FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

The Impact of Encrypted DNS on Network Security

by

Karel Hynek

A dissertation thesis submitted to
the Faculty of Information Technology, Czech Technical University in Prague,
in partial fulfilment of the requirements for the degree of Doctor.

Doctoral study programme: Informatics
Department of Digital Design

Prague, May 2023

Supervisor:
prof. Ing. Hana Kubátová, CSc.
Department of Digital Design
Faculty of Information Technology
Czech Technical University in Prague
Thákurova 9
160 00 Prague 6
Czech Republic

Co-Supervisor:
Ing. Tomáš Čejka, Ph.D.
Department of Digital Design
Faculty of Information Technology
Czech Technical University in Prague
Thákurova 9
160 00 Prague 6
Czech Republic

Abstract

The encrypted DNS is a natural response of the engineering community to the privacy concerns raised due to domain name misuse by internet service providers, threat actors, or government surveillance programs such as MORECOWBELL or QUANTUMDNS. However, visible domain names are essential for network intrusion detection, and their encryption has a great impact on security. This dissertation thesis maps the impacts of encrypted DNS on security, its state of deployment, and the possibilities of encrypted traffic analysis to overcome the reduced visibility. We partnered our research with the CESNET association, which is a Czech national education and research network operator, which kindly provided us with the anonymized flow-based telemetry that we used in our studies. In particular, the main contributions of the dissertation thesis are:

1. Evaluation of encrypted DNS adoption.
2. Description of encrypted DNS misuse by threat actors.
3. Design and evaluation of flow-based encrypted DNS detectors.

Keywords:
DNS, DNS over HTTPS, DNS over TLS, Security, Computer Networks

Abstrakt

Šifrované DNS vzniklo jako odpověď na zvýšené obavy uživatelů ohledně soukromí a zpracování dat. Masové sledování doménových jmen bylo v minulosti prokázáno u několika amerických poskytovatelů internetu, ale i u tajných složeb v rámci rámci rozsáhlých špionážních programů jako byly MORECOWBELL a QUANTUMDNS. Šifrování doménových jmen značně snižuje možnost jejich odposlechu a případného zneužití k profilování uživatelů, na druhou stranu ale zabraňuje i jejich legitimnímu použití. Analýza doménových jmen je naprosto zásadní při detekci síťové aktivity malwaru. Využití šifrovaného DNS má tedy i negativní dopad na síťovou bezpečnost. Tato dizertační práce se zabývá mapováním dopadů šifrovaného DNS na bezpečnost uživatelů a vývojem metod analýzy šifrovaného provozu, které mohou zmírnit dopady snížené viditelnosti do provozu. Práce obsahuje následující hlavní přínosy:

1. Vyhodnocování adopce šifrovaného DNS.
2. Popis zneužívání šifrovaného DNS ke škodlivým účelům.
3. Vytvoření a vyhodnocení nového detektoru šifrovaného DNS.

Klíčová slova:
DNS, DNS over HTTPS, DNS over TLS, Bezpečnost, Počítačové sítě

Acknowledgements

First of all, I would like to express my deepest gratitude to my dissertation thesis supervisors Ing. Tomáš Čejka, Ph.D., and prof. Ing. Hana Kubátová, Csc. They have always been helpful and helped me with numerous problems.

I want to acknowledge and express my deepest gratitude to the members of the Aposemat Laboratory at CTU, FEL. Mainly Sebastian Garcia and Joaquín Bogado, who completely changed my writing style, and I really like the cooperation.

I also want to thank my Bachelor's and Master's students that helped me implement some proof of concept. Special thanks go to Jan Luxemburk, Dmitrii Vekshin, Zdena Tropková, Lukáš Melcher, Daniel Uhříček, and Richard Plný. It was an honor to be their supervisor and advisor during their studies.

Special thanks also go to the staff at CESNET, mainly to Ing. Martin Žádník, Ph.D. and doc. Ing. Jan Kořenek, Ph.D., for their support and inspiration. I must also mention Armin Wasicek from Avast, who provided us with some necessary research data.

I would like to thank CESNET, CTU FIT, the European Union, the Ministry of Interior and Technology Agency of the Czech Republic for material and financial support in the following grants:

- SAPPAN, Project No. 833418 funded by European Union
- SPARTA, Project No. 830892 funded by European Union
- FETA, Project No. VJ02010024 funded by Ministry of Interior of Czech Republic
- SmartADS, Project No. TH04010073 funded by Technology Agency of Czech Republic
- PRIVILEGE, Project No. 886854-2 funded by European Union
- Student Grant Project No. SGS23/207/OHK3/3T/18 funded by Czech Technical University in Prague

Finally, my greatest thanks go to my family and my life partner Kačka for their support and patience during the bad days.

Dedication

To my supportive parents, Kačka and Morfeus
who forced me to rest and maintain my mind fit as a fiddle.
Without them I would have already finished my thesis two years ago.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

Network Protocols

DHT	Distributed Hash Table
DNS	Domain Name System
DoH	DNS over HTTPS
DoQ	DNS over QUIC
DoT	DNS over TLS
EDNS	Extended DNS
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
POP3	Post Office Protocol 3
QUIC	Quick UDP Internet Connections
SMTP	Simple Mail Transfer Protocol
SSH	Secure Shell
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
VPN	Virtual Private Network

Machine Learning and Statistics

AUC	Area Under Curve
FN	False Negatives
FP	False Positives
KNN	K-Nearest Neighbors
MI	Mutal Information Value
ML	Machine Learning
RF	Random Forest
SMOTE	Synthetic Minority Oversampling Technique
STD	Standard Deviation
TN	True Negatives
TP	True Positives

Miscellaneous Abbreviations

API	Application Program Interface
ASN	Autonomous System Number
C2	Command and Control
CDN	Content Delivery Network
CIDR	Classless Inter-Domain Routing
DPI	Deep Packet Inspection
eTLD	Effective Top-Level Domain
FEL	Faculty of Electrical Engineering
HTML	HyperText Markup Language
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
IM	Instant Messaging
ISP	Internet Service Provider
JSON	JavaScript Object Notation
LAN	Local Area Network
OS	Operating System
RFC	Request For Comments
RIPE	Réseaux IP Européens Network Coordination Centre
RTT	RoundTrip Time
SLD	Second-Level Domain
SNI	Server Name Indication
SW	Software
TLD	Top-Level Domain
URL	Uniform Resource Locator
USA	United States of America
USD	United States Dollar

Chapter 1

Introduction

The translation of human-readable domain names into machine-usable IP addresses and vice versa is an essential feature that enables a user-friendly usage of network services. Traditionally, this mechanism is performed by Domain Name System (DNS) [90, 89] in the Internet environment. DNS is one of the oldest network protocols, and, therefore, it is based on transferring unencrypted queries and answers through network links. Nevertheless, the plain text domain names leave a wide space for possible misuse. For example, studies like [56, 74] showed the possibility of user tracking over DNS and even bypassing the private mode in browsers. The fact that DNS-based tracking is possible in scale was shown by enormous DNS-based surveillance scandals such as QUANTUMDNS and MORECOWBELL, operated by governmental agencies [51]. The massive size of these scandals triggered concerns over users' privacy among the broad public. The natural response to these concerns was the privacy-preserving encrypted DNS protocol.

1.1 Motivation

Personal privacy has become one of the crucial features of modern applications in recent years. Thus, privacy-preserving technologies represent a rapidly evolving area, that is being fastly deployed to production and warmly welcomed by users. In recent years, DNS protocol finally got its privacy-preserving version—the encrypted DNS. Currently, there are three encrypted DNS versions standardized by IETF in the form of RFC: DNS over HTTPS (DoH) [59], DNS over TLS (DoT) [63], and DNS over QUIC (DoQ) [65]. Despite their relative novelty, DoH and DoT have already gained wide adoption in popular software. DoH is already supported by Windows OS [70], Apple (in MacOS and iOS) [28], and Mozilla Firefox [91]. Moreover, Chrome-based browsers support DoH by default (since version 83) [10]. DoT is currently enabled by default in Android OS (since version 9.0 Pie) [75].

The fast adoption of encrypted DNS is definitely beneficial for users' privacy. Nowadays, even users without any technical skills or knowledge about encrypted DNS can gain the benefits of increased privacy. Moreover, encrypted DNS is often enabled by default; users thus gain the benefits of increased privacy with all the comfort without any necessary actions.

Nevertheless, increased privacy always impacts security. Plaintext DNS inspection is one of the essential sources for network intrusion detection. The presence of malware in the network can be identified by generated (DGA) or blocklisted domain names. Visible domain names also play an essential role in parental control systems, policy enforcement, or lawful blocking. As a relatively novel set of protocols, it has not been appropriately studied to gather knowledge about its properties and effects on network security. The abuse possibilities and security threats arising from the lack of domain name visibility have been mostly unknown, highlighting the urgency and necessity of encrypted DNS research.

1.2 Problem Statement

The encrypted DNS represents a novel set of protocols with a potentially enormous impact on the security of users due to reduced visibility and essential information loss for security detectors. Nevertheless, no comprehensive study on important encrypted DNS properties. The researchers do not have open information about the encrypted DNS service providers and actual adoption of encrypted DNS service among users and service providers, which provides essential knowledge to understand the severity of encrypted DNS security implications.

The other challenge arising from the encrypted DNS is a possible novel set of attack vectors that must be explored. Moreover, we are unaware of the applicability of known plain DNS attack vectors data exfiltration, packet reflection, or command-and-control (C2) in the encrypted DNS domain. There needs to be a comprehensive evaluation of these abuses via encrypted DNS; especially in the face of a novel trend of secure DNS resolvers that claim security protection capabilities.

Lastly, we need to inspect the possibility of DoH detection due to its capability to blend into other HTTPS traffic, leaving the network operators and administrators unaware of its presence. DoH can thus be an easy-to-use tool that can be used even by non-experts to bypass DNS-based policy enforcement systems. Therefore, reliable DoH detection is essential for maintaining security without the need for traffic decryption with a man-in-the-middle proxy, which would have a much more negative impact on the user's privacy.

1.3 Goals of the Dissertation Thesis

1. Measurement of the encrypted DNS adoption across service providers.
2. Description and evaluation of encrypted DNS threats, particularly DoH, since it cannot be reliably detected in the network.
3. Evaluation of DoH privacy properties.
4. Research, design, and development of the detection of DoH.

1.4 Structure of the Dissertation Thesis

The thesis is organized into six chapters. The chapters are based on eight research papers authored by this thesis's author that were published at conferences (including CORE A conference) or in Q1 journals (including journals with Impact Factor=11.043). The contribution of the author of this thesis is then summarized in the following Section 1.5. The thesis chapters are as follows:

1. *Introduction*: Describes the motivation behind our efforts together with our goals.

2. *Background and State-of-the-Art*: Introduces to the necessary theoretical background and surveys the current state-of-the-art in encrypted DNS research.

3. *Encrypted DNS Adoption*: Provides results of our encrypted DNS adoption measurement.

4. *DNS over HTTPS Misuse by Threat Actors*: Provides a comprehensive summary of possible threats arising from DoH and their misuse in the wild.

5. *DoH Detection and Fingerprinting Using Side-Channel Analysis*: Describes our experiments with side-channel attacks, including DoH detection.

6. *Conclusions*: Summarizes the results of our research, suggests possible topics for further research, and concludes the thesis.

1.5 Author's Publications Used in the Dissertation Thesis and His Contribution

The text in the dissertation thesis is based on the following published journal or conference papers. This section summarizes the thesis author's contribution to each paper.

Publication [A.1]: Dmitrii Vekshin, Karel Hynek, Tomáš Čejka; DoH insight: Detecting DNS over HTTPS by machine learning

Dmitrii Vekhin (student of the thesis author) created a dataset and performed an initial preprocessing. **The author of this thesis designed the detection algorithm and performed all the experiments. Besides he also contributed to the writing.** Tomáš Čejka then helped with the writing of the initial draft and also did the internal review.

Publication [A.2]: Karel Hynek, Tomáš Čejka; Privacy Illusion: Beware of Unpadded DoH

The author of this thesis created an experimental dataset and performed all the experimental evaluations described in the publication; besides, he also contributed to the writing. Tomáš Čejka helped with the initial draft and also did the internal review of the publication.

Publication [A.3]: Karel Hynek, Dmitrii Vekshin, Jan Luxemburk, Tomas Cejka, and Armin Wasicek; Summary of DNS Over HTTPS Abuse

The author of this thesis created an extensive survey of related works and is the main co-author of the analysis of DoH use in the malware. Moreover, the author of this thesis described the inner workings of novel thread models occurring in web environments. Dmitrii Vekshin (a student of the author and an Avast employee) was responsible for DoH malware and DoH metadata gathering. Moreover, as a native Russian speaker, he translated multiple Russian documents necessary for the novel attack vector understanding. Jan Luxemburk helped with the revisions and contributed to writing the revised draft. Tomas Čejka and Armin Wasicek helped to write the initial draft and performed internal reviews.

Publication [A.4]: Sebastian García, Joaquín Bogado Garcia, Karel Hynek, Dmitrii Vekshin, Tomas Cejka, and Armin Wasicekk; Large Scale Analysis of DoH Deployment on the Internet

Sebastian García, Joaquín Bogado Garcia took care of the internet-wide scans and contributed to the writing. Moreover, they performed a trend analysis and statistical comparison of the year-apart lists. **The author of this thesis analyzed the data from the scans and also enriched them for other contextual information. Moreover, he also performed a deep analysis of the DoH providers, their certificates, and their DNS capabilities.** Dmitrii Vekshin provided data from the Avast internal threat analysis tool. Tomáš Čejka and Armin Wasicek then helped write the initial draft.

Publication [A.6]: Lukáš Melcher, Karel Hynek, and Tomáš Čejka; Tunneling through DNS over TLS providers

The design of all the experiments has been performed by the author of this thesis, who also performed the analysis of the gathered data; Moreover, he also wrote the original draft of the publication. Lukáš Melcher performed the data gathering during his Master's thesis, supervised by the author of this thesis. Tomáš Čejka performed the final review of the publication.

Publication [A.7]: Daniel Uhricek, Karel Hynek, Tomáš Čejka, and Dušan Kolář; BOTA: Explainable IoT malware detection in large networks

---

The original idea of the weak-indication principle was created by the author of this dissertation thesis, who also contributed to experimental data gathering and detection algorithm design. Moreover, he also contributed to the writing of the original draft. Daniel Uhříček created the experimental implementation and performance measurement during his Master's thesis, supervised by the author of this thesis. Moreover, Daniel Uhříček also contributed to the original draft writing. Tomáš Čejka and Dušan Kolář then performed the final review of the publication.

Publication [A.8]: Sebastián García, Karel Hynek, Dmitrii Vekshin, Tomáš Čejka, and Armin Wasicek; Large scale measurement on the adoption of encrypted DNS.

Sebastian Garcia took care of the data gathering from the University (Organization 2) and provided the analysis of the University data. Moreover, Sebastian contributed to the writing. **Author of this thesis provided statistical data from the large ISP (Organisation 1) and performed its statistical analysis. And he contributed to the writing.** Dmitrii Vekshin and Armin Wasicek provided data and its analysis from the Global Security Company (Organization 3). Tomáš Čejka contributed to writing the original draft and performed an internal review.

Chapter 2

Background and State-of-the-Art

In this chapter, we summarize the necessary background about encrypted DNS. The Section 2.1 provides detailed information about the three standardized encrypted DNS approaches (DoH, DoT, DoQ). Details of the encrypted DNS approaches have been studied and provided in papers [A.1, A.2, A.8, A.3]. Following Section 2.2 contains a summary of research works that studied the encrypted DNS from various aspects. This section is heavily based on data published in survey [A.3].

2.1 Background on the Encrypted DNS

For a long time, it has been known to the security community that domain name encryption is one of the essential features for users' privacy protection. DNS traffic can be used for user profiling and surveillance [52, 51], especially in countries without Internet freedom [22]. Users from oppressive countries would gain the most significant benefit from DNS encryption. Nevertheless, DNS encryption also has strong opposing voices due to data centralization and reduced visibility of security network monitoring.

Currently, the encrypted DNS is pushed by the tech giants such as Google or Cloudflare, who let their resolvers by default in web browser settings [A.4]. For instance, Firefox (since version 92.0) uses Claudflare Inc. by default, similar to Opera browser (from version 79.0.4143.50). In these cases, the browser bypasses OS-level DNS settings and sends the DNS requests to a third party (Cloudflare), which then has privileged access to valuable data for profiling and advertisement. These few large encrypted DNS providers are typically big tech giants and telecommunication providers (telcos) who effectively cut off smaller ISPs, small telcos, and even local administrators from accessing DNS. Moreover, bypassing OS-level DNS settings can drastically decrease user security when the OS-based DNS is connected to the security-protection system, such as DNS filters for policy enforcement in organizations or antivirus tools.

Despite the opposing voices, the deployment of encrypted DNS is successfully progressing. Currently, there are three standardized protocols—(i) DNS over HTTPS, (ii) DNS over TLS, and (iii) DNS over QUIC. We further describe them in the following sections.

2.1.1 DNS over HTTPS

The IETF adopted the DoH protocol as an RFC document (RFC 8484 [59]) in 2018. Nevertheless, there are two significantly different implementations. The RFC 8484 compliant approach uses classic DNS "Wireformat" [89] encapsulated in the HTTPS protocol. The messages are transferred either by HTTP GET or POST requests to `/dns-query` API endpoint. The other approach uses DNS messages encoded in the JSON format described in RFC 8427 [58]. However, with the JSON approach, there is no standardized API endpoint. Most well-known resolvers use the same as the RFC version [A.4]; however, for example, Google uses `/resolve` API endpoint.

The JSON data are then transferred via HTTPS GET. The primary motivation to encode the DNS query in a JSON is to increase the readability and easy data manipulation based on text-based messages. According to our observation, JSON is used primarily for a single query by applications where performance and short response time are not a priority.

2.1.1.1 DNS over HTTPS on the Packet Level

DoH follows the classic request-response scheme, with expected differences across HTTP protocol versions. Even though HTTP 1.1 is not officially recommended by RFC [59] due to performance reasons, according to our findings in [A.4], most resolvers and browsers support it. The biggest performance bottleneck of HTTP 1.1 is the missing support of multiple concurrent requests within a single connection; therefore, it always has to wait for the response before sending the following query. According to our observations (in Chrome version 94[1], and Firefox 91[2]), browsers reduce the performance penalty by creating multiple parallel connections (usually two). By switching between connections, they can perform concurrent requests. According to RFC 8484, each packet contains only one DNS query or response. Thus, network observers can reliably count the number of queries/responses transferred in the encrypted channel. Apart from that, no other information can be directly obtained from the network packets due to the TLS encryption.

From the packet-level perspective, DoH looks similar to any other HTTPS communication. It establishes a connection on port 443, performs a TLS handshake, and transfers encrypted data. This design decision prevents easy recognition of DoH in firewalls and creates a straightforward way to bypass DNS-based network protections. Detailed analysis of DoH traffic shape is provided in Section 5.1.2.

2.1.1.2 Oblivious DNS over HTTPS

As previously mentioned, the main argument against all encrypted DNS approaches is data centralization. Therefore, RFC 9230 [73] recently proposed the Oblivious DoH (ODoH) protocol, an upgrade of DoH to prevent central DNS resolver from surveillance by hiding

[1] `https://chromereleases.googleblog.com/2021/09/stable-channel-update-for-desktop_30.html`.

[2] `https://www.mozilla.org/en-US/firefox/91.0/releasenotes/`.

actual users' IP addresses. The protocol stands on the DoH principles, thus encoding DNS messages into HTTPS. However, it adds an intermediate proxy between the user and DNS resolver, which separates the content and IP address. The proxy cannot decrypt the underlying messages and only retransmits them to the DNS resolver. The DNS resolver then processes the query, but it does not know the IP address of the actual user. Even though ODoH significantly improves users' privacy, it also impacts performance due to the added latency by the intermediate proxy.

2.1.2 DNS over TLS

DNS over TLS is specified by RFC 7858 [63]. Its design is heavily based on the DNS over TCP described by RFC 7766[38]. However, instead of sending DNS wireformat messages over TCP, those are sent over a secure TLS over TCP connection. IANA reserved port 853/TCP, which all DoT clients and resolvers should use by default. Since the packets are sent over a dedicated port, a network administrator can easily recognize, block, or filter the traffic. However, it is worth noting that the RFC standard allows using DoT with ports other than 853/TCP; therefore, it is expected that DoT clients and resolvers will have the option to change the port in their configuration.

From the packet level, the DoT behaves similarly to DoH; however, the missing HTTPS headers result in smaller packets overall. Similarly, as in DoH, each packet carries only a single DNS request/response; the observer thus can reveal the number of exchanged DNS messages.

2.1.3 DNS over QUIC

The last IETF standardized encrypted DNS version is DoQ in RFC 9250 [65]. DoQ is very similar to DoT since it encapsulates DNS Wireformat messages (specified in RFC 1035 [89]) into a QUIC connection. Contrary to DoT and DoH, RFC 9250 considers the DoQ usage between the recursive resolver to the authoritative nameserver. Initially, it was proposed in the RFC draft [66] to use port 784/UDP. Nevertheless, the final standard specifies port 853/UDP as a default one, which all DoQ clients and resolvers should use. Therefore, also DoQ can be easily recognized in the network.

Since DoQ is still relatively novel, the software support is still nascent, and only a handful of resolvers already support it. Therefore, we are still determining its production-like packet behavior. Nevertheless, due to the RFC 9250 [65] requirement that the client selects a separate QUIC stream for each query, which is similar to the current DoH over HTTP/2 implementations), we might expect similar behavior as in the case of DoH and DoT. Thus a single packet would carry only a single DNS request/response.

2.2 Encrypted DNS Research

Despite the relative novelty of encrypted DNS protocols, they have already been studied by multiple researchers, including the author of this thesis. Even though the DoT is the oldest encrypted DNS approach, most of the research works are focused on DoH due to its stealthiness causing its significant impact on security. Research in DoQ is still nascent due to its novelty and limited support in the software. In this section, we will summarize related work based on the four perspectives: (i) Performance Perspective, (ii) Adoption Perspective, (iii) Privacy Perspective, and (iv) Security Perspective.

2.2.1 Performance Perspective

The latency of DNS protocol directly impacts the performance of networking applications [17]. Therefore, many researchers measured the performance consequences of encrypted DNS deployment. These studies are summarized in Table 2.1.

One of the first DoH latency measurements was published by McManus [86] from Mozilla in 2018, showing that the average additional latency caused by DoH is only six milliseconds. The following study created by Böttger et al. [15] focused on DoH overhead compared to traditional DNS. Their results show that DoH adds significant latency when the connection is used for a single query. However, the additional latency is negligible when the DoH connection is reused for multiple queries. Another study by Hounsel et al. [61] shows that DoH latency and reliability depend strongly on the selected resolver. This observation is also supported by Jerabek et al. [71], who studied DoH resolver behavior and the distribution of DoH packet sizes depending on used resolvers. According to their results, some DoH resolvers use long HTTP headers resulting in larger packets and thus creating a more considerable overhead.

A more extensive study was performed by Chhabra et al. [26], who studied DoH performance impact worldwide. Their results show that users from higher-income countries with higher-quality internet infrastructures are less likely to experience slower performance caused by DoH, resulting in a disproportionate impact on users from countries with lower economic capacity. Their findings are also supported by the studies performed by Hounsel et al. [60], Borgolte et al. [14] and Mbewe et al. [85], who also show that DoH has a negligible impact in good network condition. According to these studies [60, 14, 85], traditional DNS significantly outperforms DoH when dealing with congested or 3G mobile networks.

Lu et al. [80] conducted the DoT performance impacts measurements. Their experiments spanned ten countries worldwide. Similarly, as in DoH, they report negligible latency compared to traditional DNS when the connection is reused for multiple queries. The results are supported by Hounsel et al. [61] and Mbewe [85]. Other DoT performance measurement was done by Doan et al. [41], who measured only the latency of single-query requests. Unsurprisingly, the measured DoT overhead was significant, reaching up to 200 ms.

Lu et al. [80], Hounsel et al. [61], and Mbewe [85] also performed a comparison of DoT and DoH performance. In Lu et al. measurements, the latency of DoH and DoT queries were very similar—DoT performed slightly better with average/median performance overhead of 5 ms/9 ms (for DoT) and 8 ms/6 ms (for DoH). Nevertheless, Hounsel et al. and Mbewe et al. show that DoT significantly outperforms DoH and sometimes even traditional DNS when used over networks with higher latencies.

The only study which measures the performance of DoQ was published by Kosek et al. [76], who queried 264 resolvers via DoT, DoH, and DoQ. The results are consistent with previous measurements performed with DoH and DoT on high-quality connections. Nevertheless, the underlying QUIC protocol also supports fast 0-RTT handshake types. According to Kosek et al., the DoQ does not use fast handshakes to its full potential, and 40% of measurements show considerably higher handshake times than expected, leaving space for possible optimization in the future. Nevertheless, the authors claim that DoQ outperformed DoH and DoT; thus DoQ is the fastest encrypted DNS option.

Table 2.1: Comparison of DoH performance-related research. Measurement Setup — measurement data and its origin, Results — The main conclusions of the measurement about the DoH performance impact compared to traditional DNS.

Author	**Year**	**Protocol**	**Measurement Setup**	**Results**
McManus [86]	2018	DoH	Firefox users	Negligible impact, added latency of 6 ms.
Böttger et al. [15]	2019	DoH	Single client	Negligible impact on latency when reusing connection.
Borgolte et al. [14]	2019	DoH	Self-emulated network conditions	Selective impact, depending on network conditions.
Lu et al. [80]	2019	DoH & DoT	Generated across endpoints in 11 countries	Negligible impact, DoT and DoH perform similarly.
Hounsel et al. [60]	2020	DoH & DoT	Self-emulated network conditions	Selective impact, depending on network conditions.
Hounsel et al. [61]	2021	DoH & DoT	Generated via endpoints across North America	Selective impact, depending on used DoH resolver DoT outperforms DoH with bad network conditions.
Chhabra et al. [26]	2021	DoH	Worldwide measurement across 224 countries	Selective impact, depending on network conditions.
Mbewe et al. [85]	2021	DoH & DoT	Generated via endpoints across Africa	Selective impact, depending on network conditions.
Doan et al. [41]	2021	DoT	Generated from RIPE Atlas probes	Large impact, when used with single-query connections.
Jerabek et al. [71]	2022	DoH	Generated, single location	Selective impact depending on the used DoH resolver.
Kosek et al. [76]	2022	DoH & DoQ & DoT	Generated, single location	Negligible impact, DoQ is the fasted encrypted DNS.

2.2.2 Adoption Perspective

Encrypted DNS is gaining adoption across prominent software vendors, and popular operating systems, web browsers, and DNS resolution software already support it. Therefore we might expect the increasing usage of encrypted DNS among users and DNS service providers. Research works listed in Table 2.2 study the state of adoption of selected encrypted DNS approaches.

DoT adoption was studied by Lu et al. [80] in 2019. They performed an IPv4 internet-wide scan for opened 853/TCP port to identify the resolvers, and they found around 2250 DoT resolvers operating in the wild. Deccio et al. [36] performed such a scan across 1.2 million open DNS resolvers[3]. They found 1747 different DoT resolvers, making their adoption around 1.5‰ across open DNS resolvers. The following study performed by Doan et al. [41] replicated the methodology of Deccio et al. [36] nine months after and found 2151 dot resolvers—an increase of 23.1%. Doan et al. [41] also studied network flows from the WIDE backbone[4]. They report that DoT forms only 0.017% of all flows. In comparison, classical DNS accounts for 2.3% of all flows, meaning the share of DoT was negligible.

Compared to DoT, the DoH resolver is much more challenging to recognize since it does not use a dedicated port. In 2019, Lu et al. [80] used a large dataset of URLs from their commercial partner looked for RFC 8484 compliant URLs[5]. Their analysis found 17 DoH resolvers, which is compared to DoT negligible. Nevertheless, we need to consider their results lower-bound since they used only a limited dataset.

Deccio et al. [36] also studied the support of DoH in 2019 and used active DoH queries to open resolvers. From around 1.2 million open resolvers, only nine supported DoH. Following studies of DoH adoption were carried out in 2021 and 2022 by the author of this thesis [A.8, A.4]. We found two orders of magnitude more DoH resolvers and an increasing trend in DoH adoption across users and service providers. For more details, please refer to the Chapter 3.

The only study of DoQ adoption was published by the author of this thesis in [A.8]. According to our results, DoQ traffic is extremely rare. Nevertheless, this study was performed before the release of the DoQ standard. For more details, please refer to the Chapter 3.

2.2.3 Privacy Perspective

Since the primary benefit of encrypted DNS is the increased privacy of end-users [63, 59], it has been thoroughly studied by many researchers. The privacy-focused studies are summarized in Table 2.3. Overall, there is a general scepticism [128, 112] about the sufficiency of DNS encryption for preserving users' privacy. Therefore, the DNS protocol privacy enhancement feature called EDNS padding [84] was introduced. Clients with EDNS support send requests (via encrypted DNS) padded with random content to equalize the sizes of all packets. The padding reduces the possibility of side-channel information leakage.

[3] IP addresses that also support traditional DNS.

[4] WIDE Project data repository:`https://www.wide.ad.jp/index_e.html`.

[5] `https://*/dns-query?*`.

Table 2.2: Comparison of encrypted DNS adoption-related research. Measurement Setup — measurement data and its origin, Results — The main conclusions about the DoH adoption measurement.

Author	**Year**	**Protocol**	**Measurement Setup**	**Results**
Deccio et al. [36]	2019	DoH & DoT	Scan across Open Resolvers	DoH adoption ≤ 1‰, 1.5‰ of DoT adoption.
Lu et al. [80]	2019	DoT & DoH	IPv4 scan for DoT URL dataset for DoH	17 DoH resolvers, 1500 DoT resolvers.
Doan et al. [41]	2021	DoT	Scan across Open Resolvers	DoT adoption of 1.7‰ DoT accounts for 0.017% of all connections.
Garcia et al. [A.8]	2021	DoH & DoT & DoQ	IPv4 address space Traffic from 3 organizations	931 DoH capable IP addresses, Volume of DoH traffic is increasing, DoH is relatively rare.
Garcia et al. [A.4]	2022	DoH	IPv4 address space	Two-fold increase of DoH resolvers in one year, Most DoH resolvers are privately operated.

Website fingerprinting is one of the possible attacks which leverage side-channel information. The fingerprinting attacks are built on the assumption that connection to each website generates unique sequence packets' sizes, which the adversary can leverage to infer the transferred and encrypted content [113]. Traditional website fingerprinting is based on the observation of all victims' traffic across multiple connections. Encrypted DNS website fingerprinting is motivated by a significant reduction in the amount of data necessary to process by the adversary to infer visited websites.

Houser et al. [62] performed a website fingerprinting attack using DoT traffic only. They report a high accuracy (AUC over 0.99) for detecting victim visits to a website from a particular category—dating, insurance, gambling, and so on. More importantly, the accuracy remains high even with the deployment of padding (AUC over 0.97).

The susceptibility of DoH against website fingerprinting was studied by Bushart et al. [20] and Siby et al. [113]. Both approaches achieved an accuracy of over 86% in the detection of particular website visit using DoH traffic without padding. When the EDNS padding feature was enabled, they were still successful with more than 70% accuracy.

The author of this thesis also studied an attack similar to DoH website fingerprinting in [A.2]. However, his research aimed to infer the content of queries inside a single DoH packet. It was shown that HTTP 1.1 is much more susceptible to fingerprinting attacks, and we successfully inferred queried domain names with an accuracy of 90%. However, the method proved unusable when the EDNS padding was enabled. More details are provided in Chapter 5

The downgrade privacy attack on DoH was studied by Huang et al. [64]. They performed a downgrade attack by blocking the DoH connection, forcing the browsers to roll back to traditional unencrypted DNS without any noticeable alert in the user interface. According to the study [64], browser vendors do not consider this attack a vulnerability but rather a well-documented feature also described in RFC 8310 [39]. Since RFC 8310 is targetting mainly DoT, we might expect similar behavior also in the case of DoT. According to Huang et al. [64], the impact of a downgrade attack could be reduced by proper notification about lost privacy; however, none of the browser vendors plan to integrate it.

Table 2.3: Comparsion of DoH privacy-related research. The study scope abbreviation stands for: C — Correlation of encrypted and unencrypted DNS on recursive resolver, FP — Fingerprinting attack, DG — Downgrade attack.

Author	Year	Protocol	Scope	Outcomes
Shulman et al. [112]	2014	DNS	C	Execution of correlation attack for domain inference.
Siby et al. [113]	2019	DoH	FP	ML model for website fingerprinting. 0.908 F1 score when no defence mechanism used.
Houser et al. [62]	2019	DoT	FP	ML model for queried domain name and category inference 0.99 AUC when no defence mechanism used for category inference.
Bushart et al. [20]	2020	DoH	FP	ML model for website recognition. 86.1% accuracy when no defence mechanism used.
Hynek et al. [A.2]	2020	DoH	FP	ML model for queried domain name inference. 90.14% accuracy when no defence mechanism used.
Huang et al. [64]	2020	DoH	DG	Execution of DoH downgrade attack.

2.2.4 Security Perspective

Studies [14, 44, 19, A.3] on the impact of encrypted DNS (particularly DoH) mass deployment conclude that DNS encryption is a security problem since many existing automated network security tools rely on plain-text DNS. Attackers can leverage the increased privacy of encrypted DNS to hide their malicious activities. Even though DoH provides confidentiality of resolution, it does not protect against subversion of DNS resolution (such as DNS cache poisoning) [9] and allows the creation of DNS tunnels [A.6].

The security-based DNS research is concentrated mainly to DoH—DoT, and DoQ is still nascent. We are aware of a single research work [A.6] considering DoT, which was published by the author of this thesis. We studied security measures deployed by DoT service providers, showing a lack of protection by the most used providers, including Google DNS. More details are provided in Chapter 4.

The rest of encrypted DNS-related security research considers only DoH. These DoH studies can be divided into two categories: (i) Detection of DoH presence in the network, and (ii) Detection of malicious DoH. All encrypted DNS studies are then summarized in Table 2.4.

2.2.4.1 Detection of DoH Presence in Network

DoH decreases visibility by automated network security tools [14]; therefore, detection of DoH presence can be considered viable for maintaining situational awareness of network operators and analysts. DoH traffic in the highly restricted network might indicate a policy violation attempt or the presence of some unwanted software. Since DoH does not use any dedicated port number, it blends into other encrypted HTTPS traffic, making its recognition difficult. Most DoH can be blocked by filtering 443/TCP connections to well-known DoH providers (such as Google or Cloudflare). However, it is always possible to choose less known DoH resolver that anyone can deploy — there is already available open-source software capable of DoH to DNS translation. As shown in our research [A.4], hundreds of "unknown" DoH resolvers do not appear on public DoH blocklists.

The author of this thesis is also the author of the first study [A.1] that proposed DoH detection by its traffic characteristics. We trained several machine learning models to distinguish DoH connections from other traffic, achieving high accuracy of 99% (0.99 F1 score). The essential traffic feature for detecting DoH is the duration of the connection, its burstiness, and the number of transferred packets. However, we worked only with browser-based DoH connections, leaving a single query DoH undetected. Detailed information about our approach and evaluation are provided in Chapter 5. Following studies [131, 82, 49, 11, 12] also achieved similar results, proving that browser-based DoH has distinctive properties that can be leveraged for detection. Csikor et al. [34] expressed concern about the DoH detection possibility, arguing that it can be misused for censorship by a downgrade attack. Therefore, they have evaluated multiple DoH padding techniques, which modified the DoH traffic characteristics, making them similar to regular HTTPS. One of the evaluated techniques successfully degraded the performance of machine learning detectors to the level where its deployment would be impractical.

2.2.4.2 Detection of Malicious DoH

Traditional DNS abuse detection is a well-studied topic, targeted by multiple research works [40, 132, 13, 43, 100, 23, 111, 98]. However, none of the mentioned work can be directly applied to DoH due to the added encryption.

The security-related research in the DoH area focuses mainly on data exfiltration. MontazeriShatoori et al. [82] analyzed the DoH tunneling approaches and the possibility of their detection. They created a dataset called CIRA-CIC-DoHBrw-2020[6] and proved the usability of time-related features to detect DoH tunnels and reported an accuracy of almost 100% (F1 of 0.999).

Following studies [11, 107, 134, 12, 6, 114] then used the DoHBrw-2020 to prove the possibility of malicious DoH detection with various machine learning approaches, all of them achieving very high accuracy above 99%. However, the CIRA-CIC-DoHBrw-2020 consists of only lab-created traffic from tunneling tools that use traditional unencrypted DNS, translated into DoH using a proxy. The dataset does not include traffic from already DoH-capable malware samples or exfiltration tools. These weaknesses were addressed by studies from Kwan et al. [77], and Zhan et al. [135]. Both studies focused on a more realistic scenario of DoH tunnel detection using a DoH-capable exfiltration tool. Kwan et al. focused on simple detection techniques using only a single feature, such as throughput and achieved 93% accuracy by observing only outgoing throughput. Zhan et al. [135] executed

[6] `https://www.unb.ca/cic/datasets/dohbrw-2020.html`.

DoH-based exfiltration between various locations worldwide. They tested multiple machine learning classifiers and achieved detection accuracy above 99%.

2.3 Key Findings

This chapter presented a survey on encrypted DNS research. The main findings are:

1. Encrypted DNS can be as fast as a regular DNS.
2. There are serious concerns about EDNS privacy. It has already been exploited for website fingerprinting.
3. The adoption of encrypted DNS is rising. The author of this thesis significantly contributed to this research area.
4. It is possible to detect DoH using ML with very high accuracy. The author of this thesis proposed the first ML-based DoH detector.
5. Detection of DNS tunnels is reliable and accurate despite the encryption.

The approaches and measurements performed by the author of this thesis will be further described in the following chapters.

Table 2.4: Comparison of research considering encrypted security. The abbreviations in *Scope* column stand for D — DoH detection, E — DoH exfiltration, S — Summary. The abbreviations in *Data* column stand for: C — Custom dataset, D — DoHBrw-2020 Dataset.

Author	**Year**	**Protocol**	**Scope**	**Method**	**Data**	**Outcomes**
Vekshin et al. [A.1]	2020	DoH	D	AdaBoost	C	DoH detector with accuracy of 99% (0.99 F1 score) DoH client (browser) identification with an accuracy of 99%.
MontazeriShatoori [82]	2020	DoH	D & E	Random Forest	D	DoH detector with 0.99 F1 score. DoH exfiltration detector with 0.99 F1 score.
Banadaki et al. [11]	2020	DoH	D & E	XGBoost	D	DoH detector with claimed 100% accuracy. DoH exfiltration detector with claimed 100% accuracy.
Singh et al. [114]	2020	DoH	E	Gradient Boosting	D	DoH exfiltration detector with claimed 100% accuracy.
Wu et al. [131]	2021	DoH	D	Autoencoder	C	DoH exfiltration detector with 98% accuracy.
Casanova et al. [49]	2021	DoH	D	Bi-LSTM	D	DoH exfiltration detector with 99% accuracy.
Csikor et al. [34]	2021	DoH	D	Random Forest	C	DoH exfiltration detector with 0.97 F1 score, when no defence mechanisms used. DoH detection defence techniques, which drops the detector performace to unusable level.
Kwan et al. [77]	2021	DoH	E	Simple statistical	C	They achieved 94% accuracy by observing outgoing throughput.
Ding et al. [107]	2021	DoH	E	Autoencoder	D	DoH exfiltration detector with 0.99 F1 score.
Behnke et al. [12]	2021	DoH	D & E	LightGBM	D	DoH detector with 99% accuracy DoH exfiltration detector with claimed 100% accuracy.
Alenzi et al. [6]	2021	DoH	E	XGBoost	D	DoH exfiltration detector with 99% accuracy.
Zebin et al. [134]	2022	DoH	D & E	Balanced Stacked Random Forest	D	Single detector capable of distinquishing DoH exfiltration, DoH and non-DoH traffic with accuracy of 99%.
Zhan et al. [135]	2022	DoH	E	Random Forest	C	DoH exfiltration detector with 0.99 F1 score.
Melcher et al. [A.6]	2022	DoT	E	—	—	Evaluation of DoT resolvers' protection against covert channels.

Chapter 3

Encrypted DNS Adoption

In this chapter, we provide detailed information about the results of our research related to encrypted DNS adoption. This chapter is divided into two sections. The Section 3.1 describes the methodology and results of our Internet-wide research scan among DoH providers active on the Internet, which was published in our papers [A.8, A.4]. The Section 3.2 then contains methodology and results of our encrypted DNS flow measurement published in [A.8]. We conducted this research in cooperation with anti-virus company Avast Software s.r.o. and with Stratosphere laboratory at FEL CTU.

3.1 DoH Adoption Among Service Providers

Previous works focused mainly on DoT adoption among the service providers (as described in Section 2.2.2). Contrary to DoT, DoH does not have an assigned port and shares port 443/TCP with other HTTP services. Thus, the scan for DoH capability is much more resource-intensive due to the necessity to perform several HTTPS requests.

Nevertheless, we performed two Internet-wide scans one year apart to study the amount of DoH-capable IP addresses and their evolution in time. For the second scan, we also studied the properties of each DoH-capable IP address to provide deeper insight into DoH support on the Internet.

3.1.1 Methodology

The longitudinal analysis consists of two Internet-wide scans. The first was performed in April 2021, and the second between January and April 2022. Each exploration consisted of the following methodology steps: (i) create a list of well-known DoH resolvers; (ii) scan all the hosts on the IPv4 Internet looking for servers with port 443/TCP open; (iii) discover which of those IPs are DoH resolvers; (iv) verify that they answer DoH correctly and compile a final list; (v) enrich the IP addresses of the discovered DoH resolvers with information from threat intelligence services; (vi) verify the use of SNI; (vi) estimate the

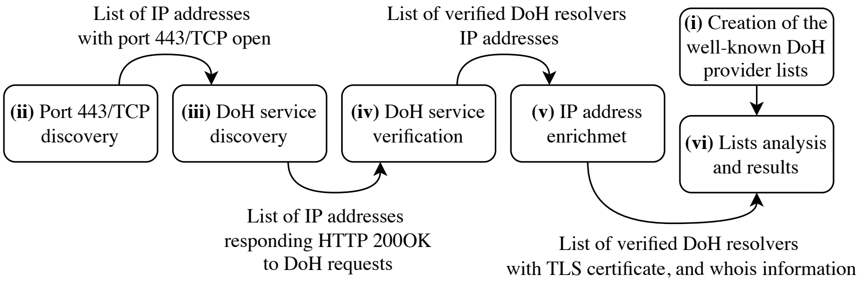

List of IP addresses
with port 443/TCP open
List of verified DoH resolvers
IP addresses
(i) Creation of the well-known DoH provider lists
(ii) Port 443/TCP discovery
(iii) DoH service discovery
(iv) DoH service verification
(v) IP address enrichmet
(vi) Lists analysis and results
List of IP addresses responding HTTP 200OK to DoH requests
List of verified DoH resolvers with TLS certificate, and whois information

Figure 3.1: Diagram of the methodology to scan the Internet to find DoH resolvers.

number of organizations providing DoH resolution services. The steps are graphically presented in Figure 3.1.

3.1.2 Creation of the Well-known DoH Resolvers Lists

Each list of well-known DoH resolvers was created by aggregating all the resolvers available in public lists, reports, documents, and academic papers. The DoH resolvers were verified using our custom Python script described in Section 3.1.2.3.

Some lists of DoH resolvers, such as the AdGuard list [2], and the curl list [35], are publicly available on the Internet. However, those lists are not comprehensive and miss some resolvers that appear on the different lists. Therefore, we created compound lists each year. The list created in 2021 is called Known2021, and the list created in 2022 is called Known2022. Both were published in public repositories [48, 69]. The original lists used for the Known2021 and Known2022 creation are included within each list repository. Moreover, IP addresses from Known2021 that worked in 2022 were also added to the Known2022.

3.1.2.1 Scan of Port 443/TCP on the Internet

We scanned the entire IPv4 address space on the Internet, looking for hosts with open port 443/TCP. It was done by dividing the IPv4 address space into 255 uniform A-class ranges to distribute the load among several scanning nodes. Each range was scanned from a different cloud virtual machine. The masscan tool performed the scan [50] with a fixed rate of 2000 packets per second. Masscan was also configured to retry each IP address three times.

The same masscan parameters were used in both 2021 and 2022.[1]. Moreover, in both scans, we used the masscan feature to scan the IP addresses in random order and limit the number of packets per second sent to service providers.

[1] Masscan command example: `masscan -p 443 --range 20.0.0.0-29.0.0.0 --rate 2000 --retries 3`.

3.1.2.2 DoH Service Discovery

Once the list of IP addresses with open port 443/TCP was collected, we needed to find DoH-capable ones. To automate the process, we created a DoH Nmap script [108]. Nmap is a well-known multifunctional network scanner that implements the Nmap Script Engine (NSE) for users to develop their scanning scripts [81]. Our DoH script checks all six different DoH methods: HTTP/1 with GET, HTTP/1 with POST, HTTP/1 with JSON, HTTP/2 with GET, HTTP/2 with POST, and HTTP/2 with JSON. This scan was executed using the same cloud setup as for port 443/TCP scan.

In order to speed up the process, the script only checks the HTTP status code in the response. It does not parse the whole HTTP response or make any more DNS resolutions. Therefore, false positives may occur, which we later filter in the DoH verification stage Section 3.1.2.3. This verification stage was implemented in a separate script to keep the Nmap script as simple and fast as possible.

The Nmap script sends six DoH requests[2] with a DNS query asking for the `example.com` domain. This domain is managed and recommended by IANA for testing purposes. For all six methods, the script sends the same query to the `/dns-query`[3] API endpoint. This endpoint is specified in RFC 8484 for the HTTP GET and HTTP POST DoH methods. Since the JSON method is not standardized by the RFC, the endpoint of DNS JSON API might differ between providers. However, many well-known providers, such as Cloudflare [32], AhaDNS [3], and Quad9 [101] use the same endpoint as defined in the RFC.

The Nmap parameters used for this stage in 2021 differ from the ones used in 2022. In 2021 we used Nmap with the most aggressive timing template (parameter -T5), allowing for a faster scan. However, this timing template is prone to packet loss, reducing the service discovery efficiency. In 2022, we used the normal Nmap timing template (parameter -T3) to obtain higher-quality results, minimizing packet loss.

Therefore, to make a fair comparison between the two scans, we estimated the number of resolvers lost in 2021. We re-scanned all the 2022 DoH resolvers using both timing parameters. Results show that the more aggressive parameters of 2021 indeed caused packet loss and resulted in a smaller number of detected DoH resolvers. From the 4354 DoH resolvers found with normal timing parameters, the nmap set with aggressive parameters found between 2851 and 3213 in repeated scans. the relative efficiency, then, of the *DoH Service discovery* in 2021 was between 65.5% and 73.8% compared to 2022.

3.1.2.3 DoH Resolver Verification

The list of DoH resolvers found in the previous stage was verified for correct DoH implementation to remove false positives. We implemented a Python script (available in [108]), which tests the correct support of three DoH methods (GET, POST, and JSON) via HTTP/1 and HTTP/2. Contrary to the Nmap DoH script, the Python script can parse

[2] HTTP/1-GET, HTTP/1-POST, HTTP/1-JSON, HTTP/2-GET, HTTP/2-POST, and HTTP/2-JSON.

[3] DNS query endpoint example: `https://1.1.1.1/dns-query?name=example.com`.

the DoH responses and check that they are valid DNS responses. This step filtered out IP addresses that responded “HTTP 200 OK” to DoH requests, but the response did not contain DNS data. The result of this stage is a list of confirmed and validated DoH resolvers and supported DoH methods. The same verification method was performed for 2021 and 2022.

At the end of this step, and from now on, the verified list of DoH resolvers of 2021 is called Scan2021, and the one from 2022 is called Scan2022.

3.1.2.4 IP Address Enrichment

The list of DoH resolvers was further enriched with related information about the discovered IP addresses. The enrichment consists of (i) the TLS certificates, (ii) information from the WHOIS service, (iii) information from VirusTotal threat intelligence feeds, including downloaded samples and URLs related to malware samples associated with the IP addresses, (iv) passive DNS data with the referred domain names for the IP, (v) DNS server type, (vi) DNS server version identification, and (vii) information about the web page if there was any. In addition, a *suspicious* flag was included in case the IP address has a high probability of being *relate to a phishing campaign* according to a set of indicators used by the Avast Web Shield feature. This set of indicators consists of keywords, domain name structure, lexical analysis results, domain hosting information, and other indicators.

The information for (v) DNS server type and (vi) DNS server version identification was obtained using the DoH inherited capabilities of traditional DNS. In DoH, as in DNS, it is possible to create a CHAOS record class with TXT requests and issue it into a `version.bind` query to identify which type of DNS software the server is using. Finally, the TLS certificate data of the DoH resolvers were analyzed to detect anomalies, such as expired or self-signed certificates. Since the IP address enrichment was implemented late in 2021, it was applied only to the DoH resolvers in 2022.

3.1.2.5 Verification of SNI Usage

The main limitation of our DoH scan is that it may not find DoH resolvers hosted on shared infrastructure—a single server hosting multiple services behind the same IP address. In such cases, to be successful, the query needs to send a Server Name Indication (SNI) and `HTTP Host` header, or `HTTP/2 :authority` header. In this case, our methodology would incorrectly mark active DoH resolvers as non-DoH since we do not know and thus do not provide domain names in the TLS handshakes.

To investigate the severity of this limitation, we estimated how many DoH resolvers were not found by performing a test with the Known2022 list of well-known DoH resolvers where domain names are present. The methodology was: (i) For each well-known DoH resolver in the Known2022 list with an IPv4 address, get its domain name; (ii) do all the six types of DoH queries providing the SNI, or HTTP/1, or HTTP/2 host headers; (iii) get the IPv4 address for that domain; (iv) do all the six types of DoH queries providing only the IPv4 address, without any SNI or HTTP header. Using these steps, we obtained the share of well-known DoH resolvers requiring a domain name for a successful connection.

3.1.2.6 Estimation of the Number of Organizations

To estimate the number of organizations that provide DoH resolution services in the Scan2021 and Scan2022 lists, we used the following methodology: (i) extract the reverse DNS of all the IPs in the Scan2021 and Scan2022 lists. (ii) extract the effective second-level domain name for each IP and consider each unique, effective second-level domain[4] of an organization. (iii) if the effective second-level domain was not available, extract the WHOIS organization name and consider each *WHOIS organization* an organization. (iv) if the *WHOIS organization* was not available, group the IP addresses by their /16 CIDR and Autonomous System Number (ASN), and consider each unique group as an organization as used by Deccio et al. [36].

3.1.3 Results

This section shows the findings that can be obtained from the Scan2021 and Scan2022 lists and a comparison of these results with the well-known lists of DoH resolvers. Then we deeply analyze the result from the Scan2022 and inspect the IP addresses from various aspects, including threat intelligence feeds.

3.1.3.1 Results of Creating Well-Known DoH Resolvers Lists

Regarding the creation of DoH resolver lists that are well-known by the community, Table 3.1 shows a summary of the main differences. The total number of well-known DoH resolvers between 2021 and 2022 increased by ∼12%. From the DoH resolvers found in 2021, only ∼67% remained active in 2022 (157 IP addresses from 234 IP addresses). In 2022 there was a ∼10% increase of IPv4 addresses, with ∼65% appearing in 2021 and 2022—∼35% of the IPv4 addresses of well-known DoH providers disappeared in 2022. Similarly, there was an increase of ∼13.4% of unique ASNs and a ∼14.5% increase in the number of unique IPv6 addresses in 2022.

The number of unique domain names slightly decreased due to the different methodologies of their collection. Contrary to the well-known DoH resolver list of 2021, the 2022 list does not contain domain names acquired by reverse DNS queries (PTR) as discussed in Section 3.1.2.

3.1.3.2 Results of DoH Scans

The port scan of 2021 found 41,022,969 IP addresses with port 443/TCP open on the Internet. Of these, 930 were verified to be actual DoH resolvers. Since this scan used a set of aggressive Nmap parameters, which reduced its efficiency, this number of DoH resolvers

[4] We used `https://publicsuffix.org/list/` for recognition of effective second-level domain name.

Table 3.1: Summary of well-known DoH resolvers in the Known2021 and Known2022 lists.

	Known2021	Known2022	Intersection	Increase
Total Unique Servers UP	234	262	157	11.9%
Total Unique IPv4 Servers	131	144	86	9.9%
Total Unique IPv6 Servers	103	118	78	14.5%
Unique Autonomous Systems	52	59	42	13.4%
Unique Domain Names	110	109	67	-0.1%

Table 3.2: Features of IP addresses of the Discovered DoH resolvers.

Feature	Scan2021	Scan2022
Total number of unique IP addresses	930 (100%)	4354 (100%)
IP addresses with domains	679 (73%)	4197 (96%)
IP addresses without domains	251 (27%)	149 (3%)
Unique SLD	171	657
Unique /16 prefixes*	115	39
Unique Autonomous System*	72	27
Estimated number of unique providers	243 – 286	684 – 696

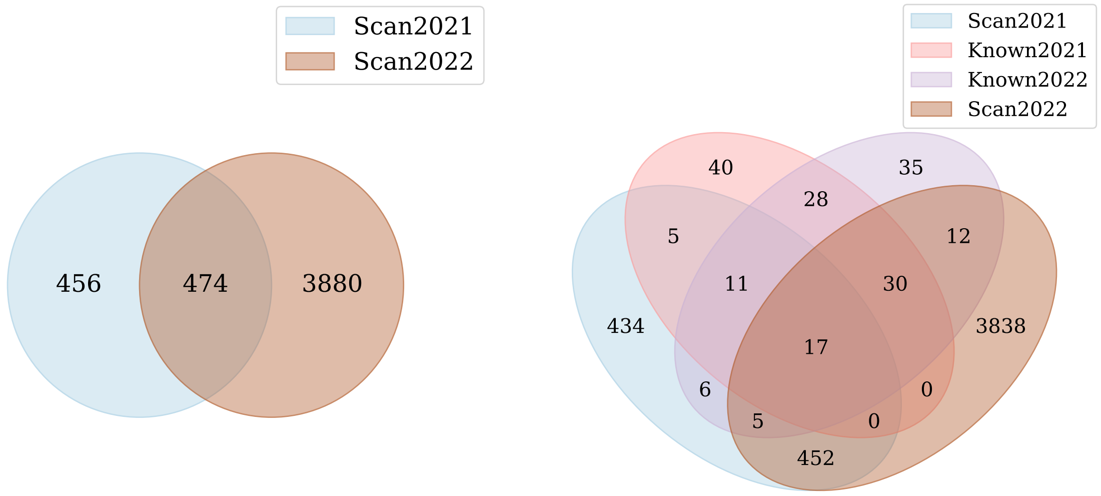
* Number calculated only from IP addresses for which we could not obtain a domain name.

could be underestimated. Our experiments show that the number of DoH resolvers in April 2021 was actually between 1173 and 1241.

In 2022, the port scan found a total of 36,035,492 IP addresses with port 443/TCP open, which represents 87.84% of the IPs found during 2021. We attribute the smaller amount of IPs to the large variability in Internet scans (packet loss, bandwidth differences, and geolocation filters) and not to an actual decrease in the number of endpoints with port 443/TCP open. The number of verified IP addresses of DoH resolvers found during 2022 and contained in the Scan2022 list is 4354. This number is ~4.8 times larger than the amount of DoH resolvers of Scan2021.

Table 3.2 summarises the total number of resolvers discovered in both scans. In Scan2022, we found four times more unique IP addresses of DoH resolvers than during Scan2021. Even when we consider the decreased efficiency of Scan2021 during the service discovery stage, the difference in the discovered DoH resolvers with Scan2022 is statistically significant with a p-value < 0.01. This result is based on a standard two-sample-one-side T-Test for the mean of a distribution [123]. Thus we observed an actual increase in the number of public DoH resolvers.

Moreover, the number of organizations providing DoH resolution services in April 2022 is 2.5 times larger than in April 2021. Figure 3.2a shows that 474 DoH resolvers were found in both scans. However, almost half of the verified DoH resolvers found in Scan2021 were not found in Scan2022. Since our methodology does not track the DoH resolvers individually, we do not know if these DoH resolvers have been moved to another IP address or ceased operations. On the other hand, the decrease in the number of unique /16 prefixes can be explained by a slight increase in the efficiency of the domain extraction process in 2022.



(a) Comparison between first and second scan. (b) Comparison with Well-known lists.

Figure 3.2: Venn diagrams of DoH resolver IP addresses distribution.

3.1.3.3 Comparison Between the Well-Known and DoH Scan Lists

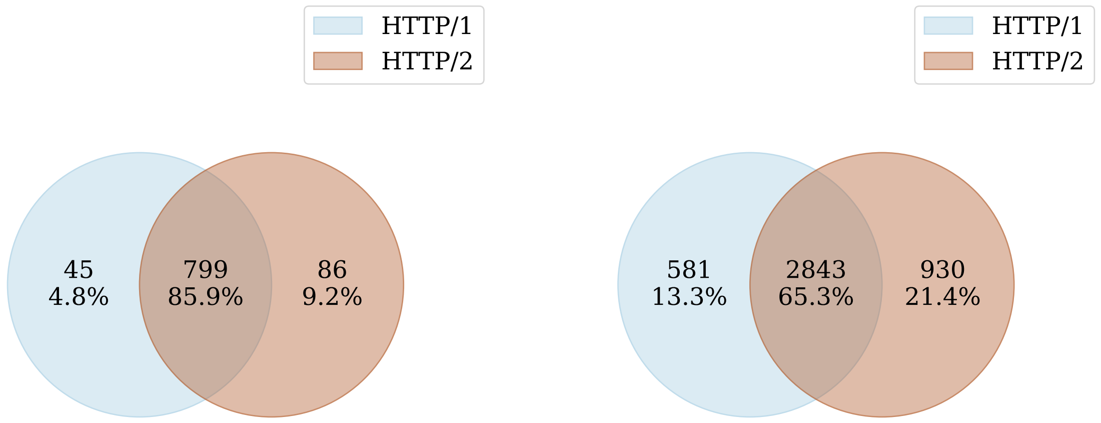
The distribution of the DoH resolver IP addresses across all lists is shown in Figure 3.2b. Reading the figure from top to bottom, we find that 40 addresses are only present in the Known2021 list, 35 only in the Known2022 list, and 28 in both. None of these addresses were found in the Scan2021 or Scan2022 lists. Meanwhile, five IP addresses present in the Known2021 list were only seen in that list, and 12 IPs present in the Known2022 list were only seen in that same list. From all IP addresses present in both well-known lists, 11 were found only during Scan2021, 30 only during Scan2022, and 17 were found during both scans. A total of 434 IP addresses were only found in Scan2021, 3838 were only found in Scan2022, and 474 were present in both scans. Nevertheless, 452 IP addresses were absent from any of the well-known lists. Most DoH resolver IP addresses did not appear on any of the well-known lists. However, the well-known lists are evolving. There are 11 servers that were in Scan2021, which were not present in the Known2021 list but are included in the Known2022 list. However, only five of these servers appear to be still active on the Scan2022 list. The rest may have been moved to another address or stopped operations.

3.1.3.4 Results of the SNI Verification

The results of the SNI verification are shown in Table 3.3. In the Known2022 list of DoH resolvers, there are only 93 that have a domain and an IPv4 address. It can be seen that around 30% of the well-known DoH resolvers require an SNI or HTTP header to work successfully. It means that our Scan2021 and Scan2022 of DoH resolvers are a lower bound, and theoretically, there could be *at least* 30% more DoH resolvers on the Internet. It should also be considered that this test was performed on well-known resolvers, in which

Table 3.3: Results of SNI verification during DoH resolvers finding.

Connected by	Successful	Successful %
Domain Name	93	100%
IP address	66	71%



(a) HTTP version support in Scan2021.

(b) HTTP version support in Scan2022.

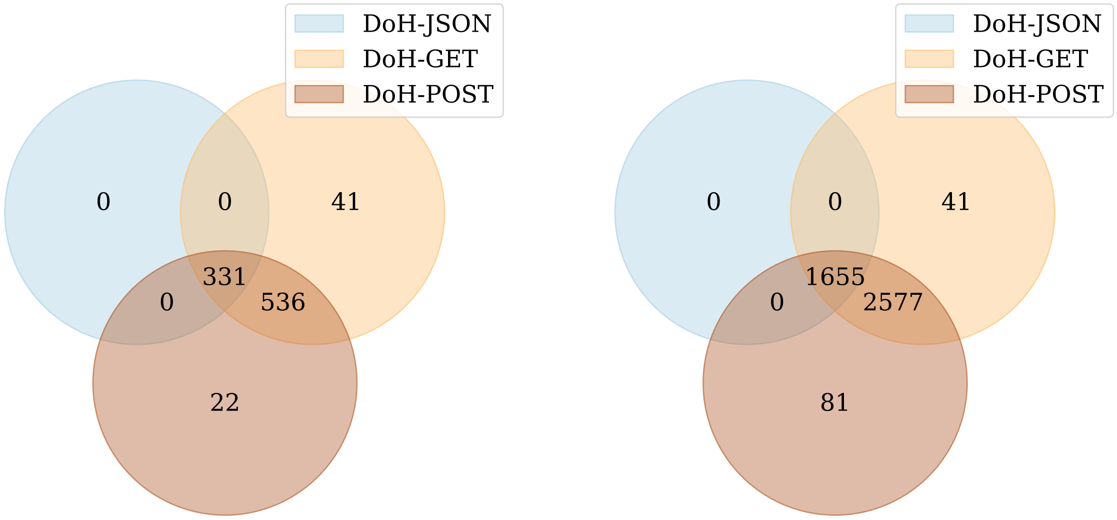
Figure 3.3: Venn diagrams of HTTP version support across DoH resolvers.

the distribution of an SNI requirement may differ from the rest of the resolvers on the Internet.

3.1.3.5 Capabilities of the DoH Resolvers Found

Since we have queried each DoH resolver multiple times, we can analyze the methods supported by the DoH resolvers. Figure 3.3 shows the HTTP version support. It can be seen that most DoH resolvers on both scans support both HTTP versions. In the 2022 scan, we observed an increased share of HTTP/2-only or HTTP/1-only resolvers with respect to the total.

The methods supported in the DoH resolvers are shown in Figure 3.4. Most resolvers support the RFC 8484 compliant versions. Some resolvers support only DoH-GET or only DoH-POST, even though the RFC 8484 specifies that the resolver must implement both methods. The IP addresses of those DoH resolvers supporting only DoH-GET are the same in both scans. The JSON approach is supported by around one-third of all resolvers. None of the resolvers supports the JSON approach exclusively.



(a) DoH methods support in Scan2021.

(b) DoH methods support in Scan2022.

Figure 3.4: Venn diagrams of supported methods across DoH resolvers.

3.1.3.6 DNS Server Identification

Table 3.4 shows the results of the DNS software identification for all the DoH resolvers that answered the specialized version query correctly (only 435 or 10%). However, most of them replied with an empty string response. The Scan2022 IP addresses were also queried using traditional unencrypted DNS over port 53/UDP. We used `nslookup` software to query the Google.com address with a 10-second timeout, and from 4354 only 1176 (∼27%) resolvers supported legacy DNS. We repeated the test three times with similar results.

3.1.3.7 Who Operates the DoH Resolvers

A total of 657 unique domain names from TLS certificates were analyzed to find out who is offering the DoH resolution services. At first, we tried to use the domain classification service NetStar[94]; however, only a negligible portion of domain names was classified. Therefore, we visited each of them manually via the web browser and classified domain names into one of 11 categories: **DNS/ISP/Cloud**—DNS providers, Internet service providers,

Table 3.4: DNS software identification of found DoH resolvers in Scan2022.

Name	#	%	Name	#	%
a) empty	113	26.0	g) AkamaiVantioCacheServe	10	2.3
b) Unbound	88	20.2	h) Q9	8	1.8
c) PowerDNS	77	17.7	i) NominumVantioCacheServe	8	1.8
d) unknown	68	15.6	j) SDNS	1	0.2
e) Bind	48	11.0	k) I-Evolve DNS	1	0.2
f) Dnsmasq	13	3.0			

Table 3.5: Share of DoH provider categories.

Name	#	%	Name	#	%	Name	#	%
a) unknown	280	41.9	e) other	24	3.5	i) security	16	2.3
b) DNS/ISP/Cloud	145	21.7	f) finance	22	3.2	j) government	11	1.6
c) personal webpage	92	13.7	g) software-provider	18	2.6	k) privacy	10	1.5
d) industry&business	34	5.1	h) education	16	2.3			

hosting providers and cloud providers; **industry&business**—manufactures, e-shops, and other types of trade business; **finance**—banks, investment advisers, and insurance companies; **software-provider**—companies providing software development services; **education**—universities, research institutes, and libraries; **security**—computer security companies; **government**—governments and governmental organizations; **privacy**—companies that focus on privacy such as VPN providers and privacy enhancement software; **personal webpage**—domain names hosting personal web site portfolio or personal blogs; **other**—companies and institutions that did not fall into any other category; and **unknown**—domain names did not host website, or that could not be categorised.

The share of each category among DoH providers is shown in Table 3.5. We were not able to categorize most of the resolvers. The web page hosted on these resolvers could not identify the website's owner, or the server did not serve web pages. When the web server responded with a web page, it usually showed a login page. Around 20% of the domain names in the category "unknown" showed a login page of AdGuard Home DNS resolver. Two servers were misconfigured and showed a directory structure of private project files.

For identification, we did not use information directly from the domain names. Although some domain names suggested that the server is operated by an individual, we also categorized it into the "unknown" category since we could not verify it. Nevertheless, we could not even estimate the owner for most domain names. Sometimes they seemed randomly generated, such as `hhgasdygqwueysbjadasghds.com` or `kasldjflkasdjf.xyz`.

The second most common category is DNS/ISP/Cloud providers, which offer DoH. A significant share of these companies might be expected since they usually provide DNS resolution as part of their services. The third most common category is private web pages. Individuals operate these resolvers, and the website usually contains the portfolio of a freelance software developer, or it was a personal blog.

3.1.3.8 TLS Certificate Analysis

We analyze the TLS certificate data of found DoH resolvers in the Scan2022. The share of certificate authorities is written in Table 3.6. The most common certification authorities across found resolvers are Let's Encrypt and ZeroSSL. Most of the DoH resolvers provided valid and trusted certificates. We found 193 ($\tilde{4}$.5%) IP addresses with expired certificates. More than 57% of those expired certificates were issued by the Let's Encrypt Certification Authority. The expiration date of the invalid certificates was mainly 2021 and 2022 (in 81% of the cases). The certificates of five resolvers expired before the DoH standardization in 2018.

Table 3.6: Share of TLS certification authorities across the found DoH resolvers in Scan2022. CA stands for Certification Authority, IIJ stands for Internet Initiative Japan Inc., and ERDC stands for Engineer Research and Development Centre.

CA Name	#	%	CA Name	#	%	CA Name	#	%
a) Let's Encrypt	1703	39.1	d) Blue Coat	106	2.4	g) IIJ	63	1.4
b) ZeroSSL	1654	38.0	e) Sectigo ltd.	103	2.3	h) WoTrus CA ltd.	36	0.8
c) other	545	12.5	f) Apple Inc.	100	2.3	i) ERDC	36	0.8

3.1.3.9 Threat Intelligence Results

From the 4354 IP addresses in the Scan2022 list inspected by the threat intelligence tools, 1502 are considered *suspicious for phishing* according to the Avast Web Shield tool. This fact does not mean that the IPs are currently malicious but were associated with phishing activities during the studied period 2021-2022. When we inspected URL links on the IP addresses websites, we found 105 addresses that reference a malicious site (According to VirusTotal). Moreover, 27 DoH-capable IP addresses were marked by VirusTotal as a source of *downloaded malware samples*—they directly hosted malware.

3.1.4 Scan Result Summary

The results presented in the previous section confirm that the deployment of public DoH resolvers is increasing. The number of well-known resolvers in 2022 increased by 12% compared to 2021. However, only 67% of the well-known DoH resolvers in 2021 remained active in 2022.

A similar phenomenon can be observed with Scan2021 and Scan2022, where only 9.8% of the IPv4 addresses were found in both scans. A possible explanation for this discrepancy is that the missing servers were for testing purposes and, as such, have been moved to a definitive address or stopped operations. Furthermore, 88% of the DoH resolvers found in Scan2022 were not previously seen by any list, nor Known2021, Known2022, or Scan2021. Nevertheless, even the Scan2022 list is incomplete. Approximately 55% of the well-known DoH resolvers in 2022 were not found in the Scan2022 performed in January 2022 due to limitations (domains requiring SNI) or other connection errors.

Moreover, the fact that so many DoH resolvers could not be found after one year and that the number of DoH resolvers is increasing suggests a great dynamism and casts doubts about the effectiveness of the blocklist-based DoH resolvers filtering.

By comparing the results of the HTTP versions supported by the discovered resolvers of the two scans, as shown in Figure 3.3, there were some changes in support of the HTTP version. There is a decrease in the percentage of DoH resolvers that support both HTTP/1 and HTTP/2; however, we can also see an increase in HTTP/1-only resolvers, even though the RFC does not recommend it due to performance reasons.

The list of differences in DoH resolvers can be summarised as follows:

- The well-known list of 2022 compared to the well-known list of 2021: 1.22x more IPv4, 28% intersection, 5x more organizations.
- DoH on Internet 2021 compared to well-known list 2021: 21x more.
- DoH on Internet 2022 compared to DoH on Internet 2021: 1.86x more (1.7x adjusted).
- DoH on Internet 2022 compared to well-known list 2021: 39.33x more.
- DoH on Internet 2022 compared to well-known list 2022: 32x more.
- 98.5% of the DoH resolvers found in Scan2022 are unknown to the community.

Only 21% of the DoH resolvers found in Scan2022 belong to DNS/ISP/cloud providers, while 44.6% belong to unknown organizations, and 12.7% belong to personal web pages. Almost 35% of the IP addresses found in our study show indicators related to phishing campaigns, and 27 of 4354 IPs were a source of malware. We expect the domain resolution service to be under constant security reviews, either if it is unencrypted through standard DNS or encrypted using DoH or some other protocols. The occurrence of DoH resolvers' IP addresses associated with malware or phishing shows that users' security and privacy could be already at risk or that these resolvers are misused for malicious purposes.

Leaving aside which of those groups can be considered trusted DoH resolvers, 77.3% of the certificates of DoH providers in Scan2022 were given by free services such as Let's Encrypt, suggesting low-cost deployment of DoH resolvers. The large number of DoH resolvers operated for private use, which are not listed in the DoH lists, allows threat actors to hide and abuse DoH in ways described in following Chapter 4.

3.2 Amount of Encrypted DNS Traffic

Compared to the previous section, which studied DoH adoption by the service providers, this section focuses on the encrypted DNS adoption across the users. Moreover, our scope is not limited only to DoH, but we also study the adoption of DoT and DoQ.

First, we describe the data source for the encrypted DNS traffic volume measurement and methodology. Then we analyze the traffic from four perspectives: (i) amount and comparison of encrypted DNS traffic in each network, (ii) analysis of the stationarity and trends of the traffic in each organization, (iii) ratios of the relationship between traffic, with their stationarity.

3.2.1 Description of the Source Networks and Datasets

The encrypted DNS was measured through a collaboration between three large organizations. The first organization is a European ISP provider, the second is a European University, and the third is a global security company.

Large European ISP Provider (Organization 1) Organization 1 is a national research and education network (NREN) infrastructure. It is an ISP that interconnects many academic institutions, research organizations, some government offices, and others. In total, the infrastructure routes the traffic from over 500,000 users and serves as a transit network for neighboring networks as well. At the perimeter, monitoring probes are equipped with FPGA-accelerated hardware cards to handle the high-speed traffic (over 100 Gb/s) and export extended IP flow data.

The flow export in this organization is configured with 5 minutes of active and 30 seconds of inactive timeout to produce unidirectional IPFIX [4] extended with custom information fields. Long connections are split by the "active timeout," and a flow record is exported every time this timeout elapses, even though the actual connection is not terminated yet. The connection is also terminated if no packet is observed within the "inactive timeout" period. The collected monitoring data are represented by unidirectional IP flows, which are different from other organizations; thus, densities and ratios were used to compare the traffic.

More importantly, Organization 1 has an internal peering (multiple lines) with cache servers of Google LLC services; however, these lines are not equipped with monitoring probes. Therefore, the traffic between internal users and Google is not measured; meanwhile, the traffic between foreign users (via the perimeter) and Google is measured.

The measurement methodology in Organization 1 can be summarized as follows:

- **DoH** traffic is filtered from IP Flows extended with TLS handshake information. It is therefore measured as the number of Client Hello packets with SNIs of well-known DoH providers transferred via port 443/TCP. The list of the providers used for DoH measurement is shown in Table 3.7.

- **DNS over TLS** traffic was also filtered from IP Flows extended with TLS information, which ensures the occurrence of Client Hello packets. However, in this case, only port filtering was used. Therefore, DoT traffic is calculated as the number of Client Hello packets observed on port 853/TCP.

- **DNS over QUIC** traffic was counted from traditional IP Flow data. Since QUIC obfuscates even the handshake packets by encryption, it is unfeasible for Organization 1 to perform DPI for QUIC recognition due to the lack of available computational power. Therefore, DoQ traffic is calculated as the number of flows observed on port 784/UDP[5].

- **DNS** traffic was measured from traditional IP Flow data by filtering by the port number of 53/UDP and 53/TCP.

- **Total** traffic was measured from traditional IP Flow data without any filtering.

[5] The 853/UDP port was assigned to DoQ more than one year after the measurement. During the measurement, 784/UDP was recommended to use for DoQ.

Table 3.7: Major DoH providers used for measurement.

	IPv4-1	**IPv4-2**	**IPv6-1**	**IPv6-2**
dns.cloudflare.com	1.1.1.1	1.0.0.1	2606:4700:4700::1111	2606:4700:4700::1001
dns.google.com	8.8.4.4	8.8.8.8	2001:4860:4860::8888	2001:4860:4860::8844
dns.aa.net.uk	90.155.62.13	90.155.62.14	N/A	N/A
dns-nyc.aaflalo.me	104.27.159.50	104.27.158.50	N/A	N/A
dns.adguard.com	104.20.31.130	104.20.30.130	2a10:50c0::ad1:ff	2a10:50c0::ad2:ff
doh.cleanbrowsing.org	192.124.249.8	N/A	N/A	N/A
nic.cz	193.17.47.1	185.43.135.1	N/A	N/A
dns.nextdns.io	104.31.88.168	104.31.89.168	2a07:a8c0::1c:7db6	2a07:a8c1::1c:7db6
dns.brahma.world	104.27.170.14	104.27.171.14	N/A	N/A
dns1.dnscrypt.ca	69.165.220.221	N/A	2620:fe::fe	2620:fe::9
libredns.gr	116.202.176.26	N/A	N/A	N/A

Large European University (Organization 2) Organization 2 is a faculty of a public University in Europe. The measured network contains approximately 2200 non-WiFi desktop computers and servers.

Organization 2 uses the Zeek Security Monitor [96] to capture traffic between its network and the Internet. The traffic is filtered so only flows *started* by computers within its IPv4, and IPv6 ranges are included. There is an official authority DNS server of the organization that works as an open resolver for the local computers and also for users from the Internet. Thus, the traffic *from* this DNS server is removed to measure only the DNS communication originating inside Organization 2.

The data from Organization 2 has the following features:

- Number of flows: Total flows generated by the organization. From Zeek conn.log.
- Number of flows to port 443/TCP: Regardless of the state. From Zeek conn.log.
- Number of flows TLS Established: Established TLS flows in any port. From Zeek ssl.log
- Number of DoQ flows: Flows to port 784/UDP. From Zeek conn.log.
- Number of DoT flows: Flows to port 853/TCP. From Zeek conn.log.
- Number of DoH flows: Flows to well-known DoH providers from Table 3.7. From Zeek ssl.log.
- Number of DNS flows: Flows going out to port 53/UDP but not to the main DNS server of the organization. From Zeek conn.log.
- Number of unique source IPs: Count of unique IPv4 and IPv6 IPs. From Zeek conn.log.

The DNS flows were filtered to discard flows going to the official DNS server of the organization because (i) the traffic from the local network to that server does not appear in the capture, and (ii) we discard spurious connections from the Internet to the main DNS server.

The DoH flows were computed by first filtering all established TLS traffic, i.e. to port 443/TCP with an established TLS session, and then by filtering the IP address of the DoH providers shown in Table 3.7.

Global Security Company (Organization 3) Organization 3 is a large, global security company protecting hundreds of millions of endpoints. To protect clients against security issues, the traffic produced in the endpoints is analyzed and blocked in the endpoints. The solution processes HTTP and HTTPS traffic from both IPv4 and IPv6. It uses URL detection algorithms to analyze and protect against threats and full content filtering to stop malware. As a result, over 300 billion URLs are checked each month.

As part of the research to identify new and emerging threats, the threat intelligence gathered from the endpoints can be queried to produce trends and statistics, for instance, to count the number of DoH and DNS requests.

For privacy reasons, Organization 3 can not share the absolute numbers for their measurements. Therefore all the analyses are done by ratios of values, such as *DoH flows per user*.

3.2.2 Amount of Encrypted DNS Traffic

The first descriptive analysis is done on the amount of traffic of every encrypted protocol that has been seen by the three organizations. First, we show the total amounts for Organizations 1 and 2, and then we compare the combinations of ratios of values between all three organizations.

The DoH traffic was filtered in all organizations using the list of DoH resolvers in Table 3.7. We selected a limited number of resolvers since Organization 1 has a technical limit in its filtering capabilities.

The traffic of Organization 1, described in Section 3.2.1, is shown in Figure 3.5. This Figure uses a logarithmic y axis and computes the daily values. The number of DoH flows per day has $mean$=181,794 and STD=78,331. The DoT flows per day has a mean of 28,395 and a STD=10,120; being on average 6.4x times smaller than DoH. Note that the amount of DoH traffic is larger than the amount of DoT by one order of magnitude. The number of DoQ flows per day has a $mean$=6235 and a STD=20,131. The amount of DoQ traffic is approximately 29 times smaller than DoH but still significant.

Regarding the percentages of each protocol for Organization 1, the percentage of DoT traffic to the total flows had a $mean$=0.00001%, and STD=0.00007. The percentage of DoH traffic to the total flows had a $mean$=0.00007% and STD=0.0004.

For the use of any encrypted DNS protocol in comparison with all DNS-related protocols, Organization 1 has, on average, a 0.01% of its DNS traffic being encrypted.

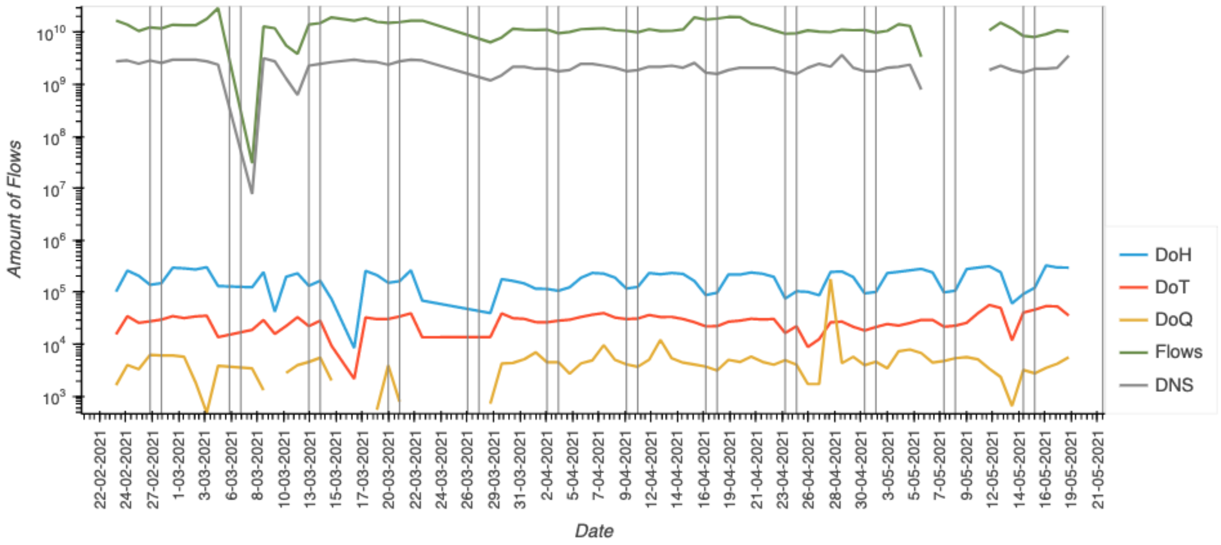


Figure 3.5: Number of flows for Organization 1 in logarithmic scale. Total flows, number of DNS flows, DoH flows, DoQ flows, and DoT flows. From February 24th, 2021 to May 15th, 2021. Vertical lines represent weekends.

The traffic of Organization 2, described in Section 3.2.1, is shown in Figure 3.6. This figure uses a logarithmic y axis and computes the daily values. It can be seen that the DoT traffic is larger than the DoH traffic, which is opposite to Organization 1. We could not reconcile this difference, and we attribute it to the potentially larger number of Android phones in Organization 2. The amount of DoH traffic is, on average, 35 times smaller than DoT. The amount of DoQ traffic is almost continually zero. The mean and standard deviation values for all the measurements are shown in Table 3.8. The STD of the number of flows is large due to the infection happening around Feb 28th and the outage happening around May 3rd.

Figure 3.6 shows that the DoT traffic had an important decrease around mid-January 2021; that is not an anomaly but a consequence of some unknown change in the process. The reasons for this strong decline could not be found, but as shown in Section 3.2.4, a similar but opposite phenomenon happened in 2020. Around February 22nd, there was a peak in the number of flows to port 443/TCP (top red line). This peak corresponds to an infected computer[6] inside the organization that scanned ports 443/TCP on the Internet. However, it can be seen that the TLS Established traffic (yellow line) does not pose a peak since most of the connections were not successful.

It can also be seen in Figure 3.6 that the number of unique IP addresses (both IPv4 and IPv6) is quite constant (red line in the middle), with a small standard deviation of 253. Around May 3rd, there was a small network outage that reduced the amount of traffic collected and added some bias to the measurements.

Regarding the percentages of each protocol for Organization 2, the percentage of DoT

[6] The infection was verified and cured during the same day.

Table 3.8: Mean and STD of the daily traffic volumes for Organization 2 from January 1st, 2021, to May 23rd, 2021. The traffic volume mean is calculated from flows. The mean of IP addresses is calculated from unique addresses.

Value	Mean	STD
DoH Traffic	50.3	237.6
DoT Traffic	1782.4	2459.0
DoQ Traffic	0	0.1
DNS Traffic	5,524,360.2	2,535,076.7
Total Flows	15,504,487	12,290,761.5
TLS Traffic	506,300.9	222,479.8
Traffic on Port 443	2,227,208.6	10,780,265.3
IP Addresses	2971.4	253.6

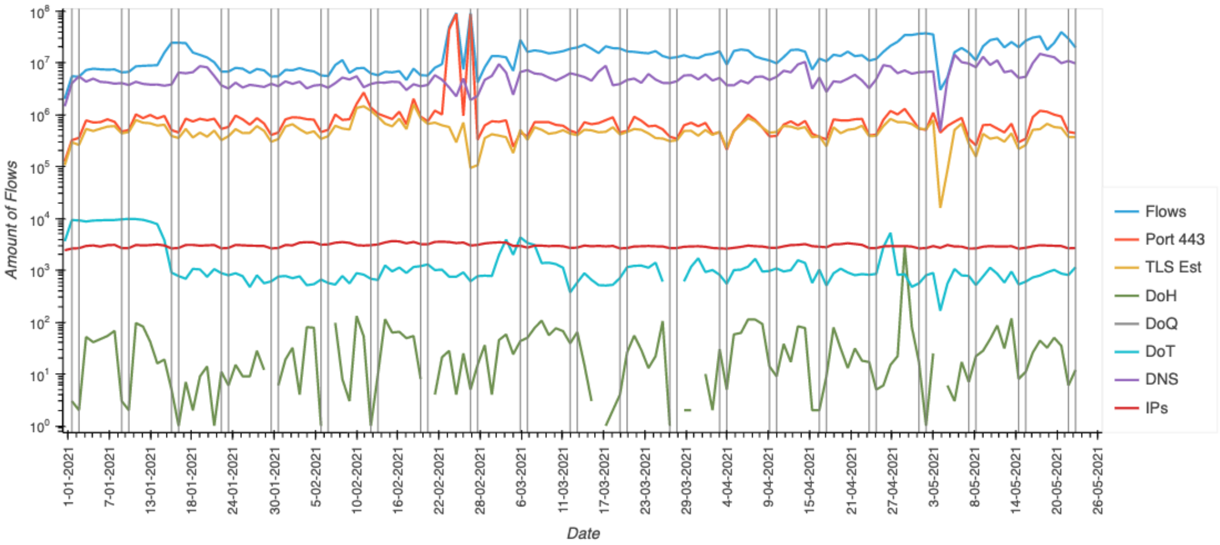


Figure 3.6: Traffic per day in logarithmic scale from Organization 2 showing the total number of flows, number of DoH flows, number of DNS flows, number of established TLS flows, number of flows to port 443/TCP, number of DoT flows, number of DoQ flows, number of DNS flows and amount of IP addresses. From January 1st, 2021 to May 23th, 2021. Vertical lines are on weekends.

traffic to the total flows had a *mean*=0.0002%, and *STD*=0.0003. Compared with Organization 1, the percentage of DoT traffic is 20 times larger in Organization 2 despite its total amount being 16 times larger in Organization 1.

The ratio of DoH flows to the total flows for Organization 2 had a *mean*=0.000003%, and *STD*=0.000007. The percentage of DoH traffic is 23x times larger in Organization 1 than in Organization 2, and its total amount is 3.6 times larger in Organization 1.

Regarding the use of any encrypted DNS protocol compared with all DNS-related protocols, Organization 2 has, on average, a 0.033% of its DNS traffic being encrypted. This is 3 times more than Organization 1.

The traffic of Organization 3, described in Section 3.2.1, is shown as ratios of other values. The first comparison is on the amount of DoH flows per unit user per country per day, normalized by the country's population. The countries used are the top 8 countries with the largest amount of DoH flows in total. This ratio is computed by dividing the DoH traffic for the whole country per day by the total number of unique users sending DoH in all countries and again dividing by the country's population.

Figure 3.7 shows the DoH values per user per country normalized by the relative population of the country. Population data taken from Datacommons[7]: Argentina 44.94 M, Brazil 211 M, Italy 60.36 M, Mexico 127.6 M, Poland 37.97 M, Russia 144.4 M, Spain 46.94 M, US 328.2 M. The normalized comparison shows that even though US and Brazil are still the top senders, they are not by a large margin, and Italy appears like a strong DoH user, followed by Argentina. The exact mean and standard deviation values are shown in Table 3.9.

We did not normalize the values with the number of users that Organization 3 has in each country, but we assume the population follows the size of the countries since Organization 3 is global.

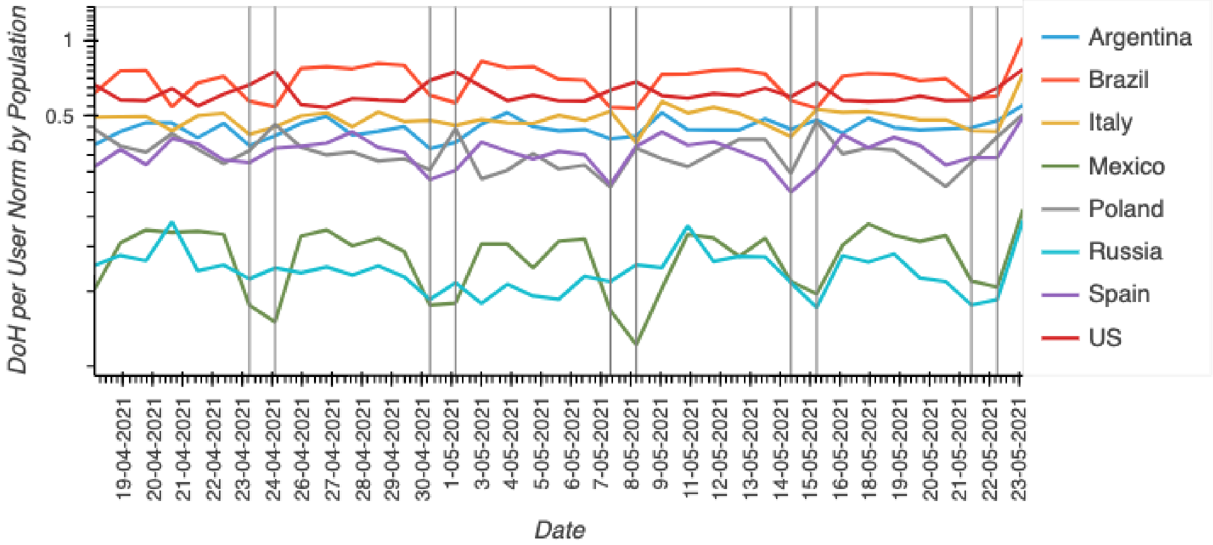


Figure 3.7: Organization 3 DoH traffic per user per country, normalized by the relative size of the population of each country. Vertical lines represent weekends. Population estimation in millions taken from datacommons.org.

Stationarity of Values To better understand if the traffic is growing, decreasing, or having any trend, we tested the stationarity of the measurements. It was tested using the Augmented Dickey–Fuller test (ADF) [92], which is the standard test of non-stationarity of time series. A stationary time series is one whose properties do not depend on the time at which the series is observed [67]. Time series that show trends or seasonality are non-stationary. The implementation used was the function *adfuller()* from the Python library *statsmodels.tsa.stattools.*

[7] `https://datacommons.org/`.

Table 3.9: Mean and STD for the top 8 countries by DoH count, of "DoH flow count-per-user, per day, per capita," in Organization 3.

Country	Mean	STD
Argentina	0.44	0.03
Brazil	0.69	0.1
Italy	0.49	0.05
Mexico	0.14	0.03
Poland	0.35	0.05
Russia	0.12	0.02
Spain	0.35	0.04
US	0.61	0.05

Table 3.10: Augmented Dickey-Fuller test for stationarity of values in the three organizations for the duration of their measurements in 2021. The confidence interval was set to 95%.

Org	Value	ADF Stat	P-value	Conclusion
1	DoH	-2.7	5.95e-02	Non-Stationary
1	DoT	-1.1	6.91e-01	Non-Stationary
1	DoQ	-8.7	2.66e-14	**Stationary**
1	Flows	-2.03	2.70e-01	Non-Stationary
1	DNS	-4.3	3.01e-04	**Stationary**
2	DoH	-11.6	1.77e-21	**Stationary**
2	DoT	-2.2	1.80e-01	Non-Stationary
2	DoQ	-12.09	2.13e-22	**Stationary**
2	DNS	-5.3	5.32e-06	**Stationary**
2	Flows	-4.6	1.27e-04	**Stationary**
2	TLS	-6.6	4.42e-09	**Stationary**
2	Port443	-3.3	1.33e-02	**Stationary**
2	IPs	-2.07	2.56e-01	Non-Stationary
3	DoH	-0.1	9.44e-01	Non-Stationary

The results of the ADF test on the raw amount of data for the three organizations can be seen in Table 3.10. The test shows that in Organization 1, the only Stationary value is the DoT traffic. The rest of the values seem to have a trend or a strong seasonality. For Organization 2, most of the values are stationary, except the number of IP addresses, according to its linear model, which seems to be growing. Note that the DoT traffic of Organization 2 is ignored since there were only two days with a value of 1.

In particular, for the number of IP addresses of Organization 2, the ADF test failed to reject the null hypothesis that the series is non-stationary with confidence of 95%. Therefore we conclude that the number of IPs is non-stationary. This result may seem contradictory with Figure 3.6 where the line for IP addresses seems *very* stationary. However, a detailed study of the time series of IP addresses shows that it has a strong cycle due to weekends. The result of non-stationarity is, therefore, correct because according to the theory, a time series of cyclic behavior (without trends or seasonality) is stationary *only if the cycles are not of fixed length.* Since the cycles have a fixed length due to the weekends, the series is therefore non-stationary [67].

3.2.3 Ratios Between Organizations

Ratio DoH per 1 Million Flows Given the different numbers of users in each organization, it was necessary to compute a ratio-per-user and a ratio-per-amount-of-flows to compare the changes in the traffic. These ratios are based on the idea that, given large enough networks, the users tend to generate a similar amount of traffic. Therefore, computing the ratios-per-user gives an idea independent of the number of users.

The first ratio comparison done is the *number of DoH flows per 1 million flows*, computed each day. This ratio gives an idea of how much DoH traffic is generated compared to the total traffic but independently of the number of users seen in each organization. Instead of comparing with the total amount of flows, which is a varying quantity, comparing with the DoH flows per 1 million flows is a much more stable and *transferable* metric.

The comparison of ratios for all three organizations is shown in Figure 3.8. Because each organization had access to a slightly different time frame, the Figure shows an overlap only in the last 40 days. More importantly, the comparison shows that the ratio of DoH flows per 1 million flows for all three organizations is *comparable* and not with large differences. For Organization 1 (top blue line), the mean is 15.2 (STD=8.9); for Organization 2 (bottom red line), the mean is 3.2 (STD 7.4); and for Organization 3 (middle yellow line), the mean is 4.25 (STD 0.2). Even though the data does not come from the same exact distribution according to an ANOVA test, the values are a good indicator and estimator of this ratio for other networks.

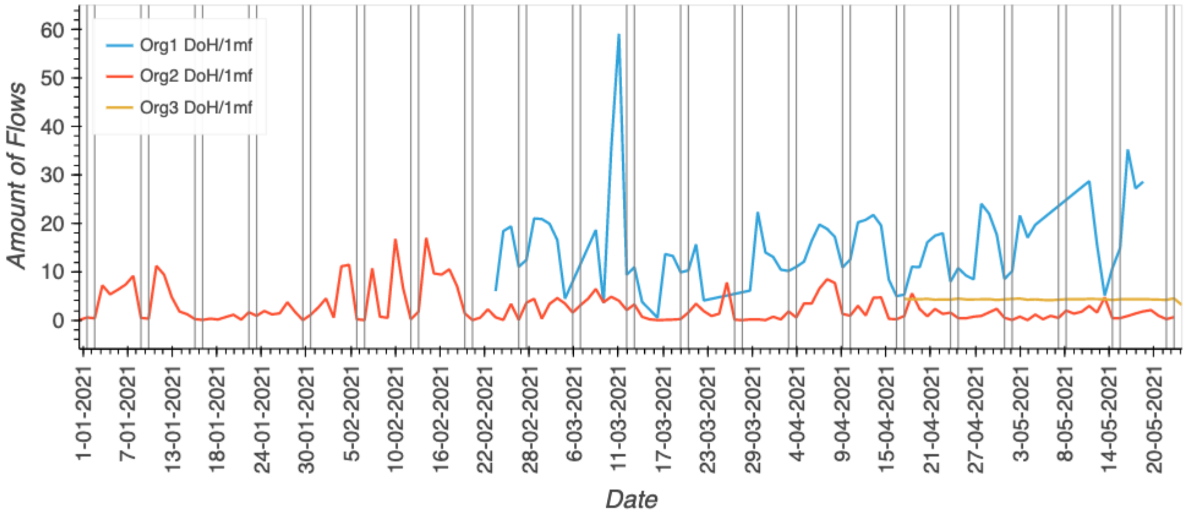


Figure 3.8: Comparison of the number of DoH flows per 1 million flows for all Organizations. The ratios were processed to discard anomalies.

The second evaluation of the ratio of DoH per 1 million flow was regarding its stationarity, and it used again the Augmented Dickey–Fuller test (ADF) (described in Section 3.2.2). The results of the test show that for Organization 1, the ratio had a statistic of -3.62, with a p-value of 5.35e-03, and therefore using a confidence interval of 95%, we can estimate that it is Stationary. For Organization 2, the ratio had a statistic of -11.28, with a p-value of 1.40e-20, and therefore using a confidence interval of 95%, we can estimate that it is Stationary. For Organization 3, the ratio had a statistic of -4.59, with a p-value of 1.34e-04, and therefore using a confidence interval of 95%, we can estimate that it is Stationary.

Ratio DoH per 1 Million DNS Flows The second ratio comparison done is regarding the amount of DoH flows per 1 million DNS flows. This ratio, similar to the last one, allows a comparison between the organizations regardless of the number of their users. In this case, we compare DoH with DNS since both protocols are intimately related, and we may even expect a small decrease in DNS if DoH grows. Figure 3.9 shows the ratio comparison. Similarly, as in the previous case, even though the data does not come from the same distribution, the numbers are *comparable* and with some similarity. For Organization 1, the mean of the ratio is 86.5 (STD 56.9), and for Organization 2, the mean of the ratio is 8.88 (STD 33.5).

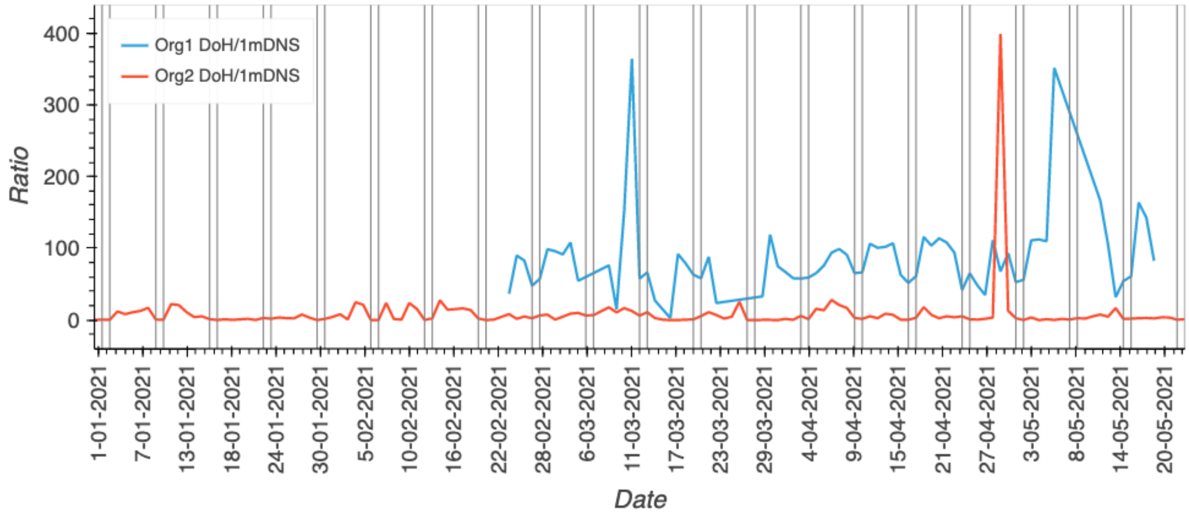


Figure 3.9: Comparison of the number of DoH flows per 1 million DNS flows for Organization 1, and 2. The ratios were processed to discard anomalies.

Regarding their stationarity, using the ADF test, for Organization 1, the statistic is -3.6, with a p-value of 5.57e-03; therefore, using a confidence interval of 95% we can conclude that its ratio is Stationary. For Organization 2, the statistic is -11.6, with a p-value of 2.43e-21; therefore, using a confidence interval of 95%, we can conclude that the ratio is Stationary.

3.2.4 Past Trends in Encrypted DNS

Given that almost five months of traffic may not be sufficient to see larger trends, we accessed past traffic captured by Organization 2 during 2020 using Argus sensors. This traffic was not used for comparison with the other organizations since they did not retain old traffic. This traffic is a measurement of DoH, DoT, and DoQ flow protocols, as shown in Figure 3.10. Apart from a sensor problem that broke the capture from 2020/09/09 to 2020/09/23, it can be seen that there are two clear growing trends for DoT and DoQ. In particular, from 2020/08/01, the DoT traffic grew 3 times in a couple of days. The DoQ traffic continued to be around 4200 requests per day from August 1st to October 10th, after which it lowered to 1100 requests per day until November 24th. On November 25th, it started to grow, reaching a peak on December 12th of 14,000 requests per day. Given that the number of IPs in Organization 2 remained almost constant, it can be seen that the *peak* ratio of DoT per IP address was around 5.6 requests per day.

This is a large change compared to the mean of the ratio DoH/IP address for 2021, which was 0.016. The peak ratio of the DoT per IP in 2021 was 350x larger than the mean value for the first five months of 2021.

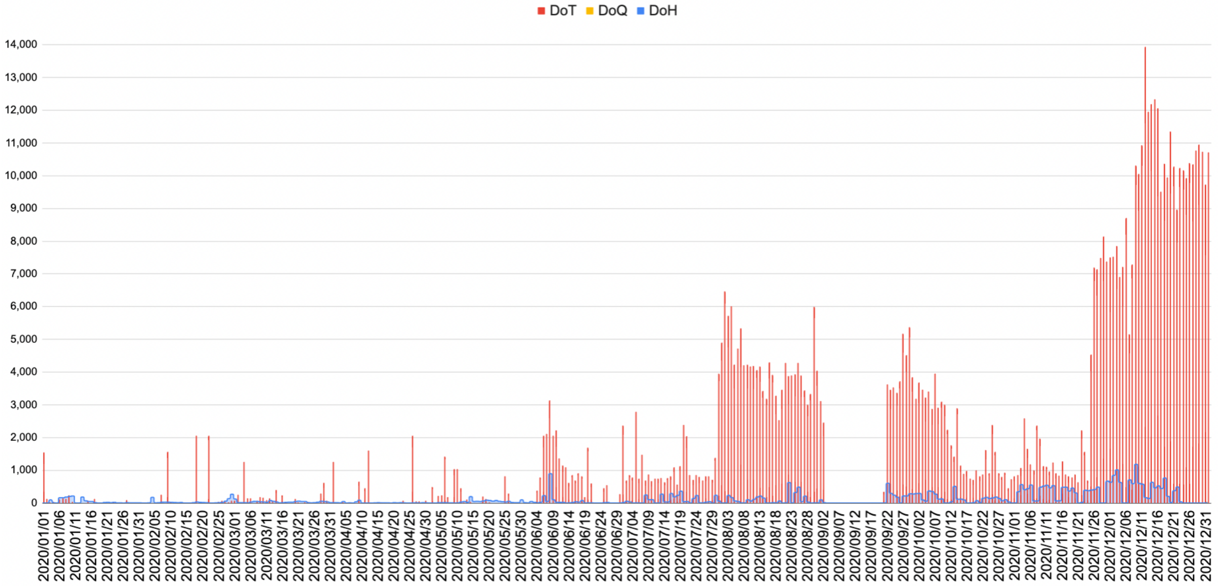


Figure 3.10: Number of DoH/DoT/DoQ requests during 2020 in the traffic of Organization 2.

3.2.5 Measurement Summary of Encrypted DNS Adoption by Users

DoH Trends DoH traffic has grown from 2020 to 2021, according to our measurements, but in the first five months of 2021, its growing ratio has become stationary. The percentage of *DoH to total flows* is larger in the ISP provider (Organization 1) than in the university (Organization 2). These results are consistent with the previous work of Lu et al. [80] (see Chapter 2). The difference between *DoH flows per 1 million flows* and *DoH flows per 1 million DNS flows* is significantly similar in the three organizations with a nearly constant mean, which supports the idea that DoH has become stationary in several locations around the world. The only apparent growth in DoH is in ISP Organization 1.

For the global security company (Organization 3), DoH has gone from stationary to slightly decreasing. However, when DoH was measured per user, by country, and normalized by population, the amount of DoH was stable. USA and Brazil have larger *DoH-per-user-per-capita*, but Italy and Argentina are close behind.

DoT Trends DoT traffic seems much larger in the ISP (Organization 2) than in the other organizations. Showing a non-stationary growth in this organization. However, it shows an actual decrease in Organization 2 in mid-January 2021. The absolute number of DoT flows is larger than DoH in all the traffic captures combined.

DoQ Trend The amount of DoQ traffic was only significant in Organization 1, with a small but stationary amount; almost zero in Organization 2, and not measured in Organization 3. It may be possible that the differences are due to artifacts in the network capture mechanisms.

Comparison between Organizations Despite the differences in size, type of organization, and scope, it is possible to say that the *ratio* of DoH traffic growth at the beginning of 2021 is similar and almost constant for all organizations. Organization 1 is the most different, showing a small growth in the DoH ratio for May 2021.

3.3 Key Findings

This chapter presented the research on encrypted DNS adoption by users and also by the DNS service providers. Moreover, the properties of DoH resolvers has been discussed. The key findings in this chapter are:

1. The amount of encrypted DNS (particularly DoH and DoT) has risen in the studied period in two organizations.

2. DoH was mostly used in the United States of America and Brazil (from the list of considered countries).

3. The lowest DoH usage has been measured in Russia (from the list of considered countries).

4. The number of DoH providers has risen between 2021 and 2022 almost two times.

5. The small personally-owned DoH resolvers represent 13.7% of all DoH resolvers on the Internet in 2022.

6. Some of the DoH resolvers IP addresses found in 2022 on the Internet took part in malware campaigns.

7. There are 32x more DoH resolvers on the Internet than listed in the open-source DoH blocklists.

The results in this chapter suggest a high impact on the network security. More and more users are starting to use DoH; thus the security monitoring tools are losing a significant amount of DNS traffic for inspection. Network intrusion detection should not rely on plain DNS data in the future. Moreover, the increasing trend in DoH resolvers deployment might be just a reaction to increasing demand from users. Nevertheless, the poor IP reputation and the history associated with malware campaigns or phishing campaigns are worrying properties. Moreover, the large number of DoH resolvers operated for private use allows threat actors to hide and abuse DoH.

Chapter **4**

Encrypted DNS Misuse

Due to the increased privacy of encrypted DNS, there is a large potential for its misuse by threat actors. In this chapter, we thus summarize our findings about currently encrypted DNS misuse, mainly focusing on DoH, which is by far the most misused approach. The chapter is divided into three sections: The first Section 4.1 surveys current proof of concept codes and actual malware families that leverage increased privacy of DoH and provide DoH abuse taxonomy, which we published in [A.3]. Following Section 4.2 explores the measures deployed by DoT providers to prevent DoT abuse. This section is based on our work [A.6], which was awarded by the CNSM 2022 conference committee. And last Section 4.3 describes novel abuse vectors used in the web environment that we found during our cooperation with anti-virus company Avast Software s.r.o. The novel abuse vectors were also published in [A.3].

4.1 Taxonomy of DoH Abuse: Tools & Malware

Since the encrypted DNS is built upon the traditional DNS, its abuse possibilities can be derived from classical DNS protocol. According to the 2016 Cisco annual security report [30], 91.3% of malware families use DNS, and the number is not decreasing. DNS is primarily abused for accessing C2 infrastructure as well as data exfiltration. Incorporating DNS into malware's infrastructure increases its resilience against threat protection systems, for instance, when combined with DGA [115] and Fast Flux [93] techniques. The resilience of malware even increases when deploying these techniques via encrypted DNS due to added encryption.

Malware creators are aware of the advantage of encryption and have started to use it to avoid detection [125]. However, not every traditional DNS abuse technique can be applied to encrypted DNS. For example, DNS amplification, a common DDoS attack vector, is a widespread problem first described by Evron et al. [104]. Fortunately, DNS amplification cannot be performed with encrypted DNS. DNS amplification attacks spoof source IP addresses such that the DNS resolver's response is sent to the victim's system. Encrypted

DNS requires establishing TCP or QUIC connection; thus, source IP address spoofing is impossible.

In this section, we survey only DoH abuse approaches since DoH cannot be reliably detected and thus blocked in the network, creating tempting opportunities for the threat actors. We are unaware of any DoT and DoQ capable malware; nevertheless, DoH is already misused in multiple malware and proof-of-concept codes. We have analyzed multiple public sources of information and related works (such as GitHub, malware analysis research blogs, and VirusTotal) to summarize the state of DoH abuse.

We divide known DoH abuse into three categories: (i) C2 Access and Communication, (ii) Covert Channels, and (iii) Unaware Usage. Table 4.1 summarizes the number of DoH-abusing code/malware samples we are aware of for each category. The following sections describe the categories and the code/malware samples in further detail.

Table 4.1: Number of DoH-capable code samples/malware strains for each category.

Category	**#**	**References**
C2 Access and Communication	10	[122, 129, 54, 125, 1, 124, 78, 116], Novel abuse scenario described in Section 4.3.2,Section 4.3.3
Covert Multipurpose Channels	4	[8, 88, 119, 45], Novel abuse scenario described in Section 4.3.1
Unaware Usage	N/A	Any SW with DNS

4.1.1 C2 Access and Communication

C2 communication is one of the most common abuses of unencrypted DNS. In the encrypted case, most malware uses DoH only to gain access to the C2 infrastructure. C2 communication itself then continues via other protocols. An example of such usage is the `PsiXbot` malware. The analysis created by the Proofpoint threat insight team [122] reveals that `PsiXbot` uses the hardcoded `dns.google.com` resolver and issues a JSON-based DoH request via HTTP 1.1 to resolve a hardcoded C2 domain. After receiving the C2 server IP, the communication between C2 and malware uses HTTP, which is unencrypted. Interestingly, the HTTP payload is encrypted using the RC4 algorithm. Similarly, banking malware `FluBot`, which targets Android devices, also relies on DoH to access its C2 infrastructure [129].

Another case of DoH abuse was published by Huntresslabs [54] describing the JSON-based TXT request for DKIM using DoH via the `dns.google.com` domain resolver. The TXT answer contained the IP addresses of external servers for downloading another payload to complete the C2 access. Both approaches exploit the fact that Google DNS is the most popular DNS resolver [103]; thus, it is probably accessible.

Overall, we are currently aware of five approaches that gain access to the C2 infrastructure using DoH [122, 129, 54, 125, 1], and all of them are slight modifications of the two mechanisms described above. All five approaches use the JSON API of DoH, and they mostly use Google's DNS resolver. The only exception is the `Godlua` malware [125], which uses Cloudflare's DNS resolver.

Malware can also utilize DoH as a channel for transmitting C2 commands. The LSD malware [124] uses DoH for accessing C2 infrastructure and downloading (via TXT records) a bootstrap script to connect to a crypto-mining pool proxy.

The other existing proof-of-concept (PoC) source codes are not yet deployed in any actual malware. One noteworthy PoC code is godoh [110, 78], which uses DoH via its JSON API to tunnel C2 conversations. A similar concept called DoHC2 [116] was implemented for the adversary simulation and red team operations software Cobalt Strike[1].

4.1.2 Covert Multipurpose Channels

Some solutions for covert channels natively support DoH. The dnstt [45] is a tool capable of data exfiltration via DoH. Similarly, the DNSExfiltrator [8] can upload files to the server via DoH with Google's or Cloudflare's resolvers. Ciampanu [29] reports that DNSExfiltrator is already used by the OilRig group, which is tracked as Advanced Persistent Threat group 34 (APT34). In addition, DoH tunnels are already covered in red team seminars and conferences like 44CON [88] or BruCON [119], where an Excel sheet downloads malware via a DoH tunnel.

Moreover, there are multiple solutions available on regular, unencrypted DNS, such as Iodine [42], DNSCat [99], or TUNS [95]. Even though these well-known and easy-to-use programs do not support DoH, they can extend their capabilities by running a DoH proxy.

4.1.3 Unaware Usage

For comprehensiveness, there is also a separate category, "Unaware Usage" which we have identified during our analysis. With the large-scale deployment of DoH in popular browsers and Operating Systems, malware DNS communication might get encrypted without the malware's intention or awareness of the encryption. Canonical examples are web browser extensions that call a browser API for domain resolution, or malware might use DoH because DoH is set as a default DNS method in the OS. As an example of the consequence, malware becomes stealthier due to encrypted communication, even though the malware itself is not aware of DoH.

From the network security perspective, these scenarios are the most challenging. We are not aware of any study that analyzed the detection possibility of malicious DNS traffic mixed with benign inside the same DoH connection. Untangling the mix is a challenging problem.

4.2 Evaluation of DoT Tunneling

As described in the Section 4.1, DNS tunneling is one of the typical DoH abuses. However, with the advance in traditional DNS tunnel detection, which has become very accurate in the last years [133], we expect that service providers would deploy security measures,

[1] https://www.cobaltstrike.com.

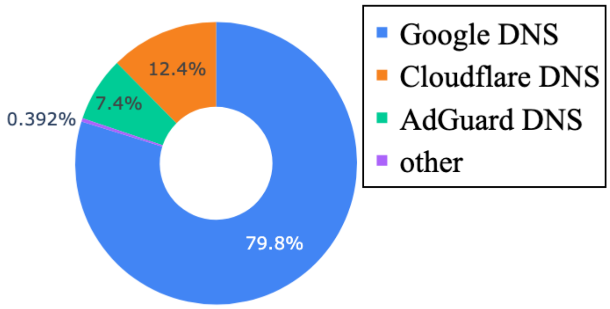


Figure 4.1: Share of DoT resolvers in CESNET network. The share was calculated from the number of DoT connections.

including protections against DNS tunneling. Since DNS service providers can process unencrypted DNS messages, they have the opportunity to mitigate these threats and perform payload-based and signature-based detection of DNS abuse, which are, according to Wang et al. [133], the most accurate approaches. Moreover, there already are DNS servers claiming anti-malware or family-friendly protection[2]. We performed the first study that evaluated their protective properties.

Compared to related works on encrypted DNS tunneling (see Section 2.2.4.2), which considered only DoH, we focused our experiments on DoT tunnels. In the following sections, we will describe the results of our experiments with the most popular DoT resolvers and evaluate the usability and, thus, threat severity of DoT tunnels.

4.2.1 Experimental Methodology

The proper methodology used for the experiments is crucial in obtaining relevant results. Therefore, we will first describe our experimental setup.

4.2.1.1 Selection of DoT Resolvers

The first step was to identify relevant DoT-capable resolvers. We decided to focus our experiments on well-known and established DNS resolvers since they are more important from a security perspective. We argue that if an attacker deploys the unprotected resolver, the DoT connection[3] will be considered suspicious due to an unknown IP address and it can be then blocked by the firewall. On the contrary, the use of well-known and established DoT resolver is much more stealthy. In the case of DoT, the main threat arises from abusing well-known services such as Google DNS, which is used by millions of benign users. Moreover, we wanted to select resolvers with global reach, making the results applicable worldwide and representing global security risks related to encrypted DNS.

To select the most popular DoT resolvers, we worked with CESNET, which provides Internet service to more than half a million users.

[2] `https://kb.adguard.com/en/general/dns-providers`.

[3] Compared to DoH, DoT can be recognized by its use of assigned port 853/TCP.

The anonymized one-month traffic from CESNET captured in January 2022 was analyzed to obtain the most popular DoT resolvers. DoT traffic was selected via port filtering since it uses port 853/TCP [63], resulting in 10 million DoT connections. Consequently, domain names of used resolvers were then extracted from Server Name Indication extension, which is transmitted during the TLS handshake. The share of individual resolvers is depicted in Figure 4.1. Resolvers marked as “other” were mainly local, operated either by CESNET itself or by universities. Since these resolvers are not used globally and are not dominant even in CESNET (only 0.4% of all DoT connections), we decided not to include them in our analysis.

The real-world analysis of DoT traffic showed only three well-established providers that represent 99.6% of all DoT traffic on CESNET. Since the analysis was only limited to traffic from Czechia, we also used a list of well-known DoT resolvers maintained by DNS privacy project[4]. Together, we evaluated 16 DoT resolvers listed in Table 4.2. All of them are operated by large global organizations, and thus we assumed they have an extensive user base. Therefore, the observed DoT communication with them usually does not raise suspicion. We also purposely selected family-filtered versions of DNS resolvers (when available), which we considered more protective. Moreover, the CleanBrowsing family filter explicitly claims security protection [31].

Table 4.2: Evaluated public DoT resolvers, marked ones allowed the creation of DoT tunnel.

Name	**Domain Name**	**IP Address**
Google DNS	dns.google.com	8.8.8.8
CleanBrowsing	family-filter-dns.cleanbrowsing.org	185.228.168.168
AliDNS	dns.alidns.com	223.5.5.5
BlahDNS	dot-de.blahdns.com	78.46.244.143
Bitdefender	ore-dns.bitdefender.net	35.247.80.47
Cloudflare	one.one.one.one	1.1.1.1
Dismail	fdns2.dismail.de	159.69.114.157
Quad9	dns9.quad9.net	9.9.9.9
AppliedPrivacy	dot1.applied-privacy.net	146.255.56.98
NextDNS	dns.nextdns.io	178.255.154.59
Adguard	dns.adguard.com	94.140.14.14
Adguard-F	dns-family.adguard.com	94.140.15.16
Bitdefender	fra-dns.bitdefender.net	35.242.226.78
Digitalcourage	dns3.digitalcourage.de	5.9.164.112
Bitdefender	ore-dns.bitdefender.net	35.247.80.47
dns.sb	dns.sb	185.222.222.222

4.2.1.2 Experimental Environment

Contrary to DoH, we are unaware of any malware or exfiltration tool that natively supports DoT. It does not mean that DoT cannot be misused—there are DNS to DoT translation proxies, which are transparent for connected devices, leaving them unaware of encryption.

[4] `https://dnsprivacy.org/public_resolvers/`.

Any software (including malware) which does not support DoT natively can then take advantage of encryption when the proxy is deployed (e.g., on a router).

Our testing setup is depicted in Figure 4.2 and follows the unaware usage scenario (from Section 4.1.3) with the DoT proxy deployed on the edge router of a small LAN. There are three main entities: (i) Router with DoT Proxy, (ii) DNS Tunnel Target, (iii) Rogue User performing DNS tunneling, and (iv) Benign Users.

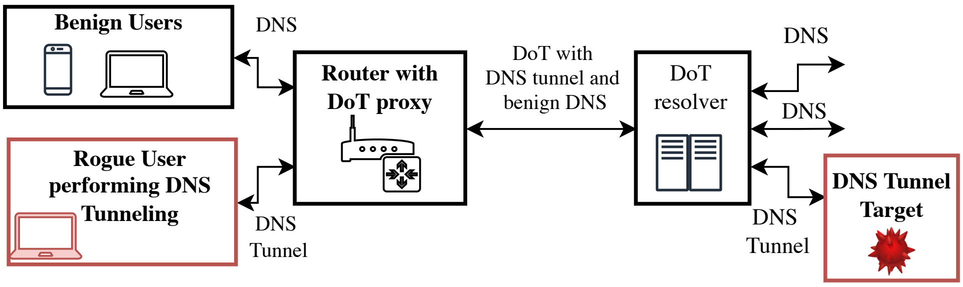


Figure 4.2: DoT tunnel measurement setup.

Router with DoT Proxy We used a router with OpenWrt[5] operating system. Since OpenWrt is Linux-based, we could install a third-party DoT proxy. Moreover, we set the router as a primary DNS resolver for connected clients.

We have installed Stubby[6] DoT proxy into the router and used it throughout all our experiments. It was operated with default configuration settings. We provided only the domain name of the used resolver as the configuration entry.

DNS Tunnel Target Represents the server side of the DNS tunnel. We registered the domain name using freenom free domain name provider[7]. We rented Virtual Private Server (VPS) with a 1 Gbps connection and set it as an authoritative DNS server for the registered domain. On the VPS, we executed the server side of the DNS tunnel. During our experiments, we used Iodine[8] and DNS2TCP[9] for tunneling.

Iodine is a well-known DNS tunneling tool. The tunnel is created on the data-link layer. Thus IP headers are also transmitted. During its start-up, Iodine created a specialized network interface that any application can use.

DNS2TCP performs tunneling on the TCP layer. Thus IP headers are not transmitted. The developers claim that contrary to IP-over-DNS approaches (such as Iodine), the lack of IP headers increases throughput. However, it cannot tunnel arbitrary traffic, and the "resource" (application listening on the tunnels server-side) needs to be specified. We

[5] We used TP-Link Archer AC1750 with `https://openwrt.org`.
[6] `https://dnsprivacy.org/dns_privacy_daemon_-_stubby/`.
[7] `https://www.freenom.com`.
[8] `https://github.com/yarrick/iodine`
[9] `https://github.com/alex-sector/dns2tcp`.

have used SSH as a resource for all of our experiments. Unlike other supported resources such as SMTP or POP3, SSH allowed us to tunnel traffic easily with various speeds and characteristics.

Rogue User Performing DNS Tunneling This entity represents the client side of the DNS tunnel. The DNS tunneling tools were executed on a Linux-based machine connected via WiFi to the *Router with DoT proxy*. Moreover, the tunneling tool was set to use a Stubby DNS proxy running on the router.

Even though Iodine is highly configurable, it also supports autodetection and selects the most suitable configuration based on the used resolver. During our measurement, we deployed Iodine in a default configuration; thus, the optimal parameters were chosen automatically.

DNS2TCP does not support autodetection, nor is it as configurable as Iodine. It allows configuration of only resolver (we used *Router with DoT proxy*), "resource" application (we used SSH), and timeout interval. The timeout interval (a maximum server's answer delay) was set to 3 seconds since it is the default value; thus, we assumed it is commonly used.

Benign Users This entity represents actual devices using the Internet and creating DNS requests to make background noise. These devices were two laptops, desktops, and four Smart Phones actively used by the users. Moreover, background traffic made the experiments more realistic, considering our setup with the DoT proxy deployed on the router.

Definition of Measured Performance Characteristics We selected four performance characteristics: (i) Tunnel stability, (ii) Packet Loss, (iii) Packet Delay, and (iv) Throughput.

Tunnel Stability It represents the time interval for which the tunnel stayed connected and was ready for use. During the measurement, we did not transfer any large volumes of data in the tunnel; instead, we only used the tunnel for C2-like communication sending short packets with only 4 B of data. The client sent a message every ten seconds, which was then immediately followed by the server's response. This measurement verified the feasibility of C2 communication. When the connection remained active for more than 360 minutes, we stopped the experiment.

Packet Loss It represents the number of packets lost in the tunnel. Since Iodine creates a tunnel on the IP layer, we could use the Ping program to measure packet loss. The packet loss was not measured for DNS2TCP since it creates a reliable transport layer.

Packet Delay It represents the Round Trip Time (RTT) of a packet transmitted via the tunnel. Similarly, as in Packet Loss, we used the Ping program to obtain these characteristics for the Iodine-based tunnel. For DNS2TCP we used TCP version of Ping[10]. Since DNS2TCP used SSH as a resource, we used port tunneling inside SSH for this measurement.

Throughput It represents the achieved throughput of the tunnel. We measured it by sending a large file (∼1 GB) via SSH using the `scp` program. Since some tunnels did not perform well and we could not transmit the whole file in a reasonable time, we always left the transmission active for at least 30 minutes. The resulting throughput was then calculated from the successfully transmitted amount of data. This measurement verified the feasibility of data exfiltration.

4.2.1.3 Experiments Execution

Altogether, the experiments took place between January 2022 and March 2022. Each resolver was tested at least three times during the day. All experiments were conducted from a single location in Prague, Czech Republic. The router was connected to the network via 30 Mbps connection, which is, according to report [21], an average connection speed in the country. Nevertheless, we assume that location and connection speed have negligible impacts on the results since the tested providers have multiple servers around the world to improve their performance.

The experiment procedure consisted of the following steps: (i) Startup of DNS tunnel server, (ii) Configuration of Stubby to use evaluated resolver, (iii) Startup of DNS tunnel client on separate Machine. (iv) Execution of performance measurement described in Section 4.2.1.2, and finally, (v) gathering the results and their interpretation.

The workflow remained unchanged for all selected DoT resolvers to maintain the comparability of results.

4.2.2 Results

We could not create a DoT tunnel through most of the selected well-known resolvers listed in Table 4.2. The tunnel was successfully created only via seven out of 16 evaluated DoT resolvers, which are written with the measured performance characteristics in Table 4.3.

Generally, Iodine was more successful in connections and outperformed DNS2TCP in all measured characteristics. Dismail offered the best throughput from all measured resolvers while being very stable. The C2-like communication was uninterrupted for the whole 360 minutes till we ended the stability experiment. Even though we achieved the highest throughput with Iodine, the DNS2TCP tunnel could not be established at all.

Google DNS also performed very well, with high throughput and high stability. The average RTT was more than 30 ms smaller than Dismal. Moreover, Google DNS is the only resolver where we successfully created a tunnel with the DNS2TCP.

[10] tcping – `https://github.com/cloverstd/tcping`.

Table 4.3: DoT tunnels performance results for resolvers, for which the tunnel was successfully established. The abbreviation in the column titles stands for: T. — Tool, E. — Tunnel was successfully established, RTT-min — minimal Round Trip Time, RTT-avg — average Round Trip Time, RTT-max — maximal Round Trip Time, RTT-std — Standard Deviation of Round Trip Times, and Through. — Throughput. The Tools abbreviations are: I — iodine, D — dns2tcp.

	T.	**E.**	**Stability**	**RTT-min**	**RTT-avg**	**RTT-max**	**RTT-std**	**Loss**	**Throug.**
Google DNS	I	**Y**	>360 min	27.9 ms	51.4 ms	3593.1 ms	147.4 ms	2.50%	176 Kbps
	D	**Y**	120 min	28.7 ms	213.4 ms	17 365.2 ms	986.4 ms	—	148 Kbps
CleanBrowsing	I	**Y**	>360 min	50.6 ms	1103.8 ms	10 821.6 ms	1489.8 ms	16.80%	7.2 Kbps
	D	**N**	—	—	—	—	—	—	—
AliDNS	I	**Y**	>360 min	161 ms	211 ms	8637.6 ms	1192 ms	57.60%	0.8 Kbps
	D	**N**	—	—	—	—	—	—	—
BlahDNS	I	**Y**	50 min	70.7 ms	618 ms	5126.6 ms	1006.5 ms	23.60%	8 Kbps
	D	**N**	—	—	—	—	—	—	—
Dismail	I	**Y**	>360 min	37.4 ms	82 ms	3345 ms	258.6 ms	2.60%	232 Kbps
	D	**N**	—	—	—	—	—	—	—
AppliedPrivacy	I	**Y**	4 min	531 ms	1241.6 ms	7161.4 ms	1198.4 ms	9.20%	—
	D	**N**	—	—	—	—	—	—	—
Quad9	I	**Y**	2 min	—	—	—	—	—	—
	D	**N**	—	—	—	—	—	—	—

The family-friendly and malware protection CleanBrowsing was also very stable when used with C2-like communication. However, the achieved throughput of only 7.2 Kbps limits the possibility for exfiltration. Similar performance was also measured via AliDNS or BlahDNS.

The tunnel created via AppliedPrivacy was very unstable. We could barely measure the RTT characteristics and packet loss with the Iodine. The throughput could not be measured at all because the tunnel collapsed when we attempted to send a large file. With Quad9, we could only measure the tunnel stability with C2-like communication. The tunnel immediately collapsed when we tried to perform RTT measurements with the ping program.

4.2.3 Response from DoT Providers

To validate our results, we emailed the tested providers, informing them about the results and asking them if they perform any DNS tunnel prevention. After one month, we got a reply only from Applied Privacy, CleanBrowsing, Quad9, and Google — through all of them, we were able to create a DoT tunnel. Unfortunately, we did not receive any reply from other service providers; thus, we could not check our results completely.

Google asked us for patience until they obtain the technical department's answer, but they did not send it even after one month of waiting. The reaction from other providers was always almost the same: They do not perform any DNS tunnel protection. Instead, they confirmed that DNS throttling is deployed to prevent the overloading of their services, resulting in reduced performance. However, our results show that even the throttled DNS tunnel could be leveraged for C2 communication.

4.2.4 Discussion

We could not establish a connection with most of the evaluated resolvers; thus, we can assume they have deployed protection against DNS tunneling. However, seven out of 16 tested resolvers can be misused for DNS tunneling, even when performed without stealthiness using Iodine. The DNS2TCP success in tunnel creation was much lower. We are not sure why the Iodine outperformed DNS2TCP in all measured characteristics. We assume that Iodine's success is caused by its autodetection feature, which tailors the configuration settings (such as maximal request size and type of request) for each resolver. However, this hypothesis needs to be further investigated, which is out of the scope of this work.

Compared to other works measuring tunnel performance over traditional DNS, DoT tunnels perform much worse. The highest observed value (232 Kbps via BlahDNS) achieved around 50% of throughput measured by Merlo et al. [87]. We assume that the performance drop is caused by the overhead created by the TCP connection since the DNS exfiltration tools are not designed for DNS via a reliable channel.

Apart from Dismail and Google DNS, most of the other resolvers performed DNS traffic throughput throttling, making the tunnel slower (around 8 Kbps); thus, less usable for sending large volumes of data. However, for five resolvers, the tunnel was very stable, and it could be used for low-throughput traffic such as C2 communication, including CleanBrowsing, which claims anti-malware protection.

Unfortunately, the tunnel created via Google DNS, the most used resolver on the CESNET network (see Section 4.2.1.1), showed very good performance and could be misused for malicious purposes such as exfiltration or long-lasting C2 channels.

Even though our experiments considered only DoT, we might expect that similar results would also be obtained via DoH and DoQ. We can see that popular resolvers (such as AdGuard or Cloudflare) perform DNS tunnel protection, but the measures deployed by Google DNS (if any) are absolutely insufficient. Given that the Google DNS market share (∼80% on CESNET network), we can conclude that encrypted DNS tunneling must be considered a serious threat till at least all well-known and established providers would deploy protections[11].

4.3 Novel DoH Abuse Scenarios

In the survey of DoH abuse in Section 4.1, we summarized known DoH abuse vectors currently used on the Internet. Nevertheless, we also want to evaluate any new threat scenarios considering DoH. Therefore, we partnered with Avast Software, a large global security company protecting hundreds of millions of endpoints. During the cooperation, we had access to a continuous feed of suspicious software, malware, and malicious website samples analyzed in a sandbox environment. The automated analysis pipeline allowed the selection of particular malware samples for further inspection. We filtered malicious samples performing DoH based on port (443/TCP) and IP addresses of known Google and

[11] Small and untrusted providers can be blocked in the networks.

Cloudflare DoH resolvers which are written in Table 4.4. We decompiled or deobfuscated the source codes of found DoH-capable samples and manually analyzed them, looking for functions processing DoH requests and responses. In some web-based samples, we spotted unconventional and unpublished use of DoH by service providers to avoid DNS-based service blocking.

Table 4.4: Used IP addresses for recognition of DoH connection during our finding of DoH-capable malicious software samples.

Resolver	IPv4	IPv6
Cloudflare	1.1.1.1,1.0.0.1, 104.16.248.248, 104.16.248.249, 104.16.249.248, 104.16.249.249	2606:4700:4700::1111, 2606:4700:4700::1001
Google	8.8.8.8,8.8.4.4	2001:4860:4860::8888, 2001:4860:4860::8844

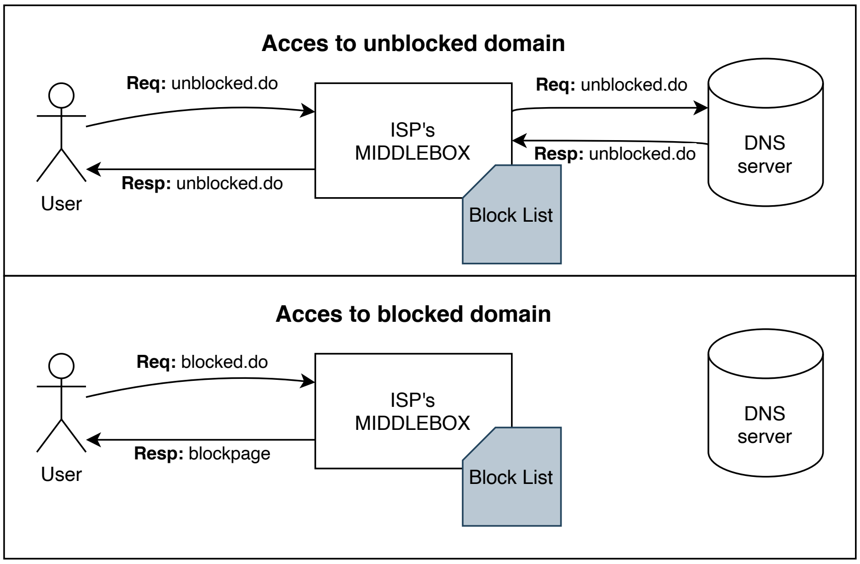


Figure 4.3: Scheme of DNS tampering procedure.

Many countries perform website censorship and blocking according to local laws. It is a common practice because our modern society considers many types of content as harmful and unacceptable. The prevention of access to some Internet resources helps to fight against child pornography, copyright infringement, and many more. There are multiple ways of implementing web content blocking [53, 37]. However, many countries implement it using DNS Tampering, i.e., a spoofed DNS answer can deny the existence of the domain name or redirect users to some block page (that can be operated by the government) with the reason of the website closure [37]. The DNS tampering procedure is depicted in Figure 4.3. Naturally, DoH effectively bypasses this blocking mechanism, which can be leveraged by service providers leaving users unaware of their illegal activity.

The rise of DoH support enabled malware authors to access an easy-to-use JSON-based DNS API through the browsers' JavaScript interpreter that can be leveraged in browser-based exploits. Specifically, multiple service providers (C2 services) were observed to take advantage of encryption and easy-to-use DNS-based C2 communication channels. All of them abuse DoH to avoid website censorship and blocking.

None of the previous studies described DoH abuse by service providers on the web, which is also a critical field related to computer security and network monitoring. Even though some of the identified threats are known or similar to traditional DNS threats, they appeared recently in the DoH domain. In this section, we present a real-world observation of their transfer into the encrypted domain, which proves an adoption of DoH abuse in web-based threats. The observations are organized in three abuse scenarios: (i) Client Modification to Access Blocked Websites, (ii) DoH in Website Redirections, and (iii) DoH Requests in Advertisements and Spam Campaigns.

4.3.1 Abuse Scenario 1: Client Modification to Access Blocked Websites

The abuse scenario assumes two entities — client and server. The client wants to communicate with the server; however, direct communication is not allowed, and its prevention is implemented by DNS tampering on the local DNS recursor. The client is modified to use DoH to bypass blocking mechanisms and obtain the working server's IP address that allows direct communication.

Even though there is almost universal support of DoH in web browsers, other types of programs still lack the support. The most straightforward modification is installing a DoH proxy that translates all local DNS requests into DoH. However, it requires much effort from the users, and we have already observed more user-friendly client modifications that use DoH only for accessing the blocked websites.

A real-world example is `sdarot.tv`, an Israeli-based website that provides video content. Due to the copyright violation, it was blocked by the Israeli government, and all local Internet service providers have to prevent access by DNS Tampering [106]. However, the website is still flourishing due to the multiple non-browser clients and their modifications. Sdarot provides a plugin written in python for the home theater software Kodi, and its short and simplified code snipped can be found below in Listings 4.1. The plugin uses *base64* encoded domain names in the translation process. After the translation, all URLs contain IP addresses directly to avoid DNS resolvers of the operating system and ISP.

Sdarot also provides Android and Android TV applications that do not use DoH. However, the applications bypass the system settings and use the Google DNS servers instead of the local DNS recursor. In addition, we analyzed the decompiled Java code, and it indeed contained code for DoH JSON-based queries. Thus, the DoH support might be enrolled soon because the simple use of some foreign DNS resolvers is already insufficient in some states [7].

Listing 4.1: DoH usage example in Sdarot Kodi plugin.

```
#b64decode domain
API = b64decode('aHR0...90LnR2')
def getIP (domain):
    #b64 encoded domain:
    #https://dns.google.com/resolve/type=A&name=
    URL='aHR0...25hbWU9'
    req = http.get(b64decode(URL)+domain);
    return json.parse(req)[IP]

def create_url(sid, season, episode):
    ipaddr = getIP(API);
    final_url = "http://"+ipaddr+"/"+sid+"/"\
    +season+"/"+episode
    return final_url
```

The Abuse Scenario 1 falls into a *Covert Multipurpose Channels* category of DoH abuse described in Section 4.1.2.

4.3.2 Abuse Scenario 2: DoH in Website Redirections

> *The abuse scenario assumes three entities – client, server, and C2 domain. The client is redirected to the server or performs willing access. On the first visit, the server modifies the client's browser by installing a redirection mechanism. Later, the server is identified as malicious, and the DNS tampering technique prevents its access. Due to the installed modification, the browser recognizes the prevention access mechanism and performs a DoH request to the C2 domain. The response contains a functional landing domain of the server that allows its access.*

During the monitoring of DoH usage in our laboratory, we found DoH requests created in web-based JavaScript by multiple websites. The websites use DoH for redirection to illegal online casinos targeting Russian citizens.

Since 2009, the gambling business has been banned in the Russian Federation, with a few gambling zones exceptions. As a result, all online casinos (even non-Russian) are prohibited in Russia. Even advertisement to gambling websites is considered illegal. The online gambling organizers risk a fine of up to 14,000 USD and website closure by the government. Despite the severe penalties, Russia's illegal gambling market is worth about 7.9 billion USD per year [83].

Online casinos are fighting the gambling ban by changing IP addresses and registering multiple domains. We have used the Security Trails Passive DNS system[12] to monitor a

[12] A system, which records the history of resolved domains and their IP addresses. URL: `https://securitytrails.com`.

domain name of a selected online casino IP address. As can be seen in Figure 4.4, more than 100 domain names point to the same website according to the Passive DNS data.

The rapid domain name-changing strategy is almost identical to malware C2 infrastructure, which uses DGA. However, the casinos depend on users who are unwilling to test the connection to hundreds of domains. Therefore, there is a redirection infrastructure in place that ensures landing on the functional unblocked casino website.

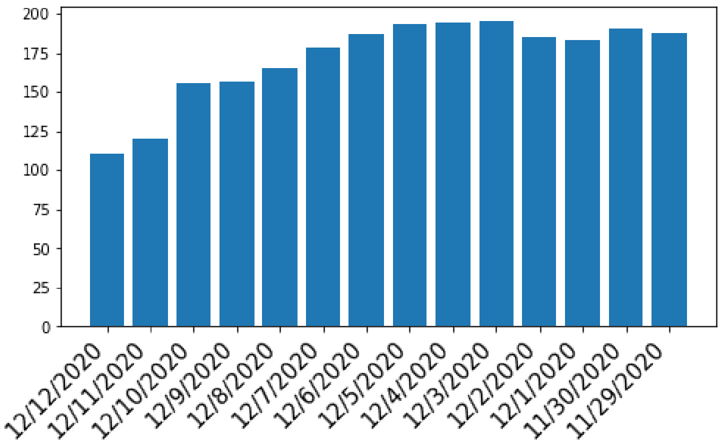


Figure 4.4: Number of unique domains pointing to the single IP address of the selected website with DoH redirection script according to Security Trails Passive DNS data.

In all observed JavaScript code samples, which performed DoH requests, the redirection occurs in the web browsers as a JavaScript Service Worker — an API that allows websites to install JavaScript code into the browser. It is like a browser plugin that can run only on domains (and all its subdomains) that installed it. When the user accesses the page, the service worker is initiated and runs in the background, separate from other websites' JavaScript code. Even though the service worker API is limited, it can register callbacks for events such as "website fetch" and modify the content similarly as a proxy.[13]

The redirector service worker is installed in the browser when the user enters the casino website. Next time, when the user wants to access, the redirector activates. In all analyzed websites, the redirector issued a DoH TXT request to a C2 domain and got a *base64* encoded JSON object. The format of the TXT answer is shown in Listings 4.2. The array contains a redirection enable flag, body substring, and the functional landing domain. The body substring distinguishes between a government block page and the actual casino webpage. It is usually a short identifier that occurs in the body tag of the webpage.

The service worker scripts in the four analyzed websites were very similar, with minor differences in function names or used API, showing that all of them were implemented separately. The example of the observed redirection script is shown in Listings 4.3. According to the instruction from C2, the service worker checks whether the domain is blocked. If not, the user proceeds to the webpage. In the other case, the user is redirected via JavaScript

[13] `https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API`.

Listing 4.2: The example of the decoded TXT answer. 1 – is the enable flag, 2 – identifier for distinguishing between block pages and the correct output, 3 – redirection domain.

```
{
    "1":1,
    "2":"VDuXmwmNZ",
    "3":"https://somedomain.com"
}
```

to the landing domain, and a new JavaScript Service Worker is installed. By this mechanism, users can remember only one URL (the first one they have visited) and are always redirected to the functional unblocked website. The whole redirection scheme is depicted in Figure 4.5.

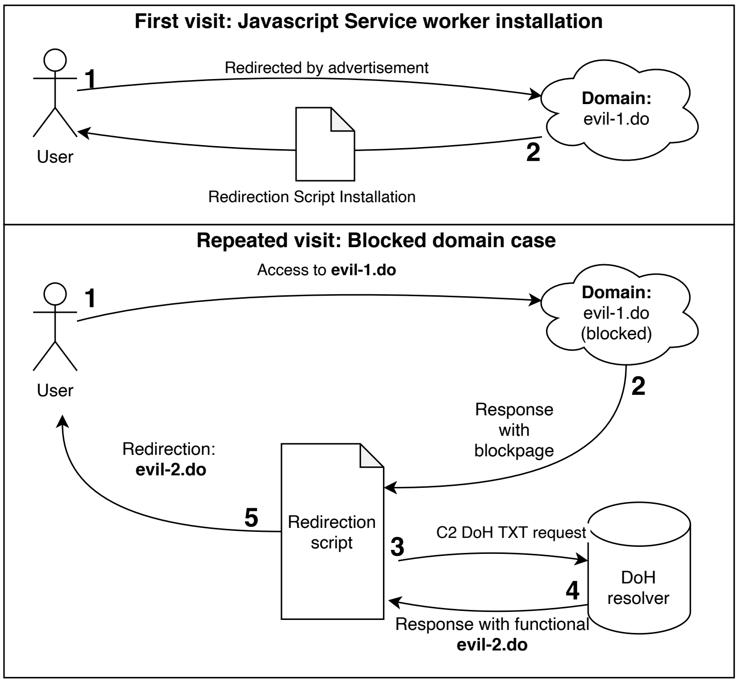

First visit: Javascript Service worker installation
User
1
Redirected by advertisement
Domain: evil-1.do
2
Redirection Script Installation
Repeated visit: Blocked domain case
Access to evil-1.do
1
User
Domain: evil-1.do (blocked)
2
Response with blockpage
Redirection: evil-2.do
5
Redirection script
3
C2 DoH TXT request
4
DoH resolver
Response with functional evil-2.do

Figure 4.5: Redirection scheme.

We have analyzed selected C2 responses in time with the Security Trails Passive DNS System[14]. All unique domain names that appeared in the responses between September 29th and November 20th in 2020 are shown in Figure 4.6. Overall, in the observed period, the landing domain name changed 35 times. It can be noticed that some of the landing domain names are very similar and differs in only a single character, which is sufficient for bypassing DNS tampering.

We have found eight different C2 domains that redirect to more than 80 websites during our research. All of them targeted the Russian market and were related to gambling.

[14] https://securitytrails.com.

Listing 4.3: DoH redirector in service worker.

```
//response arg. contains the server response
//for previous GET query
function onWebsiteFetch(response){
    [enabled, check_string, domain] = get_domain();
    if(enabled and !response.contains(check_string))
        redirect(domain);
    else
        return response;
}
function get_domain() {
    //b64encoded DoH query to C2 domain
    resp = fetch(b64decode("aHR...bWU="))
    return json.parse(b64decode(resp)).txtContent
}
```

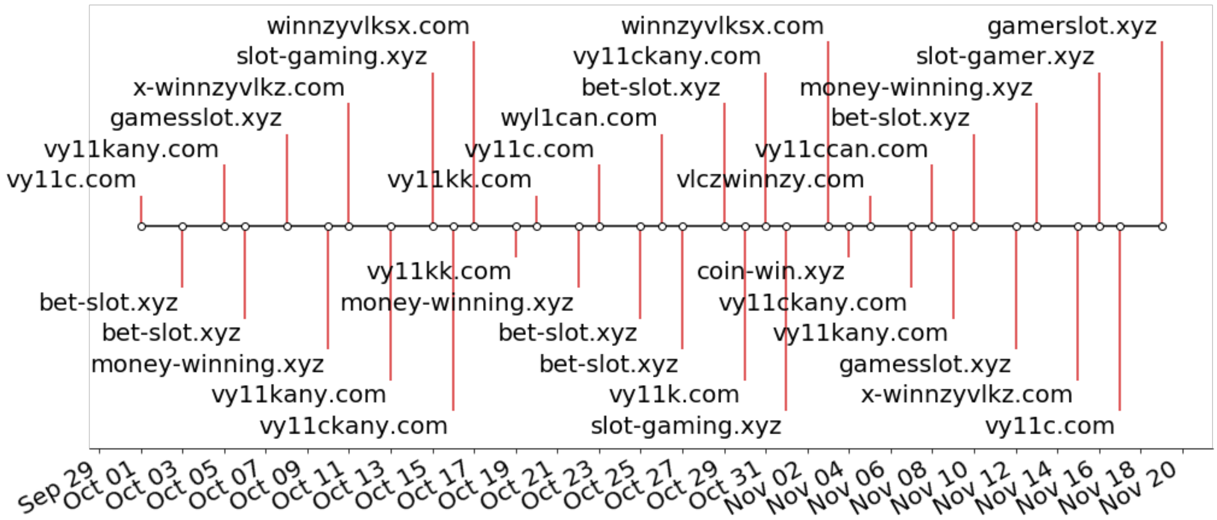


Figure 4.6: Responded landing domains in time for selected C2 between September 29th and November 20th in 2020. For data extraction we used Security Trails Passive DNS System.

However, the presented approach has enormous potential in more fields other than gambling. Besides, the presented DoH-based redirection can potentially substitute the domain fronting [46] (a technique for censorship bypass utilizing infrastructure with multiple services), which is already banned by large CDN providers [72].

The Abuse Scenario 2 falls into a *C2 Access and Communication* category of DoH abuse described in Section 4.1.1, and its mass deployment can enable a hidden web (like the dark web). Websites could change their domains and IP addresses more rapidly (in a matter of minutes) without reduced comfort for users. The only problem is the first visit, which can be performed via advertising (as described) or other services that would query the C2 domain and provide the first redirect. The state authorities are almost defenseless against this redirection principle. The C2 domain might seem like a candidate for a weak spot because its inaccessibility would cause the collapse of the whole redirection infrastructure. However, when the C2 domain is accessed solely by DoH, the access can be prevented only by the DoH resolver or by the TLD[15] operator. Even though the DoH provider can technically prevent access to a particular domain, users can always use a different one that does not perform blocking. TLD operators can perform forced domain shut-down. However, it is usually complicated to achieve.

4.3.3 Abuse Scenario 3: DoH Requests in Advertisements and Spam Campaigns

> *The abuse scenario assumes two entities – client and C2 domain. The client unwillingly initiates one or multiple DoH requests to the C2 domain – the response contains a JavaScript code or pieces of code. The client then executes the code and performs actions commanded by the C2 server.*

This scenario is observed mainly in redirection use cases, often triggered by illegal advertisements. Its usage was detected in e-mail spam campaigns; however, the same scripts can be found even on websites. All of the detected scripts utilized the same principle as in Section 4.3.2—C2 domain queried via the JSON DoH API of Google resolver; therefore, it falls into a *C2 Access and Communication* category of DoH abuse described in Section 4.1.1. However, contrary to Scenario 2, these scripts did not use JavaScript Service Worker API; instead, they fetched JavaScript source code from the C2 channel and executed it.

The samples we observed on websites received redirection JavaScript code to illegally operated web pages. The C2 communication was usually fetched right after the load or by some action, such as a button click. In the case of e-mail spam campaigns, an HTML document is delivered as an attachment (or as a MIME[16] part) and requires the mail client to open it. The pseudocode of the malicious scripts is shown in Listings 4.4. At first, the DoH TXT query to the attacker's C2 domain is performed. The attacker domain is usually encoded as a base64 string and hardcoded in the script. The DoH request can be executed directly within scripts. We also observed the utilization of public API (such as Google OAuth API), in which case the malicious code is passed as a callback function.

In the observed cases, the answer always contained the redirection script with a landing URL wrapped inside a code utilizing JavaScript *window* API. The JavaScript interpreter then executed the code and performed the redirection. Even though we observed its use only in redirection use cases, passing a JavaScript code from the C2 domain gives the attacker immense flexibility to run almost any command. Such practice could make phishing and cross-site scripting attacks more resistant because exploiting public DoH resolvers

[15] Top Level Domain.

[16] Multipurpose Internet Mail Extensions.

Listing 4.4: DoH redirector in advertisements.

```
//base64 encoded query to C2 domain
query = b64decode('aHR0...90LnR2')
function redirector(){
    // b64encoded DoH query to C2 domain
    resp = fetch(query)
    command = json.parse(resp).txtContent
    // eval(window.location.redirect(https://identification.some.domain))
    eval(command)
}
//The redirector can be directly called
redirector();
//It can utilize some API
//Example with googles OAuth API
<script src="https://accounts.google.com/o/oauth/revoke?callback=redirector()">
</script>

//Or it can be triggerd by users' action
<button onClick="redirector()"></button>
```

Table 4.5: Summary of presented abuse scenarios characteristics. The abbreviation of the abuse category stands for: CMC — Covert Multipurpose Channels (Section 4.1.2), C2 — C2 Access and Communication (Section 4.1.1).

	Scenario 1	**Scenario 2**	**Scenario 3**
Requires specialized client	✓		
DoH as hidden channel to bypass DNS	✓	✓	✓
DoH as C2 channel		✓	✓
Getting malicious code from C2			✓
Targets only Web Browsers		✓	
Website closure detection		✓	
Abuse category	CMC	C2	C2

hides them from the network traffic analysis systems, which could trigger an alarm if the JavaScript code was downloaded directly by HTTPS from a potentially suspicious domain.

Summary of Described Abuse Scenarios

The described scenarios represent working examples of mechanisms built above DoH that (i) have been observed by our malware laboratory, (ii) were not previously described in academic study, and (iii) can be very easily used for any malicious activity. Each scenario misuses DoH in a different way, Scenario 1 uses DoH to bypass restricted DNS, and Scenario 2 detects DNS tampering and website closure and then performs redirection. Moreover, Scenario 2 uses DoH as a C2 to recognize valid web pages from the block page. Scenario 3 also uses DoH as a C2; however, it uses DoH for obtaining malicious code. The differences between scenarios are also shown in Table 4.5.

4.4 Key Findings

This chapter described the existing abuse possibilities of DoH. The key findings can be summarized as:

1. There are multiple malware families leveraging DoH to hide their activity. They use it for C2 or for Covert Channels.

2. DoH created an easy-to-use JSON API that can be leveraged for C2 and Covert Channel from a web browser's JavaScript. Such practice has been observed on the Internet to avoid lawful blocking.

3. Some large DoT service providers (Google DNS, AliDNS, CleanBrowsing and more) do not block malicious DNS activity despite their content inspection possibility.

The fact that DoH and possibly other encrypted DNS approaches are already leveraged for C2 access has a significant impact on the security. Moreover, we found out that even well-established DoT providers that have access to unencrypted data do not protect the users. This chapter showed that encrypted DNS can be easily misused and thus should not be neglected by network administrators and security experts.

Chapter 5

DoH Detection and Fingerprinting Using Side-Channel Analysis

Since encrypted traffic, in general, is often challenged by statistical side-channel analysis [127] to reveal various properties about the transmitted content, we also applied these techniques to encrypted DNS, particularly DoH. As in other encrypted traffic research, side-channel analysis forms a feasible approach that would mitigate the reduced visibility and enable network IDSs to maintain security.

In Section 5.1 we describe our approach to DNS fingerprinting, which we published in [A.2] and got an award from the conference committee. Section 5.2 presents results published in [A.1], the first DoH recognition approach by traffic shape. Nevertheless, DoH recognition still has severe limitations regardless of its reliability. A possible solution for reliable DoH recognition, which is based on our approach published in [A.7], is then described in Section 5.3.

5.1 DoH Fingerprinting

Previous works studied website fingerprinting by observing just DoH or DoT connections, which showed great accuracy in the inference of visited websites. However, we were interested in an even higher level of detail from the DoH connections, such as mapping individual domains requested in encrypted channel. We thus analyzed the DoH requests and responses to infer the payload of each request.

In this section, we describe the dataset, methodology, and results of our DoH query fingerprinting approach that challenged the privacy properties of DoH. The accuracy of the DoH query fingerprinting is studied across different HTTP protocol versions and multiple browsers.

5.1.1 Datasets

Since no previous research attempted to fingerprint individual DoH queries, there was not any dataset, that would be usable for our tasks. Therefore, we created a novel dataset and made it publicly available on the Zenodo platform [68].

DoH is currently supported in almost all commonly used web browsers [91, 10]. 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 [118]. We also evaluated several Chrome-based browsers, such as Microsoft Edge and Opera, but the connections were the same as in the case of 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 significant differences in terms of packet sizes and padding. Therefore, the further text explains our experiments and analysis using only one resolver—Cloudflare (which is the default choice in multiple browsers).

To create the DoH communication datasets, we used several virtual machines with Windows and GNU/Linux operating systems. A simplified scheme is shown in Figure 5.1. We captured the traffic from the DoH-enabled web browsers using tcpdump [121]. 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 [109] (Table 5.1 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[1] 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.

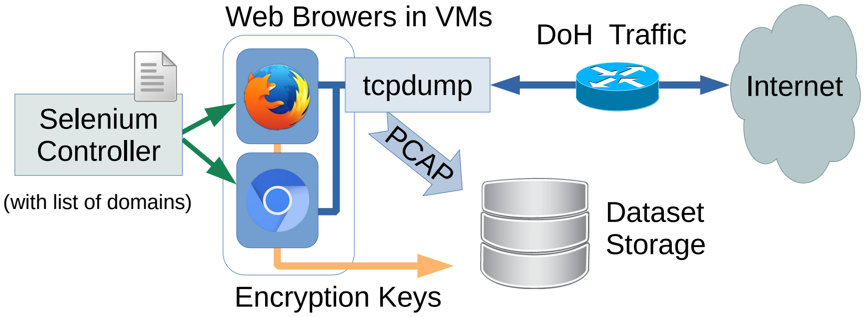


Figure 5.1: Scheme of capturing datasets using Selenium, several virtual machines with web-browsers, and tcpdump. The web browsers were forced to dump cipher keys, so the captured PCAP files can be decrypted.

Each virtual machine was configured to export TLS cryptographic keys, that was used for decrypting the traffic using Wireshark application. The encrypted content of DoH

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

Table 5.1: Versions of browsers and OS used for dataset generation. These versions were considered updated at the time of dataset creation in 2020.

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

responses was used only as a ground truth for labels at the end of the dataset preparation process.

We captured multiple datasets with different numbers 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 Table 5.2. Each dataset is composed of testing and training parts of approximately the same size. The capturing of the training and testing part was done on different days for even more realistic results.

Unfortunately, we were not able to enforce the Windows version of Firefox to use with HTTP/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. We also could not capture the traffic of the Linux version of Chrome because the DoH was still unsupported at that time.[2]

Table 5.2: Overall information about created dataset containing the number of DoH IP flows and the total number of IP flows. The abbreviations in the column names 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	1200	409
Lin-Fir-H2-50	Lin	F	2	50	230,025	2000	455
Lin-Fir-H2-70	Lin	F	2	70	356,311	2800	627
Win-Fir-H2-50	Win	F	2	50	147,839	2000	445
Win-Chr-H2-50	Win	C	2	50	37,125	2000	389
Lin-Fir-H1-30	Lin	F	1	30	110,949	1200	308
Lin-Fir-H1-50	Lin	F	1	50	186,070	2000	421
Lin-Fir-H1-70	Lin	F	1	70	272,470	2800	572
Win-Chr-H1-50	Win	C	1	50	22,787	2000	382

5.1.2 Traffic Shape of DoH

The essential step for successful DoH query fingerprinting is a deep understanding of the traffic. Therefore, we manually analyzed decrypted raw PCAP data with the DoH com-

[2] The full support of DoH in Chrome browser was from version 93 (released in July 2021, one year after our experiments).

munication, which we captured for our datasets.

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 support of EDNS padding by Google Chrome. All requests and responses coming from Chrome had the same size. At that time, Firefox did not support the EDNS padding feature. The first version of Firefox with padded encrypted DNS was version 95, which was released in December 2021, one year after our experiments.

5.1.2.1 DNS over HTTP/2

The DNS over HTTP/2 communication pattern is shown in Figure 5.2. 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 characteristic originates from stream management. Each request creates a new stream, which is then closed by the response. The queries and also 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 the 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 of DoH communication and precludes fingerprinting. However, the number of such anomalous responses is negligible.

HTTP/2 header regarding the header compression (HPACK [97]) was also identified as an important characteristic affecting fingerprinting. The header fields with nonpersistent content across all packets (such as timestamps) result in 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.

5.1.2.2 DNS over HTTP/1

The HTTP/1 is not officially recommended by [59] due to performance reasons. Moreover, performing DoH response fingerprinting is more feasible in the case of HTTP/2. By observing a single TCP connection, we are able to pair each request with an appropriate response due to the fact, that HTTP/1 does not support streams and each query is always followed by the response. Also, the DNS requests and responses are always placed in individual packets. Figure 5.3 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

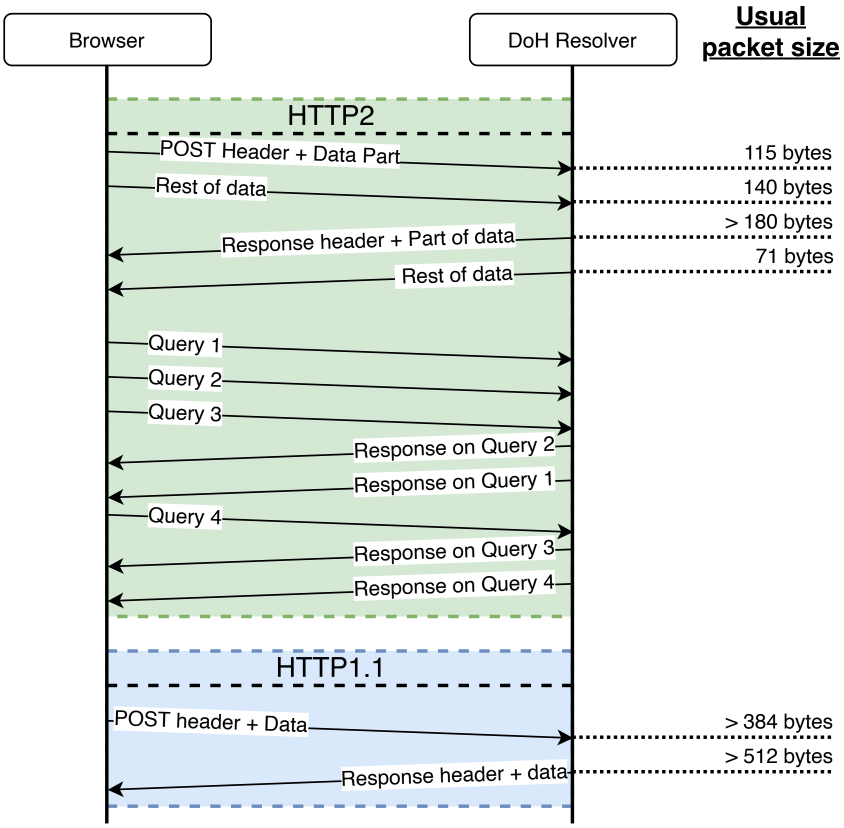


Figure 5.2: DoH communication pattern difference between HTTP/2 and HTTP/1. The abbreviations in legends stand for: H1 — HTTP/1, H2 — HTTP/2

padding. The padding effect is more noticeable in HTTP/1, where we observed only two packet sizes.

The differences among the analyzed DoH communication are clearly summarized in Table 5.3.

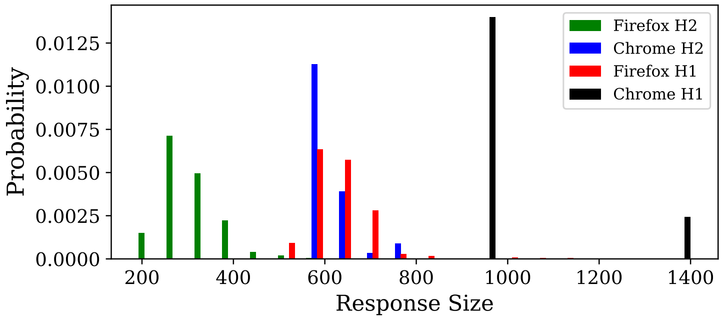


Figure 5.3: The histogram of DoH packet sizes.

Table 5.3: Summary of observed differences across analyzed browsers' DoH communication. The abbreviations in the columns label stand for: F — Firefox, C — Chrome

	HTTP/2		**HTTP/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

5.1.3 Fingerprinting Targets

During our first experiments, we observed a considerable number of DNS queries targeting and generated subdomain name[3] or subdomain name with a 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 to inferring the second-level domain (a domain name before the top-level domain), as they provide us with the most important information.

5.1.4 Feature Engineering

The website fingerprinting approaches 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 their direction.

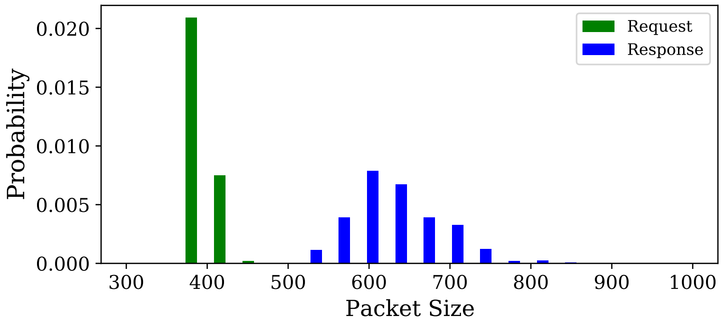


Figure 5.4: The histogram of DoH packet sizes on Firefox Linux version with HTTPS/1.

In the case of DoH, we have at least two packets associated with each DNS query—request and response. However, the shuffled order of DNS over HTTP 2 responses prevents

[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`.

pairing DNS requests with a corresponding response (see Section 5.1.2). 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 sizes, as it is shown in the Figure 5.4; Thus, the query size in the feature vector confused the classifier resulting in worse results.

Similarly to website fingerprinting, the size of transmitted packets (particularly DoH response 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 [20, 113]. The batches of DNS queries and responses are observable at the traffic level as bursts of packets in both directions.

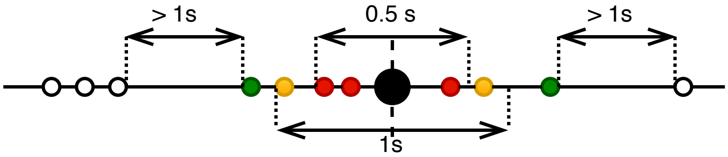


Figure 5.5: Neighborhoods of a DoH response. The black dot represents the fingerprinted DoH response. Red packets belong to *Close* neighborhood, yellow and red packets belong to *Medium* neighborhood and the green, yellow and red packets belong to *Webpage* neighborhood.

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-off between a burst and a webpage. The sizes of each neighborhood were determined experimentally and are depicted in Figure 5.5. 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 shown in Table 5.4.

Table 5.4: Calculated Mutual Information value (MI) for each extracted feature. The features with MI denoted in bold were included in our feature vector. The abbreviations in the 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

5.1.5 Algorithm Selection

We experimented with multiple supervised learning algorithms and evaluated their precision. The models such as C4.5 decision tree [102] or K-Nearest Neighbours [117] performed poorly; thus, we decided to focus on ensemble algorithms.

At first, we experimented with various stacked [130] model architectures, especially the state-of-the-art k-fingerprinting approach-based architecture. However, in our initial testing, the k-fingerprinting [55] based ensemble performed with around 40% precision. We achieved the best results using the combination of the AdaBoosted decision tree [47] and the Bagging meta-learning algorithm [18].

The AdaBoost ML algorithm sequentially learns 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 a 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 dataset overfitting.

The Hyperparameters of our classifier were set experimentally, and the most important of them are written in the Table 5.5

Table 5.5: 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
	Max. ratio of features	0.4
Bagging	Max. ratio of data	0.4
	Number of estimators	55

5.1.6 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 to 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.

5.1.7 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.

5.1.7.1 DNS Content Fingerprinting Accuracy with HTTP/2 Datasets

The detailed results of the classifier used with HTTP/2 are written in the Table 5.6. We can notice that the accuracy of our classifier on the Firefox traffic varies around 70%, which is surprisingly high. As can be seen, the classifier does not perform significantly worse with a larger number of unique web pages. The 70% accuracy and only 10% of unclassified responses might suggest that the unpadded DoH is a serious privacy threat. The classifier performs poorly on Google Chrome's padded traffic.

Table 5.6: 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 column names 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%

5.1.7.2 Classifier Precision with HTTP/1

According to our evaluation, the DoH connections without padding that use HTTP/1 are even worse for the users' privacy. In Table 5.7, 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 more web pages. However, this decrease is not linear, so it would not be substantial with larger datasets.

Table 5.7: DNS content fingerprinting accuracy with HTTP/1 datasets. The values in brackets show the ratio of class in the test part of the dataset. The abbreviations in column names 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 %

5.1.7.3 Open-World Evaluation

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. Similar results were also achieved with HTTP/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 of a potential attacker.

5.1.8 Discussion of Fingerprinting Evaluation

At first sight, the results of fingerprinting approach might seem mediocre. We also needed to perform multiple sacrifices in the evaluation methodology—multi-label approach and only SLD domain recognition. Moreover, the execution of such fingerprinting attacks would be extremely challenging, especially due to the accuracy drop in an open-world environment. Nevertheless, the accuracy proved to be high in the closed-world environment using the DoH traffic without enhanced privacy protection.

The necessity of EDNS padding seems to be essential to maintain privacy. The accuracy of non-padded queries fingerprinting is 50 percentage points higher than in the case of padded ones. The other important outcome of our experiments is that HTTP/1 is less private than HTTP/2, and the fingerprinting reached almost 90%.

5.2 DoH Detection

Since DoH does not use a specialized port, it is not straightforward to detect in the networks. As described in Chapter 4 Malware creators are upgrading their DNS-based C2 infrastructure by deploying DoH to avoid detection by network IDSs. DoH detection can thus be considered an essential task for maintaining situational awareness and security. Detection by blocklist is not reliable due to fast IP changes and incomprehensive lists (see Section 3.1). Therefore, we need to explore novel ways of DoH detection.

In the following section, we describe our approach that leverages side-channel analysis and machine learning to detect DoH. We created a novel dataset with DoH traffic, which was then used for our feature vector proposal and evaluation. Last but not least, we describe the limitations of the proposed classifier.

5.2.1 Datasets

We started to prepare our experiments on DoH recognition in early 2020. At that time, there was no publicly available dataset with DoH traffic. Therefore we created our own and made it publicly on the Zenodo platform [126].

Currently, there are only two common options for using DoH. The first option is to enable DoH in the web browser. The second way is to redirect all traditional DNS queries via a central DoH proxy, which translates DNS queries to DoH. We set up both options to produce DoH traffic, and the described scenarios are also shown in the simplified scheme for dataset creation depicted in Figure 5.6.

The left side of Figure 5.6 presents a capturing of the traffic from DoH-enabled web browsers. We installed Google Chrome and Mozilla Firefox into separate virtual machines and controlled them with the Selenium framework, which simulates the user browsing according to the predefined script. The browsers received commands to visit domains taken from Alexa's top websites list. The capturing was performed on the host by listening to the network interface of the virtual machine. This created dataset contains about 5000 web pages visited by Mozilla Firefox and about 1000 pages visited by Google Chrome.

The right side of Figure 5.6 presents a collection of DoH data using a DoH proxy. There are several DoH proxy implementations. We decided to use a DoH client developed by Cloudflare [33] — `cloudflared` — because we believed it was one of the most used solutions at the time of dataset creation. We installed the `cloudflared` software into a Raspberry Pi computer. The IP address of the Raspberry was set as the default local DNS resolver for the two independent offices at our university. This DNS resolver was a provided option by local DHCP servers, so any auto-configured device connected to the office network used this resolver by default. The Raspberry continuously captured DNS and DoH traffic created by about 20 devices, including computers, laptops, and smartphones, for around three months.

Additionally, we run several scripts to generate DNS requests to Raspberry rapidly. These scripts simulated a busy middle-sized network that generates a significant amount of DNS queries—the higher number of DoH queries in a sequence results in entirely different

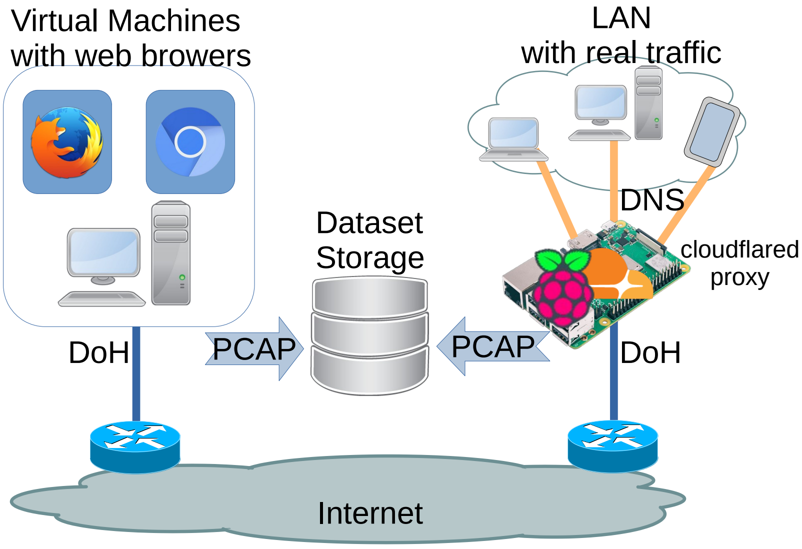


Figure 5.6: Simplified scheme of datasets creation. On the left, there is a capture of DoH traffic generated directly by web browsers captured on the machine. On the right, there is a whole LAN with several normally operating network devices and users that generate DNS; DNS is resent by the cloudflared proxy using DoH.

behavioral patterns of the connection. The aim was to add a simulated DoH traffic with as many similar parameters to ordinary HTTPS data transmission as possible. This is crucial to include in the dataset to prevent potential misclassification. The scripts generated DNS queries for domain names taken from Alexa's top websites list. We captured more than 3,845,000 DoH packets at the proxy.

We also added traffic produced by instant messaging clients (IM) into our dataset. We believe that IM traffic is the most similar to the DoH since it also follows the request-response scheme with a small amount of transmitted data.

The captured PCAP data were immediately converted into extended IP flows due to user privacy. To convert packets into extended IP Flows, we used an ipfixprobe flow exporter [24] developed by the CESNET association. More specifically, we used a particular PStats plugin, which is capable of computing additional packet-level statistics, usually called as SPLT (sequence of packet lengths and times) feature, for the first 30 packets in the flow. The resulting flows were consequently processed by Python scripts to add more computed features and annotation (ground truth) labels. The DoH labels were reliably completed according to our knowledge of proxy setup and, for manually generated traffic, according to known IP addresses of the DoH services and target HTTPS servers.

Overall, the created dataset consists of 1,128,904 flows (aggregated into bidirectional records), with around 33,000 of them labeled as DoH. To deal with the resulting imbalance between DoH and regular HTTPS classes, we used the mechanism described in Section 5.2.3. The size statistics and information about the software used in the dataset are listed in in Table 5.8.

Table 5.8: Overall information about created dataset containing the number of DoH IP Flows and the total number of IP Flows.

DoH Client + Version	Size	DoH	Total
Mozilla Firefox 73.01 Lin.	28 GB	698	523,824
Google Chrome 81.0.4044.129 Win.	8 GB	729	154,201
Cloudflared 2020.2.0 Lin.	1.5 GB	32,752	450,879

5.2.2 Analysis and Feature Selection

Feature selection is one of the most important parts because it affects the accuracy of any ML classifier. We noticed several differences from classic HTTPS traffic by looking into the raw packet data. Typical DoH connections start with a TLS handshake followed by an HTTP/2 connection preface. The rest of the communication looks like a classical request-response scheme. However, there are several differences between classical web browsing. The typical DoH connection parameters compared to other types of HTTP connections are presented in Table 5.9.

Table 5.9: The typical connection parameters of DoH compared to other types of HTTPS communications. The *Pack.* in column names stands for packets.

Name	Pack.	Bytes	Pack. A→B	Pack. B→A	Bytes A→B	Bytes B→A	Duration
DoH Firefox	55,312	7293 kB	27,822	27,490	3021 B	4271 B	2088.2 s
Facebook CDN	5893	7474 kB	996	4898	84 kB	7390 kB	164.95 s
Web Page	233	275 kB	48	185	4690 B	271 kB	5.75 s

According to our observations, a single DNS request and response has at least five packets in DoH. Therefore, we can directly mark a shorter connection as a classical HTTPS. The most significant difference between DoH and classic HTTPS is the duration of the flow. According to our observations, browsers create a single connection to the DoH server, which is then used for a longer time. During the operation, there might occur some reconnections or a completely new connection to different DoH servers; however, it does not happen very often. The longer connections can also be created by different communication, like file downloading, video streaming, and so on. However, these types of connections tend to transmit much more data in a shorter time than the DoH, ideally in the form of a continuous burst of data. This can be clearly seen in Table 5.9, where the connections to Facebook CDN are much shorter, with almost the same amount of transferred data. The row “Web Page” represents an average of typical pages from Alexa’s top websites list, i.e., from our captured dataset.

The DoH communication can also be distinguished from regular HTTP by the size of transmitted packets. In Figure 5.7, we can see that the DoH variance of response packet sizes is much lower. We can observe the same trend with the sizes of outgoing packets; however, it is less significant because HTTP requests also tend to have similar sizes.

The specific activity pattern can also reveal the DoH directly implemented in browsers. Figure 5.8 shows the example of an activity of one DoH connection, where we can see

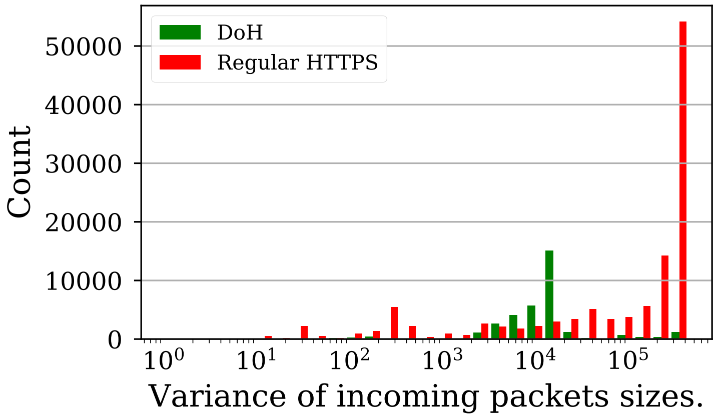


Figure 5.7: Histogram of variance of incoming packet sizes.

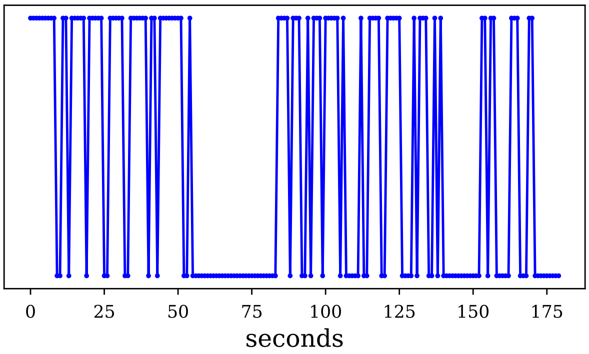


Figure 5.8: Activity in selected DoH flow record created by the Firefox web browser.

packet bursts and pauses depending on the user interaction. The number of packets inside bursts, pauses, and their ratio is included in our feature vector.

We detect a packet burst when the interpacket time is shorter than a predefined *burst threshold*. Therefore, we can count the number of packets "within a burst," which all have short interpacket times. Similarly, we detect a pause when the interpacket time is longer than a predefined *pause threshold*. We understand that the packet delays depend on a web server, user connection quality, and many other factors. Therefore, the thresholds must be considered relatively for each connection. We evaluated several HTTPS connections, and we set the *burst threshold* value as the 33.3% percentile from the inter-packet times of each connection (i.e., a bidirectional flow record). The *pause threshold* is set similarly to the 66.6% percentile.

Another identified feature represents the symmetry of the amount of incoming and outgoing data. The DNS responses, especially in DNS wireformat, have almost the same sizes as requests, and communication tends to be balanced (compared to HTTPS). We also split the sequence of packets into thirds and calculated three symmetry metrics separately. HTTPS traffic might be similar at the beginning of the connection, but it becomes strongly asymmetric later.

To measure the periodicity of the traffic, we used the autocorrelation metric as another feature. In several previous works, autocorrelation is claimed as crucial for identifying DNS traffic inside covert channels.

In total, we have identified and tested 19 traffic features. After calculating the feature importance with the Gini index, we reduced the feature list to the final 18 features for DoH recognition. All identified features are clearly outlined in Table 5.10. As seen in the table, the most important feature for DoH recognition is the duration of an IP flow. The average inter-packet delay is also essential. Surprisingly, the autocorrelation, regularly used in related work, is insignificant.

Table 5.10: Importance of the evaluated features. The features with importance typed in bold font were included in a feature vector of the corresponding usage.

Feature Name	**DoH Importance**
duration	**0.239**
minIntrPckDelay	**0.040**
maxIntrPckDelay	**0.089**
avgIntrPckDelay	**0.221**
varPktSizeIn	**0.015**
varPktSizeOut	**0.023**
bytesInoutRatio	**0.034**
pktsInoutRatio	**0.011**
avgPktSizeIn	**0.037**
avgPktSizeOut	**0.038**
medianPktSizeIn	**0.045**
medianPktSizeOut	**0.015**
burstPausesRatio	**0.049**
pktInBursts	**0.027**
pktInPauses	**0.063**
autocorrelation	**0.015**
symmetry-1thrd	**0.011**
symmetry-2thrd	0.001
symmetry-3thrd	**0.010**

5.2.3 Results

This section describes the results of our measurements of the ML-based DoH recognition using the dataset described in Section 5.1.1. Moreover, we applied methods for imbalanced learning since we do not have equally distributed classes in the dataset. Currently, applying *oversampling* and *undersampling* methods is the most common approach for dealing with imbalanced classes (e.g., according to to [79]). Specifically, we used SMOTE [25] for oversampling and NearMiss-3 [136] as an undersampling method. The ratio between the DoH class and regular HTTPS in our dataset is around 1:13. The SMOTE increases the number of minority (DoH) class samples to a ratio of 1:5. The undersampling method then reduces the number of majority (regular HTTPS) classes to the final ratio of 1:2.

The dataset balancing methods are applied only to the data given to the training phase of the algorithms (selected using the standard n-Fold cross-validation, see later) since applying it to the testing data is not recommended.

5.2.3.1 Classification Algorithm

In order to classify and recognize DoH traffic, we experimented with five ML algorithms: K-Nearest Neighbours [117] (We use 5-NN in our study), C4.5 Decision Tree [102], Random Forest [57], Naive Bayes [27], and AdaBoosted Decision Tree [47]. These algorithms are commonly used in Networking applications [16].

We used 5-Fold cross-validation to evaluate the precision of each algorithm. The hyperparameters of each algorithm were set experimentally by evaluating each parameter separately and observing the precision of the results.

The overall performance of the algorithms is very similar across all evaluated algorithms, which shows that our feature vector is robust and discriminative enough for DoH recognition and classification. The detailed results are written in Table 5.11. The Naive Bayes performs the worst in both tasks; however, its precision is still high. For further evaluations, we selected the AdaBoosted Decision tree, which has the best accuracy.

Table 5.11: Comparison of the overall precision of the evaluated ML algorithms for DoH identification in HTTPS traffic.

Algorithm Name	**Accuracy**
5-NN	99.4%
C4.5	99.4%
Random Forest	99.5%
Naive Bayes	96.8%
AdaBoosted Dec. Tree	99.6%

5.2.3.2 Detailed Evaluation of DoH Recognition

Table 5.12: Confusion matrix of DoH recognition from a regular HTTPS traffic. The table contains class accuracy and class recall for both classified classes: DoH and regular HTTPS.

		Ground Truth		**Class Accuracy**
		DoH	HTTPS	
Result	DoH	32,668	81	99.7%
	HTTPS	1511	411,791	99.6%
	Class Recall	95.5%	99.9%	

Based on the results in Section 5.2.3.1, we used AdaBoosted Decision Tree with the maximal depth set to 15 and the number of estimators set to 5—other parameters remained in their default values established in the SciKit Learn library[6]. The evaluation was done using 5-Fold cross-validation again to obtain the results. The trained model achieved an excellent result of 99.6% accuracy with an F1 score of 0.996. The detailed results are presented in the form of a confusion matrix shown in Table 5.12.

[6] `scikit-learn.org`.

5.2.3.3 Limitations

The proposed ML algorithms achieved excellent results on the created dataset. However, they also have some limitations. DoH detection is only possible with connections that contain multiple DNS queries. The proposed ML algorithm cannot recognize a DoH connection with a single query because of the similarity with another request/response API. The DoH implementation in browsers and the burst shape of packets are crucial for the correct operability of the algorithm. Therefore it is easy to mask the DoH connection from our classifier by creating new connections for each query, which would also significantly increase latency due to TLS handshake.

The attackers can also hide the use of DoH by masking the traffic shape. For example, they might synthetically create more asymmetrical connections — adding padding into DoH query packets. This type of DoH connection would be misclassified due to its similarity to the multimedia stream.

5.3 DoH Detection using Weak Indicators

The ML-based detector presented in Section 5.2 presented a first step in DoH detection that had a lot of limitations and could not be deployed into commercial-grade IDS. Unfortunately, following DoH detection studies (listed in Table 2.4) just slightly improved accuracy while still using ML-only detection on top of web-browser traffic.

Even though the number of ML-only-based detection proposals has risen in recent years, we can notice that the commercial sector is reluctant to ML-only deployment, and they are still predominantly using signature-based approaches [120]. From the network security perspective, the ML-based detectors process extensive amounts of data—even 99% of accuracy is not sufficient. Such accuracy results in one misclassification every second when deployed on a network with 100 connections per second. Since DoH detection is generally a very asymmetrical problem and the DoH class represents only a negligible fraction of all traffic, we might expect that most false classifications would be false positives. Moreover, ML-only detectors are less explainable (if any), which is an essential property for commercial IDS, since the alerts are then mostly consumed by human operators. According to Alahmadi et al. [5], alert triggers without apparent reason require extensive investigative effort, which gradually causes alarm desensitization—even professional security analysts lose their trust in the validity of the alarm resulting in ignored and unresolved alerts. Consequently, accurate IDS with unexplainable and untrustworthy alert triggers have only a limited impact on security, leading to successful intrusions or large data breaches [105].

ML-only DoH detection is not a viable approach, and we need to explore novel ways of its detection that would not rely just on ML and provide explainable and trustworthy results. In our work [A.7], we presented a concept of weak indicators for the detection of IoT malware. Nevertheless, a weak indication as a concept is general and can also be used in DoH detection.

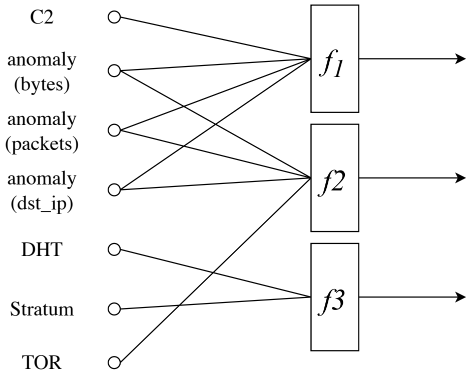


Figure 5.9: Proposed weak indicators from [A.7] and their combination for IoT malware detection.

5.3.1 Weak Indication Approach

The weak indication approach stands on the principle of weak indicators. The weak indicator can be defined as an event that might suggest the presence of targetted class that we want to detect. However, it is just an indication—it is imprecise and potentially unreliable, and we cannot make any conclusions. Nevertheless, by combining multiple weak indicators together, we can obtain accurate information that is trustworthy enough for final detection.

In [A.7], we proposed eight weak indicators targetting IoT malware detection, each of them working on a different operational principle to ensure robustness—we had ML-based, anomaly-based, and signature-based weak-indication detectors. Metaclassifiers then process the outputs of weak indicators. The proposed weak-indication detection architecture is depicted in Figure 5.9.

We use boolean functions as metaclassifiers; however, even ML-based metaclassifier could be used. Nevertheless, the boolean formulas are easily understandable and can provide reasoning for alert triggers (e.g., by describing that DHT and Stratum were seen during a short time window, which suggests the presence of peer-to-peer miners), thus saving valuable time for security personnel during the investigation.

The use of weak indicators does not just improve the explainability, but it also proved to be very effective in malware hunting. We evaluated it by using real traffic from the CESNET network and more than 100 IoT malware strains provided by anti-virus company Avast Software. The proposed weak-indication architecture achieved 100% precision and 94.286% recall due to six false negatives caused by the ML-based C2 detector.

5.3.2 Weak Indicators for DoH Detection

The weak indication principle can also be applied in the DoH detection task. Based on our thorough investigation of DoH protocol properties and its implementation on the Internet, we defined four weak indicators (depicted in Figure 5.10) of DoH usage by the client. All indicators are synchronous and work on a 30-second interval basis. Thus for each IP from

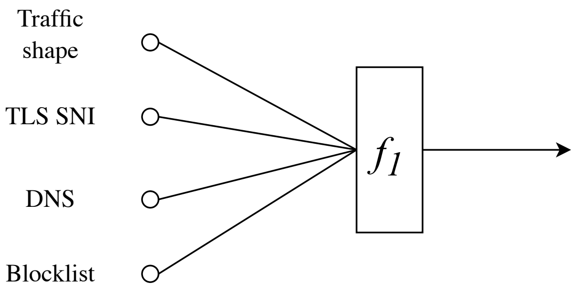


Figure 5.10: DoH detection by weak indication principle

the protected network prefix, they aggregate 30 seconds of network telemetry data (flows) and then forward the prediction to the following metaclassifier.

Traffic Shape weak indicator uses Machine Learning to recognize DoH. Machine learning is considered an unreliable source of information that only suggests that statistical properties of ongoing connection do (or do not) look like DoH. The design of this detector is going to be based on our detection model presented in Section 5.2. When a single flow in the 30-second long window is classified as DoH, the traffic shape DoH indicator outputs a positive result.

TLS SNI weak indicator performs pattern matching on domain names transmitted during TLS Handshakes. As shown in DoH blocklists [35], domain names of servers that offer DoH resolution service often contain substrings `doh` and `dns`, which are detected by this indicator. When a single flow in the 30-second long window contains targetted substrings, the TLS SNI indicator outputs a positive result.

DNS weak indicator inspects DNS queries of individual users and reports whether the user visited an IP address for which we cannot pair the corresponding DNS query/answer. When more than 10 public IP addresses are visited without a previous DNS request, the DNS weak indicator outputs a positive result. The threshold of 10 IP addresses has been evaluated experimentally. The relatively high number of IP addresses stems from the influence of local DNS caching on the endpoints.

Blocklist performs detection based on DoH resolvers blocklists. When a single destination IP address appears on the DoH blocklist, the blocklist-based weak indicator outputs a positive result.

Every 30 seconds, the weak indicators output their classification results to the metaclassifier, which we implemented as a majority boolean function—at least three out of four indicators must be positive in order to conclude DoH presence in the network.

The weak indication approach for DoH detection is still in the work-in-progress phase. The values of the parameters (30-second aggregation windows and also the minimal 10 IP addresses in the DNS weak indicator) might depend on the network and devices, and we need to explore their influence further. We already published a comprehensive collection of DoH datasets [A.5] that were created using CESNET backbone lines (which ensures the realness of the data) as well as traffic generation to 16 different providers (ensuring the comprehensiveness of the data). Nevertheless, the whole evaluation of the weak-indication approach for DoH detection is planned as future work.

5.4 Key Findings

In this chapter, we described the possible side-channel analysis of the DoH. The key findings can be summarized as:

1. The DoH used without padding is susceptible to a detailed fingerprinting attack. It allows the attacker to infer even a domain name.
2. DNS over HTTP/1 is more susceptible to fingerprinting attacks than DNS over HTTP/2
3. Detection of DoH using side-channel analysis and machine learning is very accurate. The author of this thesis was the first one who demonstrated that. Nevertheless, the ML detector's applicability is limited by the following:
   - The classifier based solely on ML is unreliable and may trigger many false-positive alerts.
   - It works well only for longer DoH connections. Single-query DoH is very hard to detect.
4. We can overcome the limitations of ML with the weak-indication principle.

Due to hard accuracy standards in the computer security area, the side-channel-based fingerprinting attack is not a viable approach to potentially detect policy violations within the encrypted DoH channel. Nevertheless, at least the detection of DoH shows much more promising results. Even though ML-based detectors still have limitations, overcoming them with a weak-indication principle is possible. Still, the side-channel analysis of DoH is nascent and cannot be used now for solving current security challenges posed by DNS encryption.

Chapter 6

Conclusions

Encrypted DNS is a relatively novel approach for maintaining users' privacy. Due to its novelty, there is a need for studies targeting its adoption across service providers, users, and, unfortunately, threat actors. The main focus of this thesis has been on DoH protocol since its mass deployment has the most extensive impact in the cybersecurity field since even its recognition in the network is challenging. Therefore this thesis focused on three main aspects of DoH. The adoption across the service providers and also across users was studied. Moreover, we researched the encrypted DNS abuse possibilities. Lastly, we focused on side-channel analysis and DoH detection.

6.1 Summary

The results of our measurements show significant and rising adoption of encrypted DNS by users and also by DNS service providers. Therefore, the impact of encrypted DNS on network security is already large. Moreover, we might expect more severe security implications from mass encrypted DNS usage in the future. Users and threat actors are becoming more familiar with encrypted DNS and thus can leverage the increased privacy to bypass DNS-based policy-enforcement tools in restricted networks. Moreover, we also described that even web-based services intentionally deploy encrypted DNS to bypass the DNS-based blocking, leaving users unaware of possible policy violations. Our research also showed that DoH providers are restrained in active DNS blocking of malicious content; thus, we cannot rely on them. Due to these facts, network administrators and security managers should not rely on information from traditional DNS to force network policies and maintain security, instead, they need to use more privacy-intrusive technologies such as deciphering proxies.

Nevertheless, in corporate or university networks, without the possibility of deploying a deciphering proxy, network administrators can prevent encrypted DNS abuse by blocking ports assigned to encrypted DNS, such as 853/TCP for DoT, and forcing the clients to downgrade to traditional DNS. In the case of DoH, we must use an ML-based detector, described in this thesis, to recognize DoH from other regular HTTPS. Nevertheless, ML-based detectors cannot be 100% reliable; thus, further research in DoH recognition using weak-indicator is necessary to achieve reliable DoH detection.

All of our results have been presented and discussed within the scientific community. Two papers [A.6, A.2] have been awarded by the conference committee. Together we published our results relevant to this thesis in four conferences (including the CORE A conference) and three journals (including a journal with Impact Factor=11.043). A lot of the works already received a considerable amount of citations, and we were also invited to write blog posts on APNIC and perform presentations for the general public.

6.2 Contributions of the Dissertation Thesis

1. We mapped the adoption of encrypted DNS across service providers and users, showing its increasing trend. Moreover, we revealed multiple properties of DoH resolvers operating on the internet, such as their association with as a source of malware binaries.

2. We also surveyed the DoH use by threat actors showing the transition of malicious software into the encrypted domain. Moreover, we demonstrated and measured the quality properties of encrypted DNS tunnels via multiple DoT providers.

3. We described DoH abuse by a web-based application that leverages DoH as an API for their C2 infrastructure to bypass governmental closure.

4. We challenged DoH with side-channel analysis showing privacy weaknesses of DNS over HTTP/1 and the necessity of EDNS padding.

5. We developed an ML-based DoH detection approach that achieved excellent accuracy of 99.6% on the created dataset. Moreover, we outlined the way to make DoH detection more accurate, robust, and explainable by using a weak indication approach.

6.3 Future Work

The author of the dissertation thesis suggests exploring the following, which are already out of the scope of this thesis:

- It would be interesting to perform several additional encrypted DNS adoption measures to show its progress over a long period of time.

- Following the study of DoH-capable malicious PoC and DoH-capable malware would be interesting to study the adoption across threat actors.

- Comparison of all DoH detection approaches on a single dataset would be interesting to reveal their properties and establish the best DoH detection approach.

- The real-world performance of a weak-indication approach for DoH detection needs to be measured to reveal if the weak indicators are the right approach to design network detectors. The design of accurate DoH detectors is essential for network security. Currently, it is challenging to map DoH-capable malware since we cannot reliably recognize DoH.

- The protective properties of encrypted DNS resolvers should be explored more. The possibility of data exfiltration via protective versions of resolvers is surprising and needs to be evaluated and communicated to the public. Assuming blocklist-based protection, it would be interesting to review the timeliness of used blocklist and their precision.

Bibliography

[1] 360 Netlab. Pink, a botnet that competed with the vendor to control the massive infected devices [online]. jul 2021. URL: https://blog.netlab.360.com/pink-en/.

[2] AdGuard software Limited. Adguard known dns providers [online]. (Accessed on 25/05/2021). URL: https://kb.adguard.com/en/general/dns-providers.

[3] AhaDNS. Dns over https (doh) [online]. URL: https://ahadns.com/dns-over-https/.

[4] P. Aitken. Specification of the IP flow information export (IPFIX) protocol for the exchange of flow information, September 2013. doi:10.17487/rfc7011.

[5] Bushra Alahmadi, Louise Axon, and Ivan Martinovic. 99% false positives: A qualitative study of SOC analysts' perspectives on security alarms. In *31st USENIX Security Symposium (USENIX Security 22)*, Boston, MA, August 2022. USENIX Association. URL: https://www.usenix.org/conference/usenixsecurity22/presentation/alahmadi.

[6] Rafa Alenezi and Simone A. Ludwig. Classifying DNS Tunneling Tools For Malicious DoH Traffic. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–9, 2021. doi:10.1109/SSCI50451.2021.9660136.

[7] Anonymous. The collateral damage of internet censorship by dns injection. *ACM SIGCOMM Computer Communication Review*, 42(3):21–27, Jun 2012. doi:10.1145/2317307.2317311.

[8] Arno0x. Arno0x/dnsexfiltrator [online]. Dec 2020. URL: https://github.com/Arno0x/DNSExfiltrator.

[9] Raj Badhwar. *Domain Name System (DNS) Security*, pages 207–212. Springer International Publishing, Cham, 2021. doi:10.1007/978-3-030-75354-2_24.

[10] Kenji Baheux. A safer and more private browsing experience with secure dns [online]. May 2020.

[11] Yaser M Banadaki. Detecting Malicious DNS over HTTPS Traffic in Domain Name System using Machine Learning Classifiers. *Journal of Computer Sciences and Applications*, 8(2):46–55, 2020.

[12] Matthew Behnke, Nathan Briner, Drake Cullen, Katelynn Schwerdtfeger, Jackson Warren, Ram Basnet, and Tenzin Doleck. Feature engineering and machine learning model comparison for malicious activity detection in the dns-over-https protocol. *IEEE Access*, 9:129902–129916, 2021.

[13] Hamad Binsalleeh, A Mert Kara, Amr Youssef, and Mourad Debbabi. Characterization of covert channels in DNS. In *2014 6th International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5. IEEE, 2014.

[14] Kevin Borgolte, Tithi Chattopadhyay, Nick Feamster, Mihir Kshirsagar, Jordan Holland, Austin Hounsel, and Paul 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.

[15] Timm Böttger, Felix Cuadrado, Gianni Antichi, Eder Leão Fernandes, Gareth Tyson, Ignacio Castro, and Steve Uhlig. An Empirical Study of the Cost of DNS-over-HTTPS. In *Proceedings of the Internet Measurement Conference*, IMC '19, page 15–21, New York, NY, USA, 2019. Association for Computing Machinery. `doi:10.1145/3355369.3355575`.

[16] Raouf Boutaba, Mohammad A. Salahuddin, Noura Limam, Sara Ayoubi, Nashid Shahriar, Felipe Estrada-Solano, and Oscar M. Caicedo. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *Journal of Internet Services and Applications*, 9(1), Jun 2018. URL: `http://dx.doi.org/10.1186/s13174-018-0087-2`, `doi:10.1186/s13174-018-0087-2`.

[17] Ilker Nadi Bozkurt, Anthony Aguirre, Balakrishnan Chandrasekaran, P. Brighten Godfrey, Gregory Laughlin, Bruce Maggs, and Ankit Singla. Why is the internet so slow?! In Mohamed Ali Kaafar, Steve Uhlig, and Johanna Amann, editors, *Passive and Active Measurement*, pages 173–187, Cham, 2017. Springer International Publishing.

[18] Leo Breiman. Bagging predictors. *Mach. Learn.*, August 1996. `doi:10.1023/A:1018054314350`.

[19] K. Bumanglag and H. Kettani. On the Impact of DNS Over HTTPS Paradigm on Cyber Systems. In *2020 3rd International Conference on Information and Computer Technologies (ICICT)*, pages 494–499, 2020. `doi:10.1109/ICICT50521.2020.00085`.

[20] Jonas Bushart and Christian Rossow. Padding ain't enough: Assessing the privacy guarantees of encrypted DNS. In *10th USENIX Workshop on Free and Open Communications on the Internet (FOCI 20)*. USENIX Association, August 2020.

[21] Cable.co. Worldwide broadband speed league 2021 [online]. 2021. URL: `https://www.cable.co.uk/broadband/speed/worldwide-speed-league/#regions`.

[22] Patricia Callejo, Rubén Cuevas, Narseo Vallina-Rodriguez, and Angel Cuevas Rumin. Measuring the Global Recursive DNS Infrastructure: A View From the Edge. *IEEE Access*, 7:168020–168028, 2019. `doi:10.1109/ACCESS.2019.2950325`.

[23] Tomas Cejka, Zdenek Rosa, and Hana Kubatova. Stream-wise detection of surreptitious traffic over DNS. In *2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Athens, Greece, 2014. IEEE. `doi:10.1109/CAMAD.2014.7033254`.

[24] CESNET, a.l.e. ipfixprobe - IPFIX flow exporter [online]. 2022. URL: `https://github.com/CESNET/ipfixprobe`.

[25] Nitesh Chawla, Kevin Bowyer, Lawrence Hall, and W. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)*, 16:321–357, 01 2002. `doi:10.1613/jair.953`.

[26] Rishabh Chhabra, Paul Murley, Deepak Kumar, Michael Bailey, and Gang Wang. Measuring DNS-over-HTTPS Performance around the World. In *Proceedings of the 21st ACM Internet Measurement Conference*, IMC '21, page 351–365, New York, NY, USA, 2021. Association for Computing Machinery. `doi:10.1145/3487552.3487849`.

[27] C. K. Chow. An optimum character recognition system using decision functions. *IRE Transactions on Electronic Computers*, EC-6(4):247–254, 1957.

[28] Catalin Cimpanu. Apple adds support for encrypted dns (doh and dot) [online]. 2020. URL: `https://www.zdnet.com/article/apple-adds-support-for-encrypted-dns-doh-and-dot/`.

[29] Catalin Cimpanu. Iranian hacker group becomes first known APT to weaponize DNS-over-HTTPS (DoH) [online]. Dec 2020. URL: `https://www.zdnet.com/article/iranian-hacker-group-becomes-first-known-apt-to-weaponize-dns-over-https-doh/`.

[30] Cisco 2016 annual cybersecurity report [online]. Dec 2020. URL: `http://mkto.cisco.com/rs/564-whv-323/images/cisco-asr-2016.pdf`.

[31] CleanBrowsing. Free dns content filtering [online]. 2022. URL: `https://cleanbrowsing.org/filters/`.

[32] Cloudflare, Inc. Dns over https — using json [online]. URL: https://developers.cloudflare.com/1.1.1.1/encryption/dns-over-https/make-api-requests/dns-json/.

[33] Cloudflare, Inc. Cloudflared proxy [online]. Dec 2020. URL: https://github.com/cloudflare/cloudflared.

[34] Levente Csikor, Himanshu Singh, Min Suk Kang, and Dinil Mon Divakaran. Privacy of dns-over-https: Requiem for a dream? In *2021 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 252–271, 2021. doi:10.1109/EuroSP51992.2021.00026.

[35] curl DNS over HTTPS. (Accessed on 25/05/2021). URL: https://github.com/curl/curl/wiki/DNS-over-HTTPS.

[36] Casey Deccio and Jacob Davis. Dns privacy in practice and preparation. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, CoNEXT '19, page 138–143, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3359989.3365435.

[37] Ronald Deibert, John Palfrey, Rafal Rohozinski, and Jonathan Zittrain. *Access denied: The practice and policy of global internet filtering.* The MIT Press, 2008.

[38] John Dickinson, Sara Dickinson, Ray Bellis, Allison Mankin, and Duane Wessels. DNS Transport over TCP - Implementation Requirements. RFC 7766, March 2016. URL: https://www.rfc-editor.org/info/rfc7766, doi:10.17487/RFC7766.

[39] Sara Dickinson, Daniel Kahn Gillmor, and Tirumaleswar Reddy.K. Usage Profiles for DNS over TLS and DNS over DTLS. RFC 8310, March 2018. URL: https://www.rfc-editor.org/info/rfc8310, doi:10.17487/RFC8310.

[40] C. J. Dietrich, C. Rossow, F. C. Freiling, H. Bos, M. v. Steen, and N. Pohlmann. On Botnets That Use DNS for Command and Control. In *2011 Seventh ECCND*, pages 9–16, 2011. doi:10.1109/EC2ND.2011.16.

[41] Trinh Viet Doan, Irina Tsareva, and Vaibhav Bajpai. Measuring DNS over TLS from the edge: Adoption, reliability, and response times. In *Passive and Active Measurement*, pages 192–209. Springer International Publishing, 2021. doi:10.1007/978-3-030-72582-2_12.

[42] Erik Ekman, Bjorn Andersson, and Anne Bezemer. yarrick/iodine [online]. Dec 2020. URL: https://github.com/yarrick/iodine/tree/iodine-0.7.

[43] Wendy Ellens, Piotr Żuraniewski, Anna Sperotto, Harm Schotanus, Michel Mandjes, and Erik Meeuwissen. Flow-Based Detection of DNS Tunnels. In Guillaume Doyen, Martin Waldburger, Pavel Čeleda, Anna Sperotto, and Burkhard Stiller, editors,

Emerging Management Mechanisms for the Future Internet, pages 124–135, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[44] Andy Fidler, Bert Hubert, Jason Livingood, Jim Reid, and Nicolai Leymann. DNS over HTTPS (DoH) Considerations for Operator Networks. Internet-Draft draft-reid-doh-operator-00, Internet Engineering Task Force, March 2019. Work in Progress. URL: `https://datatracker.ietf.org/doc/html/draft-reid-doh-operator-00`.

[45] David Fifield. dnstt [online]. 2020. URL: `https://www.bamsoftware.com/software/dnstt/index.html`.

[46] David Fifield, Chang Lan, Rod Hynes, Percy Wegmann, and Vern Paxson. Blocking-resistant communication through domain fronting. *Proceedings on Privacy Enhancing Technologies*, 2015(2):46–64, 01 Jun. 2015. URL: `https://content.sciendo.com/view/journals/popets/2015/2/article-p46.xml`, `doi:https://doi.org/10.1515/popets-2015-0009`.

[47] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *ICML*, 1996.

[48] Sebastián García, Tomáš Čejka, and Veronica Valeros. Dataset of DNS over HTTPS (DoH) Internet Servers, 2021. `doi:10.17632/ny4m53g6bw.2`.

[49] Lionel F. Gonzalez Casanova and Po-Chiang Lin. Generalized Classification of DNS over HTTPS Traffic with Deep Learning. In *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1903–1907, 2021.

[50] R Graham. Masscan: the entire internet in 3 minutes [online]. 2013. URL: `http://blog.erratasec.com/2013/09/masscan-entire-internet-in-3-minutes.html`.

[51] Christian Grothoff, Matthias Wachs, Monika Ermert, and Jacob Appelbaum. Toward secure name resolution on the internet. *Computers & Security*, 77:694–708, August 2018. `doi:10.1016/j.cose.2018.01.018`.

[52] Saikat Guha and Paul Francis. Identity trail: Covert surveillance using dns. In Nikita Borisov and Philippe Golle, editors, *Privacy Enhancing Technologies*, pages 153–166, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. `doi:10.1007/978-3-540-75551-7\_10`.

[53] Joseph Lorenzo Hall, Michael D. Aaron, Stan Adams, Ben Jones, and Nick Feamster. A Survey of Worldwide Censorship Techniques. Internet-Draft draft-hall-censorship-tech-07, Internet Engineering Task Force, March 2019. Work in Progress. URL: `https://datatracker.ietf.org/doc/html/draft-hall-censorship-tech-07`.

[54] John Hammond. Hiding in plain sight —— part 2 [online]. Sep 2020. URL: `https://blog.huntresslabs.com/hiding-in-plain-sight-part-2-dfec817c036f`.

[55] Jamie Hayes and George Danezis. k-fingerprinting: A robust scalable website fingerprinting technique. In *25th USENIX Security Symposium*, pages 1187–1203, 2016.

[56] Dominik Herrmann, Christian Banse, and Hannes Federrath. Behavior-based tracking: Exploiting characteristic patterns in dns traffic. *Computers & Security*, 39:17–33, 2013. 27th IFIP International Information Security Conference. URL: `https://www.sciencedirect.com/science/article/pii/S0167404813000576`, `doi:https://doi.org/10.1016/j.cose.2013.03.012`.

[57] Tin Kam Ho. Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, ICDAR '95, page 278, USA, 1995. IEEE Computer Society.

[58] Paul E. Hoffman. Representing DNS Messages in JSON. RFC 8427. URL: `https://rfc-editor.org/rfc/rfc8427.txt`, `doi:10.17487/RFC8427`.

[59] Paul E. Hoffman and Patrick McManus. DNS Queries over HTTPS (DoH). RFC 8484, October 2018. URL: `https://rfc-editor.org/rfc/rfc8484.txt`, `doi:10.17487/RFC8484`.

[60] Austin Hounsel, Kevin Borgolte, Paul Schmitt, Jordan Holland, and Nick Feamster. *Comparing the Effects of DNS, DoT, and DoH on Web Performance*, page 562–572. Association for Computing Machinery, New York, NY, USA, 2020. URL: `https://doi.org/10.1145/3366423.3380139`.

[61] Austin Hounsel, Paul Schmitt, Kevin Borgolte, and Nick Feamster. Can Encrypted DNS Be Fast? In Oliver Hohlfeld, Andra Lutu, and Dave Levin, editors, *Passive and Active Measurement*, pages 444–459, Cham, 2021. Springer International Publishing.

[62] Rebekah Houser, Zhou Li, Chase Cotton, and Haining Wang. An investigation on information leakage of dns over tls. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, CoNEXT '19, page 123–137, New York, NY, USA, 2019. Association for Computing Machinery. `doi:10.1145/3359989.3365429`.

[63] Zi Hu, Liang Zhu, John Heidemann, Allison Mankin, Duane Wessels, and Paul E. Hoffman. Specification for DNS over Transport Layer Security (TLS). RFC 7858, May 2016. (Accessed on 05/25/2021). URL: `https://rfc-editor.org/rfc/rfc7858.txt`, `doi:10.17487/RFC7858`.

[64] Qing Huang, Deliang Chang, and Zhou Li. A Comprehensive Study of DNS-over-HTTPS Downgrade Attack. In *10th USENIX Workshop on Free and Open Communications on the Internet (FOCI 20)*, 2020.

[65] Christian Huitema, Sara Dickinson, and Allison Mankin. DNS over Dedicated QUIC Connections. RFC 9250, May 2022. URL: `https://www.rfc-editor.org/info/rfc9250`, `doi:10.17487/RFC9250`.

[66] Christian Huitema, Allison Mankin, and Sara Dickinson. Specification of DNS over Dedicated QUIC Connections. Internet-Draft draft-ietf-dprive-dnsoquic-00, Internet Engineering Task Force, 04 2020. Work in Progress. URL: `https://datatracker.ietf.org/doc/draft-ietf-dprive-dnsoquic/00/`.

[67] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[68] Karel Hynek and Tomas Cejka. Dataset used for fingerprinting of DNS over HTTPS responses., Sep 2020. `doi:10.5281/zenodo.4039587`.

[69] Karel Hynek, Sebastián García, Joaquín Bogado, Tomas Cejka, Dmitrii Vekshin, and Armin Wasicek. Dataset of dns over https (doh) internet servers [online]. May 2022. `doi:10.5281/zenodo.6517360`.

[70] Tommy Jensen. Windows Insiders can now test DNS over HTTPS [online]. May 2020. URL: `https://techcommunity.microsoft.com/t5/networking-blog/windows-insiders-can-now-test-dns-over-https/ba-p/1381282`.

[71] Kamil Jerabek, Ondrej Rysavy, and Ivana Burgetova. Measurement and characterization of DNS over HTTPS traffic, 2022. URL: `https://arxiv.org/abs/2204.03975`, `doi:10.48550/ARXIV.2204.03975`.

[72] Alex Johnson. Domain fronting: Making backdoor access look like google requests, 2018. URL: `https://www.cs.tufts.edu/comp/116/archive/spring2018/ajohnson.pdf`.

[73] Eric Kinnear, Patrick McManus, Tommy Pauly, Tanya Verma, and Christopher A. Wood. Oblivious DNS over HTTPS. RFC 9230, June 2022. URL: `https://www.rfc-editor.org/info/rfc9230`, `doi:10.17487/RFC9230`.

[74] Amit Klein and Benny Pinkas. DNS cache-based user tracking. In *Proceedings 2019 Network and Distributed System Security Symposium*. Internet Society, 2019. `doi:10.14722/ndss.2019.23186`.

[75] Erik Kline and Ben Schwartz. Dns over tls support in android p developer preview [online]. Apr 2018. URL: `https://android-developers.googleblog.com/2018/04/dns-over-tls-support-in-android-p.html`.

[76] Mike Kosek, Trinh Viet Doan, Malte Granderath, and Vaibhav Bajpai. One to rule them all? a first look at DNS over QUIC. In *Passive and Active Measurement*, pages 537–551. Springer International Publishing, 2022. `doi:10.1007/978-3-030-98785-5_24`.

[77] Carmen Kwan, Paul Janiszewski, Shela Qiu, Cathy Wang, and Cecylia Bocovich. Exploring simple detection techniques for dns-over-https tunnels. In *Proceedings of*

the ACM SIGCOMM 2021 Workshop on Free and Open Communications on the Internet, pages 37–42, 2021.

[78] Jacobs Leon. Waiting for godoh [online]. Dec 2020. URL: `https://sensepost.com/blog/2018/waiting-for-godoh/`.

[79] Octavio Loyola-González, Milton García-Borroto, Miguel Medina-Pérez, José Francisco Martínez-Trinidad, Jesús Carrasco-Ochoa, and Guillermo De Ita. An empirical study of oversampling and undersampling methods for lcmine an emerging pattern based classifier. In *Lecture Notes in Computer Science*, volume 7914, pages 264–273, 06 2013. `doi:10.1007/978-3-642-38989-4_27`.

[80] Chaoyi Lu, Baojun Liu, Zhou Li, Shuang Hao, Haixin Duan, Mingming Zhang, Chunying Leng, Ying Liu, Zaifeng Zhang, and Jianping Wu. An end-to-end, large-scale measurement of DNS-over-encryption. In *Proceedings of the Internet Measurement Conference*. ACM, October 2019. `doi:10.1145/3355369.3355580`.

[81] Gordon Fyodor Lyon. *Nmap network scanning: The official Nmap project guide to network discovery and security scanning*. Insecure. Com LLC (US), 2008.

[82] M. MontazeriShatoori et al. Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic. In *2020 IEEE (DASC/PiCom/CBDCom/CyberSciTech)*, pages 63–70, 2020. `doi:10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00026`.

[83] Virve Marionneau. Gambling in Russia: policies, markets, and research. *International Journal of Russian Studies*, 9(2):122–134, 2020.

[84] Alexander Mayrhofer. The EDNS(0) Padding Option. RFC 7830, May 2016. URL: `https://www.rfc-editor.org/info/rfc7830`, `doi:10.17487/RFC7830`.

[85] Enock S. Mbewe and Josiah Chavula. On QoE Impact of DoH and DoT in Africa: Why a User's DNS Choice Matters. In Rafik Zitouni, Amreesh Phokeer, Josiah Chavula, Ahmed Elmokashfi, Assane Gueye, and Nabil Benamar, editors, *Towards new e-Infrastructure and e-Services for Developing Countries*, pages 289–304, Cham, 2021. Springer International Publishing.

[86] Patrick McManus. Firefox nightly secure dns experimental results [online]. Aug 2018. URL: `https://blog.nightly.mozilla.org/2018/08/28/firefox-nightly-secure-dns-experimental-results/`.

[87] Alessio Merlo, Gianluca Papaleo, Stefano Veneziano, and Maurizio Aiello. A comparative performance evaluation of DNS tunneling tools. In *Computational Intelligence in Security for Information Systems*, pages 84–91. Springer Berlin Heidelberg, 2011. `doi:10.1007/978-3-642-21323-6_11`.

[88] Leszek Miś. In & out – the network data exfiltration techniques training [online]. Dec 2020. URL: https://44con.com/44con-training/inout-the-network-data-exfiltration-techniques-training/.

[89] Paul Mockapetris. Domain names - implementation and specification. RFC 1035 (Internet Standard). URL: https://www.rfc-editor.org/rfc/rfc1035.txt, doi:10.17487/RFC1035.

[90] Paul Mockapetris. Domain names - concepts and facilities. RFC 1034, November 1987. URL: https://www.rfc-editor.org/info/rfc1034, doi:10.17487/RFC1034.

[91] Mozilla Foundation. Firefox DNS-over-HTTPS [online]. 2020. URL: https://support.mozilla.org/en-US/kb/firefox-dns-over-https.

[92] Rizwan Mushtaq. Augmented dickey fuller test. *Econometrics: Mathematical Methods & Programming eJournal*, 2011.

[93] J. Nazario and T. Holz. As the net churns: Fast-flux botnet observations. In *2008 3rd International Conference on Malicious and Unwanted Software (MALWARE)*, pages 24–31, 2008. doi:10.1109/MALWARE.2008.4690854.

[94] NetSTAR Inc. Netstar url/ip lookup [online]. (Accessed on 15/05/2022). URL: https://incompass-branch.netstar-inc.com/urlsearch.

[95] Lucas Nussbaum. lnussbaum/tuns [online]. Dec 2020. URL: https://github.com/lnussbaum/tuns.

[96] Vern Paxson. Bro: A system for detecting network intruders in real-time. *Comput. Netw.*, 31(23–24):2435–2463, December 1999. doi:10.1016/S1389-1286(99)00112-7.

[97] Roberto Peon and Herve Ruellan. HPACK: Header Compression for HTTP/2. RFC 7541, May 2015. URL: https://www.rfc-editor.org/info/rfc7541, doi:10.17487/RFC7541.

[98] Tomás A. Peña. *A Deep Learning Approach to Detecting Covert Channels in the Domain Name System*. PhD thesis, Capitol Technology University, 2020.

[99] Tadeusz Pietraszek. DNScat [online]. Dec 2020. URL: http://tadek.pietraszek.org/projects/DNScat/.

[100] Cheng Qi, Xiaojun Chen, Cui Xu, Jinqiao Shi, and Peipeng Liu. A bigram based real time DNS tunnel detection approach. *Procedia Computer Science*, 17:852–860, 2013. doi:10.1016/j.procs.2013.05.109.

[101] Quad9 Foundation. Doh with quad9 dns servers [online]. URL: https://www.quad9.net/news/blog/doh-with-quad9-dns-servers/.

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

[103] Roxana Radu and Michael Hausding. Consolidation in the dns resolver market – how much, how fast, how dangerous? *Journal of Cyber Policy*, 5(1):46–64, 2020. doi:10.1080/23738871.2020.1722191.

[104] Gadi Evron Randal Vaughn. DNS amplification attacks [online]. Mar 2006. URL: http://web.archive.org/web/20110717124634/http://www.isotf.org/news/DNS-Amplification-Attacks.pdf.

[105] Michael Riley, Ben Elgin, Dune Lawrence, and Carol Matlack. Missed alarms and 40 million stolen credit card numbers: How target blew it [online]. Mar 2014. URL: https://www.bloomberg.com/news/articles/2014-03-13/target-missed-warnings-in-epic-hack-of-credit-card-data#p1.

[106] Eliran Rubin and Nati Tucker. Technation: Pitango launches seventh venture fund with projected $175 million [online]. Dec 2020. URL: https://www.haaretz.com/israel-news/business/technation-1.5474071.

[107] S. Ding et al. Encrypt dns traffic: Automated feature learning method for detecting dns tunnels. In *2021 IEEE ISPA/BDCloud/SocialCom/SustainCom*, pages 352–359, 2021. doi:10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00056.

[108] Garcia Sebastian, Karel Hynek, Dmtrii Vekshin, Tomas Cejka, and Armin Wasicek. Doh research scripts for cvut/cesnet/avast doh project, 2022. (Accessed on 25/01/2022). URL: https://github.com/stratosphereips/DoH-Research.

[109] Selenium automates browsers. that's it! [online]. 2020. URL: https://www.selenium.dev/.

[110] SensePost ethical hacking team. sensepost/godoh [online]. Dec 2020. URL: https://github.com/sensepost/godoh.

[111] Stephen Sheridan and Anthony Keane. Detection of dns based covert channels. In *European Conference on Cyber Warfare and Security*, page 267. Academic Conferences International Limited, 2015.

[112] Haya Shulman. Pretty bad privacy: Pitfalls of DNS encryption. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, pages 191–200, 2014.

[113] Sandra Siby, Marc Juarez, Claudia Diaz, Narseo Vallina-Rodriguez, and Carmela Troncoso. Encrypted DNS –> Privacy? A Traffic Analysis Perspective, Dec 2020. arXiv:1906.09682.

[114] Sunil Kumar Singh and Pradeep Kumar Roy. Detecting Malicious DNS over HTTPS Traffic Using Machine Learning. In *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*, pages 1–6, 2020. doi:10.1109/3ICT51146.2020.9312004.

[115] A. K. Sood and S. Zeadally. A taxonomy of domain-generation algorithms. *IEEE Security Privacy*, 14(4):46–53, 2016. doi:10.1109/MSP.2016.76.

[116] SpiderLabs. Spiderlabs/dohc2 [online]. Dec 2020. URL: https://github.com/SpiderLabs/DoHC2.

[117] Craig Stanfill and David Waltz. Toward memory-based reasoning. *Commun. ACM*, 29(12):1213–1228, December 1986. doi:10.1145/7902.7906.

[118] Browser market share worldwide, May 2020. URL: https://gs.statcounter.com/browser-market-share.

[119] Didier Stevens. Downloading Executables Over DNS: Capture Files [online]. Dec 2020. URL: https://blog.didierstevens.com/2019/08/07/downloading-executables-over-dns-capture-files/.

[120] Kazi Abu Taher, Billal Mohammed Yasin Jisan, and Md. Mahbubur Rahman. Network intrusion detection using supervised machine learning technique with feature selection. In *2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST)*, pages 643–646, 2019. doi:10.1109/ICREST.2019.8644161.

[121] Tcpdump Group. Tcpdump/libpcap public repository [online]. 2020. URL: https://www.tcpdump.org/index.html.

[122] The Proofpoint Threat Insight. PsiXBot Now Using Google DNS over HTTPS and Possible New Sexploitation Module: Proofpoint US [online]. Dec 2020. URL: https://www.proofpoint.com/us/threat-insight/post/psixbot-now-using-google-dns-over-https-and-possible-new-sexploitation-module.

[123] The SciPy community. Scipy two sample t-test [online]. 2022. (Accessed on 15/05/2022). URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html.

[124] Anomali Threat. Illicit cryptomining threat actor rocke changes tactics, now more difficult to detect [online]. Dec 2020. URL: https://www.anomali.com/blog/illicit-cryptomining-threat-actor-rocke-changes-tactics-now-more-difficult-to-detect.

[125] Turing, Alex et al. An Analysis of Godlua Backdoor [online]. 2019. URL: https://blog.netlab.360.com/an-analysis-of-godlua-backdoor-en/.

[126] Dmitrii Vekshin, Karel Hynek, and Tomas Cejka. Dataset used for detecting dns over https by machine learning., May 2020. `doi:10.5281/zenodo.3818004`.

[127] Petr Velan, Milan Čermák, Pavel Čeleda, and Martin Drašar. A survey of methods for encrypted traffic classification and analysis. *International Journal of Network Management*, 25(5):355–374, July 2015. `doi:10.1002/nem.1901`.

[128] Tim Wicinski. DNS Privacy Considerations. RFC 9076, July 2021. URL: `https://www.rfc-editor.org/info/rfc9076`, `doi:10.17487/RFC9076`.

[129] Luca Winter. FluBot malware – All you need to know &; to act now [online]. Dec 2021. URL: `https://www.threatmark.com/flubot-banking-malware/`.

[130] David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, January 1992. `doi:10.1016/s0893-6080(05)80023-1`.

[131] Jiating Wu, Yujia Zhu, Baiyang Li, Qingyun Liu, and Binxing Fang. Peek inside the encrypted world: Autoencoder-based detection of doh resolvers. In *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 783–790, 2021. `doi:10.1109/TrustCom53373.2021.00113`.

[132] K. Xu, P. Butler, S. Saha, and D. Yao. DNS for Massive-Scale Command and Control. *IEEE Transactions on Dependable and Secure Computing*, 10(3):143–153, 2013. `doi:10.1109/TDSC.2013.10`.

[133] Y. Wang et al. A comprehensive survey on dns tunnel detection. *ComNet*, 197, 2021. `doi:https://doi.org/10.1016/j.comnet.2021.108322`.

[134] Tahmina Zebin, Shahadate Rezvy, and Yuan Luo. An explainable ai-based intrusion detection system for dns over https (doh) attacks. *IEEE Transactions on Information Forensics and Security*, 17:2339–2349, 2022. `doi:10.1109/TIFS.2022.3183390`.

[135] Mengqi Zhan, Yang Li, Guangxi Yu, Bo Li, and Weiping Wang. Detecting DNS over HTTPS based data exfiltration. *Computer Networks*, 209:108919, 2022. URL: `https://www.sciencedirect.com/science/article/pii/S1389128622001104`, `doi:https://doi.org/10.1016/j.comnet.2022.108919`.

[136] J. Zhang and I. Mani. KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction. In *Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets*, 2003.

Reviewed Publications of the Author Relevant to the Thesis Topic

[A.1] Dmitrii Vekshin, Karel Hynek, and Tomas Cejka; DoH insight: Detecting DNS over HTTPS by machine learning In: *Proceedings of the 15th International Conference on Availability, Reliability and Security.* New York: ACM, 2020 ISBN 978-1-4503-8833-7.

The paper **has been cited 47x** (excluding self-citations).

[A.2] Karel Hynek and Tomas Cejka; Privacy Illusion: Beware of Unpadded DoH In: *11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON).* Montreal: IEEE 2020 p. 621-628 ISSN 2644-3163 **Awarded Paper**

The paper **has been cited 5x** (excluding self-citations).

[A.3] Karel Hynek, Dmitrii Vekshin, Jan Luxemburk, Tomas Cejka, and Armin Wasicek; Summary of DNS Over HTTPS Abuse In: *IEEE Access.* 2022, p. 54668-54680, ISSN 2169-3536.

The paper **has been cited 3x** (excluding self-citations).

[A.4] Sebastian García, Joaquín Bogado Garcia, Karel Hynek, Dmitrii Vekshin, Tomas Cejka, and Armin Wasicek; Large Scale Analysis of DoH Deployment on the Internet In: *Computer Security - ESORICS 2022.* Cham: Springer International Publishing 2022, p. 145-165, ISBN 978-3-031-17142-0.

The paper **has been cited 1x** (excluding self-citations).

[A.5] Kamil Jeřábek, Karel Hynek, Tomáš Čejka, Ondřej Ryšavý: Collection of datasets with DNS over HTTPS traffic In: *Data in Brief Journal.* Netherlands: Elsevier, 2022, ISSN 2352-3409.

[A.6] Lukáš Melcher, Karel Hynek, and Tomáš Čejka; Tunneling through DNS over TLS providers In: *Proceedings of 2022 18th International Conference on Network and Service Management (CNSM)*. New York: IEEE, 2022, p. 359-363, ISBN 978-3-903176-51-5. **Awarded Paper**

[A.7] Daniel Uhricek, Karel Hynek, Tomáš Čejka, and Dusan Kolar; BOTA: Explainable IoT malware detection in large networks In: *IEEE Internet of Things Journal*. New York: IEEE, 2022, ISSN 2327-4662.

The paper **has been cited 1x** (excluding self-citations).

Remaining Publications of the Author Relevant to the Thesis Topic

[A.8] Sebastián García, Karel Hynek, Dmitrii Vekshin, Tomáš Čejka, and Armin Wasicek: Large scale measurement on the adoption of encrypted DNS. Arxiv Preprint 2021.

The paper **has been cited 14x** (excluding self-citations).

[A.9] Karel Hynek, Tomáš Čejka, and Dmitrii Vekshin DoH Detection: Discovering hidden DNS In: *Proceedings of the 8th Prague Embedded Systems Workshop.* Praha: Czech Technical University in Prague, 2020. p. 14-16. ISBN 978-80-01-06772-7..

The paper **has been cited 1x** (excluding self-citations).

[A.10] Dmitrii Vekshin, Karel Hynek, and Tomáš Čejka Brief Summary of Observations and Research on Encrypted DNS In: *Proceedings of the 9th Prague Embedded Systems Workshop.* Praha: Czech Technical University in Prague, 2021. ISBN 978-80-01-06858-8.

Remaining Relevant Reviewed Publications of the Author

[A.11] Karel Hynek, Tomas Benes, Matej Bartik, Pavel Kubalik: Ultra High Resolution Jitter Measurement Method for Ethernet Based Networks In: *The 9th IEEE Annual Computing and Communication Workshop and Conference (CCWC).*, New York: IEEE p. 847-851, 2019, ISBN 9781728105543.

The paper **has been cited 2x** (excluding self-citations).

[A.12] Dominik Soukup, Tomas Cejka, Karel Hynek: Behavior Anomaly detection in IoT networks In: *Proceeding of the International Conference on Computer Networks, Big Data and IoT (ICCBI - 2019).*, Cham: Springer International Publishing, 2020. p. 465-473, 2020, ISSN 2367-4520.

The paper **has been cited 1x** (excluding self-citations).

[A.13] Karel Hynek, Tomas Benes, Tomas Cejka, Hana Kubatova: Refined detection of SSH brute-force attackers using machine learning In: *ICT Systems Security and Privacy Protection.*, Cham: Springer International Publishing, p. 49-63, 2020, ISSN 1868-4238.

The paper **has been cited 6x** (excluding self-citations).

[A.14] Karel Hynek, Tomas Cejka, Martin Zadnik, and Hana Kubatova: Evaluating Bad Hosts Using Adaptive Blacklist Filter In: *Proceedings of the 9th Mediterranean Conference on Embedded Computing - MECO'2020.*, New York: IEEE p. 306-310, 2020, ISSN 2637-9511.

The paper **has been cited 1x** (excluding self-citations).

[A.15] Tomas Benes, Michal Kekely, Karel Hynek, Tomas Cejka: Pipelined ALU for effective external memory access in FPGA In: *Proceedings of the 23rd Euromicro Conference on Digital Systems Design.*, IEEE Computer Soc. p. 97-100, 2020, ISBN 978-1-7281-9535-3.

The paper **has been cited 3x** (excluding self-citations).

[A.16] Jan Luxemburk, Karel Hynek, and Tomas Cejka: Detection of HTTPS Brute-Force Attacks with Packet-Level Feature Set In: *11th Annual Computing and Communication Workshop and Conference (CCWC2021). Piscataway (New Jersey)*, New York: IEEE p. 115-123, 2021, ISBN 978-0-7381-4394-1.

The paper **has been cited 5x** (excluding self-citations).

[A.17] Dominik Soukup, Peter Tisovcik, Karel Hynek, and Tomas Cejka: Towards Evaluating Quality of Datasets for Network Traffic Domain In: *Proceedings of the 2021 17th International Conference on Network and Service Management.*, New York: IEEE, p. 264-268, 2021, ISSN 2165-963X.

The paper **has been cited 1x** (excluding self-citations).

[A.18] Zdena Tropkova, Karel Hynek, and Tomas Cejka: Novel HTTPS classifier driven by packet bursts, flows, and machine learning In: *Proceedings of the 2021 17th International Conference on Network and Service Management.*, New York: IEEE, p. 345-349, 2021, ISSN 2165-963X.

The paper **has been cited 2x** (excluding self-citations).

[A.19] Martin Zadnik, Jan Wrona, Karel Hynek, Tomas Cejka, and Martin Husak: Discovering Coordinated Groups of IP Addresses Through Temporal Correlation of Alerts In: *IEEE Access*, 10, p. 82799-82813, 2022, ISSN 2169-3536.

[A.20] Richard Plny, Karel Hynek, and Tomas Cejka: DeCrypto: Finding Cryptocurrency Miners on ISP networks In: *Secure IT Systems*, Cham: Springer, 2022, ISSN 0302-9743.

[A.21] Jan Luxemburk, Karel Hynek, Tomas Cejka, Andrej Lukacovic, Pavel Siska: CESNET-QUIC22: A large one-month QUIC network traffic dataset from backbone lines In: *Data in Brief Journal*, Netherlands: Elsevier, 2023, ISSN 2352-3409.

Remaining Relevant Publications of the Author

[A.22] Richard Plný, and Karel Hynek: Detection of Cryptomining in High-speed Networks In: *Proceedings of the 10th Prague Embedded Systems Workshop.* Praha: Czech Technical University in Prague, 2022. ISBN 978-80-01-07015-4.

[A.23] Dominik Soukup, Karel Hynek, and Tomáš Čejka QoD: Ideas about Evaluating Quality of Datasets In: *Proceedings of the 8th Prague Embedded Systems Workshop.* Praha: Czech Technical University in Prague, 2020. ISBN 978-80-01-06772-7.

[A.24] Matěj Bartík, Tomáš Beneš, and Karel Hynek An Example of PCB Reverse Engineering - Reconstruction of Digilent JTAG SMT3 Schematic In: *The 7th IEEE Workshop on Advances in Information, Electronic and Electrical Engineering* Piscataway, New Jersey: IEEE 2019. ISBN 978-1-7281-6730-5.

The paper **has been cited 1x** (excluding self-citations).

[A.25] Karel Hynek, Tomáš Beneš, Tomáš Čejka, and Hana Kubátová Future approaches to monitoring in high-speed backbone networks In: *Proceedings of the 7th Prague Embedded Systems Workshop.* Praha: Czech Technical University in Prague, 2019. ISBN 978-80-01-06607-2.