) with software scripts for data generation and post-processing. The list and short description of supplementary files, are provided in Tab. 7.



(a) Application protocols' share (extracted from APLN extension)

(b) DoH resolvers' share

Fig. 2: The share of DoH resolvers and application protocols presented in the real-word dataset.

Table 6: Total stats of captured data.

Name	Value
Total Data Size	179 GB
Total Time	~10 Days
Connections	~420 M
Number of unique Client IP addresses	116,263
Number of unique Server IP addresses	9343
Number of unique DoH Resolver's IP addresses	142

2. Experimental Design, Materials and Methods

In this section, we describe the data acquisition environments. At first, we describe the environment used for traffic generation (*generated data* part), then we provide information about *Real-World Data* acquisition procedures, anonymization and its additional postprocessing.

2.1. Generated Data

Web browsers supporting DoH were used for making web requests that included DoH resolutions. We included Chrome and Firefox web browsers because of their most advanced implementation of the DoH feature. Since we primarily target DoH and Non-DoH traffic generation, we have to cover all possible settings that can affect the DoH traffic characteristics (such as packet sizes) generated by those applications.

Chrome browser is based on Chromium code-base that shares DoH resolution implementation with other Chromium-based browsers such as Edge[5], Brave[6].

Firefox browser provides even possibility of choosing DoH HTTP method, either GET or POST. Firefox provides more management options than Chrome, such as whether to force DoH usage or not and which server to use (default is Cloudflare's DoH service).

[5] https://www.microsoft.com/en-us/edge
[6] https://brave.com/

Table 7: The list and description of supplementary files.

Name	Description
`doh-resolvers`	The list of IP addresses of known DoH resolver, which was used during the *real-world* part creation. The list is in the CSV file format.
`pcap-anonymizer`	Python scripts used for the anonymization of real captures.
`firefox|chrome/scripts/`	Folder containing scripts and other files used for data generation.
`firefox|chrome/domains/<num>.csv`	Files containing the domains used for data generation.
`doh_server_urls.txt`	File containing DoH server urls used in generation process.

The *generated data* samples were created by visiting a collection of websites that are part of the Majestic million list [13]. The URLs were taken from the beginning of the list and each website was visited by one of the browsers.

DoH traffic characteristics are not only governed by client implementation but rather client-server interaction. DoH servers can employ different implementations and be deployed with different configurations. Moreover, the HTTPS ecosystem on the server-side can consist of proxies, firewalls, load-balancers, and other systems that impact network communication. To cover various DoH traffic patterns, we included 16 different DoH servers in our dataset. The servers were chosen from our DoH monitoring tool[7], supporting different features and each server deployed by a different provider.

The CSV files with flow data enriched for TLS information were then created with open-source flow exporter ipfixprobe[8]. We used the ipfixprobe TLS plugin to create flow extended for TLS information.

2.1.1. Firefox Browser Traffic Generation

At first, the Firefox browser in version 76.0.1 was used in the Docker container. The use of the Docker container does not influence the traffic with specific properties; however, it simplified the generation process automation and provided an isolated application environment. The platform utilized in the data generation process using Firefox browser was Supermicro SuperTwin2 6026TT-TF server equipped with eight Intel (R) Xeon E5520 @ 2.26 GHz. The cluster consists of 4 nodes. The nodes were equipped with 48 GB RAM and 16 CPU cores. The platform was directly connected to the university network. Network traffic was captured directly in the container using the tcpdump command-line tool.

Python and Selenium libraries were used for the automation of the process. To be able to provide more realistic behavior of the browser, we used X-virtual frame buffer, which implements the X11 display server protocol where all graphical operations are made in virtual memory without graphical output [14]. The browser than works with a standard graphical output even within the container. In addition, we restricted the container to use 4 CPUs and 4 GB of RAM at maximum.

[7] https://doh.sigman.io/
[8] https://github.com/CESNET/ipfixprobe.git

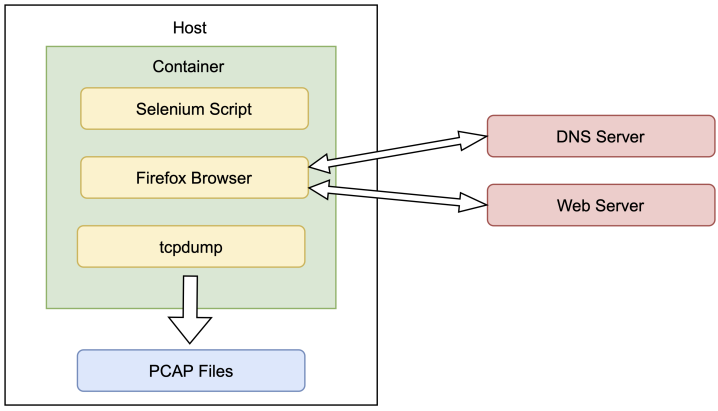


Fig. 3: Firefox browser traffic generation schema.

The Firefox had disabled cache during data generation that helped us obtain more network traffic by always fetching complete website content. The process consists of opening the browser, fetching the website, timeout, and waiting for loading the website, followed by browser closing. The schema of this generation is depicted in Fig. 3. This process was repeated 1000 times for DoH POST and GET method and for each 16 DoH servers. Together 32,000 different websites were fetched.

2.1.2. Chrome Browser Traffic Generation

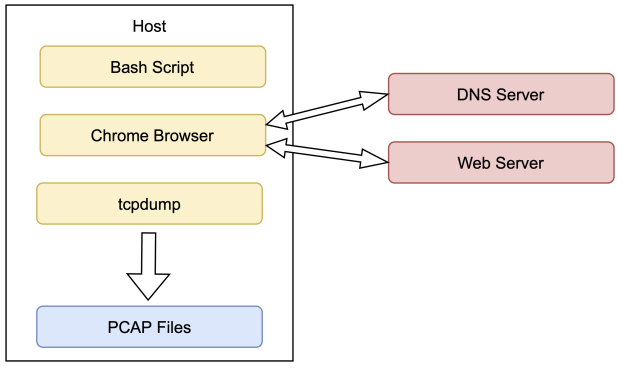


Fig. 4: Chrome browser traffic generation schema.

Since used version of Chrome browser (94.0.4606.81) does not support DoH on Linux OS, we could not reuse the same generation environment as in the case of Firefox. The Chrome browser generation was performed on a separate machine running Windows OS equipped with 3rd generation Intel Core i5, 8 GB RAM, 128 GB SSD also connected directly to the university network.

The traffic generation process itself was similar to Firefox. It consists of opening the browser with the given website, timeout, and waiting for loading the website, followed by browser closing. This generation is shown in supportive schema in Fig. 4. The automation was handled by a

python script with a manual setting of custom DoH servers. The same 16 DoH servers as in the case of Firefox were used. Chrome does not allow users to set HTTP GET or POST methods for querying DoH. Hence, the amount of fetched websites is similar to the Firefox case: 2×1000 only with the default Chrome method used. In total, 32,000 different websites were fetched.

2.2. Real-World Data

The *Real-World data* of the dataset was captured on the monitoring points of CESNET organization, which is the Czech National Research and Education Network operator. CESNET operates the backbone network infrastructure called CESNET2, which is used by half-million users. CESNET2 provides internet connectivity to universities, campuses, research centers, schools, hospitals, and selected governments. The topology of the CESNET2 backbone network is depicted at Fig. 5.

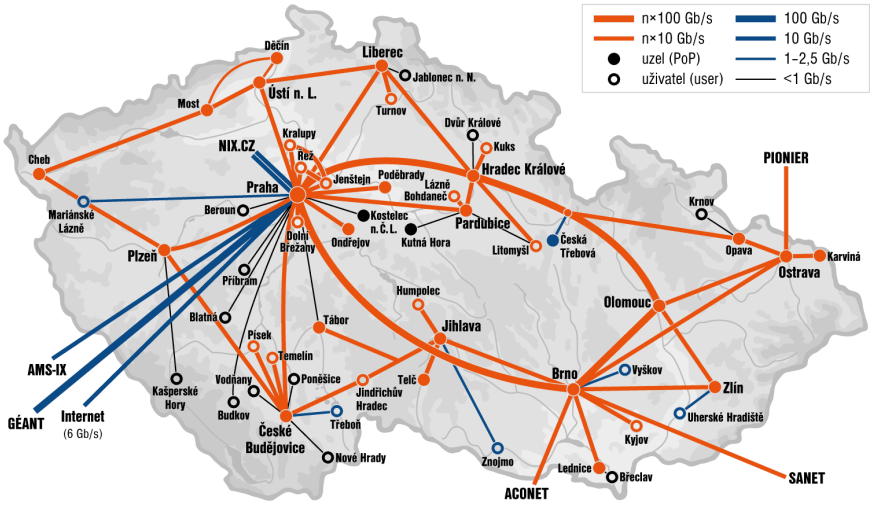


Fig. 5: The topology of CESNET2 network.

The six monitoring points are located at the perimeter of the CESNET2 network in three locations in the Czech Republic — Prague, Brno, and Ostrava. Each of them is connected to one or multiple 100 Gbps lines via passive optical tap. Since the packet capture is distributed across multiple points and locations, we need to deal with time-shifts on each monitoring server resulting in inaccuracies in packets timestamps. The points are synchronized via Network Timing protocol, with an accuracy smaller than 10 ms to minimize the time inaccuracies.

The time shift between monitoring probes does not affect connections that are routed symmetrically, meaning that all packets belonging to the one TCP connection pass through the same monitoring point. According to long-term measurement in the CESNET2 network, more than 60 % of all 443/TCP connections are routed symmetrically. The asymmetrically routed connection might have inaccurate interarrival time values between packets in the opposite direction — packets with the same direction are always routed via the same monitoring points. However, the inaccuracies are small, and they do not cause any significant disturbances in packet order within the connection (HTTPS requests are always before HTTPS responses). Moreover, the inaccuracy in order of 10 ms can be neglected due to other natural factors in the computer networks that

cause more considerable inter-arrival time variance (network jitter), such as network congestion or poor Wireless connection.

2.2.1. Capturing Process

Since the CESNET2 network is a large infrastructure with multiple monitoring points, we have created custom scripts for concurrent full-packet capture on each point using parallel SSH[9]. Fig. 6 shows a scheme of the overall capturing process. As a first step, the capturing controller distributes the command to the monitoring points. The command contains a filter and maximal duration of capturing. When the packet capture exceeds the maximum duration, the pcap file is distributed back to the controller.

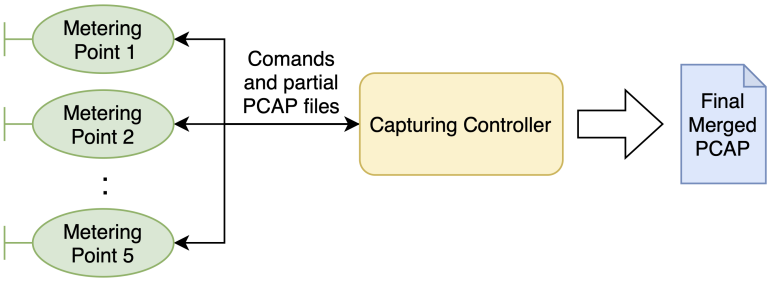


Fig. 6: The schema of capture distribution.

The monitoring points can process 100 Gbps traffic; however, it is not possible to capture all ongoing traffic due to bandwidth limitation and writing speeds of the disk arrays, which would result in missing packets in the dataset. Therefore it was necessary to create a capturing filter with this limitation in mind. Since the use of DoH is still small, we could capture all DoH traffic by filtering based on the DoH resolver's list (which is included with the dataset as supplementary file — described in Sec. 1.3).

In the case of HTTPS traffic, we have limited the capturing to the selected /24, and /22 IPv4 address ranges assigned to the university campus. Additionally, we have monitored the whole capturing process for packet drops to ensure that all transmitted packets in filtered connections were saved on the disks and the points were not overloaded; thus, no artificial packet drops were added.

2.2.2. Captured data Postprocessing

After capturing and partial packet merging, the data were automatically processed by performing packet deduplication, anonymization, and flow export.

Packet Deduplication. Since a backbone ISP network can also carry transit traffic, there is a chance of capturing duplicate packets on distributed monitoring infrastructure. The deduplication was performed with editcap[10] utility with a time window of 10 ms. Thus if there are two, or more duplicate packets within 10 ms time window, only the first packet will be maintained.

[9] https://linux.die.net/man/1/pssh
[10] https://www.wireshark.org/docs/man-pages/editcap.html

Anonymization and Flow Export. Since we are using data from real ISP backbone lines, we needed to remove sensitive information. The anonymization of captured traffic was performed automatically; thus, nobody could view or analyze raw source data.

At first, we anonymized all addresses (IPv4, IPv6, and MAC), except the IP addresses of DoH resolvers, which remained unmodified. The other addresses were substituted by part of the SHA256 hash (first N bytes, where N is the length of the corresponding address). The hashing algorithm ensures that a particular address is always mapped onto the same anonymized value.

PCAP files with anonymized IP addresses were then used to create TLS enriched flows with ipfixprobe flow exporter. The flows were created the same way as in the generated part.

After the flow export, we also anonymized the payload from all packets by substituting every byte with the letter 'X'. Removing packets' payload has only a negligible impact on the value of the data (since they are mostly encrypted anyway); however, it ensures the future privacy of the real users.

Ethics Statement

Privacy of users is our essential priority, so our whole research was done with extreme carefulness. The indisputable advantages of real traffic generated by hundreds of thousands of people come with the cost of potential privacy abuse of real users. Therefore, we used only automatic data processing with immediate data anonymization. With this, we declare that we did not analyze or manually process deanonymized data, and we did not perform any procedures that could lead us to the user's identity. Moreover, there is no possibility of revealing real users' identities from the provided dataset.

CRediT Author Statement

Kamil Jeřábek: Writing-Original draft preparation, Conceptualization, Methodology, Software, Data curation. **Karel Hynek:** Writing-Original draft preparation, Conceptualization, Methodology, Software, Data curation. **Tomáš Čejka:** Writing-Reviewing and Editing. **Ondřej Ryšavý:** Writing-Reviewing and Editing.

Acknowledgments

This research was funded by the Ministry of Interior of the Czech Republic, grant No. VJ02010024: Flow-Based Encrypted Traffic Analysis 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 also by Brno University of Technology, Faculty of Information Technology internal grant FIT-S-20-6293, and also by Technology Agency of the Czech Republic, grant No. FW03010099: Context-based Encrypted Traffic Analysis Using Flow Data.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships which have, or could be perceived to have, influenced the work reported in this article.

References

[1] Y. Banadaki, Detecting malicious dns over https traffic in domain name system using machine learning classifiers, Journal of Computer Sciences and Applications 8 (2020) 46–55.

[2] Y. Li, A. Dandoush, J. Liu, Evaluation and optimization of learning-based dns over https traffic classification, in: 2021 International Symposium on Networks, Computers and Communications (ISNCC), 2021, pp. 1–6. doi:10.1109/ISNCC52172.2021.9615659.

[3] T. Böttger, F. Cuadrado, G. Antichi, E. L. a. Fernandes, G. Tyson, I. Castro, S. Uhlig, An empirical study of the cost of dns-over-https, in: Proceedings of the Internet Measurement Conference, IMC '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 15–21. URL: https://doi.org/10.1145/3355369.3355575. doi:10.1145/3355369.3355575.

[4] A. Hounsel, K. Borgolte, P. Schmitt, J. Holland, N. Feamster, Comparing the effects of dns, dot, and doh on web performance, in: Proceedings of The Web Conference 2020, WWW '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 562–572. URL: https://doi.org/10.1145/3366423.3380139. doi:10.1145/3366423.3380139.

[5] W. Pan, G. Cheng, Y. Tang, Wenc: Https encrypted traffic classification using weighted ensemble learning and markov chain, in: 2017 IEEE Trustcom/BigDataSE/ICESS, IEEE, 2017, pp. 50–57.

[6] P. Velan, M. Čermák, P. Čeleda, M. Drašar, A survey of methods for encrypted traffic classification and analysis, International Journal of Network Management 25 (2015) 355–374.

[7] M. Piskozub, F. De Gaspari, F. Barr-Smith, L. Mancini, I. Martinovic, Malphase: Fine-grained malware detection using network flow data, in: Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security, 2021, pp. 774–786.

[8] M. Gohari, S. Hashemi, L. Abdi, Android malware detection and classification based on network traffic using deep learning, in: 2021 7th International Conference on Web Research (ICWR), IEEE, 2021, pp. 71–77.

[9] G. Harris, M. Richardson, PCAP Capture File Format, Internet-Draft draft-gharris-opsawg-pcap-02, Internet Engineering Task Force, 2021. URL: https://datatracker.ietf.org/doc/html/draft-gharris-opsawg-pcap-02, work in Progress.

[10] I. Akbari, M. A. Salahuddin, L. Ven, N. Limam, R. Boutaba, B. Mathieu, S. Moteau, S. Tuffin, A look behind the curtain: Traffic classification in an increasingly encrypted web, Proc. ACM Meas. Anal. Comput. Syst. 5 (2021).

[11] P. Matoušek, I. Burgetová, O. Ryšavý, V. Malombe, On reliability of JA3 hashes for fingerprinting mobile applications, in: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Springer International Publishing, 2021, pp. 1–22. URL: https://doi.org/10.1007/978-3-030-68734-2_1. doi:10.1007/978-3-030-68734-2_1.

[12] VMware, Inc, Tcp segmentation offload, 2020. URL: https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.networking.doc/GUID-E105A601-9331-496C-A213-F76EA3863E31.html.

[13] Majestic, The majestic million, 2021. URL: https://majestic.com/reports/majestic-million, (Accessed July 2021).

[14] I. David P. Wiggins, The Open Group, Xvfb, 2010. URL: https://www.x.org/releases/X11R7.6/doc/man/man1/Xvfb.1.xhtml, (Accessed September 2021).