FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of master's thesis

Title:	Detection of DNS over HTTPS abuse
Student:	Bc. Dmitrii Vekshin
Supervisor:	Ing. Karel Hynek
Study program:	Informatics
Branch / specialization:	Computer Security
Department:	Department of Information Security
Validity:	until the end of summer semester 2022/2023

Instructions

Study DNS encryption methods and focus on the DNS over HTTPS (DoH). Summarise the current knowledge about its abuse. Explore available DoH datasets suitable for DoH abuse with DNS tunnels. Design an algorithm for automatic detection of DoH tunnels' presence in the network based on the observed network traffic. Develop a software prototype capable of processing real network traffic. Test and evaluate the prototype for its accuracy and other performance characteristics such as processing speed.

Electronically approved by prof. Ing. Róbert Lórencz, CSc. on 24 February 2022 in Prague.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Master's thesis

Detection of DNS over HTTPS abuse

Bc. Dmitrii Vekshin

Department of Information Security
Supervisor: Ing. Karel Hynek

June 23, 2022

Acknowledgements

I would like to thank my family and friends for support during writing this thesis.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

In Prague on June 23, 2022 . . . . . .

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Vekshin, Dmitrii. *Detection of DNS over HTTPS abuse.* Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2022.

Abstrakt

Tato práce poskytuje hlubokou analýzu protokolu DNS-over-HTTPS se zaměřením na malware související s DNS-over-HTTPS. Další část této práce se zaměřuje na problematický tunel DNS-over-HTTPS s vytvořením prototypu pro jeho detekci. Přínosem této práce je představení nového přístupu zachycování dat tunelu DNS-over-HTTPS a aplikace tohoto přístupu k vytvoření celosvětové distribuované infrastruktury pro zachytávání pro simulaci skutečného chování DNS-over-HTTPS v různých prostředích.

Klíčová slova DNS-over-HTTPS, Tunel, Síť, Sledování, Tok, Strojové učení

Abstract

This study is provide deep analysis of DNS-over-HTTPS protocol with focusing on DNS-over-HTTPS related malware. Another part of this work focuses on DNS-over-HTTPS tunnel problematic with creating a prototype to detect it. The contribution of this work is introducing of novel approach of DNS-over-HTTPS tunnel data capturing and applying this approach to create world wide distributed capturing infrastructure to simulate action of real DNS-over-HTTPS behaviour in different environments.

Keywords DNS-over-HTTPS, Tunnel, Networking, Monitoring, Flow, ML

Contents

List of Figures

List of Tables

Introduction

The Domain Name System protocol is the one of basic protocols which is widely used on the Internet. However, the protocol was designed in 1983, and since this year Information Technology (IT) world significantly changed. Domain Name System (DNS) technologies do not satisfy modern security and privacy requirements and that makes DNS protocol highly vulnerable and unreliable.

This problematic leads to the idea, that global wide enhance or replacement of this important network protocol is unavoidable to cover nowadays requirements of networking.

During the last years we observed a trend to study and develop the encrypted DNS solutions. One of this kind of solution is the DNS over HTTPS (DoH). Huge IT organizations participated in research and development of DoH or rolled out their own DoH infrastructure. Almost all biggest browsers introduced default support of the DoH protocol. More and more news related to DoH are appearing into information field of IT community.

This synergy of demand, prevalence, companies interest, and relative youth of technologies gives a strong motivation to contribute in research of encrypted DNS area, particularly in scope of DoH protocol.

The DNS over HTTPS protocol is designed to solve security and privacy issues of DNS protocol. This protocol provides additional encryption level upon original DNS. In comparing with other encrypted DNS protocols, the addition beneficial feature of DoH protocol is utilization of HTTPS protocol. The HTTPS layer allows DoH to remain hidden between other HTTPS connections. Unlike other encrypted DNS protocols, DoH does not run on a specific port. Both mentioned features increase complexity of simple determining of DoH presents.

Encryption of DoH solves several security problems of basic DNS and enhance the privacy makes Internet a safer, but at the same time it creates additional space to malicious actors and provides new possibilities to enhance existing attacking techniques by abusing the protocol.

In this study this dual nature of protocol is considering as a challenge. This challenge rises the motivation to research DoH technology, its impact as well as methods of adaptation to new threat landscape.

This thesis focuses on detection of DoH tunnels and developing of software prototype capable of detection this kind of malware. The DoH tunnels are considered as critical threat for enterprises. And this work propose novel approach to simulate behaviour of DoH tunnels in real-word environment. This new approach is applied during the dataset collection to cover issues missing in other works related to DoH tunnels detection.

The thesis is structured in the following order. Chapter 1 provides background information about technologies related to DoH, discuss network traffic monitoring techniques and provides description of all encrypted DoH protocols with focus on DoH. Chapter 2 introduce classification of DoH malware, describes DoH malware groups and in particular explain modus operandi of DoH related malicious programs. Chapter 3 examines public available existing datasets of DoH traffic, estimates their suitability to be used in this work. Also this chapter presents DoH tunnels data capturing approach and describes dataset creating process. Chapter 4 is dedicated to design, creating and evaluating of software prototype of DoH tunnels detector.

Chapter 1

Background

1.1 Domain Name System

Domain Name System is one of the main services in the networking. DNS is the naming system represented as hierarchical distributed database. DNS maintains records for the Internet as well as a private networks. The most important function of DNS is to translate domain names into corresponding identifiers represented by numerical Internet Protocol (IP) addresses. Process of receiving of IP address related to a domain name is called domain name resolution. IP addresses take very important part in the network communication, but these addresses are hard to remember for users and DNS solves this problem. The current specification is presented in [1] and [2].

1.1.1 DNS architecture

As mentioned above, DNS is the hierarchical distributed database and it has hierarchical structure and it maintained by many different organisations. DNS architecture is represented by a hierarchical tree structure as it shown on the figure 1.1. The structure consists of zones and each zones is separated by dots. From right to left it includes Root name servers, then Top Level Domain (TLD) name servers which usually represent countries e.g. CZ - Czech Republic and authoritative name servers - Second Level Domain (SLD). The Root server is the top node of the domain tree and in contains information for all TLD domains. It is represented as "." which is displayed only in Fully Qualified Domain Name form. Root has 13 IP addresses used to query the different root server networks. Roots servers are own by 12 organizations supervised by Internet Corporation for Assigned Names and Numbers (ICANN). The root servers are the critical part of the Internet and to provide high reliability and availability is ensured by anycast routing.

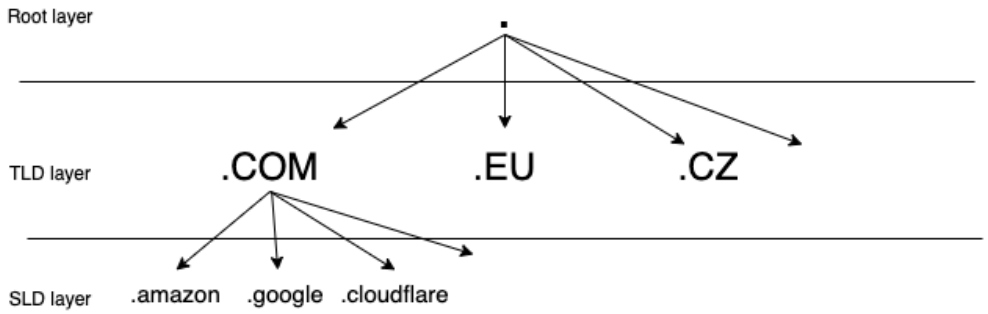

Figure 1.1: Diagram shows DNS hierarchical tree structure

1.1.2 DNS protocol

DNS protocol in application layer protocol by TCP/IP model. Typically DNS works upon User Datagram Protocol (UDP) but it can use another transport layer protocol, particularly Transport Control Protocol (TCP). By specification DNS protocol has to listen on port 53. In turn, majority of encrypted versions of DNS use stateful TCP, between discussed ecrypted DNS protocols, only DNS over QUIC use UDP as a transport layer protocol.

Functionality of DNS consist as not only the name-address mapping between domain name and IPv4 addresses, but also as lot different functions. Each function has specific Resource Record (RR) type. The most used and relevant for this work RR types presented in Table 1.1. In details AAAA records are used for IPv6 address mapping in domain resolution process. TXT records are used in different algorithms, such a Domain Key Identified Mail (DKIM) signatures or Sender Policy Framework (SPF), both are used in electronic mail communication to ensure security and integrity. TXT records also uses in many different algorithms and technologies. CNAME records are used as an alias.

At the present time, a lot of different other network technologies rely on the DNS and use it as part of itself, it its infrastructure. That makes replace or significant changes of DNS very complicated.

DNS messages include five section, presented in Figure 1.2. **The header** section provides information which sections are presented. For request and responses the header section is different. The section contains identifier represented by *16-bit ID*. identifier serves to pair query and responses. The header contains QR type, flag that is set to *'0'* for query and for *'1'* for responses. TR flag notifies if message is to go beyond maximum length of *512* bytes. Recursive Desired (RD) field tells if recursive resolution is preferred. Recursive Available (RA) flag appears in responses, to notify requesting side that recursive resolution is possible. The last part of a header section provides information about number of resource records in other sections.

The question section contains information about the query. Section con-

Table 1.1: DNS Resource Record types

Value	Type	Description
A	*1*	IPv4 domain address
AAAA	*32*	IPv6 domain address
CNAME	*5*	Alias of one name to another
DNSKEY	*48*	The key record used in DNSSEC
DS	*43*	Identify the DNSSEC signing key
DLV	*32769*	DNSSEC Lookaside Validation record
NS	*2*	Name server record
MX	*15*	Mail exchange record
TXT	*16*	Text record
PTR	*12*	PTR record. Pointer to a canonical name.
NSEC	*47*	Next Secure record, DNSSEC
NSEC3	*50*	Next Secure record v.3, DNSSEC

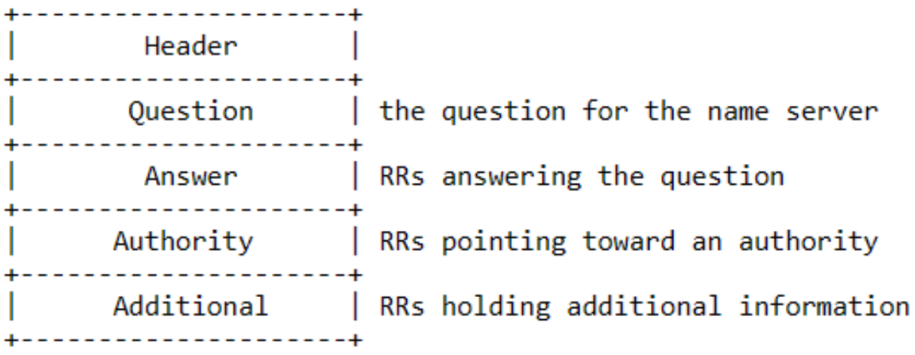

Figure 1.2: Diagram shows DNS message structure, according to RFC 1035 [1]

sists of QNAME, QTYPE and QCLASS parts. QNAME contains query name and uses as privacy DNS feature that limits the sending of the full domain destination to the upper layer nameservers. QTYPE specifies resource record requested by query. Examples are illustrated in Table 1.1. The last part QCLASS provides information about a query class.

The last sections have same structure with variable number of RRs for the responses. And additional information such as Time To Live (TTL) - TTL time in seconds, that the record is available in cache.

1.2 Network traffic monitoring

This section provides information about modern techniques and trends in network traffic monitoring from network security perspectives.

Network monitoring is critical highly important part of the cybersecurity industry. This direction is growing and developing year by year, along with network threats which also are also changing, elevating and widely spreading. In this work, passive network monitoring approaches are focused. Passive network monitoring follows traffic passed a measurement point, and therefore works with a traffic generated by users.

1.2.1 Deep packet inspections

Deep Packet Inspection (DPI) bases on extraction of information from packets, or complete inspection. DPI aggregates analysis of the packet header and data. It allows to notice a deviance in a protocol and protect against different network threats.

In addition, DPI helps to find redirected packets. In other words, in comparison with conventional packet filtering, DPI provides bigger opportunities in localization, detection, categorization, blocking and rerouting of suspicious packets.

In turn, DPI is a form of packet filtering and often performed by a firewall.

DPI performs evaluation of packets contents routed through the monitoring point. DPI takes a decision based of the predefined rules in real time.

Nowadays many of different network security tools and techniques are based on the DPI. Several of them are presented bellow.

1.2.2 Intrusion Prevention Systems

An Intrusion Prevention System (IPS) ensures network protection by continuous monitoring of network. This kind of systems performs detection and prevention of harmful activities.

Because of the ability to take action against malicious activities and notify an administrator, the IPS are considered as a powerful network security tool. Such a system can be included to next-generation firewall (NGFW) or unified threat management (UTM) solution.

When the IPS recognizes malicious behaviour, then automated actions are taken. This includes alerting, packets dropping or blocking.

1.2.3 Pattern or signature matching

One approach to using firewalls that have adopted IDS features, pattern or signature matching, analyzes each packet against a database of known network attacks. The downside to this approach is that it's effective only for known attacks, and not for attacks that have yet to be discovered.

Patterns or signatures mean set of distinguishing features of communication with comparing to other. For network security, they are represented by attack network patterns, channels, or peer-to-peer communication. Signatures

and patterns are create based on knowledge of the same attack in the past. In other words. this technique analyses each packets and matches it with records in database of network attacks, which are already known.

1.2.4 Application Layer Monitoring

Application Layer Monitoring covers plenty of network protocols. Relevant information can be extracted almost from each protocol, but some protocols are commonly used for this protection technique. One of such protocol is DNS. The DNS is critical for networking, is not encrypted and provides a lot of useful information. The monitoring of the DNS with a blocklists is extremely useful in terms of detection botnets or Domain Generation Algorithm based threats. DNS is monitored by majority of enterprises and in many couturiers on the government level. One more protocol that is widely used in Application Layer Monitoring is HTTP.

Since HTTP is web protocol used to interact with web resources, very often this protocol is abused to transfer malicious payload. HTTP requests contain Uniform Resource Identifier that already can reveal threats associated with the resource, but also both request and responses have a lot of data in their headers. Such information as request method, HTTP version, HTTP headers, encoded payload size or entropy as far as User-Agents can reveal malicious traffic or at least suspicious anomaly.

The next crucial protocol in terms of network monitoring is Transport Layer Security (TLS).

TLS provides security layer of encryption for the communicating channel. The main features provided by the protocol are authentication, confidentiality, and integrity. The monitoring of TLS can be implemented in two different ways. The first one is when monitoring point acts as man-in-the-middle (MITM) and breaks TLS connection to get access to plain text data.

In situation when MITM is not possible, another opportunity is to analysis encrypted traffic. In this case DPI focuses on the handshake protocol. During the TLS handshake client sends set of attributes. Then after short negotiation between the client and the server - version, cipher suite, session ID, and compression method are chosen for the communication. After attributes setting, server sends its certificate. And DPI system makes a decision by applying techniques such as Certificate blocklisting or JA3/JA3S TLS fingerprinting to mentioned attributes and the certificate.

1.2.5 IP Flows monitoring

IP Flows provide higher-level view of the connections between hosts in monitored network. IP Flows contain information about transmitted packets and bytes, headers and meta-information. Flow monitoring focuses on aggregated data statistics called flow.

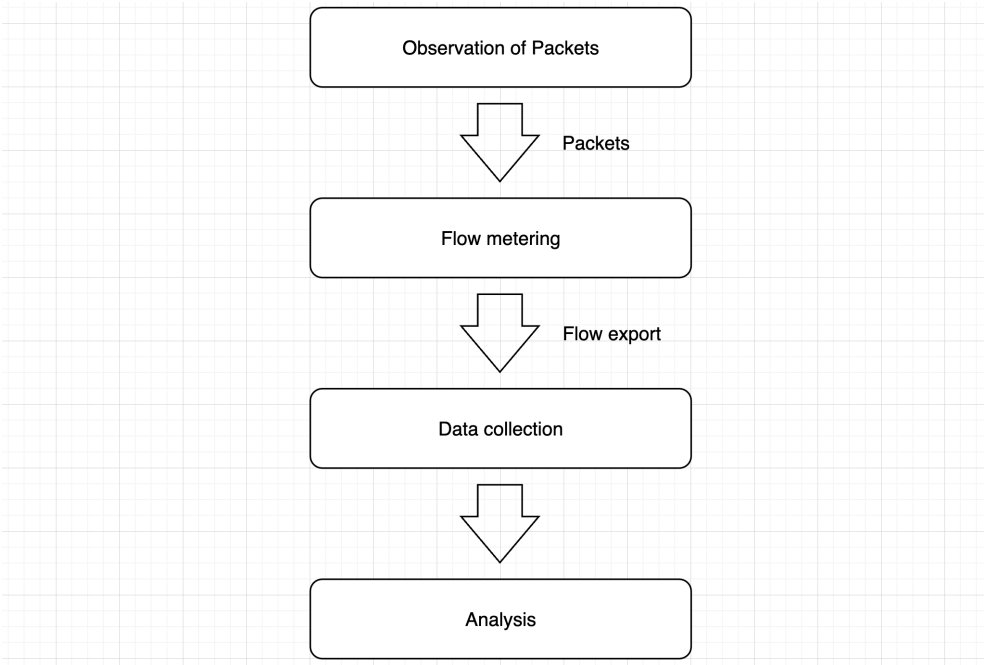

Figure 1.3: The figure illustrates architecture of flows monitoring

Advantages of flow monitoring for network security are discussed by Luca Deri et al. on work [3] and Hofstede Rick et al. in article [4]. In both works authors provides analysis of stages of a flow monitoring setup and shares of recommendations of implementation of flow monitoring.

Traditional IP flow monitoring based on creation of a flow record for a unidirectional IP flow aggregated by same header values, same source and destination IP address and ports, as well as the incoming network interface.

Since IP flow monitoring approaches work with aggregated statistics instead of packets based DPI approaches, flow monitoring significantly reduce amount of analyzed data. This is big advantage, particularly for high-speed computer networks.

1.2.5.1 Flow monitoring architecture

Mostly flow monitoring systems share several stages, which are presented in the Figure 1.3. Flow monitoring starts with Packet Observation. In this stage packets are capturing and preprocessed on the data monitoring point.

The next stage is called Flow Metering. In this phase packets are aggregated into flows. When the particular flow ends a flow record is exported. That means that that the record is filled into a data structure defined by used flow export protocol. Flow records typically contain of flow information such a IP addresses, port numbers, numbers of packet and bytes passed while flow was measured.

The third stage is Data Collection. In this stage ensures storage and preprocessing of flow data. Preprocessing may include operations of aggregation, filtering, compression, and summarizing.

The last stage is Data Analysis. This stage consists of study on captured data including correlation, profiling, classification, anomaly detection etc.

1.2.5.2 IP Flow monitoring in anomaly-based detection

Anomaly-based methods are able to effectively detect the existing attacks and attacks which are have not been observed in past. Ability to detect irregularities is a highly important aspect of network traffic monitoring for detecting the suspicious activities in the network. Network traffic categorization is crucial here. Machine learning becomes a common technique to do this in mainstream networks.

Wide integration of AI (Artificial Intelligence) to different field of IT opens new capabilities. For the network security complete study of AI applying is provided by Yuan Qingmin and Tan Xin in article [5]. Authors suggest a strategy of using AI for network security defence porpoises. Also Barradas Diogo et al. in work [6] present a system called FlowLens that allows more efficient collection of features of packet distributions and flows classification for ML-based Network Security Applications.

N.Satheesha et al. in work [7] use Flow base intrusion anomaly detection. Authors use SDN-based anomaly detection framework with Machine Learning and Deep Learning techniques. They proofed the capability of the approach to detect existing network threats and provide explanation that this approach can be used for unknown threats as well.

1.3 Encrypted DNS protocols

This section describes group of protocols aggregated by a common goal - to enhance DNS protocol. As far the DoH is a focused by this woks, DoH protocol is described significantly deeply them others.

1.3.1 DNS-over-HTTPS

The DNS-over-HTTPS protocol serves to mitigate privacy issue of DNS [2]. Unlike the DNS that performs Domain Name resolution in plain text, DoH is encrypted. That allows to keep sensitive user information such as visited websites, browsing habits and used services. Moreover, as it's mentioned in introduction, DoH uses HTTPS protocol on port 443 to hide yourself between other HTTPS traffic. The techniques that allows profile and fingerprint users using DNS information does not work with users using DoH. The protocol consists of the two parts - client and server and each of them have to be considered separately, because these parts are independent. Client establish connection

to the server, encrypts and sends DNS queries to server via HTTPS. In turn, server performs domain name resolution and sends back a encrypted responses with required data. DoH clients mainly can be implemented in three ways 1.4:

- Browser implementation. This is most common and easiest way to configure and use DoH. All biggest browsers have this feature [8] [9], and for Google, Mozilla, Internet Explorer, Edge and Brave the DoH was enabled by default. But since 2021 year, default support is off and instead of it browsers provides users with setting in configuration window.
- DoH stub implementation. This type is represented by using DNS to DoH translator as a proxy, sometimes in combined with add-blocking software. DNS is redirected to server in local network, that performs translation from DNS to DoH. Examples of this implementation are Cloudflare and Pi-hole. [10]
- Operation system (OS) implementation. Windows has implemented DoH support on the system level since Windows 10. Also easy configurable setting, that can be set in DNS settings in OS. Linux systems also has it.

Servers part of the protocol ensures three responsibilities - decryption of DoH requests from clients, domain name resolution via DNS and encrypted response to clients. Different DoH providers use different implementations. The good illustration is implementation of security technique called padding. Padding means filling all packets with dummy data to maximal size. It ensures the same length for all queries and it increase the entropy. For majority of implementations uses this technique, except the Cloudlfare implementation. This is significant issue, which is discussed in detail in conference paper [11] by Karel Hynek et al.

1.3.1.1 Specification

Very important side to analyze the protocol is technical description and specification. In other words, this section contains detailed technical description of the protocol, based on [12] standard and real implementations of most significant DoH provides. Specification contains decryption of implementation approach and provides with recommendation. Based on this document HTTP/2 is a minimum recommended version. For HTTP protocol, DoH can be run though GET and POST requests as well [13].

Figures 1.5 and 1.6 show POST DoH request to resolve www.example.com and the responses. For the request, in HTTP header and the MIME type is added in the Content-Type field. In this case Content-Type is *application/dns-message*. Request body contains an encrypted DNS request. The Accept flag in the HTTP header can be set. This flag specialize which response type is

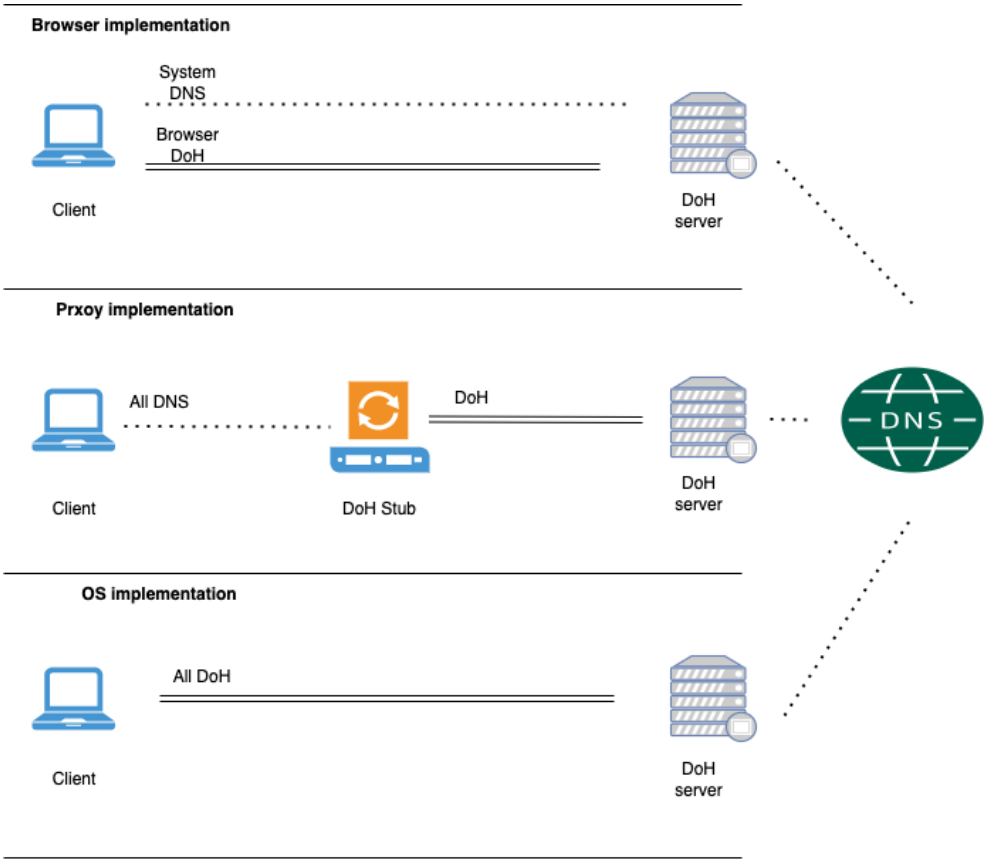

Figure 1.4: Diagram presents three possible DoH implementations

expected e.g. binary or JSON. Response HTTP header also contains Content-Type while response body contains encrypted DNS response.

Using of HTTP GET method to perform DoH request is presented on Figure 1.7. In this case body of the request is empty instead the domain is encoded in BASE64url format.

In provided specification is highlighted that both, client and server have to support *application/dns-message*, and this is only one defined media type. Also Google JSON implementation [14] allows one more specific format - JSON DoH. In this implementation *application/dns-message* is used for binary DNS and *application/x-javascript* uses for explicit JSON text. Figure 1.7 illustrates response in JSON format. After Google introduce support of JSON format, Cloudflare [15], Quad9 [16] and AhaDNS [17] also provide users with the opportunity to use JSON format.

```
:method = POST
:scheme = https
:authority = cloudflare-dns.com
:path = /dns-query
accept = application/dns-message
content-type = application/dns-message
content-length = 33

<33 bytes represented by the following hex encoding>
00 00 01 00 00 01 00 00  00 00 00 00 03 77 77 77
07 65 78 61 6d 70 6c 65  03 63 6f 6d 00 00 01 00
01
```

Figure 1.5: DoH POST request

```
:status = 200
content-type = application/dns-message
content-length = 64
cache-control = max-age=128

<64 bytes represented by the following hex encoding>
00 00 81 80 00 01 00 01  00 00 00 00 03 77 77 77
07 65 78 61 6d 70 6c 65  03 63 6f 6d 00 00 01 00
01 03 77 77 77 07 65 78  61 6d 70 6c 65 03 63 6f
6d 00 00 01 00 01 00 00  00 80 00 04 C0 00 02 01
```

Figure 1.6: DoH POST response

1.3.1.2 Privacy

The protocol allows to hide data from ISPs and therefore prevent data leaks, profiling and selling of users the DNS data as well as it allows users to bypass country level censure.

Since the communication via DoH is encrypted and visible only to the user and DoH server, profiling is almost impossible. Besides the encryption of DNS data, DoH moves the visibility of resolved domains names from local DNS providers to more centralized DoH providers. Positive impact is described in [18] .

```
:method = GET
:scheme = https
:authority = dnsserver.example.net
:path = /dns-query? (no space or Carriage Return (CR)
        dns=AAABAAABAAAAAAAAAWE-NjJjaGFyYWN0ZXJsYWJl
        bC1tYWtlcy1iYXNlNjR1cmwtZGlzdGluY3QtZnJvbS1z
        dGFuZGFyZC1iYXNlNjQHZXhhbXBsZQNjb20AAAEAAQ
accept = application/dns-message
```

Figure 1.7: DoH GET query

Privacy is a significant advantage of the protocol. User DNS data can be monitored, stored and sold by owner of public network or an ISP. Also some websites can be censored or blocked during domain name resolution process by an ISP. The law covering this issues and data privacy entirely is different for different countries and it's arduous to control compliance with the law. DoH solves data selling issue and non-legal selling problem, by hiding domain resolution process from third parties, but leaves a place to bypass censorship and be abused. The negative opinion is expressed in [19].

1.3.1.3 Security

DoH also helps to prevent DNS spoofing and eavesdropping. In other words, because the session between end-user and a DoH server is encrypted, nobody can alter the resolution request results to point the user's browser toward a fraudulent website. Also, it makes harder for an attacker to perform a DNS hijacking attack, because when the DoH is used, attacker has only one possible place to provide DNS hijacking - behind DoH server, where during domain name resolution, but this makes impossible to perform personalized DNS hijacking attack.

DoH is developed to enhance security and privacy. The first aim is privacy layer, and Karel Hynek and Tomas Cejka, show in their work [11] the possible attack to unpadded DoH and get information about resolved domains without description DoH traffic. The second aim is security level. In work [20] Qing Huang et al. demonstrate the possibility to attack protocol itself by making it downgrade to plain DNS. In this case, both layers - privacy and security are affected without getting user know about it.

1.3.1.4 Stealth ability

DoH does not use specific port and disappears between other HTTPS traffic going to port 443. This behaviour allows to achieve better privacy - DNS data are encrypted and the fact of using DNS and DoH is hidden.

```
{
  "Status": 0,
  "TC": false,
  "RD": true,
  "RA": true,
  "AD": true,
  "CD": false,
  "Question": [
    {
      "name": "example.com.",
      "type": 28
    }
  ],
  "Answer": [
    {
      "name": "example.com.",
      "type": 28,
      "TTL": 1726,
      "data": "2606:2800:220:1:248:1893:25c8:1946"
    }
  ]
}
```


Figure 1.8: DoH JSON query

For some browsers implementations, the presence of DoH can be disclosed in moment of receiving of DoH severs IPs, because to obtain these IPs plain text DNS is used. Also for when the DoH provider is one from well-known DoH providers list the the presence of DoH can be released. But the researches described in section DoH prevalence shows how many different unknown providers are exist and their number is growing.

The following list of papers [21], [22] and [23] targets detection of presence of DoH traffic. Machine learning approach is very useful in this case. Regardless, DoH uses port 443 like other HTTPS traffic, it has very specific signature of behaviour. DoH can be detect by estimating time between packets, number of packets, ration of numbers in/out packets and other features which don't require decryption. In the following paper [21], we achieved accurate results not only for determination of DoH presence but in recognizing of DoH provider as well.

1.3.1.5 Reliability

Most of DoH implementations fallback to normal DNS in case of connection errors, and the resolution process is performed via normal DNS protocol. For the other hand, it leads to strong reliability and domain name is resolved in any case, but also it is an vulnerable place and this mechanic can be used for attacks. Such kind of attack is perfectly illustrated by Qing Huang et al. in the article [20].

1.3.1.6 DoH prevalence

Another important aspect of DoH is the prevalence and speed of spreading this technology around the word. This field is not well studied that leads to the lack of information. Reason is that it is not that easy to distinguish DoH traffic from other HTTPS traffic because it run on the same port as browsers traffic. In the article [24] published in 2019, the adoption of DoT and DoH is measured by analysis of open resolvers. From 1.2 million open DNS resolvers found on the Internet, only 9 (0.007%) supported DoH. Result shows the adoption goes extremely slowly, but in this study authors find DoH and DoT servers between DNS servers, and consequently resolvers which manages only DoH resp. DoT requests are missing. The study by Lu et al. [25], in 2019, also scanned well-known open DNS resolvers and found only 17 DoH resolvers. However, in our studies [26] and [27] we scan the whole internet for open 443 port and check every obtained IP for all different DoH format. First we did it in 2021 and secondly in May 2022. And after analysing the data we indicate a trend of growing the number of DoH resolvers. In May 2022 the numbers are 3,838 IPs of DoH servers, and after analysis of their TLS certificates, we found 657 unique domain names. This number is significantly bigger then 200 domains in public available list of DoH servers [28].

1.3.1.7 DoH from market perspectives

Many big IT companies rolled out they own DoH infrastructure. For now based on data in [28] we can find out that more then 200 organisations participate in DoH propagation. In Czech Republic - Avast and cz.nic have DoH servers. Mozilla and Cloudflare started looking at how to implement and distribute DoH for users in 2018. Significant point on DoH time line is 2019 when Google run their DoH service and enabled DoH by default in Chrome browser [29]. After it marked changed and since this moment more then 80% 1.9 of DoH traffic is going to Google DoH servers. Data are obtained based on analyse of traffic of 400 millions end-points.

The next game changer for DoH occurs after Comcast enters to the industry, in 2020. They started to run Trusted Recursive Resolver [30] and invited Mozilla and Cloudflare as praters. All companies participated in this program has strong verified privacy policy and ensures for user trust to their resolvers.

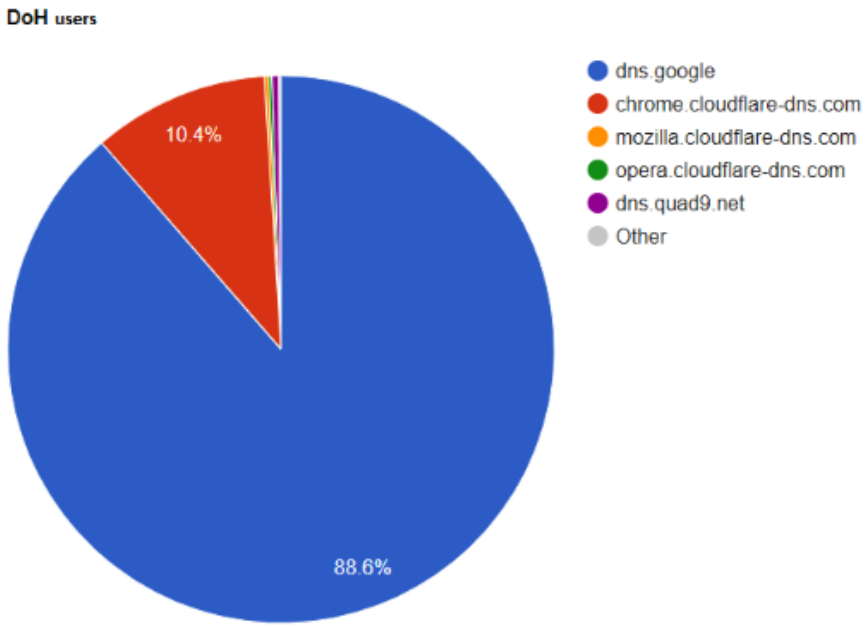

Figure 1.9: Diagram illustrates DoH market shared parts between DoH suppliers

Then an essential role in DoH propagation played OSes. Windows provides support for DoH [31] for all version higher then Windows 10. Android and IOS also became a part of DoH market with their support for smartphones.

Today DoH technology is widely supported in OSes, Browsers and DNS solutions. Also majority of DNS service providers add DoH service.

1.3.1.8 Demand on the protocol

In modern trends of global traffic encryption and privacy DoH protocol suits very well. In details, demand on the protocol proceed with combination of needs in security for vulnerable plain text DNS and wanted privacy to prevent selling users data by ISPs.

1.3.1.9 Usability

DoH technology is extremely user-friendly. A lot of guidelines, info, recommendation are provided and easy accessible. Some products contain DoH enabled by default.

1.3.1.10 DoH performance

In works [32], [33], [34], [35] and [36] provides with detailed performance analyses for DoH and comparing of DoH, DoQ, DoT, DoHoT, ODoH and DNS. Also experiments are performed in different environments: location (USA, EU countries, Australia, Korea), Encrypted DNS provider (Cloudflare, Google, Quad9), ISP, implementation (ODoH different implementations, DoH stub and DoH browser).

All works show that the difference between DoH and DNS is not significant. In certain conditions DoH can even outperform DNS. For other Encrypted DNS protocols DoH is faster, except DoQ, where DoQ is noticeably faster then DoH or DNS.

1.3.1.11 DoH Impact

The impact of the DoH protocol is ambiguously. Positive and negative sides are considered separately.

DNS is a decentralised system; this is a valuable feature, but behaviour of DoH by design leads to data centralization. That means that all data are go through shot list of DoH suppliers [37]. DoH supplier have to be trustful and reliable, but even trough it, the decentralized DNS system is more stable for attacks and failures than DoH system. Decentralized systems typically more divers and centralization leads to reduction recursive resolvers that makes a list of aims for attacks also smaller.

Also DoH allows to bypassing of restriction determined be government, company or parental control, because many of them use DNS to block unwanted resources [38]. It leads to increasing of non-legal activities. Not only domain block lists are affected, but many different security tools, because DNS monitoring is a common practice in cybersecurity. In common DoH can be abused by malware and makes it more stealthy.

On the other hand, DoH enhances privacy for users. It protects against profiling and user data selling. Also encryption protects against eavesdropping, DNS cache poisoning attack and adds resistance to DNS hijacking. Possible avoiding of blocked websites can be a good for non-legal blocking or for states with strong censorship policies.

1.3.1.12 DoH evolution

The first RFC DoH protocol draft published in 2018 and contained specification only for browsers. Since this time the protocol evolve and get support on OS level by Windows 10 and higher and many Linux distributions. Google take a significant part in enhancing process of the protocol with a json format and DNS-over-QUIC variation. Also research IT community did a retrospective and fixed found problems by creating Oblivious DNS-over-HTTPS [36].

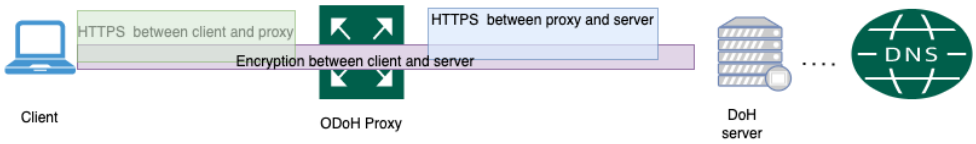

Figure 1.10: Diagram illustrates ODoH protocol

1.3.2 Oblivious DNS-over-HTTPS

Oblivious DNS-over-HTTPS is a modified version of DoH. The main distinguishing feature is adding a proxy between DoH client and server. This is required because when the normal DoH is used - users data are centralized and DoH provide has full information which user resolved which domain and can link all client activities. This behaviour could be considered as a privacy issue. ODoH solves it with a following solution.

Client establish HTTPS connection with ODoH proxy. Proxy establish connection with DoH server. Client encrypts data with key provided by DoH server and sends them to proxy. Proxy transmits data to DoH server. The same flow for the responses from DoH server 1.10:.

ODoH ensures two very important features: DoH server doesn't obtain clients private information, such as the IP, because the server receives data from proxy and can see only proxy IP. Proxy can decrypt DoH query and responses and therefore hasn't access to payload.

Both this features ensures high level of privacy and security for the DoH clients.

1.3.3 DNS-over-TLS

DNS-over-TLS is protocol, which performs domain name resolution via TLS connection to a DoT server. DoT RFC specification published in 2016 [39]. By weaponizing TLS DoT achives security and easy OS level support, but in the comparison with DoH is ensure poorer privacy. The privacy issue here is DoT requires specific port 853 and because of this, even analyse of encrypted traffic, can easily determine presence of DoT and tell amount of DNS queries send by user. Regardless privacy side of the protocol, it widely supported by majority of IT service providers, including biggest browsers and OSes. DoT is used by many Android systems even during the updates or by default as system domain name resolution method.

1.3.4 DNS-over-QUIC

DNS-over-QUIC (DoQ) [40] published in May 2022, is very similar to DoH, but use as a encryption level QUIC protocol. DoQ protocol is created by google, and is enabled by default in Chrome browser. Since QUIC is based on UDP and implements the congestion control, reliability to errors and more

useful features of TCP in the userspace instead of in the kernel, the DoQ is faster then other encrypted DNS protocols based on TCP.

DoQ protocol solves the problem of head of line blocking by multiplexing the connection. Then the query leaded to error or slow don't affect other queries.

1.3.5 DNSSEC

DNS Security Extensions protocol provides additional level of integrity and authentication for DNS protocol [41]. However DNSSEC don't implement encryption for the communication but can be used as additional enhancement for other protocols presented in the section.

DNSSEC use digital signatures based on public key cryptography.
DNSSEC means that DNS data is signed by the owner of the data. It allows to verify that data is received actually from the zone where it believes the data originated and also verify that data isn't modified during the transmission.

DNSSEC builds trust anchors - public key changes from zone to their parental zones, up to root zone's public key.

1.3.6 DNSCrypt

The DNSCrypt [42] is a earliest Encrypted DNS protocol, but it still doesn't has RFC. DNSCrypt Clients and resolvers use the protocol over UDP and over TCP as well. by default the protocol run on 443 port.

DNSCrypt rely on the public key infrastructure. DNSCrypt session begins when Client sends non-authenticated DNS query with encoded the certificate and public identifier of the provider, to a DNSCrypt-enabled resolver. The resolver sends signed certificates, and client has to verify this certificate using a previously distributed public key, known as the provider public key.

Each certificate contains a magic number .The client use this number as a prefix in queries to notify resolver which certificate is chosen by the client.

The encryption algorithm, resolver public key and magic number are used by client to encrypt queries. Client queries also contain client public key which along with chosen certificate and resolver secret key are used to decrypt client requests and encrypt responses on the resolver side.

1.3.7 Cloudflare additional solutions

Cloudflare can be considered as an ambassador of encrypted DNS. Cloudflare supports all possible kinds of encrypted DNS including experimental solutions [43]. However, these solution aren't recommended to be used for the production or for the other critical uses.

DNS-over-HTTPS-over-TOR

Table 1.2: Encrypted protocol features

Protocol	Encryption	Real IP	Spec. Port	Authentication
DoH	*Yes*	*Yes*	*No*	*No*
ODoH	*Yes*	*No*	*No*	*No*
DoQ	*Yes*	*Yes*	*No*	*No*
DoT	*Yes*	*Yes*	*Yes*	*No*
DNScrypt	*Yes*	*Yes*	*No*	*No*
DNSSEC	*No*	*Yes*	*No*	*Yes*
DoHoT	*Yes*	*Yes*	*No*	*No*

This solution employs resolver hidden in Tor onion service. This allows to keep user IP private and hide domain name resolution from ISP. This approach is similarly to ODoH hides real IP address of client.

DNS-over-Discord
DNS-over-Discord requires to invite Cloudflare DNS bot to user discord server and after resolution process is performed via communication with this bot in the application. Communication can go with Discord API to ensure amortisation.

DNS-over-Google Sheets
DNS-over-Google Sheets is implemented as a function, that user has to insert in the Google sheet. Then every time domain and DNS type are inserted in two nearest cells, function performs domain name resolution and place the result in the result cell.

DNS-over-Telegram
DNS-over-Telegram requires to add Cloudflare DNS bot to user telegram contacts. After this approach is very similar to DNS-over-Discord.

1.3.8 Summarize

Encrypted DNS protocols offers choice of different protocols to make domain name resolution more private and secure. In the table 1.2 protocols features are summarized. Features analysis shows that ODoH is most interesting protocol and ensures maximal privacy and security for clients, however DoQ as far as based on extremely fast QUIC protocol can ensure better user experience for clients. In additional, using of ODoH along with DNSSEC, covers all known problems of normal DNS. In turn of DoH, the advantage is worldwide support of most valuable companies in IT industry. Also DoH allows to encrypt browser produced DNS queries and let system ones in plain text.

Chapter 2

DNS-over-HTTPS Malware

In this work the term DoH malware is used related to any malware that uses DoH as part of it malicious infrastructure. As far as one of main features of DoH is privacy, the protocol is of interest malware creators as well as normal users. Privacy side of the protocol allows to hide the presence of malware by avoiding common DNS monitoring. Consequently it leads to creating of new malware techniques and enhancing of existing ones.

2.1 Challenges and warnings

For cybersecurity point of view, DoH brings new challenges and the most significant is that with DoH many network monitoring techniques based reading DNS packers are not longer possible. However for some security model DNS monitoring is important part.

DoH rises a warning that the protocol allows to bypass domain based censorship. Some countries and companies has implemented censorship on the DNS level. The technique is called DNS Tampering - sending instead of real DNS response a fake one, which can hide existence of domain or redirect users to a block page with explanation of blocking reason. In country wide case - this technique can be used to prevent access to illegal websites blocked by government. In company one, some websites can be blocked by internal rules of company e.g. some organization disallow their employees to access social networks as well as some companies, especially enterprises has own network security infrastructure and DoH allows to bypass firewall.

From endpoint security point of view, another challenge is that when browser setup is used, ssh keys for TLS connection used by DoH are stored by browser, and they are not easily accessible from other processes.

2.2 Classification

DoH malware can be aggregated to several category, in this work performed classification separate malware into three categories. The first one is Command and Control (C&C) and malicious programs are part of some C&C infrastructure. Malware in this category is usually represented by botnets, where malicious server sends commands to infected users machines. Server typically sends small amount of data just to orchestrated actions of its clients. Good example here is botnets targeted to DDOS attacks, where server sends only aim and order to start the attack. The second category is tunneling malware. This kind of malware establish covered channel between the victim and the attacker server, and then sends or receives big amount of data. This kind of malware uses to exfiltrate data and bypass network monitoring or to secretly access the network and acts from this network. And the last category is more common, this category describes programs and system where DoH is used to avoid a restriction or blocking. In this group, the DoH is an element of accessing technique to forbidden or unwanted resources.

2.2.1 Command and Control

The first category contains C&C malwares. Malicious programs in this category uses DoH in three different ways. The first is to hide domain name resolutions of important parts of C&C infrastructure. The second way is using as an encrypted channel to hidden communicate between C&C clients and servers. And the last observed way is using DoH responses as a transport medium for downloading a part of C&C system.

The first ever malware using DoH is detected in less than half of year after protocol has been released. It takes a place on April 24, 2019. The malware gets a name Godlua [44] malware form C&C malware group. Godlua abuses DoH TXT query to receive URL where Lua bytecode is stored as a PNG file. Also it uses DoH as beacon to say to C&C server that client is in touch.

Rocke, a China-based cryptomining threat actor also used DoH to enhance C&C infrastructure for the LSD malware. Authors from Anomaly describes evolution and behaviour of the LSD malware in the blogpost [45]. In this case DoH TXT queries are used to distribute a setup script. Attackers placed parts of script encrypted by AES-128 inside encrypted DoH responses. The AES key can be derived from domains name used to get this key itself.

Another example is Pink botnet with 5 million associated IPs, described in the blogpost [46] by 360 NetLab. Malware resolves all domain name DoH server, which IP is stored in encrypted configuration file.

One more malware abuses DoH to access C&C infrastructure is PsiXbot. Researcher from Proofpoint threat insight team [47] publish analysis of this malware. PsiXbot uses the hardcoded Google resolver and performs JSON-based DoH request via HTTP1.1 to get the IP address of C&C domain.

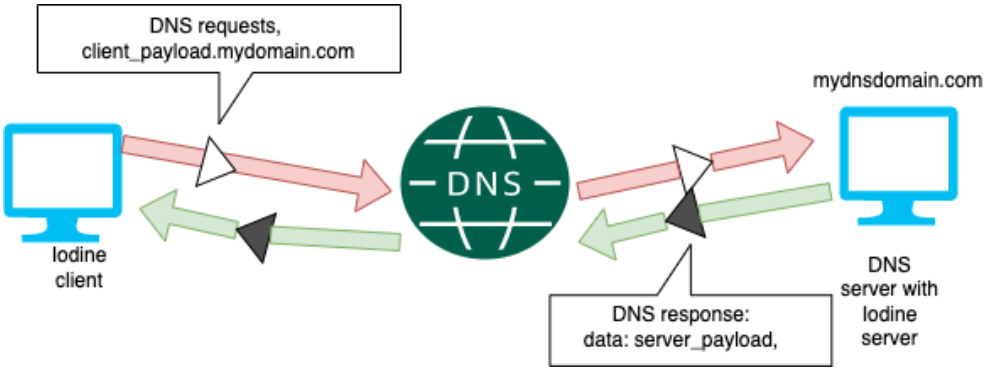

Figure 2.1: Diagram shows work flow of Iodine appellation

Banking malware FluBot for Android devices, also weaponizes DoH to access C&C infrastructure [48].

Researchers from Huntress labs publish two blogposts [49] and [50] about malware performed JSON-based DoH TXT request. Also this malware is describe in [51] blogpost published by Bleepingcomputer. The malware also uses Google DoH resolver. The answer contains data masqueraded as DKIM-signature. But on the first look normal and innocent DKIM signature is a double BASE64 encrypted and IPs in decimal number format.

Except real malware, possibilities to us DoH in C&C malware are shown in proof-of-concept (PoC) project DoHC2 [52] and godoh [53], [54]. PoC godoh uses DoH JSON queries as a transport medium for C&C communication. The project DoHC2 implements almost same behaviour and implemented by red team operations software Cobalt Strike13.

2.2.2 DoH tunnels - Exfiltration

The second category describes malware and tools serve to establish data exfiltration tunnels between victim and malicious server. This group contain directly DoH based applications as well as applications use DNS, which can be easily upgraded to DoH.

Iodine [55] is client-server application that allows to establish covered tunnel between the client and the server. The application uses DNS responses and requests to transfer data inside the tunnel. To use the Iodine tunnel owner have to registered own DNS server and enter an NS record for it in the domain control panel. NS record have to point your registered server domain name to IP of the Iodine server. The client creates A type DNS requests to tunnel owner domain with high frequency. During the domain name resolution third-party resolvers recursively send the query to your registered DNS server. And server replies with empty DNS responses. While this process is going connection is established and tunnel is ready to transmit data. To send the data clients adds payload into the DNS requests as subdomain. Iodine

server receives data and parses data from requests. If server wants to send data, server adds data to DNS responses, also as a plain text. The basic workflow of Iodine is shown on the figure 2.1. Iodine allows to setup maximum size of DNS packets used to data transfer. As soon as tunnel is established, it appears as a new network interface on the client machine as well as on the server one.

Another DNS tunnel tool is called dnsscat2 [56]. It works similarly to Iodine, and can be used in the same purposes. The benefit of dnsscat2 is that adds additional encryption for payload delivered trough the tunnel.

The first advanced persistent threat that abused DoH is APT34, said authors from ZDenet in the blogpost [57]. The Iranian hacker group APT34 as known as Oilrig uses tool DNSExfiltrator [58] to establish a covered channel and move data inside victim's internal networks, and then exfiltrate to controlled by attackers server.

Dnstt [59] is very similar to DNSExfiltrator tool. It also allows to establish end-to-en encrypted tunnel with authentication. Dnstt is design to ensure high performance during data transferring.

Also DoH tunnels are discussed in red team conferences BruCON [60], authors create a prototype of tunnel that is able to download Mimikatz or other executables via DNS-over-HTTPS with an Excel sheet.

2.2.3 Avoiding of blocking

Hynek Karel et al. in their work [38] reveal several types of abuse DoH protocol to bypass official restrictions and in other malicious proposes. In this paper authors find several ways where DoH is used to allow user to access gambling websites in country where online gabbling is forbidden. In this case after first visit of landing page, user allows to modify the his browser by installing a redirection mechanism. After when user tries to access gambling website, mentioned mechanism performs a DoH request to domain owned by gambling providers. The response contains actually functional domain of the gambling services. This technique allows to bypass censorship and blocking by domain name.

The second way is using in application Sdarot.tv that allows to access video resources with copyright violence. in this case, client tries to communicate with server blocked on local DNS resolvers. Since the direct communication is not allowed, the client uses DoH to bypass DNS blocking and obtain current working IP address of the server.

The last observed abuse is downloading of JavaScript code delivered in DoH TXT responses. This behaviour is observed in spam campaign. The HTML file was attached to the emails and after clicking on it DoH query is sent. The DoH responses contain JavaScript redirect script with landing URL.

2.3 DoH malware prevalence

The prevalence of DoH malware is also very important metric to complex analysis. To estimate the prevalence databases of VirusTotal and Avast's internal database are used. The approach consists of following steps. The first one is to create a list of DoH resolvers IPs. Then use this list to produce YARA rules [61]. And the last one is to apply this YARAs in searching API for mentioned databases.

The result of this research is 120 samples from VirusTotal and 97 samples from Avast. As far as databases of VirusTotal and Avast are huge, and only VirusTotal contains more then 80 millions of samples, it seems like the analysing of this sources provides a picture of malware creation trends in the world. However, the result depends on the list of DoH servers IPs and capabilities to detect DoH in Avast and malware hunting community in VirusTotal case.

2.4 Countermeasures

DoH brings new problems for cybersecurity industry, and countermeasures against using the protocol in malicious purposes becomes necessary. Drew Hjelm, the researcher from SANS institute in his study [23] of DoH issue provides following recommendations for organisation. Firstly, try to block DoH by configuration of network devices such are routers or firewalls to prevent access to public known DoH providers. But attackers can use some personal or not smaller DoH servers that are not published in any lists of the DoH providers. Also the warning here is that technique called Domain Fronting can be used to obfuscate real destination of HTTPS traffic, e.g. GoDoH [54] allows manage C&C instruction using DoH queries to *https://www.google.com/resolve* but after queries are forwarded to *https://dns.google.com/resolve* what is valid URL of Google DoH resolver. This approach also recommended by Sean Hutchison in the blogpost [62]. The next recommendation is to do application whitelisting and configuration standards. This means that only trusted application, even they can use DoH, have to be allowed to run across the organisation. The second part is to do unified configuration of all application and OSes and disallow using of the DoH protocol. Such configuration are presented in majority of browsers supported DoH as well as in OSes configuration. And also configure applications to produce logs, where it possible. Logs have to include information about DoH usage.

Another list of recommendation is published by the US National Security Agency (NSA) in the manual [63] covering adopting Encrypted DNS in Enterprise Environments. The key points of published manual are provided in the list bellow.

- Stay using DNS internally, but use DoH to recursive queries

- Enforce using of system DNS resolvers
- Use enterprise policy configurations for browsers
- Use HTTPS inspection, to break SSL and block DoH
- Use application whitelisting
- Use heuristic and anomaly-based detection to prevent data exfiltration

To summarize all recommendation above, the DoH is considered as unsafe for the enterprises and have to be block in enterprise environments.

Chapter 3

Dataset

Nowadays there are several public available datasets with DoH traffic. The dataset CIRA-CIC-DoHBrw-2020 [64] is used in many papers, [65], [22], [66], [67], [68] but this dataset contains issues such as that for the whole dataset the same location and same network conditions - e.g. bandwidth, latency. The clean traffic in this dataset is generated straightway without mimicry to human actions. And dataset doesn't contain mixed benign and clean DoH connection, but this type of connection seems like challenging to detect. Others public available datasets [69] and [70] are well captured but they don't contain any malicious DoH traffic.

However, Kwan Carmen and et al. in work [71] and Zhan M. and et al. in conference paper [72] take into account weak points of CIRA-CIC-DoHBrw-2020 and fixed in datasets used in mentioned works. Both datasets contain DoH based exfiltration traffic. In additional the dataset used by Zhan M. and et al. in [72] is captured around the world with different locations of tunnel client and with different DoH providers. However, both datasets don't contain connections with mixed benign and malicious DoH traffic.

Another dataset used by Nguyen Tuan Anh at el. in work [73] is captured with simulation of enterprise infrastructure, but still this work doesn't cover case when DoH tunnel traffic along with benign DoH traffic go through the same DoH proxy.

3.1 Setup

In this work the dataset of benign and malicious DoH traffic is captured. Benign DoH traffic is produced by web browsing. The malicious DoH traffic is a mix if browser traffic and traffic produced by DoH tunnel. Setup overview is presented on figure 3.1.

DoH-stubs run on Kali Linux 2020.4. It plays role of a local installed proxy that translates DNS to DoH overview and vice-versa. Also the data capturing

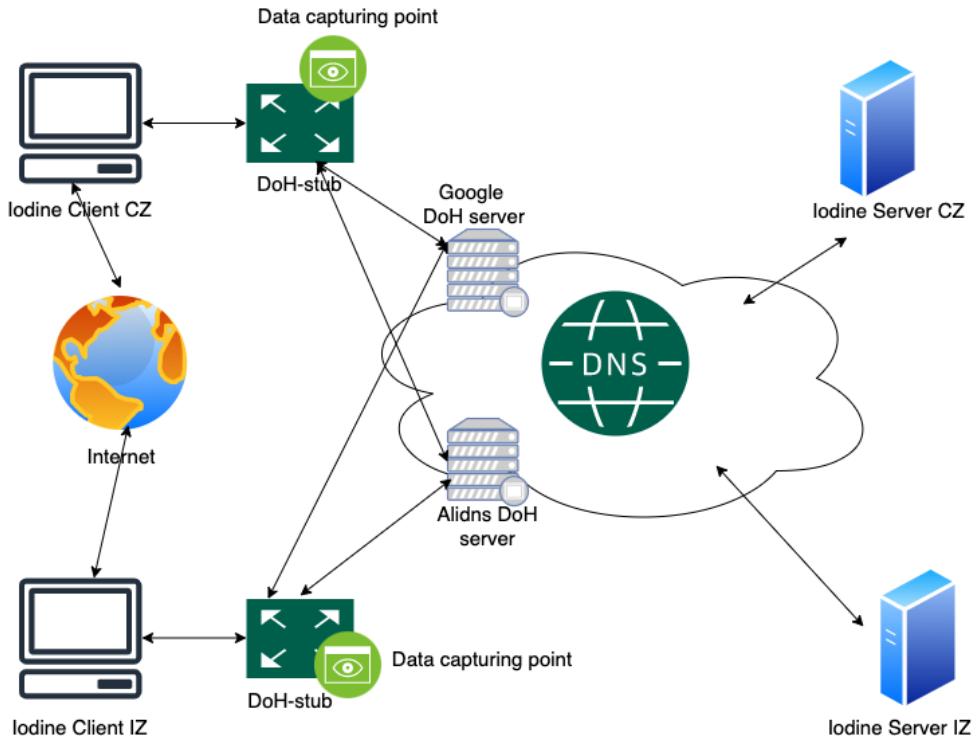

Figure 3.1: Diagram presents overview of data capturing infrastructure components

is performed on this machine, because all DNS and DoH traffic goes through DoH-stub.

Iodine tunnel clients run on Ubuntu 20.04.4 LTS. This component simulates user computer, where user is browsing web and simultaneously malicious tunnel is installed and exfiltrats user data. DNS traffic from this machine goes to DoH-stub installed in the same network, but HTTPS traffic goes directly to the internet.

Iodine servers runs in others network and other physical locations then the clients and the DoH-stubs - servers are distributed to different laboratories. For setup in Czech Republic, server uses CentOS Linux 7. For Izrael setup server installed on Ubuntu 20.04.4 LTS. Servers represent malicious tunnel server owned by attacker, which receive exfiltrated data.

3.1.1 Locations

Part of real malware programs can act from different locations - countries or even continents. Is very common practice when APTs act strictly outside their own country. To simulate this behaviour in this work, data capturing are implemented on two different continents. Similar setups are implemented

Table 3.1: List of DoH providers tried to be used in the setup with decision and explanation about using in the setup.

Provider	**Using**
Alidns	*Accepted*
Google	*Accepted*
Cloudflare	*Rejected*, after few minutes tunnel is blocked
Quad9	*Rejected*, after few minutes tunnel is blocked
Adguard	*Rejected*, extremely low speed, majority of packets lost
openDNS	*Rejected*, after few minutes tunnel is blocked
aha-dns	*Rejected*, extremely low speed, majority of packets lost

in Prague and ... laboratories.

Prague laboratory is Network Traffic Monitoring Lab (NETMON). The laboratory's goal is research and development (R&D) in monitoring of high-speed network traffic and network infrastructures, Internet of Things (IoT), analysis of network traffic and anomalies detection, i.e. malicious or suspicious traffic.

3.1.2 DoH providers

After analyzing 10 different DoH provides, to perform the data capturing only two are chosen. The list of tried providers and decision to use with explanation are provided in table 3.1.

Only Google and Alidns candidates are accepted. Other providers cut tunnel connection or show inappropriate performance. Alidns - DoH resolver owned by Chinese Cloud computing company Alibaba Cloud Computing (Beijing) Co., Ltd. Google DoH resolver is most used DoH provider in the word.

3.1.3 Used tools

Selenium
All browser traffic is produced by Selenium framework [74] run by python script. Selenium is an open-source project that serves to web browsers automation. Selenium allows programmers to use single interface for test scripts in different programming languages. Then browser-driver executes test scripts on the instance of selected browser.

The main part of Selenium is WebDriver that uses browser drivers to execute scripts. Every browser requires to use specific browser driver. WebDriver consists of following parts:

- Application Programming Interface (API) - Translate scripts in diffrent program languages to Selenese. Selenese is a scripting language created for Selenium.

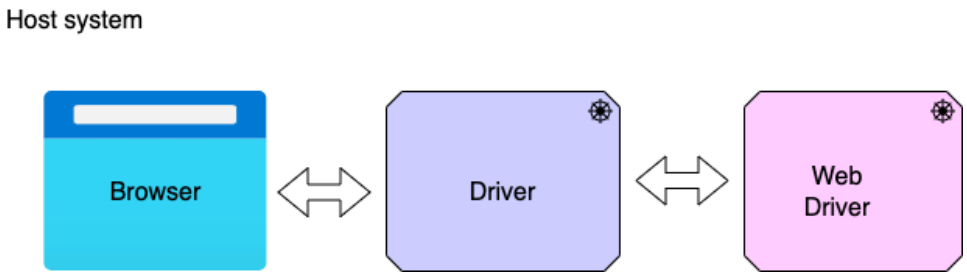

Figure 3.2: Diagram shows workflow of Selenium WebDriver

- Library - Contains the API and language-specific bindings.
- Driver - Browser specific module to run instance of the browsers and testing scripts.
- Framework - Supproting of libraries for integration with other test frameworks.

WebDriver communicate with a browser through a driver. Communication is bidirectional, WebDriver sends commands to the browser and Browsers sends information to WebDriver, in both cases via Driver as in shown in figure 3.2. Driver different for different browsers, e.g. GeckoDriver is used for Mozilla's Firefox but for Google Chrome, ChromeDriver is required, etc. The driver and the browser have to be placed on the same system.

Tcpdump

To capture packets Tcpdump [75] is used. Tcpdump is a tools that is able capture and analyze traffic on the network. Tcpdump includes many different options as well as filters. The tool provides user with a Command line interface. Tcpdump is based on libpcab library written in C/C++ programming languages. When Tcpdump is run, the libcap process starts in performs capturing of network packets and displays content information or directly stores to the file.

In this setup Tcpdump used in data capturing point and run on the DoH-stub machine.

Vagrant

Since the capturing process performed from different physical locations, to faster deployment and unification of distributed setups, Vagrant [76] is used. Vagrant is a utility for creating and managing virtual machines and environments.

Vagrant provides users with ability to create configuration file that is called Vagrantfile. In this file user can specify OS version, all software components

as well as configurations. And after the one configuration file can be used to create different machines with exactly same configurations, installed software.

Two Vagrantfiles are used in setup shown on figure 3.1. The first one is used to build Iodine tunnel clients run on Ubuntu 20.04.4 LTS.

The second for DoH-stub that is run on Kali Linux 2020.4. This Vagrantfile also contains additional instruction to run configuration bash script, provided along with Vagrantfile, This script ensures installation of DoH proxy and advanced configurations for the DoH-stub.

Tunnel payload generators
Tunnel payload is generated by sending files trough the tunnel. Two different file transfer tools are used.

The first tool used to fulfill the DoH tunnel with payload secure copy (SCP) bash command is used. SCP allows to secure transferring files between two devices connected to the network. SCP uses SSH it creates SSH tunnel and requires authentication.

The second tool, is custom Python File transfer. Simple client-server application is created for the setup. This application is able to select network interface with python socket programming, to adjust a speed rate and send a file to certain IP. Speed rate adjustment is implemented id following way - the program sends data by blocks, and after each block, it counts ration of amount of sent data and used time. If this ration is bigger then required speed limit, then application adds additional delay.

Iodine
Iodine [55] is a core tool for the whole setup. DNS tunnel is created from the client network to the server. Iodine is enhanced by DoH proxy, and since this, can be considered as a DoH tunnel.

dnscrypt-proxy
For translation of DNS queries to DoH queries the tool named dnscrypt-proxy [77] is used. The proxy runs on DoH-stub machine as a system service.. The tools works as a local DNS server that translates DNS requests to DoH requests, resolves via DoH, and then returns DNS responses.

3.2 Data capturing

To create dataset we used two different approaches, the first represents of activities of machine infected by DoH tunnel. In this case, captured traffic contains browser traffic and tunnel traffic. And both types of traffic goes through DoH proxy, therefore proxy sends in the same connection to the DoH resolver malicious tunnel traffic and benign browser one. Data capturing point is placed on DoH-stub because this is the place where all relevant network

Table 3.2: The table shows locations of the client and the server during the data capturing.

Client location	**Server location**
Czech republic	Czech republic
Czech Republic	Israel
Israel	Czech Republic
Israel	Israel

traffic - DNS and DoH is visible by monitoring on different network interfaces. The second approach represents normal normal benign traffic produced by web browsing.

Also as far as the capturing system elements are same for both locations, elements for one location can be replaced by elements of the other. Combinations of server and client locations are present in the table 3.2.

3.2.1 Browser traffic capturing

Browser traffic is generated with the idea to mimicry human behavior on the internet and to achieve it two different techniques are used. The first technique produce traffic with the following conditions. Every random period of time in range between 30 and 90 sec. the script via Selenium gets google search page. Then it types random letter and do search, after the script takes first 10 websites in search engine results and visit random number between 2 and 4 of websites. The second technique uses Top 1000 websites list from Amazon Alexa's Top. Every random time interval between 30 and 60 sec. the script takes between 2 and 5 websites from mentioned list of websites and opens them one by one every second. This two techniques supposed to simulate behaviour of user doing web browsing.

DNS requests produced by both approaches are sent to DoH-stub, and translated to DoH and send to one of DoH resolvers. Then DoH-stup receives DoH responses, translates it back to DNS and sends it black to the client machine. Data capturing process and directions of traffic are presented on the figure 3.3 Both, DoH and DNS are captured by Data capturing point run on DoH-stub machine.

3.2.2 Tunnel traffic capturing

Tunnel traffic is produced by Iodine tool. When the tunnel is established, DNS produced by Iodine is going to DoH-stub where DoH proxy translate it to DoH and then DoH requests are sent to the DoH server. On the DoH server side, DoH is decrypted back to the DNS and during the domain name resolution process finally rich the Iodine server. Answers of Iodine server route exactly the symmetric way back. To DoH server firstly, when they are encrypted, to

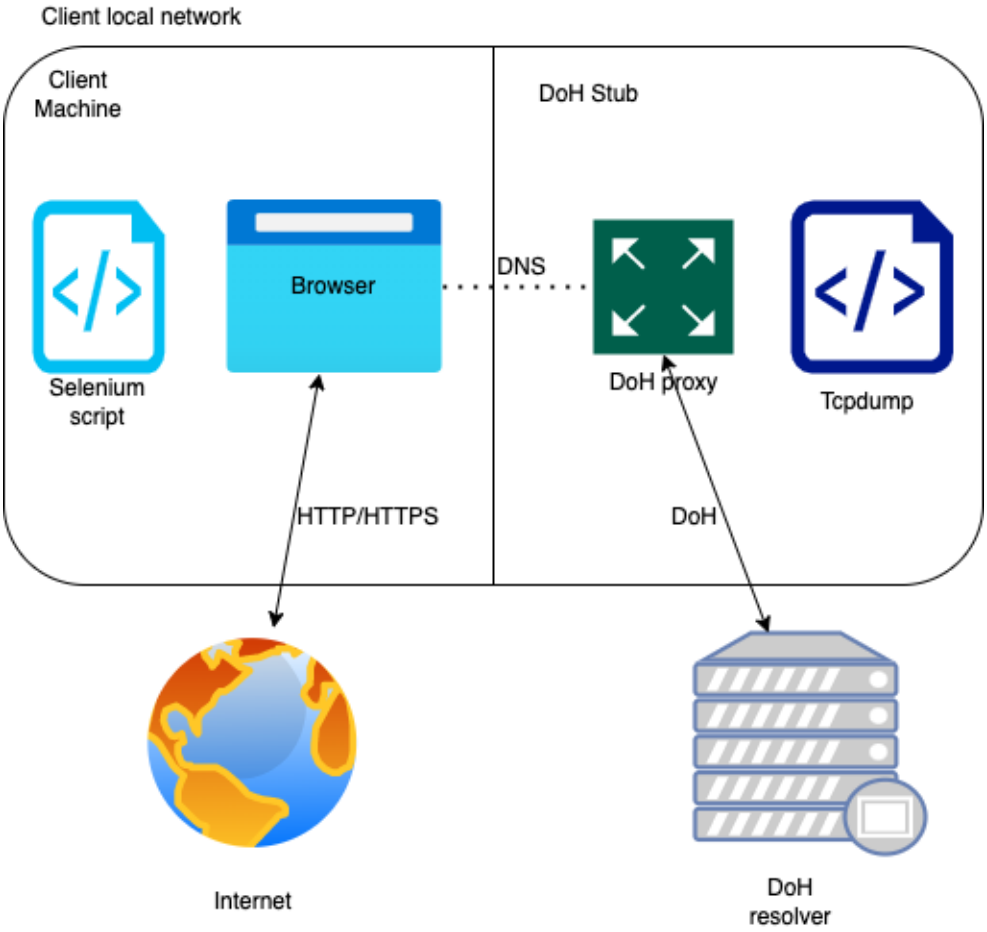

Figure 3.3: Diagram shows workflow of capturing browser traffic

the DoH-stub, where DoH proxy again decrypts it and translates to the normal DNS and after all, DNS answer are sent from DoH-stub to Iodine client. This part is the same for all captured tunnel traffic, except DoH server - for some captures Google DoH server is used and then capturing was repeated with Alidns DoH server. All configuration are captured three times according to different tunnel settings - 10 bytes, 100 bytes and 256 bytes of maximum payload size. Inside Iodine tunnel payload is created with two different approaches. The first one is a creating of secure shell tunnel by using SCP utility. In this case, data transferred through the tunnel are encrypted and the transferred content can't be discoursed. The second approach is by using Python File Transfer tool. For this approach 9 series of capturing are performed. For each maximum payload size - 10, 100, 256 bytes, three different Speed rate limits are applied - 50, 100, 500 kB/s. Unlike SCP, this configuration do not ensure encryption of data inside the tunnel and the content of tunnel can be

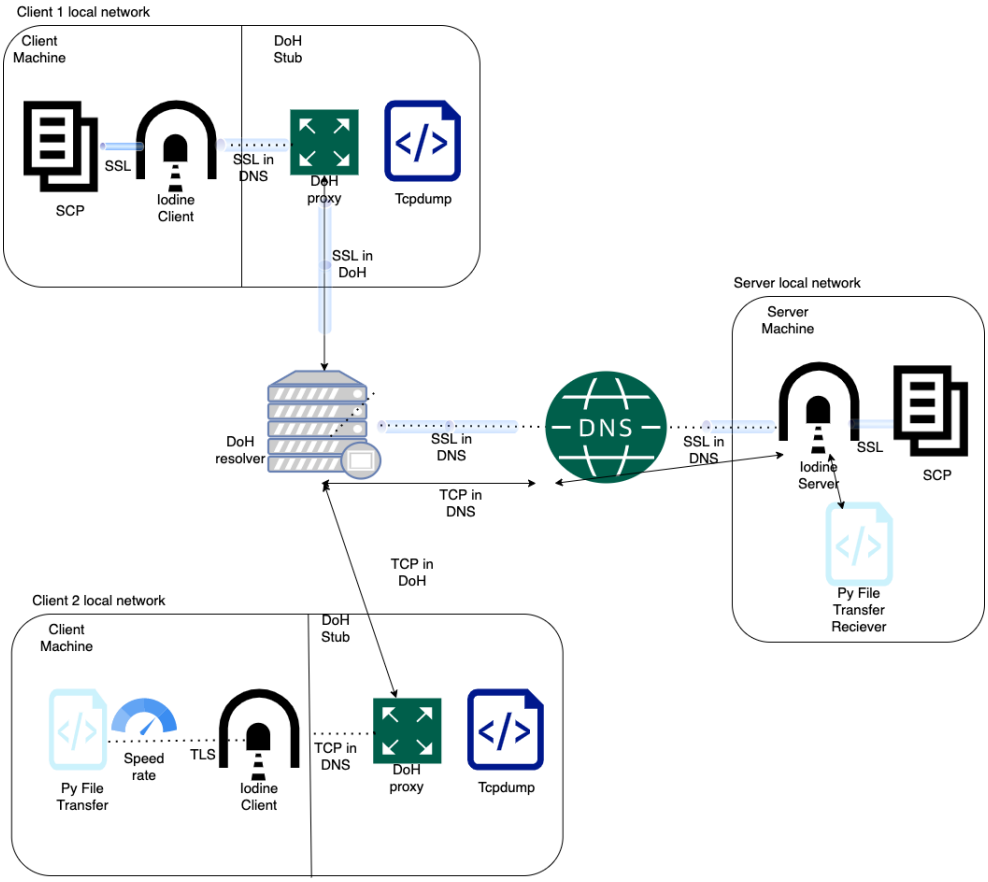

Figure 3.4: Diagram shows workflow of capturing tunnel traffic and traffic routes for two clients, one using SCP inside the tunnel and the second using Python File Transfer.

easily revealed by simple eavesdropping.

3.3 Data labeling

As far as benign and malicious data are generated separately, data are explicit annotated since the capturing. Pcaps contained browser and tunnel traffic are labeled as *malicious*. The others pcap, which contained only traffic produced by browser have label *clean*.

3.4 Dataset summary

After the capturing process and stripping of all unrelated data. Resulting pcaps are parsed by flow exporter ipfixprobe [78] into IP flows.

The client and the server captured from two different location each, with two different providers of DoH, with 3 different maximum sizes of DNS packets, and using SCP for tunnel payload generation and Python File Transfer with 3 different speeds. In summary during the dataset capturing *96* configuration of setup simulating infected with DoH tunnel machine are used. For simulation of traffic produced by normal browsing process, *4* configurations are used.

Chapter 4

Tunnel detection

For deeper understanding of DoH tunnels problem, firstly is required to address to DNS tunnels. Also, existing DNS tunnels can be upgraded to DoH tunnels. In work [79] Anna L. Buczak er al. provides good explanation and description of DNS tunneling and introduce prototype of ML based model capable to detect DNS tunnel.

Many researchers and organisations participate in the study of the DoH security and ways of abuse the protocol. Alenezi Rafa and Ludwig Simone A. in article [80] performed recognition of DoH tunnels with diffrent ML models, along with works [65], [22], [66], [67], [68] authors use public available dataset [64] and directly target DoH tunnels detection. In mentioned papers authors attempts to detect DoH tunnels with different ML algorithms Random Forest Algorithm, Decision Tree, Extra Tree, Gradient Boosting, LGBM, XGBossting, Support Vector Machine, 2D CNN and C4.5. Each of them archive a good score and provide the knowledge to choose the best option for tunnel detection. Based on experiments described in the papers above, Support Vector Machine, Random Forest Algorithm and XGBoost with almost 100% accuracy.

In terms of protection against DoH tunnels, the trend of using more efficient and stable technologies is observed. The newest papers use advanced AI-based approaches, and it allows to mitigate issues faced in previous works. In work [81] Anh Tuan Nguyen and Minho Park implement Semi-supervised machine learning classifier and achieved high results. Bangling Li et al. in [82] article use federated-learning based [83] approach with convolutional neural network (CNN). The model shows good result even with comparison with centralized learning model. The benefit here is ability to detect DoH tunnels without sharing traffic. Du, X. Liu, D. et al. in their work [84] create Autoencoder-based Anomaly Detection for the DoH traffic. Using of module based on nff-go framework [85] and module based bidirectional Long and Short-Term Memory to build an autoencoder network allows authors to gather features in high-speed Ethernet environment and achieve high accuracy. In the work [73] Nguyen Tuan Anh and Park Minho aim to detect DoH tunnels in

enterprise networks. Authors capture a dataset that simulate enterprise network, capture dataset with DoH produced by proxy and run Deep Learning Model Transformer. They archive high accuracy.

4.1 Assumptions

Legal DNS traffic is irregularly distributed in time. Some times user starts to do web surfing, some times run applications produce DNS requests, etc. But in case of DNS tunnel, traffic is produced regularly, with time patterns and with almost similar sizes of packets. The main assumptions is that behaviour of tunnel traffic have to show up in mixed tunnel and browser traffic connections as correlation to time based and bytes to packets based patterns.

4.2 Data analysis

According to suggested assumption data analyses showed that most essential differences between traffic produced by normal and infected machine lay in time of arriving and packet sizes based statistics. The most significant deviation for time related statistics are observed during analysing of intervals between arriving packet into the flow. On the Figure 4.2 is clearly showed, that even with a different speed rates on tunnel payload script, malicious traffic has almost shorter intervals and is not that varied as for benign traffic. Analyzing of standard deviations, variance and auto-correlation features of benign and malicious traffics proved previous hypothesis. On the Figure 4.2 the impermanence of time intervals length for benign traffic and opposite for malicious is very well illustrated.

The next part of assumption aims to packet lengths, since the DNS-crypt proxy doesn't use padding technique, analysing of lengths is possible and even very profitable. On the Figure 4.3 visualize the difference between means value of packet sizes per each flows. The next very significant otherness is found during the analysis of standard deviations for traffic classes, the illustration is provided with Figure 4.4.

The another observation related to size of packets is absolute number of bytes sends in outgoing direction. After deeper analysis confirmed the presence of this trend. Trend is shown on Figure 4.5.

The presented observations contain the most significant dissimilarities between malicious and benign traffic. During the data analyses phase the not significant differences are found also in TCP flags and delays.

4.2.1 Feature engineering

All knowledge gained during data analyses are applied in feature extraction phase. The main requirement to potential features is to use relative values.

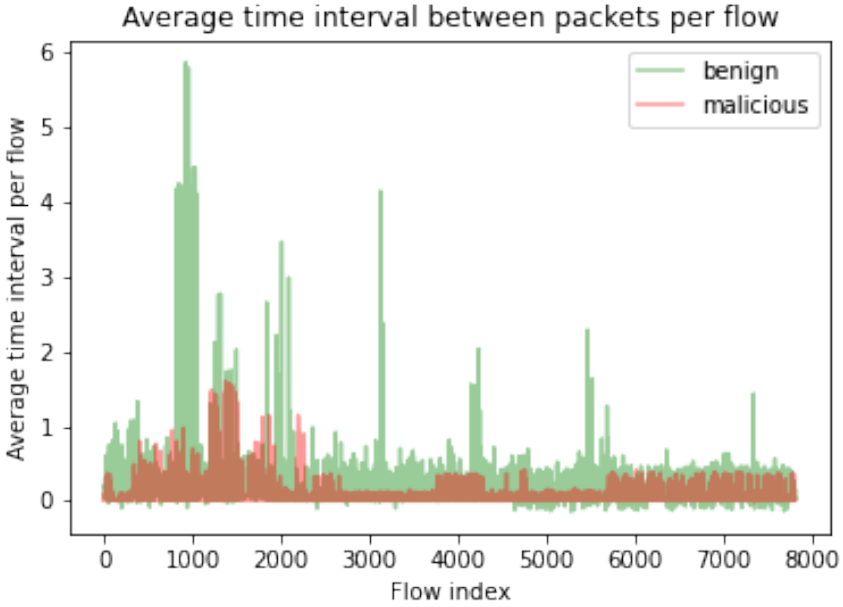

Figure 4.1: This histogram shows variances of time intervals between packets within the IP flows. Diagram contains malicious and benign traffics metrics

Table 4.1: Table of features used by prototype

Feature code	**Description**
F1	*Incoming/Outgoing bytes ration*
F2	*Incoming/Outgoing packet counts ration*
F3	*Packet size std*
F4	*Packet size mean*
F5	*Time intervals variance*
F6	*Std time between incoming packets*
F7	*Std time between outgoing packets*
F8	*TCP flags / count packets*
F9	*Ration of delayed packets / packets in a row*
F10	*Count packets variance*

The prototype has to be able to work in different networks, with different environments and conditional, but using of absolute values of traffic leads to learning the model in a way to be very personalized for dataset which is used to train, validate and estimate the model.

The features described in Table 4.1 are selected to be used in the experiment phase.

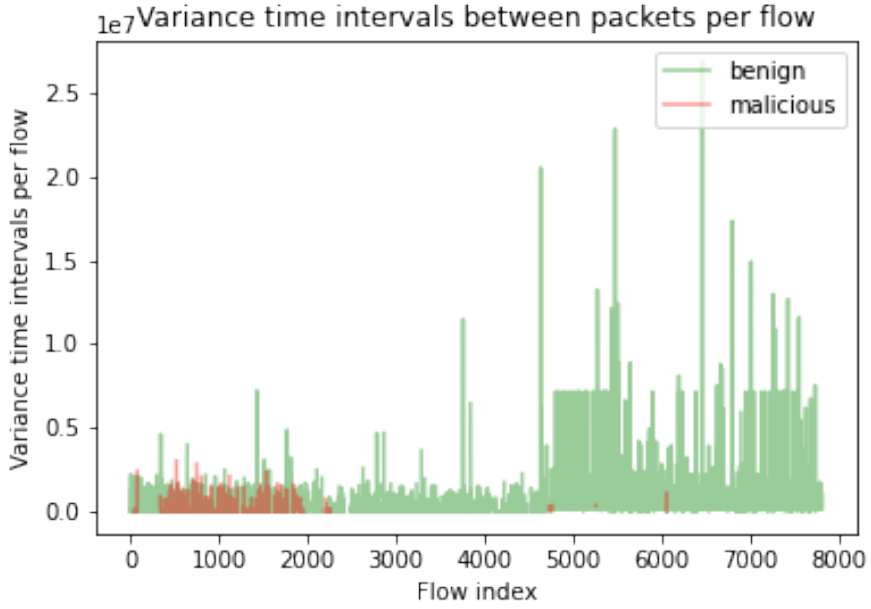

Figure 4.2: This histogram shows variances of time intervals between packets within the IP flows. Diagram contains malicious and benign traffics metric

4.2.1.1 Model choosing

After the analysis of previous works, results that have been achieved in solving of similar problem and internal principles of ML algorithms, the following three models are chosen as a candidates to be used in prototypes.

Support Vector Machine
The models are supervised, linear, binary classifiers. The model allows assign a one of two classes to each sample. This assignment is deterministically set and depends only on the feature set of the samples. It is one of the most robust methods for binary linear classification. Given a set of sample points of the form $(x_0, x_1, ..., x_n)$ with a class known, the model fits the hyperplane that maximizes the separation between both classes.

Random Forest
It is an supervised ensemble method for regression or multi class classification.The model generates multiple decision trees randomly during the training phase. For a particular sample, the model will select the same class selected by the majority of the trees.

XGBoost
XGBoost extreme gradient boosting is a library that implements the regular-

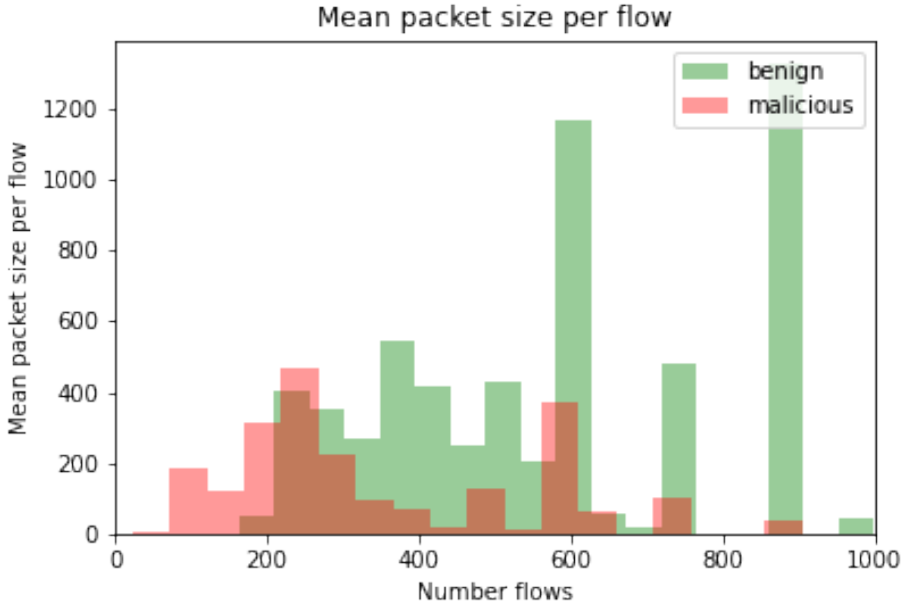

Figure 4.3: This histogram shows means of packets sizes for malicious and benign traffics

ized gradient boosting method for classification. The method combines several weak learners, while minimizing the loss function of the learners through the gradient descent method. XGBoost has gain much attention lately, mainly because was the chosen algorithm of the winners of several Machine Learning competitions.

4.3 Implementation

All data preparation and analysis performed via jupyter notebook with standard Machine Learning set of tools. The detection program is written in python.

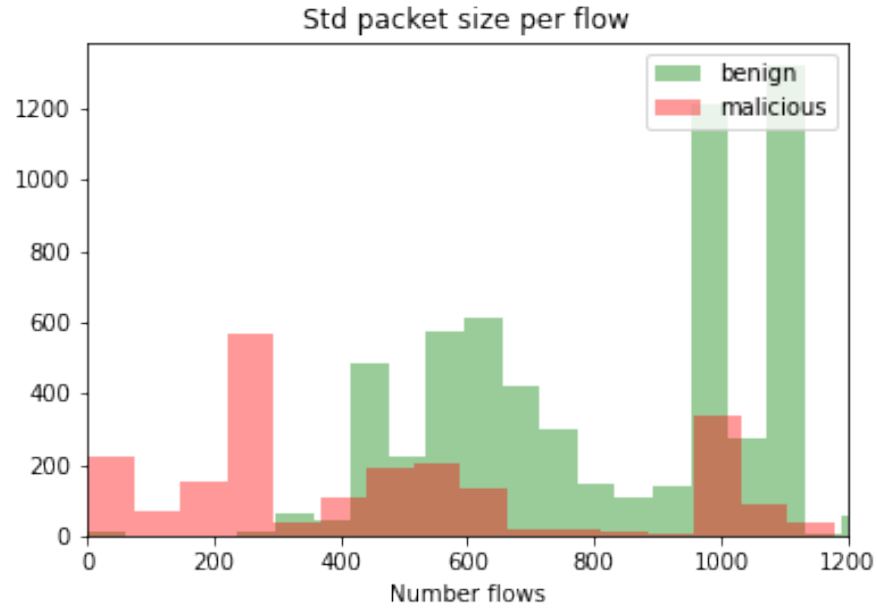

Figure 4.4: This histogram shows standard deviation of packets sizes for malicious and benign traffics

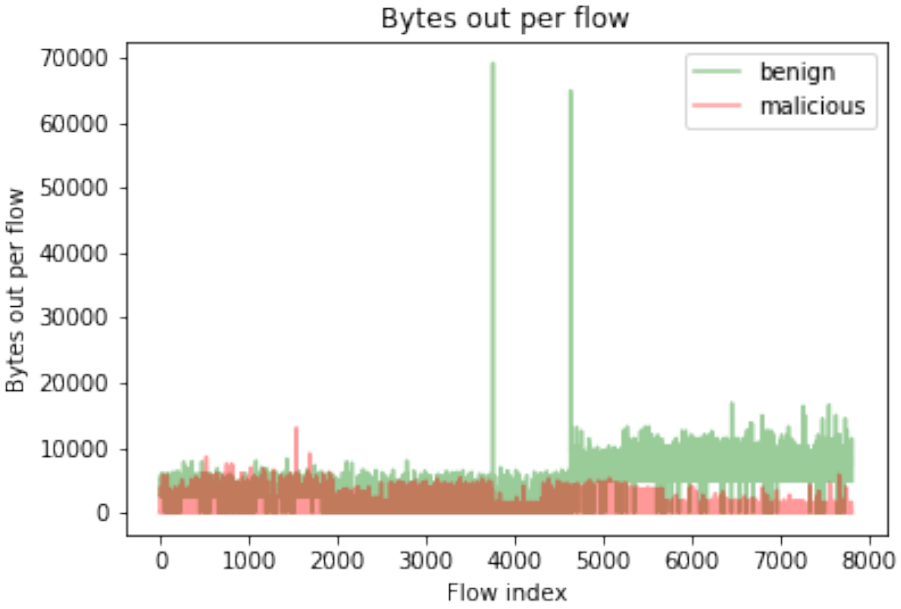

Figure 4.5: This histogram presents the difference of number of outgoing bytes in flows.

Conclusion

Historically, Domain Name Resolution has been an operative system responsibility. Moreover, though the use of a specific port and plain text messaging, the protocol is extremely manageable, allowing system administrators to filter access to certain resolution servers, to monitor the DNS queries, etc. DoH come to shift the paradigm allowing to access to Domain Name Resolution services at the application layer, using an encrypted channel and a port that can not be filtered without denying access to other lawful resources. These change have both positive and negative implications. On one hand, the use of an encrypted channel provides a protection against eavesdropping, protecting users' privacy. On the other hand, it left the door open to new vulnerabilities. Malware can now use an encrypted channel to communicate with a command and control server, or tunnels to extract data using already known techniques that have been used for a long time with the DNS protocol. As DoH is opaque to system administrators and tools that monitor DNS traffic, users don't have yet a mean for defend themselves against these new threats.

Bibliography

[1] Mockapetris, P. *DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION*. Nov. 1987. Available from: `https://www.rfc-editor.org/rfc/rfc1035.txt`

[2] Mockapetris, P. *Domain names - concepts and facilities*. Nov. 1987, doi:10.17487/RFC1034. Available from: `https://rfc-editor.org/rfc/rfc1034.txt`

[3] Deri, L.; Martinelli, M.; et al. Realtime High-Speed Network Traffic Monitoring Using ntopng. In *28th Large Installation System Administration Conference (LISA14)*, Seattle, WA: USENIX Association, Nov. 2014, ISBN 978-1-931971-17-1, pp. 78–88. Available from: `https://www.usenix.org/conference/lisa14/conference-program/presentation/deri-luca`

[4] Hofstede, R.; Čeleda, P.; et al. Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX. *IEEE Communications Surveys and Tutorials*, volume 16, no. 4, 2014: pp. 2037–2064, doi:10.1109/COMST.2014.2321898.

[5] Yuan, Q.; Tan, X. Research on Application of Artificial Intelligence in Network Security Defence. *Journal of Physics: Conference Series*, volume 2033, 09 2021: p. 012149, doi:10.1088/1742-6596/2033/1/012149.

[6] Barradas, D.; Santos, N.; et al. FlowLens: Enabling Efficient Flow Classification for ML-based Network Security Applications. 02 2021, doi:10.14722/ndss.2021.24067.

[7] Satheesh, N.; Rathnamma, M.; et al. Flow-based anomaly intrusion detection using machine learning model with software defined networking for OpenFlow network. *Microprocessors and Microsystems*, volume 79, 2020: p. 103285, ISSN 0141-9331, doi:

https://doi.org/10.1016/j.micpro.2020.103285. Available from: https://www.sciencedirect.com/science/article/pii/S0141933120304440

[8] Google. *DNS-over-HTTPS (DoH)*. Available from: https://developers.google.com/speed/public-dns/docs/doh

[9] Cloudflare. *Connect to 1.1.1.1 using DoH clients*. Available from: https://developers.cloudflare.com/1.1.1.1/encryption/dns-over-https/dns-over-https-client/

[10] Pi-hole. *DNS-over-HTTPS (DoH)*. Available from: https://docs.pi-hole.net/guides/dns/cloudflareda

[11] Hynek, K.; Cejka, T. Privacy Illusion: Beware of Unpadded DoH. In *2020 11th IEEE Information Technology, Electronic and Mobile Communication conference (IEMCON)*, 2020.

[12] Hoffman, P. E.; McManus, P. *DNS Queries over HTTPS (DoH)*. Oct. 2018, doi:10.17487/RFC8484. Available from: https://rfc-editor.org/rfc/rfc8484.txt

[13] Cloudflare. *Using DNS Wireformat*. Available from: https://developers.cloudflare.com/1.1.1.1/encryption/dns-over-https/make-api-requests/dns-wireformat//

[14] Google. *DNSJSON API for DNS over HTTPS (DoH)*. Available from: https://developers.google.com/speed/public-dns/docs/doh/json

[15] Cloudflare, Inc. DNS over HTTPS — Using JSON. Available from: https://developers.cloudflare.com/1.1.1.1/encryption/dns-over-https/make-api-requests/dns-json/

[16] Quad9 Foundation. DoH with Quad9 DNS Servers. Available from: https://www.quad9.net/news/blog/doh-with-quad9-dns-servers/

[17] AhaDNS. DNS over HTTPS (DoH). Available from: https://ahadns.com/dns-over-https/

[18] Dickson, B. Does Google Chrome's DNS-over-HTTPS (DoH) feature enhance your privacy? Dec. 2019. Available from: https://bdtechtalks.com/2019/12/11/google-chrome-dns-over-https-privacy/

[19] Cimpanu, C. DNS-over-HTTPS causes more problems than it solves, experts say. Oct. 2019. Available from: https://www.zdnet.com/article/dns-over-https-causes-more-problems-than-it-solves-experts-say/

[20] Huang, Q.; Chang, D.; et al. A Comprehensive Study of DNS-over-HTTPS Downgrade Attack. In *10th USENIX Workshop on Free and Open Communications on the Internet, FOCI 2020, August 11, 2020*, edited by R. Ensafi; H. Klein, USENIX Association, 2020. Available from: https://www.usenix.org/conference/foci20/presentation/huang

[21] Vekshin, D.; Hynek, K.; et al. DoH Insight: Detecting DNS over HTTPS by Machine Learning. ARES '20, New York, NY, USA: ACM, 2020, ISBN 9781450388337, doi:10.1145/3407023.3409192. Available from: https://doi.org/10.1145/3407023.3409192

[22] Banadaki, Y. M. Detecting Malicious DNS over HTTPS Traffic in Domain Name System using Machine Learning Classifiers. *Journal of Computer Sciences and Applications*, volume 8, no. 2, 2020: pp. 46–55, ISSN 2328-725X, doi:10.12691/jcsa-8-2-2. Available from: http://pubs.sciepub.com/jcsa/8/2/2

[23] Hjelm, D. A New Needle and Haystack: Detecting DNS over HTTPS Usage. 2019, (Accessed on 06/17/2020). Available from: https://www.sans.org/reading-room/whitepapers/dns/needle-haystack-detecting-dns-https-usage-39160

[24] Deccio, C.; Davis, J. DNS Privacy in Practice and Preparation. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, CoNEXT '19, New York, NY, USA: Association for Computing Machinery, 2019, ISBN 9781450369985, p. 138–143, doi:10.1145/3359989.3365435. Available from: https://doi.org/10.1145/3359989.3365435

[25] Lu, C.; Liu, B.; et al. An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come? In *Proceedings of the Internet Measurement Conference*, IMC '19, New York, NY, USA: Association for Computing Machinery, 2019, ISBN 9781450369480, p. 22–35, doi:10.1145/3355369.3355580. Available from: https://doi.org/10.1145/3355369.3355580

[26] García, S.; Bogado, J.; et al. List of DNS over HTTPS resolvers on the internet. Jun 2021, doi:10.5281/zenodo.4923370.

[27] Hynek, K.; García, S.; et al. Dataset of DNS over HTTPS (DoH) Internet Servers. May 2022, doi:10.5281/zenodo.6517360. Available from: https://doi.org/10.5281/zenodo.6517360

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

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

[30] RESCORLA, E. *More details on Comcast as a Trusted Recursive Resolver*. Jun 2020. Available from: https://blog.mozilla.org/en/mozilla/more-details-on-comcast-as-a-trusted-recursive-resolver/

[31] Jensen, T. *MoreWindows Insiders can now test DNS over HTTPS*. May 2020. Available from: https://techcommunity.microsoft.com/t5/networking-blog/windows-insiders-can-now-test-dns-over-https/ba-p/1381282

[32] Batenburg, B. Performance of DNS over QUIC. February 2022. Available from: http://essay.utwente.nl/89441/

[33] Hounsel, A.; Borgolte, K.; et al. Comparing the Effects of DNS, DoT, and DoH on Web Performance. 04 2020, pp. 562–572, doi:10.1145/3366423.3380139.

[34] Callejo, P.; Bagnulo, M.; et al. Measuring DoH with web ads. *Computer Networks*, volume 212, 05 2022: p. 109046, doi:10.1016/j.comnet.2022.109046.

[35] Google. *DNS-over-HTTPS performance*. Available from: https://www.samknows.com/blog/dns-over-https-performance

[36] Singanamalla, S.; Chunhapanya, S.; et al. Oblivious DNS over HTTPS (ODoH): A Practical Privacy Enhancement to DNS. 2021, pp. 575–592, doi:10.2478/popets-2021-0085. Available from: https://www.sciendo.com/article/10.2478/popets-2021-0085

[37] Livingood, J.; Antonakakis, M.; et al. *DNSCentralized DNS over HTTPS (DoH) Implementation Issues and Risks*. Mar. 2019. Available from: https://tools.ietf.org/id/draft-livingood-doh-implementation-risks-issues-03.html

[38] Hynek, K.; Vekshin, D.; et al. Summary of DNS over HTTPS Abuse. *IEEE Access*, 2022: pp. 1–1, doi:10.1109/ACCESS.2022.3175497.

[39] Hu, Z.; Zhu, L.; et al. Specification for DNS over Transport Layer Security (TLS). RFC 7858, May 2016, doi:10.17487/RFC7858, (Accessed on 05/25/2021). Available from: https://rfc-editor.org/rfc/rfc7858.txt

[40] Christian Huitema, A. M., Sara Dickinson. *DNS over Dedicated QUIC Connections*. May 2022. Available from: https://datatracker.ietf.org/doc/rfc9250/

[41] Eastlake, D. *Domain Name System Security Extensions*. Mar. 1999. Available from: https://datatracker.ietf.org/doc/html/rfc2535

[42] DNSCrypt. *CDNSCrypt A protocol to improve DNS security*. Available from: https://www.dnscrypt.org/

[43] Cloudflare. *Other ways to use 1.1.1.1*. Available from: https://developers.cloudflare.com/1.1.1.1/other-ways-to-use-1.1.1.1/

[44] of Godlua Backdoor, A. A. Jul. 2019, (Accessed on 1/06/2022). Available from: https://blog.netlab.360.com/an-analysis-of-godlua-backdoor-en/

[45] Illicit Cryptomining Threat Actor Rocke Changes Tactics, N. M. D. t. D. Oct. 2019, (Accessed on 1/06/2022). Available from: https://www.anomali.com/blog/illicit-cryptomining-threat-actor-rocke-changes-tactics-now-more-difficult-to-detect

[46] Pink, a. b. t. c. w. t. v. t. c. t. m. i. d. Oct. 2021, (Accessed on 1/06/2022). Available from: https://blog.netlab.360.com/pink-en/

[47] TEAM, T. P. T. I. PsiXBot Now Using Google DNS over HTTPS and Possible New Sexploitation Module. Sep. 2019, (Accessed on 1/06/2022). Available from: https://www.proofpoint.com/us/threat-insight/post/psixbot-now-using-google-dns-over-https-and-possible-new-sexploitation-module

[48] to Know, F. M. A. Y. N.; to Act Now. May 2021, (Accessed on 1/06/2022). Available from: https://www.threatmark.com/flubot-banking-malware/

[49] Ferrell, J. Hiding In Plain Sight. Jun. 2020, (Accessed on 1/06/2022). Available from: https://blog.huntresslabs.com/hiding-in-plain-sight-556469e0a4e

[50] Hammond, J. Hiding In Plain Sight Part 2. Aug. 2020, (Accessed on 1/06/2022). Available from: https://blog.huntresslabs.com/hiding-in-plain-sight-part-2-dfec817c036f

[51] Sharma, A. Attackers abuse Google DNS over HTTPS to download malware. Sep. 2020, (Accessed on 1/06/2022). Available from: https://www.bleepingcomputer.com/news/security/attackers-abuse-google-dns-over-https-to-download-malware/

[52] DoHC2. (Accessed on 1/06/2022). Available from: https://github.com/SpiderLabs/DoHC2

[53] godoh. (Accessed on 1/06/2022). Available from: https://github.com/sensepost/godoh

[54] Jacobs, L. *Waiting for goDoH*. Oct. 2018. Available from: https://sensepost.com/blog/2018/waiting-for-godoh/

[55] iodine. (Accessed on 1/06/2022). Available from: https://github.com/yarrick/iodine/tree/iodine-0.7

[56] dnscat2. (Accessed on 1/06/2022). Available from: https://github.com/iagox86/dnscat2

[57] Cimpanu, C. *Iranian hacker group becomes first known APT to weaponize DNS-over-HTTPS (DoH)*. Aug. 2020. Available from: https://www.zdnet.com/article/iranian-hacker-group-becomes-first-known-apt-to-weaponize-dns-over-https-doh/

[58] Deckelmann, S. DNSExfiltrator. https://github.com/Arno0x/DNSExfiltrator, April 2018, (Accessed on 11/03/2020).

[59] bamsoftware. *dnstt*. Available from: https://www.bamsoftware.com/software/dnstt/index.html

[60] Stevens, D. Downloading Executables Over DNS: Capture Files. 2019 Aug., (Accessed on 1/06/2022). Available from: https://blog.didierstevens.com/2019/08/07/downloading-executables-over-dns-capture-files/

[61] YARA. *Welcome to YARA's documentation!* Available from: https://yara.readthedocs.io/en/stable/index.html

[62] Hutchison, S. DNS Over HTTPS: 3 Strategies for Enterprise Security Monitoring. Aug. 2021, (Accessed on 1/06/2022). Available from: https://insights.sei.cmu.edu/blog/dns-over-https-3-strategies-for-enterprise-security-monitoring/

[63] Agency, U. N. S. Adopting Encrypted DNS in Enterprise Environments. Jan. 2021, (Accessed on 1/06/2022). Available from: https://media.defense.gov/2021/Jan/14/2002564889/-1/-1/0/CSI_ADOPTING_ENCRYPTED_DNS_U_OO_102904_21.PDF

[64] Yu, A.; Xiaohong, X.; et al. CIRA-CIC-DoHBrw-2020. July 2012. Available from: https://www.unb.ca/cic/datasets/dohbrw-2020.html

[65] Montazeri Shatoori, M.; Davidson, L.; et al. Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic. In *2020 IEEE Intl Conf: DASC/PiCom/CBDCom/CyberSciTech*, 2020, pp. 63–70, doi: 10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00026.

[66] Haddon, D. A. E.; Alkhateeb, H. Investigating Data Exfiltration in DNS Over HTTPS Queries. In *2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3)*, 2019, pp. 212–212, doi:10.1109/ICGS3.2019.8688016.

[67] Zebin, T.; Rezvy, S.; et al. An Explainable AI-based Intrusion Detection System for DNS over HTTPS (DoH) Attacks. 01 2022, doi:10.36227/techrxiv.17696972.v1.

[68] Ahmed Almusawi, H. A. DNS Tunneling Detection Method Based on Multilabel Support Vector Machine. 2018, (Accessed on 11/06/2020). Available from: `https://doi.org/10.1155/2018/6137098`

[69] K, J.; ; et al. DNS Over HTTPS network traffic. 2021, doi:10.21227/96ea-2055. Available from: `https://dx.doi.org/10.21227/96ea-2055`

[70] Vekshin, D.; Hynek, K.; et al. Dataset used for detecting DNS over HTTPS by Machine Learning. May 2020, doi:10.5281/zenodo.3906526, This work was supported by the European Union's Horizon 2020 research and innovation program under grant agreement No. 833418 and also by the Grant Agency of the CTU in Prague, grant No. SGS20/210/OHK3/3T/20 funded by the MEYS of the Czech Republic. Available from: `https://doi.org/10.5281/zenodo.3906526`

[71] Kwan, C.; Janiszewski, P.; et al. 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*, FOCI '21, New York, NY, USA: Association for Computing Machinery, 2021, ISBN 9781450386401, p. 37–42, doi:10.1145/3473604.3474563. Available from: `https://doi.org/10.1145/3473604.3474563`

[72] Zhan, M.; Li, Y.; et al. Detecting DNS over HTTPS Based Data Exfiltration. *Comput. Netw.*, volume 209, no. C, may 2022, ISSN 1389-1286, doi:10.1016/j.comnet.2022.108919. Available from: `https://doi.org/10.1016/j.comnet.2022.108919`

[73] Anh, N. T.; Minho, P. DoH Tunneling Detection System for Enterprise Network Using Deep Learning Technique. *Applied Sciences*, volume 12, no. 5, 2022, ISSN 2076-3417, doi:10.3390/app12052416. Available from: `https://www.mdpi.com/2076-3417/12/5/2416`

[74] Selenium. *Selenium automates browsers.* Available from: https://www.selenium.dev/

[75] Tcpdump. *MAN PAGE OF TCPDUMP.* Available from: https://www.tcpdump.org/manpages/tcpdump.1.html

[76] Vagrant. *Vagrant Development Environments Made Eas.* Available from: https://www.vagrantup.com/

[77] dnscrypt proxy. (Accessed on 1/06/2022). Available from: https://github.com/DNSCrypt/dnscrypt-proxy

[78] NEMEA. *ipfixprobe - IPFIX flow exporter.* Available from: https://github.com/CESNET/ipfixprobe

[79] Buczak, A. L.; Hanke, P. A.; et al. Detection of Tunnels in PCAP Data by Random Forests. In *Proceedings of the 11th Annual Cyber and Information Security Research Conference*, CISRC '16, New York, NY, USA: Association for Computing Machinery, 2016, ISBN 9781450337526, doi: 10.1145/2897795.2897804. Available from: https://doi.org/10.1145/2897795.2897804

[80] Alenezi, R.; Ludwig, S. Classifying DNS Tunneling Tools For Malicious DoH Traffic. 12 2021, doi:10.1109/SSCI50451.2021.9660136.

[81] Nguyen, A.; Park, M. Detection of DoH Tunneling using Semi-supervised Learning method. In *2022 International Conference on Information Networking (ICOIN)*, Los Alamitos, CA, USA: IEEE Computer Society, jan 2022, ISSN 1976-7684, pp. 450–453, doi:10.1109/ICOIN53446.2022.9687157. Available from: https://doi.ieeecomputersociety.org/10.1109/ICOIN53446.2022.9687157

[82] Li, B.; He, S.; et al. Detecting DoH tunnels with privacy protection using federated learning. In *International Conference on Network Communication and Information Security (ICNCIS 2021)*, volume 12175, edited by Y. Liu; F. Wen, International Society for Optics and Photonics, SPIE, 2022, pp. 133 – 141, doi:10.1117/12.2628461. Available from: https://doi.org/10.1117/12.2628461

[83] Furukawa, N.; Jeno, G. *What is Federated Learning?* 06 2022.

[84] Du, X.; Liu, D.; et al. Design of an Autoencoder-based Anomaly Detection for the DoH traffic System. 05 2022, pp. 763–768, doi:10.1109/CSCWD54268.2022.9776029.

[85] for Go (former YANFF), N. F. F. (Accessed on 1/06/2022). Available from: https://github.com/intel-go/nff-go

Appendix A

Acronyms

DNS Domain Name System

DoH DNS over HTTPS

HTTPS Hypertext transfer protocol

IT Information Technology

IP Internet Protocol

TLD Top Level Domain

SLD Second Level Domain

ICANN Internet Corporation for Assigned Names and Numbers

TCP Transmission Control Protocol

RR Resource Record

DKIM Domain Key Identified Mail

SPF Sender Policy Framework

TTL Time To Live

DPI Deep packet inspection

NGFW Next-generation firewall

UTM Unified threat management

TLS Transport Layer Security

MITM Man-in-the-middle

IT Information Technology

IT Information Technology

OS Operation system

Appendix B

Contents of enclosed CD

- `readme.txt` ..... the file with CD contents description
- `exe` ..... the directory with executables
- `src` ..... the directory of source codes
  - `wbdcm` ..... implementation sources
  - `thesis` ..... the directory of LaTeX source codes of the thesis
- `text` ..... the thesis text directory
  - `thesis.pdf` ..... the thesis text in PDF format
  - `thesis.ps` ..... the thesis text in PS format