Detection of HTTPS Encrypted DNS Traffic

Frank Nijeboer
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands

ABSTRACT

The Domain Name System (DNS) is one of the cornerstones of the Internet. However, DNS requests are performed without encryption, resulting in privacy and security issues such as the possibility for eavesdropping and spoofing the DNS response. These are tackled by DNS protocol extensions such as DNS over HTTPS (DoH) that provide encryption over HTTPS for DNS queries. DoH has been around since 2018 and since then some browsers such as Firefox and Chrome have been experimenting with it. Therefore, it is time to introspect the privacy and security that is provided by DoH. This research provides an analysis of the privacy that is provided by DNS over HTTPS.
In this research, Firefox is used to connect to a set of DoH resolvers over multiple test sessions. Then, the captured traffic is analyzed based on temporal features and packet sizes to detect DoH traffic.
This research uncovers a technique to filter DoH queries from other HTTPS traffic using packet size related features. Furthermore, an initial step is shown that enables outside listeners to determine queried websites based on patterns in DoH packet sizes. Lastly, this research also provides suggestions for improving DoH by adding padding to the queries to possibly enhance privacy benefits provided by DoH.
The findings in this research show that DNS privacy still faces challenges and that a thorough analysis of the threats that face DoH privacy is required.

Keywords

DNS, DNS over HTTPS, Encryption, Privacy, Security

1. INTRODUCTION

The increasing privacy awareness of the public has driven the Internet Engineering Task force to enhance the privacy of the Domain Name System (DNS). DNS queries contain information about the websites that are visited by actors on the Internet. This can be correlated to obtain insights in user behaviour such as: what websites the user visits, what applications are used by the user and sometimes also the people that the user corresponds with [1]. Only in recent years has the DNS changed to provide confidentiality for the information that is embedded within DNS packets. Two of the measures that are taken to ensure a more privacy friendly DNS are facilitated by the combining Transport Layer Security (TLS) and Hypertext Transfer Protocol Secure (HTTPS) with DNS. The outcomes of those combinations are DNS over TLS (DoT) [2] and DNS over HTTPS (DoH) [3].

This research will focus on DoH, due to the fact that DoT has its own port (853) and can therefore be easily distinguished among other network traffic. DoH, on the other hand, makes use of port 443 along with the rest of HTTPS traffic [4], concealing it more from listeners. Furthermore, DoH is currently being implemented by the major browsers Firefox [5] and Google Chrome [6] and also Apple has stated that they will enable DoH on their iPhones [7].

There exists controversy about the implementation of DoH within the cyber security community, some stating that DoH could create problems for organisations that use the DNS to fight malware on their network by blocking certain DNS requests and more risks which are described by Livingood et al. [8]. By exposing the limitations of DoH, this research can provide a useful argument in the discussion about the implementation of DoH, because some researchers argue that DoH might not provide the privacy benefits that it is intended for while it does cause issues for network managers [9].

The main focus of this research is to find an answer to the question:

To what extent does DNS over HTTPS prevent on-path devices from eavesdropping and interfering with DNS requests?

An analysis of network traffic, gathered from a Virtual Machine using Firefox with DoH, provides the answer to this question. In this research we show that:

1. *Usage of DoH can be detected*
2. *DoH Traffic can be detected among other HTTPS traffic. Also the packet statistics that point to DoH are uncovered.*
3. *Pattern analysis on multiple DoH queries could provide insight in visited websites.*
4. *How DoH can be improved to enhance its privacy benefits.*

The first contribution is discovered by analyzing captured network traffic and looking at regular DNS behaviour in the capture.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
33rd Twente Student Conference on IT July 3rd, 2020, Enschede, The Netherlands.
Copyright 2020, University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science.

We show the second contribution by analyzing packet lengths that point to DNS over HTTPS. Also, a script is created that filters DoH traffic from other HTTPS traffic.

The third contribution is shown by creating a program that analyzes network traffic and patterns in the DoH packets that are filtered by the second contribution.

The fourth contribution suggests improvements for DoH as well as the implications that this research has for security professionals.

Furthermore, Section 2 provides background information regarding DoH, Section 3 discusses related literature and Section 4 explains the techniques that are used to gather the results. We display the results in Section 5, followed by a critical analysis in Section 6 and finally conclude this paper in Section 7.

2. BACKGROUND

This Section reviews the background of DNS and DoH by exploring available work on DNS and DoH.

2.1 DNS

The Domain Name System, in its elemental form, translates human readable text to an IP address which can be understood by computers when a user accesses a website [10][11]. DNS queries are formed from a maximum of 5 data types. Two of these data types are always present in a DNS message: the Header, which contains information about the DNS query, and a Question to the DNS server. Furthermore, in a successful DNS response the optional Answer field, containing the answer to the query, is generally present. To determine an IP address based on input from a human, the browser sends a query to a DNS resolver, which will then find the IP address that answers the question in the query. To find the answer, the DNS resolver asks one or more authoritative name servers, which it finds by asking root name servers and TLD (Top Level Domain) name servers for the query question [12]. This research focuses mostly on the connection between the client and the recursive resolver.

2.2 DNS over HTTPS

Before 2018, most DNS queries were performed by sending plaintext messages over the UDP or TCP protocol on port 53, following the guidelines of RFC1035 in 1987 [10]. Without encryption, Internet Service Providers (ISPs) can log these DNS queries and any other listener on the network can eavesdrop on the queries that are performed by users of the Internet.

Besides just eavesdropping, DNS responses can also be manipulated by third parties to return wrong answers to queries, possibly leading users to malicious websites, as mentioned in RFC 7626 [13].

DoH provides better privacy and security by using HTTPS, which uses TLS to encrypt the packets that are sent over the connection between the client and the DNS resolver, and additionally hiding the DNS queries between regular HTTPS traffic. Especially the latter should make it harder for third parties to determine that DNS queries are made.

DoH encapsulates DNS in a HTTP GET or HTTP POST method. In the case of GET, the query information will be presented in the URI, while a POST DoH request has the information as message body. Also the client should include a HTTP Accept header to indicate the kind of response that it understands. The default method within browsers is the HTTP POST method.

It should be noted that there also exists DNS over TLS (DoT), which uses a dedicated port (853) and is always controlled by the operating system, whereas, in the case of DoH, the operating system is circumvented. DoT provides the same encryption as DoH does, however due to their experience with HTTPS, most browsers opt to utilize DoH instead of DoT. Some companies, such as Apple enable their users to choose between DoT and DoH [7]. The fact that DoH is more actively adopted at the moment and one of its main features is to hide DNS queries to prevent blocking makes it better suited for this research.

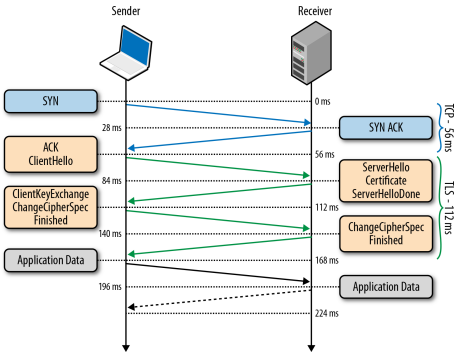

Figure 1. Visualization of HTTPS Handshake and Data Exchange [14]

2.3 HTTPS

HTTP over TLS or Hypertext Transfer Protocol Secure or HTTPS as it is most often called by users of the protocol, is a protocol that secures HTTP traffic to prevent third parties from eavesdropping or altering content that is returned as the HTTP response. Before HTTPS, ISPs could add their own advertisements to any site by altering the HTTP response data. They were also able to see all the website data that was queried, being able to see exactly what their clients were doing on the Internet.

HTTPS makes those practices impossible by using TLS to encrypt the HTTP content that is sent between user and website. To do this, the client and server first have to perform a TLS handshake as described in RFC2818 [4].

Figure 1 shows the HTTPS protocol in steps. The top part describes the TCP connection setup phase. Once the connection is acknowledged by both client and server, the TLS phase starts. During this phase the server presents the client with its certificate, which the client can check. Furthermore, a session key is generated to perform symmetric encryption during the later data exchanges. This session key is encrypted on the client side with the server's public key, to ensure that only the server is able to decrypt the session key. After this phase, HTTP data can be exchanged using the session key to ensure integrity and confidentiality. All the steps in this process are also performed by DoH and could result into behaviour which can be used to detect DoH.

2.4 Firefox and DoH

Firefox defaults users in the US to have a DoH resolver to shield them from online tracking. Its default resolver is 1.1.1.1 from Cloudflare, but Firefox resorts to regular DNS when an address cannot be obtained with DoH [5].

Furthermore, Firefox uses the canary domain *use-application-dns.net*. If Firefox cannot access this domain, it will resort to regular DNS which can be controlled by the network provider [15].

3. RELATED WORK

In this Section, related studies to this work are briefly discussed. Useful information from HTTPS detection, TOR traffic detection and VPN detection papers are used to gather insight in promising approaches for this research.

In *A New Needle and Haystack* [16], Hjelm discusses the cyber security implications of DoH, mostly focusing on Command and Control messages which are send over DoH. Furthermore, he provides some analysis tools which prove useful for detecting out of order behaviour on the network which could point to DoH. In the end though, he mostly focuses on the IP addresses which are called by the clients to detect DoH, which could become less relevent as more DoH servers are deployed on the Internet.

In *Automated Website Fingerprinting through Deep Learning* [17], Rimmer et al. research the possibility of fingerprinting websites that are accessed via HTTPS in the TOR browser. In their research they start by capturing their own HTTPS traffic to a website. This is fed into a deep learning Artificial Intelligence Agent for training purposes. This traffic results in a *fingerprint* that they use to analyze traffic which is captured from other computers in the network. Using this fingerprint they can de-anonimize the HTTPS traffic and see what the clients in the network are doing, as long as the deep learning agent has been trained with data of the visited website.

Di Martino et al. [18] also investigate the possibility for website fingerprinting, but their focus is on social networks. Their research shows that this method can be used for social networks as well. This shows that fingerprinting can be used to detect HTTPS traffic content, which could prove useful in this research.

In *An Investigation on Information Leakage of DNS over TLS*, Houser et al. analyze the confidentiality and integrity of DoT [19]. In their research they show that they can infer the visited websites based on temporal patterns and packet sizes of the DoT requests. They show that this method can be highly effective to deduce visited websites and that information leakage with DoT is possible. This technique can prove useful when evaluating the privacy that is provided by DoH as well.

4. METHODOLOGY

Based on the background and related work, this research will be an examination of DoH traffic within a lab environment. This approach is best suited for this research, because the parameters on both the DNS client and server can be controlled to generate different datasets suitable for analysis. Additionally, this approach ensures that no privacy violations occur, since no real traffic is used.

4.1 Dataset Generation

To gather a representative dataset of real DoH traffic, a test bed will be set up for this research. This test setup contains a client (browser) and a DNS resolver.

The client visits 50 websites that are generated by the Alexa Top 50. Alexa ranks web-pages based on their popularity: they are calculated based on the average daily time spent on the site and the number of page views within the past month [20]. The list of domain names chosen for this research consists of the Alexa Ranking on May 25th 2020. We publish this list here [21].

Having the client visit these websites generates a data set of traffic that tries to resemble real user traffic that happens within a network

4.1.1 Client

Firefox Version 76.0.1 is the web browser of choice during this research. This is the most relevant browser for DoH analysis, since it turns on DoH by default for its users in the United States. Furthermore it allows for extensive configuration of DNS settings so that multiple different browser setups can be tested. Also, it is further than Chrome in the adoption of DoH, therefore granting better insight in the final DoH solution.

The client that is observed and of which the traffic will be captured is a Virtual Machine running Xubuntu 20.04 LTS [22].

We use the Geckodriver [23] to control the visited websites and to make sure that the every test is carried out in the same manner.

4.1.2 Resolver

During this research several DoH resolvers are used for data generation. These are: Cloudflare, NextDNS, Google, Knot Resolver and a regular DNS resolver. Knot Resolver stands out from this list, as it is a self hosted resolver that supports regular DNS as well as DoH.

Cloudflare is the default DoH resolver for Firefox and is therefore interesting to look at. Due to the support from Firefox, the Cloudflare resolver will probably obtain a very large market share in the handling of DoH traffic in the future. Also at the moment Cloudflare is used as DoH resolver for Firefox users in the United States as mentioned in Section 2.4.

NextDNS is the second option that is also marked as a Trusted Recursive Resolver by Firefox, making it one of two resolvers that Firefox users can select as DoH resolver from the regular settings panel.

The Google resolver is currently the most used DNS resolver with almost 15% of all DNS queries going to 8.8.8.8 [24]. Google has also started to support DoH and the current market share that the Google resolver has indicates that also their DoH resolver will probably have a large market share in the future.

Knot Resolver is a self-hosted resolver that is running in another Virtualbox VM on Ubuntu Server.

For the regular DNS resolver we use a default local ISP DNS resolver to compare the DoH traffic to regular DNS traffic.

4.1.3 Traffic Capturing

When using an external resolver as DoH resolver, Wireshark is used to capture the traffic between the VirtualBox client and the Internet. Some traffic does not leave the Virtual Machine's host computer and will therefore not be captured by tools such as Wireshark or tcpdump. Therefore traffic is also captured by VirtualBox's traffic capturing tool.

As shown in Figure 2, the captured traffic is located between the client and the resolver to simulate a real situation in which ISPs are between the client and the DNS resolver. This setup will grant the researcher the same vantage point as an ISP would normally have.

4.2 Analysis

After gathering the network traffic data, it is analyzed based on visual and statistical characteristics. Useful data includes packet properties such as packet length, destination port and destination IP. On this data, statistical analysis is performed for examination. Also, techniques,

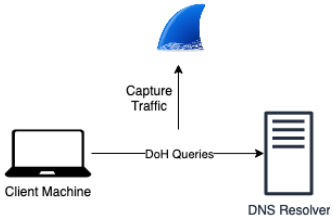

Figure 2. Test Setup

such as fingerprinting the TLS handshake, described in Section 3 are used.

Then, based on those findings, a script is created to detect DoH packets and save those to a new file which is compared to decrypted traffic to determine the detection accuracy.

Furthermore, when visiting the same websites, a DoH setup should have a similar number of DoH queries to the number of DNS queries in a regular DNS setup. These numbers are also compared to see if the DoH suspect is the real DNS server.

4.2.1 Decrypting DoH Traffic

To gather more insight into DoH traffic and to show data more clearly, some DoH packets can also be decrypted. For this, Wireshark is used and a file is kept to hold the SSL keys from the client side. Packets send from the client can then be decrypted and inspected in Wireshark.

4.2.2 Fingerprinting

Often an IP address shows much information about a website or a DNS resolver. For example, the IP addresses 8.8.8.8 and 8.8.4.4 are easily identified as the Google DNS resolvers.

However, applications on a computer can implement their own DoH resolver, independent from the operating system's DNS resolver. This means that a client could have multiple apps that all use a different DoH resolver, all on different IP addresses [9]. However, the DoH resolver implementation might be the same on different IP addresses. For example, Knot Resolver can be hosted by multiple companies, each on a different IP address.

Fingerprinting could provide an answer to find DoH resolvers indepent of the IP address that they use. JA3 and JA3s are both Python programs, developed by Salesforce, that allow for fingerprinting specific programs and clients [25].

To use it, a .pcap file is read and fed into the JA3 algorithm. This algorithm then determines all the fingerprints, which can later be used to detect the same program again.

Fingerprinting with JA3 uses features from TLS Client Hello packets during the TLS handshake phase, described in Section 1 [25]. These packets are used whenever a new TLS connection has been set up, in the case of this DoH: a new HTTPS connection. JA3 gets the values from certain fields and generates a hash based on those values. The fields used are: Version, Accepted Ciphers, List of Extensions, Elliptic Curves, and Elliptic Curve Formats.

For this research a .pcap file, described in Section 4.1.3, are read by the JA3 algorithm. The DoH resolver's IP address in this file is known to the researcher and the fingerprint is determined. Then the same fingerprint is used to detect the same resolver, when it is found on a different IP address.

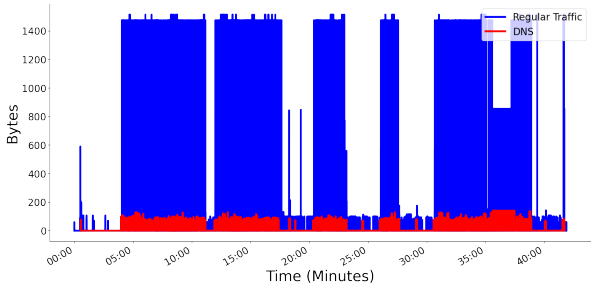

Figure 3. Outgoing DNS traffic when DoH is not used

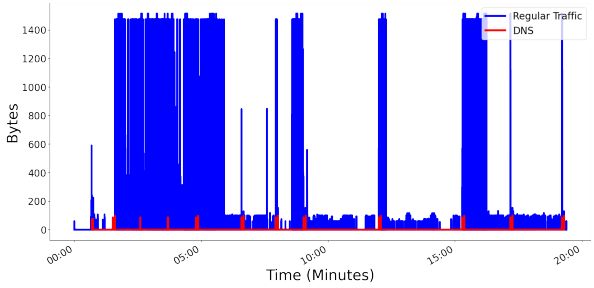

Figure 4. Outgoing DNS Traffic when DoH is used

4.3 Test Procedure

Testing is done in steps to reproduce the same environment on every test run. For every resolver, described in Section 4.1.2 these steps are performed:

1. Start Virtual Machine
2. Run script that starts Firefox with the current resolver as DoH/DNS resolver
3. Automatically visit the top 50 websites from Alexa
4. Quit Virtual Machine

For testing, browser cache has been turned off and in the case of a DoH resolver, fallback to DNS has been disabled. Enabling fallback would mean that NXDOMAIN queries will be resend as regular DNS. This fallback would also grant network monitors insight into traffic by analyzing the mistyped domains.

5. RESULTS

In this Section, the relevant results to the questions from in 1 are answered. Sub-questions are answered first, as this delivers a solid foundation for answering the main research question: *To what extent does DNS over HTTPS prevent on-path devices from eavesdropping and interfering with DNS requests?*

5.1 Detecting DNS over HTTPS

DoH can be recognized among other HTTPS traffic, when using Firefox as a browser. An observer can be notified to this by observing the results from the next Sections.

5.1.1 Lack of regular DNS

When using DoH in Firefox, very little regular DNS traffic is being generated, compared to a setup with regular DNS. As seen in Figure 3, there is the usual DNS traffic that correlates to the other traffic in the network: more active during an increase of other traffic and less active during a decrease of other traffic. However, when looking at Figure

Header	Value
Method	POST
Path	/dns-query
Authority	mozilla.cloudflare-dns.com
Schema	https
accept	application/dns-message
accept-encoding	*empty*
content-type	application/dns-message
content-length	*Variable per query*
cache-control	no-store, no-cache
pragma	no-cache
te	trailers

Table 1. Decrypted DoH Headers from a setup with Cloudflare as DoH Resolver

DNS Resolver	Header Length
Cloudflare	110-114
Google	122-126
NextDNS	115-119
Knot Resolver	117 -121

Table 2. Header sizes per DoH resolver

4, we see that the amount of DNS traffic has drastically decreased. The number of DNS messages decreased by 99.82% and 99.10% when the client machine used DoH instead of DNS for web-browsing the same websites. This could be an obvious indicator that DoH is being used by a client.

5.1.2 *Packet Size Indication*

Packets that are send by DoH always have packet lengths that differ largely from other traffic on port 443. The explanation for this is that a DoH request will always have the same format, which is also described in Section 2.2, but in more detail here.

Header Packets

The first packet in a DoH request will always send the headers to the DoH receiver. A decrypted example of the headers can be found in Table 1.

Depending on the DoH resolver that is chosen these headers result in a packet with a static size. When testing with Cloudflare these 'Header Packets' result in a packet size of either 110 or 114.

Table 2 shows the header size per DoH resolver. The table shows that the header length differs per resolver and that it can have two values. This is because the *Content-Length* header field can be encoded in two ways, which differs by 4 bytes and results in one of the two values.

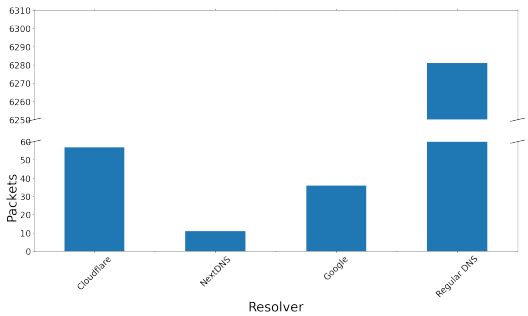

Figure 5. Number of DNS queries in capture file grouped by resolver

DoH Query Packets

Closely after sending the DoH Header packet, Firefox will send the actual DoH query packet. This packet contains a regular DNS query with all the information that is possible to send inside a DNS query. Most of the data in these packets have a static size. These include (with Cloudflare as DoH Resolver): Transaction ID (2 bytes), Flags (2 bytes), Number of Questions (2 bytes), Number of Answer RRs (2 bytes), Number of Authority RRs (2 bytes), number of Additional RRs (2 bytes), Additional records (19 bytes). Finally, there is one field of variable length: Queries.

Depending on the length of the name that is queried the packet size can be smaller or larger. The maximum size of a name is 255 bytes. In the data set, there is a DoH query of length 142, where the name is 26 bytes. Therefore we can set an upper bound of a DoH query packet length to $142 + (255 - 26) = 371$. The lower bound will then be $142 - 26 = 116$. However, the upper and lower bounds are extremes and most queries have a length between 133 and 170.

Detecting DoH

The fact that the packet size of DoH queries and the header packets have a relatively static packet size can be exploited by an observer to filter out DoH queries and see what IP address the clients in the network are using as a DoH resolver.

A simple script is therefore enough to filter DoH queries from a .pcap file, or a live monitor session. For this research the following algorithm was developed:

Algorithm 1 Find DoH packets

```
Require: capture // List of packet in a capture file
Require: minimum // Minimal header packet length
Require: maximum // Maximum header packet length
  getnext = False
  result = empty list
  for packet in capture do
      if getnext then
          if 120 <= packet.size <= 220  then
              // The packet is within DoH boundaries
              result = result + packet
          end if
          getnext = False
      end if
      if minimum <= packet.size <= maximum then
          // This is probably the header for DoH
          getnext = True
      end if
  end for
  return result
```

In this script, the variable result will be filled with packets that are marked as DoH packets, based on header packet sizes. The input is a list *capture*, consisting of all packets in a capture file. Furthermore, a *minimum* and *maximum* size are given as a parameter to accurately filter out the header packets. For example, if Cloudflare is the DoH resolver than *minimum* should be set to 110 and *maximum* should be set to 114 to get the most accurate results.

This method of detecting DoH query suspects delivers accurate results as seen in Figure 6. This Figure shows the DoH query suspected traffic in red, while other network traffic is blue. This Figure closely resembles the DNS graph from Figure 3.

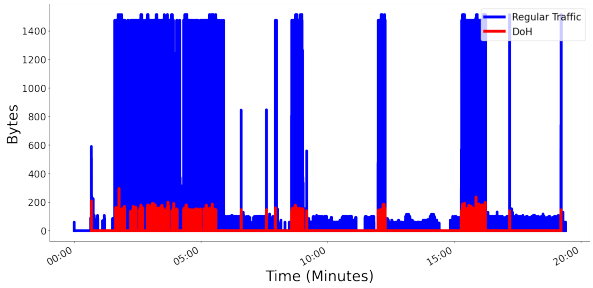

Figure 6. DoH Traffic Mapped to All Traffic

Resolver	Number of Recognized Outgoing Queries
Regular DNS	4202
Cloudflare	3181
NextDNS	2974
Google	3026
Knot Resolver	3089

Table 3. Outgoing DNS packets in capture file

Table 3 shows the number of outbound DNS/DoH queries that the algorithm recognized per Resolver.

Furthermore, we carried out an analysis of this script and compared it to decrypted DoH traffic to determine what percentage of DoH packets had been correctly identified and which packets had not been detected by the script. On average, 2.13% of the DoH queries was missed when determining DoH requests to 4 resolvers. No false positives were found during these tests. Figure 7 shows the results for each DoH resolver.

5.1.3 Fingerprinting

Section 4.2.2 describes JA3 fingerprinting techniques to use for detecting DoH resolvers. In this Section, the JA3 results are analyzed.

When running the JA3 algorithm the JA3 digest of the known Cloudflare DoH resolver has been determined[1].

The default Cloudflare DoH resolver has IP addresses: *104.16.248.249*, *104.16.248.248*, *104.16.248.248*, and also *1.1.1.1* points to the Cloudflare DoH resolver.

The JA3 digest corresponds to all of those resolvers 100% of the time. Furthermore, it matched with 213 other servers that were not Cloudflare's DoH resolver in a dataset that consisted of 281 connections. Because of this abundance of false positives, fingerprinting based on Client Hello messages is not a suitable tool to determine DoH resolvers.

[1]Digest for Cloudflare: b20b44b18b853ef29ab773e921b03422

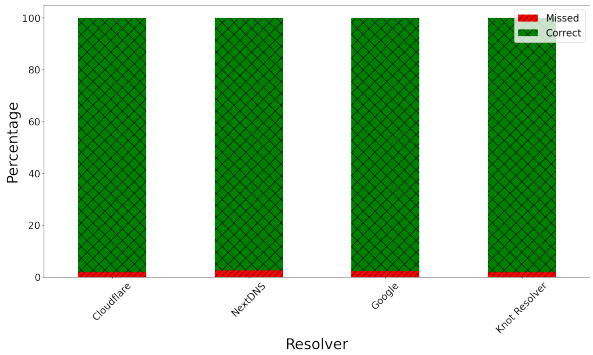

Figure 7. Percentage of correct and missed DoH packets

DoH Query	Packet Length
A youtube.com OPT	133
A www.youtube.com OPT	137
AAAA www.youtube.com OPT	137
A i.ytimg.com OPT	133
AAAA i.ytimg.com OPT	133
A accounts.google.com OPT	141
AAAA accounts.google.com OPT	141
A yt3.ggpht.com OPT	135
AAAA yt3.ggpht.com OPT	135

Table 4. DoH Query pattern for *youtube.com*

5.2 Detecting Individual Queries

Breaking HTTPS was not in the scope of this research. As such, no technique is discovered to detect the contents of individual DoH queries.

However, a pattern exists between a website and the DNS queries that are performed when such a website is visited. This technique is also described by Houser et al. in their research about the provided privacy by DoT [19]. *Youtube.com* is taken as an example here.

When visiting this website, Firefox must first perform the DoH query for *youtube.com*. However, before loading the website, multiple other DoH queries are done to gather dependencies for the website. This creates a pattern that is always the same when a website is initially loaded. The order in which these queries occur is also fixed. For *youtube.com* this is shown in Table 4.

Once these patterns are known, visited websites can be deduced independently from SNI or other indicators outside of DNS/DoH.

As a proof of concept a simple algorithm has been written that is able to detect whether or not *youtube.com* has been queried in a .pcap file. This algorithm takes as input the packet sizes of the packets that must be queried: in this case the first 5 packet sizes from Table 4. Then it preprocesses the .pcap file to filter out the DoH queries by using the technique from Section 5.1.2, after which it looks through the output to find the pattern for *youtube.com*.

This technique has been tested on 5 .pcap files that were generated by running the script described in Section 4.3 with the Cloudflare resolver. Then the results of the algorithm were evaluated by comparing it to the decrypted traffic. The algorithm was able to deduce that *youtube.com* had been queried for 4 out of those file types, giving it an accuracy of 80%.

6. DISCUSSION

6.1 Packet Sizes

One of the advantages that DoH has over DoT is that it is shrouded between other HTTPS traffic and should therefore provide enhanced privacy. Cloudflare writes this on their website when discussing the differences between DoT and DoH: "(...) from a privacy perspective, DoH is arguably preferable. With DoH, DNS queries are hidden within the larger flow of HTTPS traffic. This gives network administrators less visibility but provides users with more privacy." [26].

However, Section 5.1.2 shows that packet sizes grant an observer much insight into the use of DoH by a user. This

means that with some simple filtering, DoH queries will not be hidden within the flow of other traffic. That would mean that one of the main benefits that DoH provides over DoT does not apply with the current way that DoH works.

Using padding in DoH is a way that DoH could be changed to avoid the detection that is enabled by the strategy mention in Section 5.1.2.

In the current state of DoH, the packet sizes are substantially different from regular HTTPS traffic. Adding padding to packets would change this and make it more difficult for an outside observer to determine whether a connection is transporting DoH traffic. Making packets larger by including padding has been described in RFC 7830, and could easily be added to DoH to potentially make the packet harder to distinguish from other traffic. [27]

A possible negative side effect of adding padding is that it would require more bandwidth to enable all users on the Internet to use DoH with padding. Currently in some lossy networks, DoH performance is higher than regular DNS, since DNS can be send over UDP, which requires frequent resending when packet loss occurs regularly. Requiring more bandwidth possibly removes the performance benefit from DoH.

6.2 Fingerprinting

Section 5.1.3 shows that JA3 fingerprinting is not a suitable method for detecting DoH traffic. Many servers nowadays use an open source implementation of an Application Server such as Nginx. JA3 is much more suitable for recognizing individual clients and detecting malware along that route.

JA3 fingerprinting can be used as an additional check to make sure that the suspect is indeed a DoH resolver, however it is not suited for initially filtering out connections, since it has a large amount of false positives.

6.3 Pattern Analysis

Section 4 shows the results of a proof of concept implementation of a pattern analysis tool for detecting websites, purely based on DoH packet size patterns. If future research shows that this is indeed trivially implemented as a network check in the form of a Machine Learning algorithmn than DoH loses its main benefit of providing privacy to its users.

However, it must also be noted that if the suggestions from Section 6.1 are implemented, it could remove the possibility for this pattern analysis.

6.4 DoH vs DoT

Showing that the current state of DoH lacks some of the benefits that it promises could give the reader the idea that DoT should be used over DoH. This research shows that the current implementation can be blocked to the same degree as DoT. However, most of the promises of DoH still apply at the time of writing this research: individual queries cannot be read by an outside listener, spoofing DoH is significantly more difficult than regular DNS and fast service is provided by the considerable infrastructure around HTTPS.

Many security professionals have already expressed their concern over DoH and argued that it does not provide enough benefits to outweigh the costs [9] [28]. Some argue that network operators must always have the ability to control DNS traffic out of a security standpoint and therefore deem DoT to be superior over DoH [29]. This research provides new arguments in that debate and might lead some to rethink the decision of DoH over DoT.

7. CONCLUSION

In this research the research question *To what extent does DNS over HTTPS prevent on-path devices from eavesdropping and interfering with DNS requests?* has been evaluated.

We have shown in this research that, while eavesdropping of individual queries has not been evaluated, it is probably possible to deduce a visit to a specific website by looking at patterns in DoH packet sizes.

Furthermore, interfering with DoH traffic by manipulating responses might not be possible, but detecting DoH resolvers and thereby blocking DoH is possible.

As a consequence, the promised privacy protection of DoH is debatable, and the advantage of DoH against DoT is getting smaller.

This has some consequences for the users that turn on DoH in their browsers: they should not rely solely on DoH to protect their privacy. Furthermore, for network operators, this means that DoH can easily be detected and blocked if it is undesirable to have clients in the network use DoH, hence forcing them to use the desired resolver. Lastly, for attackers that want to use DoH to perform lookups in their malicious software it might make it more difficult to do so with a standard implementation of DoH, since that might easily be recognized and blocked. Attackers might be able to make use of padding or another technique to let their DoH queries blend in more with other HTTPS traffic, since for an attacker the performance cost of extra bandwidth is not a concern.

7.1 Future Work

This research answers some questions about DoH, but also raises some new issues and questions that can be answered in Future Work:

1. More research can be performed on the pattern related features discussed in Section 5.2. A Machine Learning algorithm can be implemented to find more patterns for other websites based on the proof of concept in this research.

2. RFC 7830 [27] could be implemented by Firefox to further enhance the privacy benefits of DoH. Then, more research must be done to evaluate whether that is enough to hide DoH properly between HTTPS traffic and if the benefits of this outweigh the cost of extra bandwidth. If so, browsers can look to implement this into their DoH methods.

7.2 Acknowledgements

Special thanks goes to the Supervisor Moritz C. Müller for always providing me with great advice and new directions for this research. Also, I would like to thank Dr. Suzan Bayhan for Supervising the Dependable Networks group. The meetings between have always been of great help and the communication was always very clear.

8. REFERENCES

[1] Fernando Gont. Networks & Trust Introduction to DNS Privacy. (January):1–14, 2019. URL `https://www.gont.com.ar/papers/deploy360-dns-privacy-intro-v1.0.pdf`.

[2] Z. Hu, L. Zhu, J. Heidemann, USC/ISI, A. Mankin, Independent, D. Wessels, Verisign Labs, P. Hoffman, and ICANN. RFC 7858 - Specification for DNS over Transport Layer Security (TLS). Technical report, 2016. URL https://tools.ietf.org/html/rfc7858.

[3] P. Hoffman, ICANN, P. McManus, and Mozilla. RFC 8484 - DNS Queries over HTTPS (DoH), 2018. URL https://tools.ietf.org/html/rfc8484 https://datatracker.ietf.org/doc/rfc8484/.

[4] E Rescorla. RFC 2818 - HTTP Over TLS. In *Network Working Group, IETF*, pages 1–8, 2000. doi: http://tools.ietf.org/html/rfc2818. URL https://tools.ietf.org/html/rfc2818.

[5] Support Mozilla. Dns over https faq. URL https://support.mozilla.org/en-US/kb/dns-over-https-doh-faqs.

[6] Chromium Blog: Experimenting with same-provider DNS-over-HTTPS upgrade, 2019. URL https://blog.chromium.org/2019/09/experimenting-with-same-provider-dns.html.

[7] Apple Inc. Enable encrypted dns - wwdc 2020 - videos. URL https://developer.apple.com/videos/play/wwdc2020/10047/.

[8] J. Livingood, Comcast, M. Antonakakis, Georgia Institute of Technology, B. Sleigh, BT Plc, A. Winfield, and Sky. Centralized DNS over HTTPS (DoH) Implementation Issues and Risks, mar 2019. URL https://tools.ietf.org/id/draft-livingood-doh-implementation-risks-issues-03.html.

[9] Catalin Cimpanu. DNS-over-HTTPS causes more problems than it solves, experts say | ZDNet. URL https://www.zdnet.com/article/dns-over-https-causes-more-problems-than-it-solves-experts-say/.

[10] Paul V. Mockapetris. RFC 1035 - Domain names - Implementation and Specification. Technical report, Internet Engineering Task Force, 1987. URL https://www.ietf.org/rfc/rfc1035.txt http://tools.ietf.org/html/rfc1035.

[11] Paul V. Mockapetris. RFC 1034 - Domain names - Concepts and Facilities. Technical report, Internet Engineering Task Force, 1987. URL https://tools.ietf.org/html/rfc1034 http://tools.ietf.org/html/rfc1034.

[12] What Is DNS? | How DNS Works | Cloudflare, . URL https://www.cloudflare.com/learning/dns/what-is-dns/.

[13] S. Bortzmeyer and AFNIC. RFC 7626 - DNS Privacy Considerations. Technical report, aug 2015. URL https://datatracker.ietf.org/doc/html/rfc7626.

[14] Ilya Grigorik. High-performance browser networking, 2016. URL https://hpbn.co/.

[15] Moritz Müller. DoH in Firefox: veel klachten, weinig actie? | SIDN Labs. URL https://www.sidnlabs.nl/nieuws-en-blogs/doh-in-firefox-veel-klachten-weinig-actie.

[16] Drew Hjelm. A New Needle and Haystack: Detecting DNS over HTTPS Usage. Technical report, 2019.

[17] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom Van Goethem, and Wouter Joosen. Automated Website Fingerprinting through Deep Learning. (June), 2018. doi: 10.14722/ndss.2018.23105.

[18] Mariano Di Martino, Peter Quax, and Wim Lamotte. Realistically fingerprinting social media webpages in HtTPS traffic. *ACM International Conference Proceeding Series*, 2019. doi: 10.1145/3339252.3341478.

[19] Rebekah Houser, Zhou Li, Chase Cotton, and Haining Wang. An investigation on information leakage of dns over tls. *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, 2019. doi: 10.1145/3359989.3365429.

[20] Alexa.com. Alexa - Top sites, 2020. URL https://www.alexa.com/topsites.

[21] NijeboerFrank. Research project frank nijeboer git. URL https://github.com/NijeboerFrank/research-packet-analysis/.

[22] Xubuntu. URL https://xubuntu.org/.

[23] GitHub - mozilla/geckodriver: WebDriver for Firefox. URL https://github.com/mozilla/geckodriver.

[24] Z Nykolas. DNS Market Share Analysis — Identifying the Most Popular DNS providers, 2018. URL https://medium.com/@nykolas.z/dns-market-share-analysis-identifying-the-most-popular-dns-providers-80fefb2cfd05.

[25] John Althouse. TLS Fingerprinting with JA3 and JA3S - Salesforce Engineering, 2019. URL https://engineering.salesforce.com/tls-fingerprinting-with-ja3-and-ja3s-247362855967.

[26] DNS over TLS vs. DNS over HTTPS | Secure DNS | Cloudflare, . URL https://www.cloudflare.com/learning/dns/dns-over-tls/.

[27] A. Mayrhofer. RFC 8467 - Padding Policies for Extension Mechanisms for DNS (EDNS(0)), 2018. URL https://tools.ietf.org/html/rfc8467.

[28] Durren Anstee. Disappearing DNS: DoT and DoH, Where one Letter Makes a Great Difference | 2020-02-06 | Security Magazine, feb 2020. URL https://www.securitymagazine.com/articles/91674-disappearing-dns-dot-and-doh-where-one-letter-makes-a-great-difference.

[29] Patrick Nohe. What is the difference between DNS over TLS & DNS over HTTPS?, 2018. URL https://www.thesslstore.com/blog/dns-over-tls-vs-dns-over-https/.