Investigating the effectiveness of major Internet protocols in addressing DNS privacy concerns

AHMED AL-JABRI

April 2022

Abstract

DNS is a crucial and ubiquitous protocol in this age of computer networks and given the lack of privacy features in the protocol's design, efforts to encrypt DNS exchanges using a number of protocol suites such as DNS over TLS(DoT), DNS over HTTPS(DoH) and DNS over QUIC(DoQ) have been adopted to enhance confidentiality of DNS exchanges. In this report, the investigation assessed how effective are the various encrypted protocols in providing confidentiality to DNS exchanges and also compared their domain name resolution performance. The study first showed that DoH is the most privacy enhancing from a traffic analysis perspective scoring notably lower against a K-NN classifier. Nevertheless, although the adopted solutions prevent on-path observers from simply reading the requested domains in plain-text, an observer situated between the victim and the recursive resolver is able to study the packet features of encrypted DNS exchanges and de-anonymize encrypted traffic using a simple classification model. Secondly in terms of performance, DoQ shorter handshake showed some improvements to DoT and DoH however the margin is expected to increase in the recursive to authoritative side. As for page load times and DNS resolutions, DoQ and DoH performances were interchangeable and consistent but notably better than DoT that provided erratic results.

Contents

1 Introduction

Domain Name System (DNS) is a vital protocol to the operation of the Internet in today's world. Due to the core and foundational role that the Internet Protocol (IP) plays, the beginning of almost every interaction consists of a DNS lookup that in essence requests the IP address of the destination machine [6]. As a result of the DNS protocol's ubiquity and importance in today's world, it is sometimes difficult to even a imagine a time where the Internet did not need a service such as DNS.

Nevertheless, during the very early days of the Internet there was no need for a system such as DNS to simplify the process of obtaining IP addresses of other nodes since there were only a few nodes to begin with. Through distributed and regularly updated `HOSTS.txt` files those connected machines had no issues knowing associated IP addresses of other machines [2, 23]. Nevertheless as more nodes joined networks and the Internet exposure expanded from the private to public domain, the traditional host files became difficult to maintain and no longer user-friendly. That was when an efficient system of translating memorable domain names to IP addresses was called for which later came to be what we know as DNS. Ever since, DNS has been and remains a crucial protocol to the operation of the Internet [30, 18].

When DNS was first standardised in 1986 (three years after its initial creation) [23], the main goal for this protocol was to be efficient and scalable in dealing with the ever growing number of machines in the Internet [7]. Throughout the protocol's life, DNS underwent various updates that enhanced its efficiency nevertheless, the general details of how DNS operated remained the same for the majority of its existence [14]. This can be attributed to the brilliant engineering work that went behind building this protocol which allowed it to remain functional, scalable and generally unchanged until today.

Unfortunately DNS's operational success across multiple decades meant that it's original weaknesses were overshadowed and barley received any attention from the community until only very recently. To this day, generally DNS lacks security and privacy features. DNS is vulnerable to hijacking and man-in-the-middle attacks due to the absence of authenticity and integrity checks. On the privacy note DNS is still sent on the clear allowing eavesdroppers to sniff users DNS traffic and possibly build models that reassemble their web activities [14, 28].

DNS's privacy and security weaknesses attracted a lot of attention recently, for security enhancing measures *DNS Security Extension* (DNSSEC) was introduced to provide integrity and authenticity to the original workflow of the protocol [30]. While various encrypted approaches to DNS were proposed such as *DNS-over-TLS* (DoT), *DNS-over-HTTPS* (DoH) and *DNS-over-QUIC* (DoQ) to provide confidentiality to DNS exchanges.

In this report the aim is to investigate the extent to which the encrypted solutions provide confidentiality to DNS exchanges and also evaluate the performance of each proposal. The investigations will assess the degree to which attackers can determine the visited domain from observing encrypted DNS exchanges. In terms of performance the investigation will compare and contrasts various performance related measures of each protocol.

2 Background

Before discussing how DNS operates through an example of a DNS query life cycle, it is recommended to first describe broadly the DNS infrastructure and understand some of the different servers within.

2.1 DNS Infrastructure

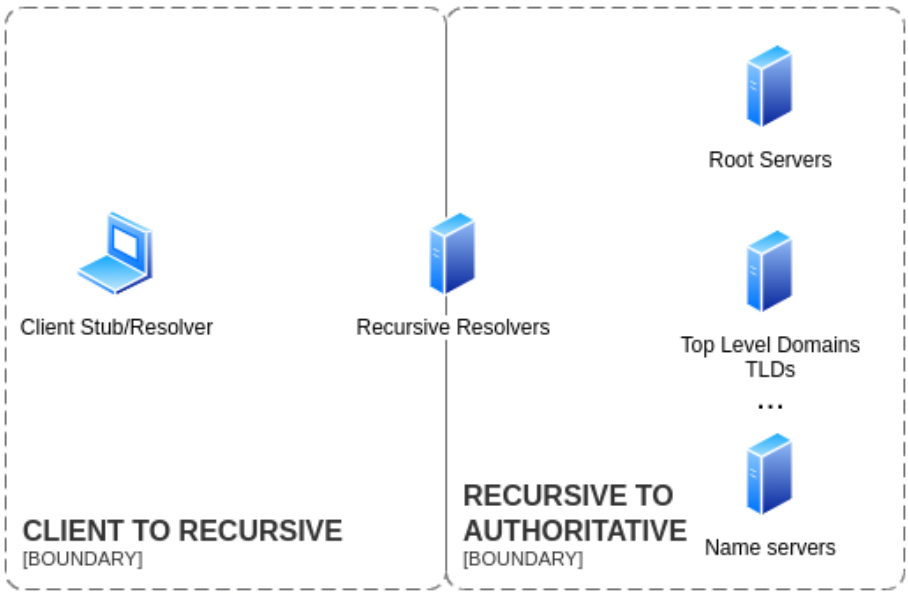

Figure 1: DNS Infrastructure

- Client Stub to Recursive Resolver

  On the client side we observe from figure 1 the client stub which is in essence the DNS software in the user's machine that sends DNS lookups to a DNS server. In Linux systems an example of a client stub would be the `systemd-resolved` daemon that operates as a proxy and forwards queries to a specified DNS resolver [3]. The applications within a machine

that create DNS queries such as a browser can also be referred to as a client stub that also forward queries to a server upstream.

- Recursive Resolver to Authoritative:

  Next looking at the recursive to authoritative end, a DNS recursive resolver lies upstream of the client stub as illustrated in figure 1 that is responsible for resolving client's queries from the authoritative side. The recursive resolver begins looking for answers by contacting the rest of the DNS authoritative side beginning with the root servers followed by the top level domains TLD, second level domains until it locates the requested name server that hosts the answer to the client's query.

2.2 How DNS works

Keeping in mind the DNS infrastructure illustrated in figure 1, this section looks at the real world operation of DNS by using an example of a DNS query triggered by a client and follow its resolution path across the DNS infrastructure. Noting that in the following example, the illustration in figure 2 assumes that there has been no caching of the requested domain name prior to this lookup either in the client or recursive resolver. Additionally the example assumes the implementation of *QNAME Minimisation* as per RFC 7816 [4].

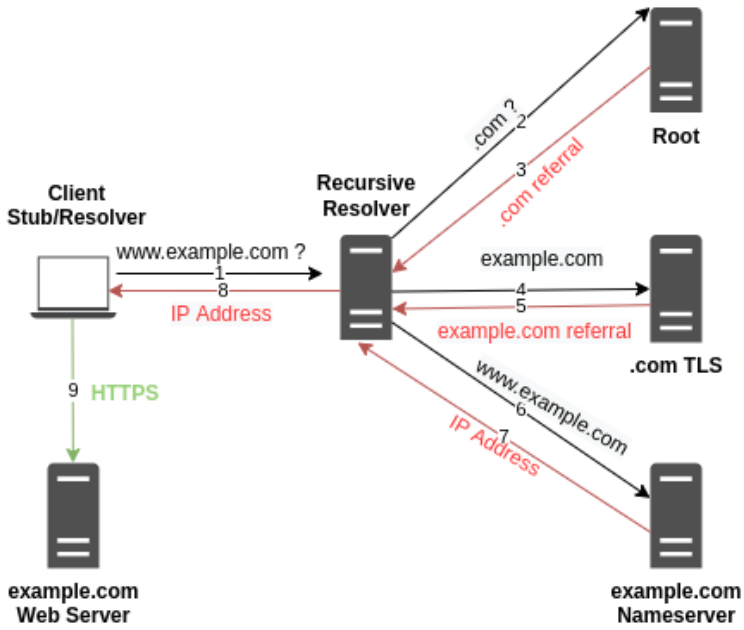

Figure 2: DNS query resolution life cycle prior to a web connection given that Qname minimisation is implemented and no caching

As evident in figure 2, a typical DNS lookup begins with a DNS client a.k.a *Client Stub* sending a message to resolve the IP address of a requested domain name for instance `www.example.com`. This message is sent to a DNS server and specifically a DNS recursive resolver which then has the responsibility of obtaining the IP address associated with `www.example.com`. The DNS resolver achieves that goal (assuming the answer is not in the server cache) by contacting each of the necessary DNS servers on the authoritative side (illustrated in figure 2). The DNS server will begin with the root servers to identify the IP address of the .com Top Level Domain (TLD). Following that the resolver will query the `.com` TLD for the `example.com` name server. Finally the resolver is able to make a final query for the requested domain to the `example.com` name server that ideally would respond with the IP address associated with the domain `www.example.com`.

Given that this DNS lookup was sent by a web client such as a browser, it then can proceed with establishing a HTTP connection with the obtained IP address of the web server.

3 DNS privacy concerns

Similar to many other protocols in their early days, DNS was first created in 1983 with no security or privacy considerations[19, 22]. The protocol was designed mainly with goal of achieving scalability and efficiency. All of which DNS has showed over the years to deliver on what it was intended to achieve. This initial disregard for security or privacy features in traditional protocols is no surprise by any means. In a similar and very popular case the HTTP protocol's initial design fit what it was intended to achieve in the early days of the web. Back when world wide web consisted of simple documents without the need for complex interactions from the client side, the HTTP protocol was sufficient enough for the purpose of fetching and serving simple resources. However, as web resources started becoming more complex and potentially sensitive interaction from the client side such as online shopping, the lack of security and privacy features were no longer disregarded paving the way for the implementation of a secure version of HTTP that was later termed HTTPS. Ultimately, this trend in protocol implementations to develop as is necessary is not new.

Nevertheless, one might ask why DNS remained until very recently before receiving any attention on its security and privacy weaknesses. Some ideas as to why that may be are discussed below:

- DNS is an 'enabler'

  DNS is often seen as an *'enabler'* and not a *'service'*, this is due to the fact that DNS resolution usually occurs to facilitate a succeeding interaction. For example, when an end user sends an email, a DNS lookup takes place

to identify the IP address of the destination mail server. Ultimately, the DNS exchange enabled the mail service to take place. Similarly in the web a DNS lookup occurs to enable the normal web exchanges through HTTP. Given that *'DNS is not a service'* perspective, the privacy and security flaws of DNS were not seen as worthy of wide attention until very recently.

- Data in DNS is public

  The second idea is around the fact that DNS does not hold any private information. Unlike HTTP that may contain passwords, credit card details and messages, DNS contains the domain names and IP addresses.

  However, although the data inside DNS exchanges are public, the individual exchanges themselves are not [28].

This leads to the discussion regarding the areas where the privacy concerns lie with DNS given its state today. Next, the report goes through the areas of concern for DNS privacy that attracted much of the attention to this matter in recent years.

3.1 DNS main areas of concern

3.1.1 Data in a DNS requests

The `source IP address` and `source port number` fields in a DNS request will indicate the user whom the request originated from, specifically the source port can be used to identify a users behind a *Network Address Translation* (NAT).

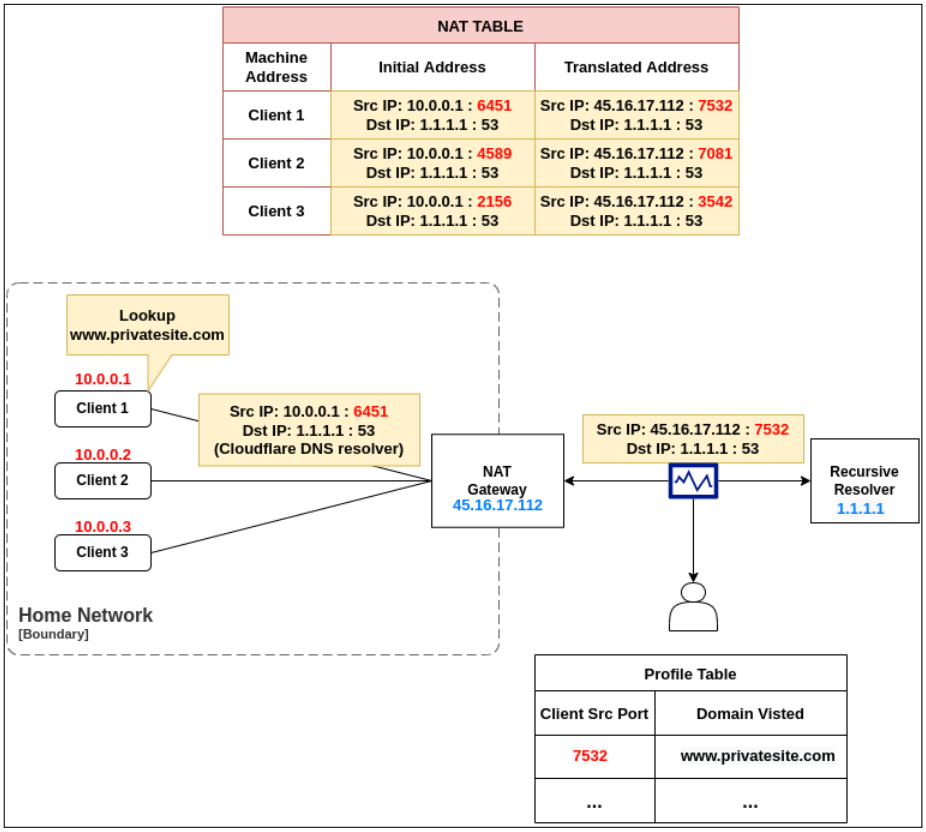

Figure 3: Identifying users behind a NAT

Figure 3 demonstrates how this threat exists. Observing client 1's lookup within the home network boundary, the packet headers are modified by the NAT gateway before forwarding the request to the recursive resolver. As private addresses can only operate within private networks, the main function of a NAT is observed as it provides a shared public IP address for clients 1 to 3 to access the public Internet. However as the NAT gateway modifies passing traffic, it keeps a record of the translations made to the initial requests. In the example in figure 3, the NAT gateway records that client 1's initial request of `Source IP: 10.0.0.1:6541` is translated to `Source IP: 45.16.17.112` (Gateway's public address) `:7532`(new unique assigned source port). If client 2 and 3 make future requests to the same destination (illustrated in the NAT table), the NAT gateway will assign different unique source ports to their requests.

With this unique assignment of source port numbers, an adversary in between the client and recursive resolver is able to identify queries made by different machines behind a NAT. Ultimately, downstream of the recursive resolver, the data in DNS requests can identify the domains requested to the actual users.

3.1.2 Data in Transmission

Arguably the main concern with the privacy issues of DNS lies within the basic transmission of DNS packets. As mentioned earlier, DNS remains unencrypted today which consequently allows for any on-path observer in between the client stub and recursive resolver (portrayed in figure 4) to identify what domains are being visited and by whom. Hence, the recent attention surrounding this issue that called for providing confidentiality to DNS transactions through encrypted protocol stacks as the document will discuss further in later sections.

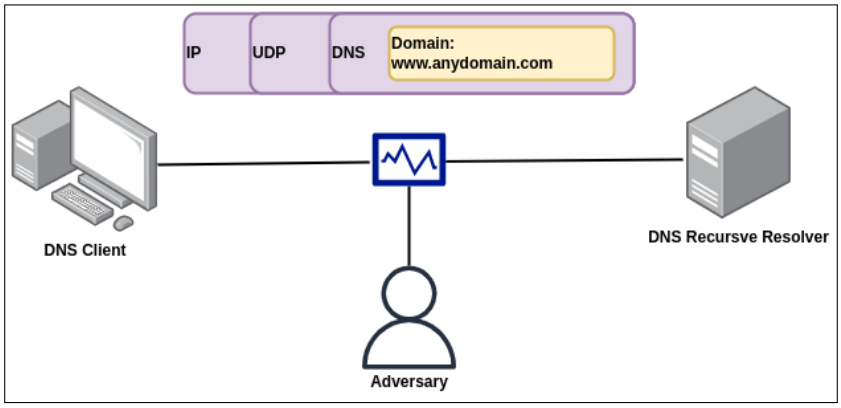

Figure 4: Lack of confidentiality with traditional DNS exchanges

3.1.3 Choice of Recursive Resolver

Another concern with regards to DNS privacy is the choice of recursive resolver. As seen earlier with the DNS architecture, the DNS resolver is typically positioned between the client and authoritative ends. In this case the resolver is in position to receive queries directly from the client machine allowing to know who exactly is making this request(as described in 3.1.1 & 3.1.2). Additionally, the way in which the DNS recursive resolver decides to behave with the DNS data is outside the control of the client, weather the resolver will log DNS traffic of particular individuals and share that data to advertisers is a total possibility. In the end, the resolver is a major trust point in the DNS architecture and the privacy risk of users is highly dependant on the policies and the way the resolver handles their data.

Furthermore, the mechanism by which a DNS recursive resolver is assigned to clients is in itself concerning. As stated in RFC 9076:

"*The choice of network has historically determined the default system DNS resolver selection; the two are directly coupled in this model.*"[28]

This is a another major concern that can be examined in a real world cof-

fee shop example. A coffee shop customer connecting to the shop's Wi-Fi access point will typically not only trigger the DHCP protocol to assign a private IP address but also to point to an available DNS server. That server is usually the ISP's DNS server however, the concern in this case lies in the fact that the coffee shop Wi-Fi administrator is able to route whoever connects to the Wi-Fi to his own rogue DNS server.

3.1.4 DNS Query Behaviour

The extent of ways in which DNS queries can occur are often misunderstood. As the example in section 2.1 uses a user initiated query for `www.example.com` to describe how DNS works, that is not the only way DNS queries occur. As Wicinski [28] classifies three types of DNS requests:

- Primary Requests: Queries initiated by the client (Entering a domain name in a browser, clicking a link or sending an email)
- Secondary Requests: Queries initiated by an application without the direct knowledge of the user.
- Tertiary Requests: Queries initiated by the DNS system behind the scenes such as a TLD referral by a root server.

The concern in this regard is that there are more DNS transactions occurring than what is obvious to the client. Over time an adversary situated between the client and recursive resolver can build client profiles based on their activities on the Internet. In single interactions, the concern is less evident however over time, various information about users (including sensitive) can become apparent such as where they live or shop, buy their medication, read the news and more.

3.2 Project Focus

Ultimately, the concerns discussed above are only a subset of the full picture concerning DNS's issues and do not address the security concerns of DNS. This is because DNSSEC as mentioned briefly earlier has already been deployed to address the security weaknesses of DNS. Even though the deployment of DNSSEC has not made considerable progress below the TLD servers due to its complexity [12, 9], DNSSEC does provide integrity and authenticity to DNS exchanges.

Hence, in this case, the report is focused on investigating the privacy concerns regarding the lack of confidentiality of DNS exchanges. This is a crucial aspect since the deployment of DNSSEC provides independent enhancements to DNS, meaning that the threats and major concerns discussed in section 3 will not be addressed [30].

Given that the scope of this report focuses on the privacy concerns, the document does not address security issues in full detail. With that being said, the

following section addresses the main proposed DNS privacy solutions and how they aim to achieve that.

4 Proposed DNS Privacy Solutions

The *Internet Engineering Task Force* (IETF) proposed and continues to develop several protocol standards that aim to mitigate the privacy concerns by providing confidentiality to DNS exchanges. The *DNS PRIVate Exchange* (DPRIVE) working group is a dedicated group within the IETF that have the purpose of designing and proposing solutions to the pervasive monitoring of DNS [1]. The three most popular proposals and the ones investigated in this report are listed in the coming subsections.

It is important to note that all current solutions are focused on implementing encrypted transport solutions downstream of the recursive (stub to recursive) [6]. The reason being that DNS is a complex architecture and attempting to encrypt the full scope is a difficult task [12]. This would typically entail deploying proposed solutions across the recursive to authoritative end from root servers down to the name-servers (see figure 2). Not only will this task be time consuming to complete as every DNS server will need to be able to understand the new protocols. More so, given the nature of the Internet and the likely hood of communication failures, packet loss and other types of network issues, efforts to secure the recursive to authoritative end will need to establish mechanisms to deal with such possibilities.

As Dickinson [7] explains in her talk that the goal at this point in time is to implement solutions downstream the recursive first and learn from that before attempting to tackle the more complex recursive to authoritative end.

As a result of this limited approach, this leaves encrypted DNS traffic susceptible to correlation attacks that monitor and match traffic on both ends of a resolver. As evident in figure 5, an adversary with the ability to monitor traffic inbound and outbound of the recursive resolver is likely to be able to identify the requested domain by observing the unencrypted traffic heading to the authoritative servers.

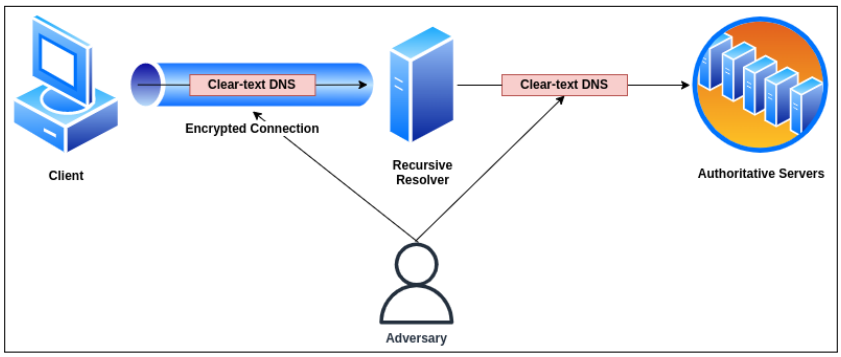

Figure 5: Correlation attack illustration

This attack however is much harder to implement accurately [6, 12]. Reasons being that the recursive resolver cache DNS records and hence not all incoming DNS requests will be forwarded to the authoritative end. Secondly, recursive resolvers typically have a vast amount of clients and will provide a sense of collective anonymity making it harder to map queries to clients and essentially reducing the severity of this risk [6]. For example, Cloudflare's popular `1.1.1.1` DNS resolvers serves thousands of clients. In such cases an observer on both ends of the DNS resolvers will not be able to directly match an encrypted query to an unencrypted query outbound of 1.1.1.1. It is significantly harder to achieve based on simple passive monitoring on busy traffic. Not to mention that there might not be any outgoing traffic from `1.1.1.1` since the answer might be fetched from the server's cache. With that being said, as Siby et al. [24] describes that this many-to-one relationship between clients and server provides some anonymity to clients where their queries are less likely to be susceptible to correlation attacks.

4.1 DoT

DNS-over-TLS or DoT as described in RFC 7858 [13] looks to pipeline normal DNS traffic through an encrypted TLS tunnel. This protocol has been assigned the port 853 and has seen some deployment in both client and server side software [12].

4.2 DoH

DNS-over-HTTPS or DoH as standardised in RFC 8484 [11] allows for DNS exchanges to be encapsulated within a normal HTTPS tunnel. Alternatively to DoT, DoH traffic will be passing through the traditional HTTPS port 443 and hence from a network operators perspective will be no different from any other web traffic [30].

The illustration in figure 6 encapsulates the main differences between DoT and DoH:

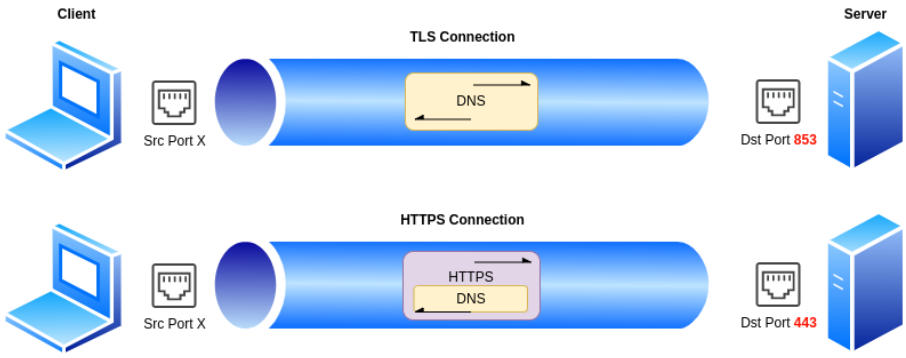

Figure 6: DoT vs DoH

Figure 6 illustrates the main differences between DoH and DoT. While DoT simply forwards plain-text DNS queries or responses through an established TLS connection, DoH embeds DNS packets either in `HTTP GET` or `POST` methods as per the specifications of RFC 8484 [11]. Hence from a traffic analysis perspective, DoH traffic is no different to normal HTTPS traffic on port 443 unlike DoT that is assigned a dedicated port (853) that can be used to distinguish DoT traffic from any other.

4.3 DoQ

With the standardisation of *QUIC* in RFC 9000[17] in May of 2021, a new alternative transport for DNS emerged. While DoT and DoH provided TCP-based solutions to encrypt DNS transactions, QUIC offered an encrypted UDP based solution. Ultimately this promised better performance results for DNS since it's design relies on the same simple transport protocol (UDP) that the original DNS of 1983 and until today was built on top of for efficient and low latency domain name resolutions services.

To understand how QUIC actually is a promising solution to encrypt DNS, the following section looks at how QUIC evolved, it's main features and how that can be very useful to implement with DNS.

4.3.1 QUIC background and main features

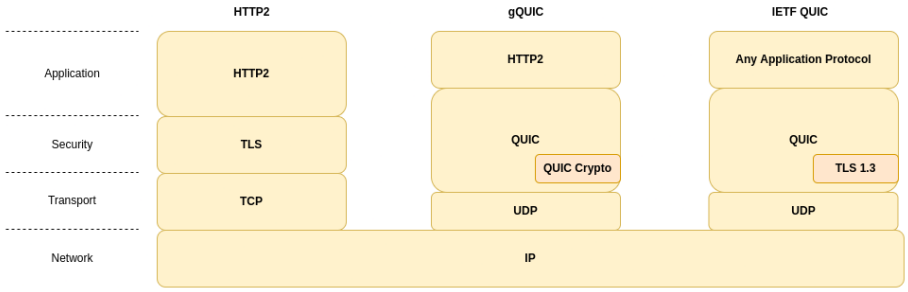

Figure 7: QUIC variations

QUIC was initially built by Google in 2013, that version was referred to as *gQUIC*. gQUIC was created to enhance the performance of the HTTP protocol that suffered from limitations such as head of line blocking caused by it's underlying transport TCP [20]. gQUIC's stack consisted of HTTP2 over QUIC over UDP as figure 7 illustrates. As gQUIC provided promising results showing slight performance advantages on high speed networks and significant improvements on slow speed networks compared to traditional HTTP2, Google handed gQUIC to the IETF in 2015 for standardisation efforts [20]. Hence the protocol standardised in RFC 9000 is IETF QUIC which is an independent transport unlike gQUIC and can be used as transport to other application layer protocols such as DNS in this case.

Long before the standardisation of QUIC in May of 2021, the IETF were aware of the promising prospect of using QUIC as transport for DNS. As a result, an evolving draft currently in its 11th stage exists to specify the implementation of DNS over dedicated QUIC connections [15]. On the bases of this evolving draft, some open source implementations including the one used in this report have built client and server side software.

The main features that QUIC offers which distinguish it from other TCP based implementations such as DoT and DoH are discussed next:

- Shorter Handshake

  The QUIC handshake is of 1-RTT which is 1 full RTT less than TLS's 2-RTT (TCP handshake plus TLS) which both DoT and DoT will be subjected to. What this means for DNS clients using DoQ will cost 1-RTT less than DoT and DoH every time a connection needs to be made with the recursive resolver.

- Minimum support of TLS 1.3

As portrayed in figure 7, TLS 1.3 is embedded by design into QUIC which means that no previous versions such as TLS 1.2 can be used. Whereas with DoT and DoH the TLS version is negotiated and could be TLS 1.2 or 1.3. This is a major advantage for QUIC since TLS 1.3 is the only TLS version that can encrypt the client hello message containing the SNI field which is another major privacy leak in transmission described in section 5.2.

- Zero Round Trip Time (0-RTT)

  QUIC also allows for subsequent connections from a client to send application data from the first go without the need to establish another QUIC connection [20, 17]. This feature provides significant performance boost as clients do not need to establish a new connection to the DNS recursive resolver every now and then.

- QUIC streams

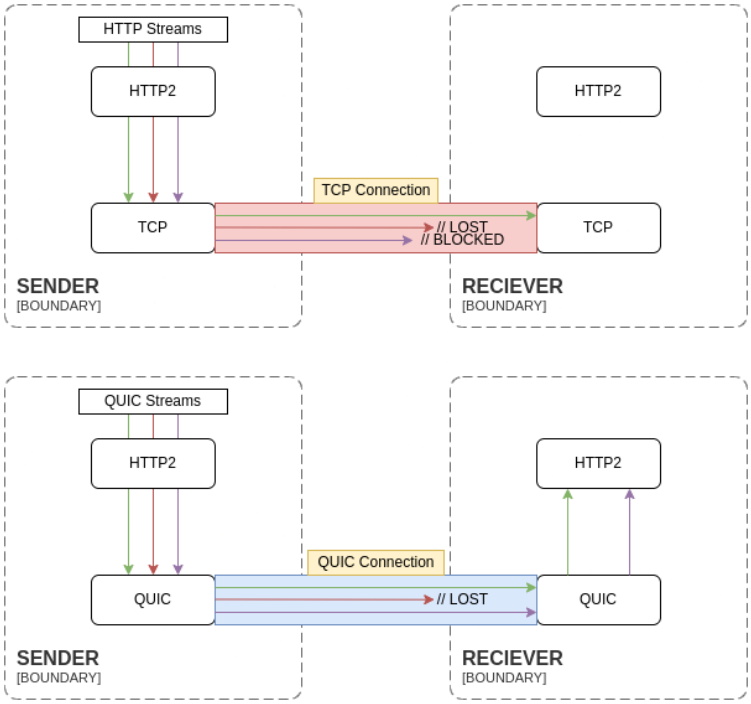

Figure 8: Illustration of how QUIC removes the burden of TCP head-of-line blocking

QUIC connections allow for data to be sent in independent streams where in case of any losses, only the stream affected is halted and waits for

retransmission. As figure 8 illustrates, this is particularly solving the issue with HTTP2 where independent HTTP2 streams where treated as a single stream down at the TCP layer, this eventually meant that if any HTTP2 stream was affected by packet loss, all other streams where halted at the receiver end until that loss was recovered. QUIC streams can also be of benefit to DNS queries where multiple queries from an initial lookup can be assigned dedicated QUIC streams and if one query is slowed by packet loss the other queries will not be affected resulting in an overall quicker DNS resolution.

- Connection Migration

  Unlike with TCP based solutions that were directly linked to the underlying network protocols in a way that a change of IP address would terminate the existing connection. QUIC has support for the ability to shift existing connections to different networks without terminating the client's connection [20, 17]. Although this feature may be beneficial to DNS clients with changing IP addresses, there are notable privacy concerns that are discussed later on the report.

As a result of the concepts discussed above, QUIC provides a promising alternative to provide confidentiality to DNS transactions without the need to endure the performance burdens of TCP's legacy features.

5 Methodology

This report aims to investigate two main concerns with the recent DNS privacy proposals. First to assess the extent to which implementing those solutions provides actual returns in protecting the privacy of users. Secondly to evaluate the performance burdens that these solutions incur on DNS's ability to provide efficient and scalable domain resolution services.

5.1 Test Environment

The investigation uses *dnsproxy*[21], a DNS proxy created by Adguard supporting DoT, DoH and DoQ. This implementation was used based on the fact that at the time of writing it was was one of the very few available implementations that support all three protocols. More so, the setup was simple and provided various configuration options through the command line. The tests also use Adguards DNS server as the main recursive resolver as illustrated in figure 9. The reason for this is that also at the time of writing, that was the only public DNS recursive resolver that had support for DoQ and since the investigation aimed to test all three protocols (DoT, DoH and DoQ) it was ideal and fair to use the same recursive resolver to test all three desired protocols.

The Operating System (OS) of choice was Linux and specifically Ubuntu v20.04

distribution. This is simply because Linux offers more control over the OS configurations compared to closed source alternatives such as Windows or MAC. Due to this selection a few system changes had to be made regarding how Linux resolves domain names by default. In Linux systems, the `systemd-resolved` daemon exists to provide domain name resolution for local applications. This daemon by default forwards DNS queries to the gateway which will ultimately send queries to the ISP's DNS resolver. When installing `dnsproxy` along side `systemd-resolved`, dnsproxy's resolutions are overridden even after configuring the network to use `dnsproxy` over `systemd-resolved`. For that reason, the setup disabled `systemd-resolved` completely (see figure 9) to avoid the issue.

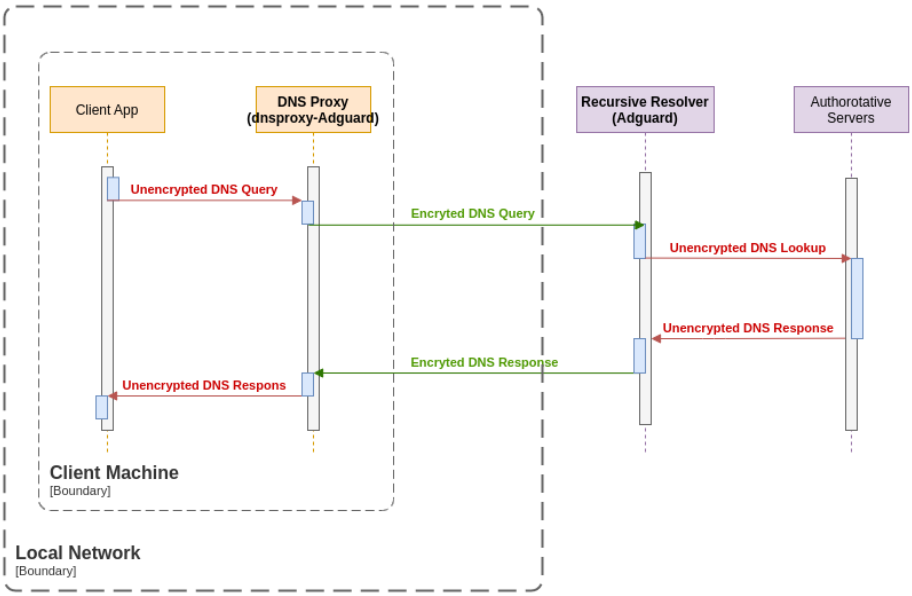

Figure 9: System setup

5.2 Privacy Testing

In this area of testing, one of the main objectives was to investigate whether or not encrypting DNS exchanges is enough to prevent passive monitoring attacks from identifying the visited domains. There is complete awareness that other related works has established that passive monitoring attacks can derive the visited domains based on the analysis of encrypted DNS features as presented in [24, 25, 26]. Nevertheless the investigation took a step further to compare how successful pervasive monitoring attacks perform against each of DoT, DoH and DoQ.

When it comes to privacy leaks on the wire, there are a few components that contribute to that risk other than DNS which have been addressed prior to this investigation.

- IP leaks

  The IP protocol is a central component for routing packets in the Internet and simply cannot be hidden away from middle boxes in between the endpoints (client and server). When a packet is sent it must contain in it's header in clear text the destination IP address so that routers are able to route the message to the intended receiver. In the case of DNS, an observer between the client and server will always be able to identify the IP address of the resolver regardless of whether DNS is encrypted or not. Furthermore the attacker is able to follow up by observing the subsequent HTTP traffic to that DNS lookup and extract the IP address of the web server contacted. That by itself can uncover the domain visited in that interaction. However that is not always the case, public IP addresses are expensive and it is typical that multiple domain names will reside under a shared public IP address. In that case the uncovering of the IP address will not directly uncover the domain visited and will instead be down to probability.

  To follow up with an example, if we consider a DNS lookup to the following domain name: `https://www.bewater.la`, at the time of writing the answer contains this IP address: `185.247.225.40`. If we then follow up with a Reverse DNS Lookup tool such as the *Reverse IP API*, we will find that about 1800 domain records exist under that same address. This consequently means that the chances for an adversary to guess the visited domain based solely on the IP address would be 1/1800.

  Ultimately, the probabilities vary depending on the pool size of sharing domain names, in some cases and particularly with popular domains the IP address would be dedicated or at least be shared with only a few or related domains making the privacy leak in this case more probable. relating to Hoang et al. [10] work assessing the privacy benefits of domain name encryption, they evaluate that 20% of the domains (these are usually popular domains) they studied will not gain any privacy benefits since they have a one-to-one IP to domain mapping. While 30% will gain significant benefits since they share a public IP address with over 100 domains.

- SNI leaks

  Following up on the points above, in a typical scenario where many domains share a public IP address, the *Server Name Indication* (SNI) extension of the TLS protocol is necessary for the server to identify the incoming request is for which domain. Using the same domain that was used to explain IP leaks above (`www.bewater.la`), in a situation where a

client attempted to establish a HTTPS connection with the hosting web server (`185.247.225.40`) to view this domain. The server needs to know which domain of the total 1800 domains hosted within this public IP address is requested by the client. This is where the SNI extension comes in play, prior to encrypting the communication between the client and server, the client will send the name of the requested domain (`www.bewater.la` in this case) to the server along with the client hello message in a traditional TLS handshake. With this information the server can identify which domain among the available 1800 is needed by the client and will ultimately use the appropriate certificate to encrypt the communication and be able to serve the requested web content.

The following figure summarises both cases where the IP and SNI exist on the wire:

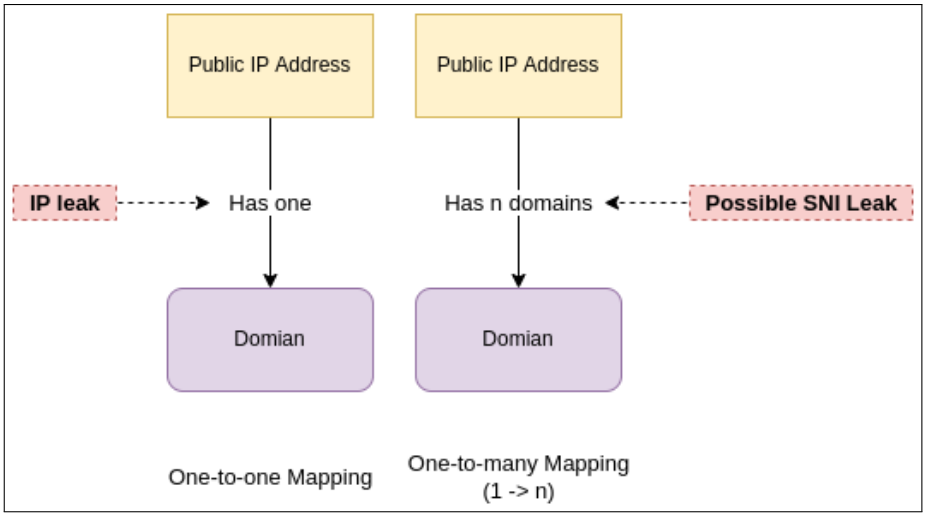

Figure 10: IP and SNI risks

Figure 10 illustrates on the left side where a one-to-one mapping exist the IP address alone will be enough to identify the visited domain. Whereas on the instance presented on the right side of figure 10, the IP address will not be enough for an observer to identify the requested domain but the SNI variable might. The reason why figure 10 specifies that in a shared hosting example the SNI leak is possible and not certain is because *Encrypted Server Name Indication* (eSNI) exists to encrypt this field in transmission so that on-path observers will be unable to read the domain name in plain-text form on the wire.

The eSNI extension was specified to be added to TLS 1.3 meaning that encrypted communications using older versions of TLS (including TLS 1.2 which is still used today) will be susceptible to SNI leaks [16, 27].

5.2.1 Assumptions

For the following tests the investigation assumes that the SNI variable is encrypted (eSNI), leaving only the IP address exposed to an eavesdropper. The test also assumes that the domain visited does not have a dedicated public address assigned to it and instead shares an address with other domains. This ultimately defines the objective of the tests to attempt to uncover the visited domains based solely on the features from the encrypted traffic observed.

An additional assumption involves the nature of the threat source. In this case the threat considered is that of an adversary that shares the network with the victim and has the ability to capture ongoing and outgoing traffic. In real terms this can be an eavesdropper on a cafe network or the network administrator.

5.2.2 Threat Model

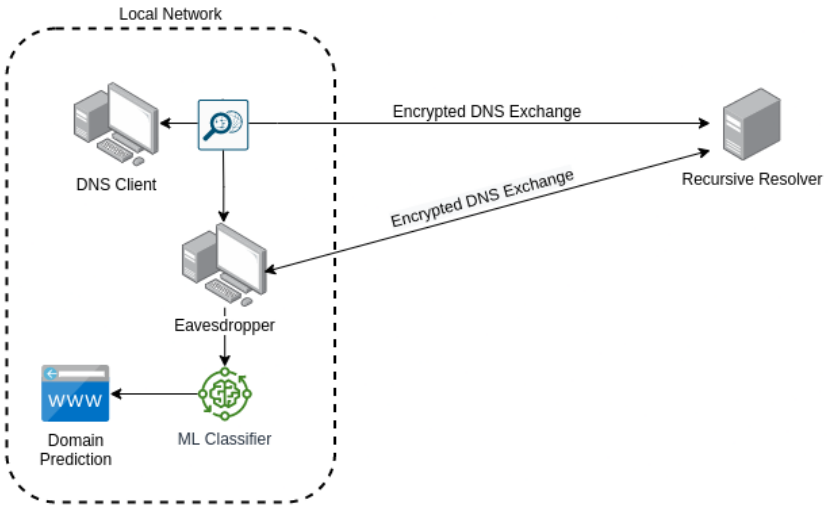

Figure 11: Threat model

Figure 11 shows an overall view of the considered privacy threat for this investigation. Given the test assumptions stated in section 5.2.1, the threat portrayed in figure 11 describes a scenario where an adversary situated within the same network of the victim, is able to identify the victim's visited domains based solely on packet features of encrypted DNS traffic (DoT, DoH or DoQ). More specifically in this case, the adversary would select a public IP address that hosts multiple domains and study the packet features of encrypted DNS exchanges on each domain. With that data, the eavesdropper then selects an appropriate *Machine Learning Classifier* to builds a *Classification Model* that will classify the set of collected features into individual categories (domain A, domain B, domain C etc). With the help of that model, the eavesdropper is

now able to capture and extract packet features of the victim's encrypted DNS traffic and feed that to the classification model to identify the visited domain among the set of studied domains.

5.2.3 Machine Learning Classifier:

In efforts to create a suitable classification model, the *K-Nearest Neighbour* (K-NN) algorithm is selected as a fitting classifier to investigate the pervasive monitoring risk with encrypted DNS. The K-NN algorithm is considered to be an appropriate algorithm to use in this instance since the test assumes there are a set of possible domains from the available public IP address. Those domains will act as the classification categories (neighbours), while the individual feature set (see section 5.2.5) are used to determine the closeness of unlabelled data to those classifications.

The objectives of the classifier can be illustrated using figure 12 below:

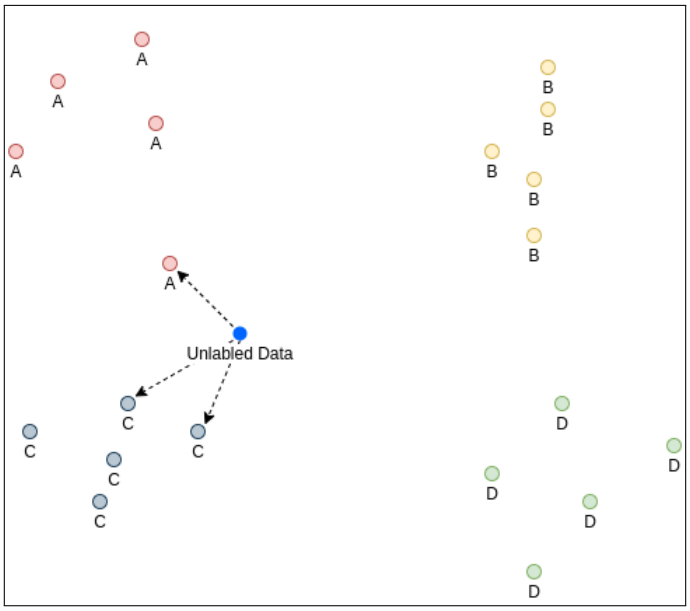

Figure 12: How K-NN classifies unlabelled data using a K value of 3

The classifier will take the set of encrypted DNS traffic features studied for a group of domains (A to D). That will then allow the classifier to build classification model that will establish some idea regarding what set of attributes are observed when domain A, B, C and D are visited. Following that, when the eavesdropper captures traffic features from the victim latest visit and feeds that to the classification model (represented as a blue data point in figure 12. The model will calculate the closest K points to the unlabelled data point and

determine the classification based on a majority vote. Similarly, in the example in figure 12 that uses a K value of 3, the classification result will be C (2 votes for C, 1 vote for A). There are multiple ways in which the K value van be derived however in the case of this report the *Square Root* method is used which takes the square root of the total data samples available and subtract 1 if the value is even to avoid possible draws.

5.2.4 Test Setup

Drawing from the threat model in figure 11, the test environment setup below was set to replicate the threat model in order to test the effectiveness of pervasive monitoring attacks on encrypted DNS traffic.

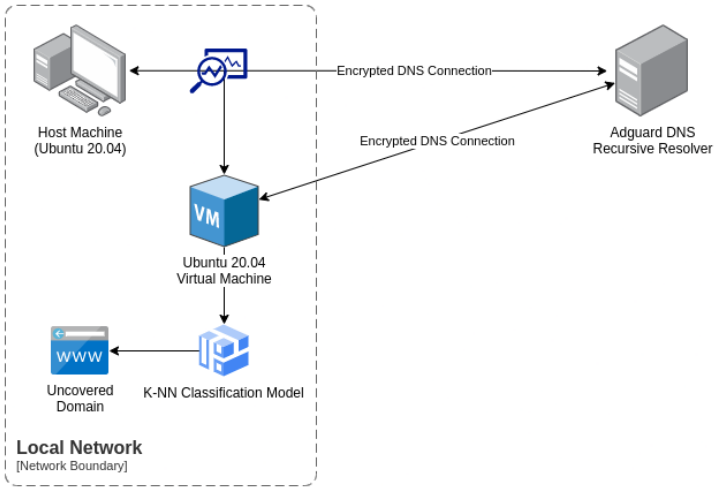

Figure 13: Testing setup

Using the set up in figure 13, the following is the list of tools was used to conduct the experiment:

- Tools used:
  - ***Wireshark***: Used as the main packet capture program, has support for many functionalities that aid the collection of the necessary features in section 5.1.4.
  - ***Selenium IDE*** extension for Chrome: Selenium IDE is a web testing tool that allows to run automated scripts simulating user like behaviour.

    Initially the experiment built a bash script using the *dig* tool to attempt to simulate real-like DNS traffic on the web. Nevertheless,

utility tools such as dig in Linux does not simulate real-like DNS traffic from browser clients. The dig commands only runs queries for the specified domain in the command while in reality, visiting domains through browsers may trigger multiple DNS queries from embedded objects within the domain's web page. For this reason the test design finalised on using Selenium IDE to simulate typical DNS traffic from web page visits.

– ***Juypter Notebook***: Juypter Notebook is a practical web application that provides live interpretive environment to run code and display illustrations.

  Juypter Notebook is used to implement the *K-Nearest Neighbor* Algorithm using Python's *scikit-learn* Library and create visualisations of the resulted outcomes such as the confusion matrix and classification report.

5.2.5 Packet Features

With the experiment solely focused on using extracted features from the encrypted DNS traffic, the experiment considered the following list of features for the attack:

- Query and Response Tuple Sizes:
  The tuple values of a DNS query and it's associated response is considered as a valid feature to collect because the contents of a query formulated by an application can be different from a domain to another based on the length of the domain name. Similarly the responses can be varying due to the content that different nameservers include in response messages.

- Cumulative DNS exchange size:
  This feature can also be used to distinguish between domains since the total number of DNS queries triggered after the initial DNS request can vary from one domain to another. Additionally even if two domains names result in a total of 4 DNS queries after the initial request, the sizes of queries and responses may differ and hence this feature can be used to distinguish among a set of domains.

- Time interval:
  The reason for collecting this feature is to investigate whether or not the time it takes for a query to be resolved for a particular domain is notably different to others and if there are any trends evident.

- Total Number of queries and responses:
  This feature is also noted since it is typical for domain names to trigger varying numbers of subsequent queries depending on the amount of subdomains and external servers contacted within a domain. In that case, this feature is likely to differ between domains and ultimately could be used to differentiate them.

- Total resolution time:
  This feature refers to the total time taken to resolve all the queries associated with a domain name, this can be an effective feature to distinguish among domains that have a similar number of queries and responses.

- Page Load Time:
  Although page load times are not a directly attributed to DNS, they are affected by the time it takes for all DNS queries within a web page to be resolved. In a typical case where a webpage consists of embedded links and content such as images that are hosted from external locations, such objects cannot be fully loaded to the user before resolving the domains for where such items reside. For that reason this feature is likely to differ with different domains and is therefore considered.

5.2.6 Test Procedure

The privacy tests procedure first involved a selection of a set of domains (A to D) that share a public IP address. Then an encrypted DNS implementation with Adgaurd's dnsproxy was used to runs each selected domain five times with each protocol (10 times in the case of DoQ only). All tests where made sure that the query was made when the protocol handshake has already been initiated to avoid having the handshake packets affect the attributes and flow of the DNS traffic.

Wireshark was used as to capture the packets and extract the features in section 5.2.5, once all captures are made the results are fed into the classification algorithm along with the ground truth values. Based on the data set collected the model takes the unlabelled data set and attempts to classify the set of features for each test instance and lastly compares that with the ground truth values given earlier.

5.3 Performance Testing

In regards to investigating the performance of the proposed protocols, the investigation analysed three main features:

- DNS Lookup Time

  The DNS lookup simply refers to the time it takes for the response to return to the client after the query has been initiated. This is collected using Wireshark by setting the query packet as the time reference and extract the time difference of the response packet. In some cases the response may be too large to accommodate a single packet and is instead sent in two, in this case the Lookup time measurement considers the end of the lookup time to be the time of arrival for the first response packet.

- Total DNS Lookup Time

This aspect looks at the total time taken to resolve all DNS queries related to the initial DNS request. Since DNS lookups may trigger subsequent requests, this way in which DoT, DoH and DoQ handle multiple queries is a factor to consider when it comes to performance.

- Handshake Establishment

  For this feature the test collects the time taken for the DNS client and server to establish a secure connection before sending any application data. This component was selected to investigate the burdens of TCP based solutions in comparison with QUIC.

- Page load time

  Although page load times are not entirely DNS related, the time it takes for all DNS queries to be resolved affects the page load time of a webpage and hence is considered as a feature of performance.

Using the same setup as in figure 9, the experiment ran multiple tests on a constant domain, in this case `aston.ac.uk` across Do53 (traditional DNS over port 53), DoT, DoH and DoQ. The experiment also sets a constant recursive resolver (Adguard's server).

- Tools used:
  - ***Wireshark***
  - ***Page Load Time*** tool: A chrome extension that uses the *Navigation Time API* to test the performance of various aspects of web pages.
  - ***Dig***: Dig is a known DNS utility that is used to gather DNS information and also initiate DNS queries to specified domains.

5.3.1 Test Procedure

The performance testing procedures are described individually below within their respective section.

5.3.2 Handshake Testing

The handshake testing procedure ran queries after connection timeouts expired. For TCP based solutions such as DoH once the first query was run we observed the TCP Keep Alive packets until the timeout period expired. Then we ran the next query and collected the needed features. As for DoQ the procedure involved waiting for a certain period of time until running the subsequent query because there was not other indication to when the connection was going to be terminated. More so dnsproxy did not implement 0-RTT which was convenient in this case.

5.3.3 Page Load Time Tests

As for this component we simply ran DNS queries through selenium IDE on chrome and let the page load time tool calculate the total time taken to load the page with encrypted DNS solutions.

5.3.4 DNS Lookup Time

In terms of DNS Lookup times the procedure was simply about setting the query packet time as reference in wireshark and extracting the time of arrival for the following response.

5.3.5 Total DNS Lookup Time

Alternative to a single DNS lookup time, this area was intended to investigate the way in which the protocols handled multiple DNS queries and measure the time taken for all DNS traffic resolved from the initial request to be resolved.

6 Findings and Comparative Analysis

Following the methodology discussed in the earlier section. The outcomes of the investigations for both privacy and performance are presented next and compared to the various protocols implemented:

6.1 Privacy component

The results of this section are concerned with the K-NN classifier implemented to provide a classification model based on the captured features of each encrypted DNS protocol stack. Figure 14 and 15 below displays the *Confusion Matrices* obtained the privacy tests.

A Confusion Matrix is a table that represents how well an algorithm has performed in predicting a classification problem [5]. In this case the confusion matrices displayed in figures 14 and 15 evaluate how well the K-NN algorithm was able to predict the selected domains A to D on each different protocol implementation (padded and unpadded DoT, DoH and DoQ).

In terms of the sample sizes, DoQ classification test consisted of 40 total samples (10 for each domain) while tests using DoT and DoH had 20 samples each (5 for each domain). Based on those sample sizes the selection of the K value for the K-NN classifier was derived in this case by calculating the square root of the total data set and where the result is an even number, 1 is subtracted from the answer. So for DoQ's test the K value was $5(\sqrt{40} - 1)$ and for DoT and DoH, K is equal to $3(\sqrt{20} - 1)$.

6.1.1 Confusion Matrix

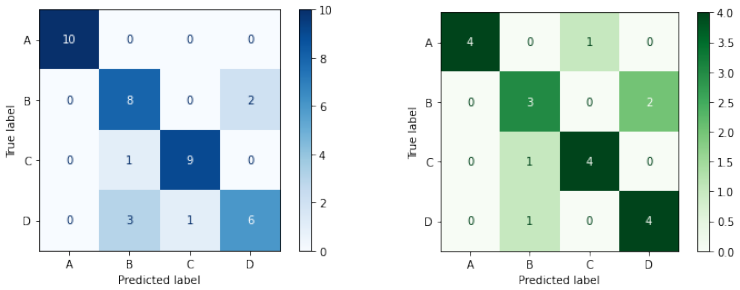

Figure 14: Confusion Matrix illustration presenting the DoQ results on the left and DoH on the right.

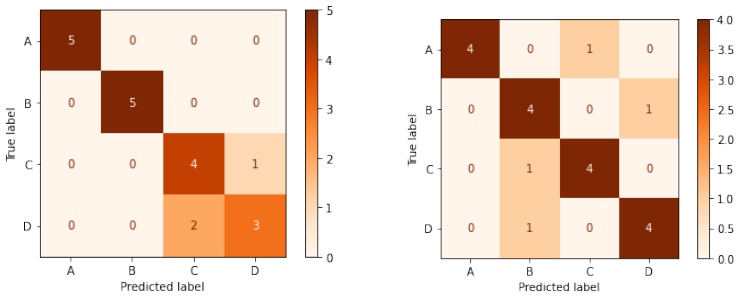

Figure 15: Confusion Matrix portraying the results of unpadded DoT on the left and padded DoT on the right.

With the obtained confusion matrices above, the classification report next summarises the obtained results from the conducted privacy tests:

6.1.2 Classification report

PROTOCOL	Domain Specific				Overall					
	DOMAIN	Precision	Recall	F1	Accuracy	Precision	Recall	F1	FP	FN
DOQ	A	1	1	1	0.82	83.00%	82.50%	0.825	17.50%	17.50%
	B	0.67	0.8	0.73						
	C	0.9	0.9	0.9						
	D	0.75	0.6	0.67						
DoT	A	1	1	1	0.85	85.50%	85.00%	0.85	7.50%	7.50%
	B	1	1	1						
	C	0.67	0.8	0.73						
	D	0.75	0.6	0.67						
DoH	A	1	0.8	0.8	0.75	76.75%	75.00%	0.7325	12.50%	12.50%
	B	0.6	0.6	0.6						
	C	0.8	0.8	0.8						
	D	0.67	0.8	0.73						
DoT (Padded)	A	1	0.8	0.89	0.8	81.75%	80.00%	0.805	10%	10%
	B	0.67	0.8	0.73						
	C	0.8	0.8	0.8						
	D	0.8	0.8	0.8						

Figure 16: Classification report

Elaborating on the report in figure 16, the model's performance among DoT, DoH and DoQ is observed and the effect of *padding* packets is also presented in this case with DoT. Padding efforts aim to unify to an extent the size attributes of DNS queries and responses in order to prevent passive observers from being able to narrow down the visited domains based solely on such size features [6].

Accuracy:
First observing the accuracy scores of each implementation, this measure looks at the degree to which the model has made correct predictions (both true positives and true negatives over total predictions). This is an important metric since it shows the extent to which the algorithm used was able to successfully determine the visited domain based on the set of features.

In this regard, the results show that the modelled attack to identifying visited domains from features of encrypted traffic has performed better against DoT than the other protocols resulting in an accuracy score of 0.85. However when using a padded implementation of DoT the accuracy of the model drops by around 5% (0.80 accuracy). This reiterates Bushart and Rossow [6] findings that indicate that padding solutions reduce the performance of such models but are not enough to prevent them from uncovering visited domains with high levels of confidence.

In terms of DoQ and DoH the model performs less accurately with a score of 0.82 for DoQ and 0.75 for DoH. From the accuracy results the investigation showed that DoH's encrypted traffic was notably harder to be classified between domains A to D by the model nevertheless the overall accuracy is still high.

Precision and Recall:
With regards to precision and recall, precision looks at the amount of correctly identified positives (true positives) over the total predicted positives (true positives + false positives), while recall focuses on the amount of correctly identified positives (true positives) over the total actual positive (true positive + false negatives). In some cases, investigating precision or recall can be more useful than another. For example in cases where the cost of false positives is high, precision is a more useful metric however, where false negatives are more costly recall is the more appropriate metric to focus on [29].

In the case of this investigation however, both false negatives and false positives are important. From the perspective of the attacker, false positives and negatives are costly since they can mislead the findings. On the other hand form a privacy standpoint evaluating the anonymity provided by the implemented protocols, both false negatives and positives are important they show how well the protocols can mislead observers regarding the visited domains. Hence the comparative analysis of in this report focuses more on the F1 score which is a metric that combines recall and precision.

F1 score:
Observing the F1 scores from the classification report in figure 16, we observe similar trends compared to the accuracy observations. Unpadded DoT scored the highest F1 score of (0.85) followed by DoQ with (0.825). Padded DoT comes next with a score of 0.805 and lastly DoH with the lowest score of 0.7325. From such results the findings show again that pervasive monitoring attacks that aim to deanonymize visited domains based on features of encrypted DNS traffic perform better against unpadded DoT implementations than any other. The padded DoT implementations similarly resulted in a notable difference (0.045) however it is still considerably a high score (0.805). DoQ's performance lied approximately in between unpadded and padded DoT (0.825) showing that the protocol does not provide major anonymity enhancements compared to DoT. DoH alternatively performed considerably better with a difference of 0.0725 to the closest implementation (padded DoT). More so with a padded DoH implementation the F1 and accuracy scores are expected to go even lower.

Ultimately, the obtained results presented above indicate that DoT (padded and unpadded) and DoQ implementations are very susceptible to classification attacks using basic size and time features of encrypted packet exchanges. This is drawn from the fact the accuracy score was over 80% and F1 scores above 0.8. As for DoH's implementation, the accuracy and F1 scores are still high (over 70% for accuracy and 0.7 for F1) but is considerably lower than the other implementations. More so, while size and time packet features may be unique to different domains, using padded implementations to unify size attributes is not a sufficient solution. Another observation is that DoH performed notably better than DoQ and DoT mainly due to the similar time and size observations among different domains from a traffic analysis perspective.

6.2 Performance component

With regards to performance, this section presents the findings from the investigations discussed in section 5.3. In the figures below, the y axis represents the time (either in seconds or milliseconds depending on what is stated in the axis title) while the x axis represents the test number. For example, in figure 17 the test was carried out 10 times, hence the x axis lists lists in the data points collected in order from 1 to 10.

6.2.1 Connection Handshake

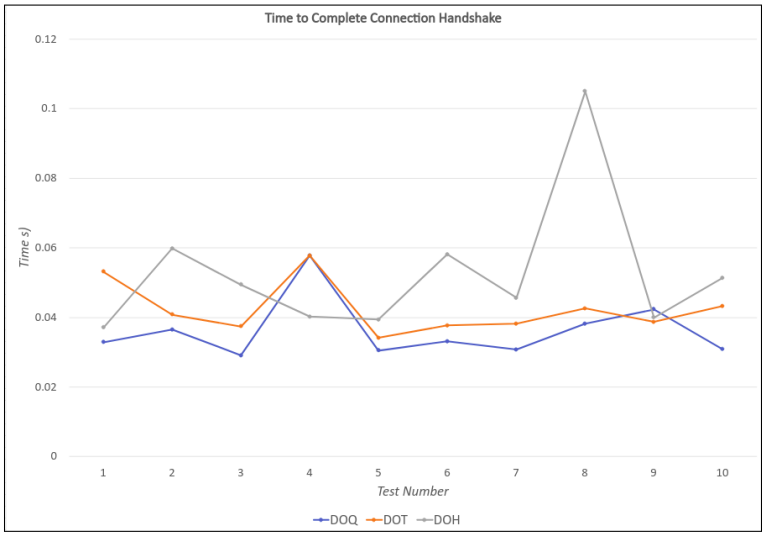

Figure 17: Line graph comparing the amount of time it takes for DoQ, DoT and DoH to establish a secure connection before sending application data

Looking at figure 17, there are notable differences between the implemented protocols. First with regards to consistency, DoT appears to provide the most consistency in handshake time given that all its values lie within approximately 0.06 to 0.375 seconds, Followed by DoQ's scores that ranges between 0.03 and 0.06 and lastly DoH with more than 0.06 seconds between the highest and lowest data point (0.04 - 0.1). However if we disregard DoQ's relative outlier at data point 4, DoQ would appear as most consistent with ranging values only between 0.04 and 0.03 seconds.

Regarding the actual duration of the handshake DoQ is observably faster at completing the secure connection prior to sending the DNS query. This is understandable given the shorter initial design of QUIC that eliminates an extra round trip [20]. DoT and DoH's scores show that the extra TCP handshake at the start is slowing down the connection establishment by an observable mar-

gin. Nevertheless DoT has also performed faster and more consistent than DoH regardless of the same underling protocol stack beneath the application layer.

6.2.2 Page Load Time

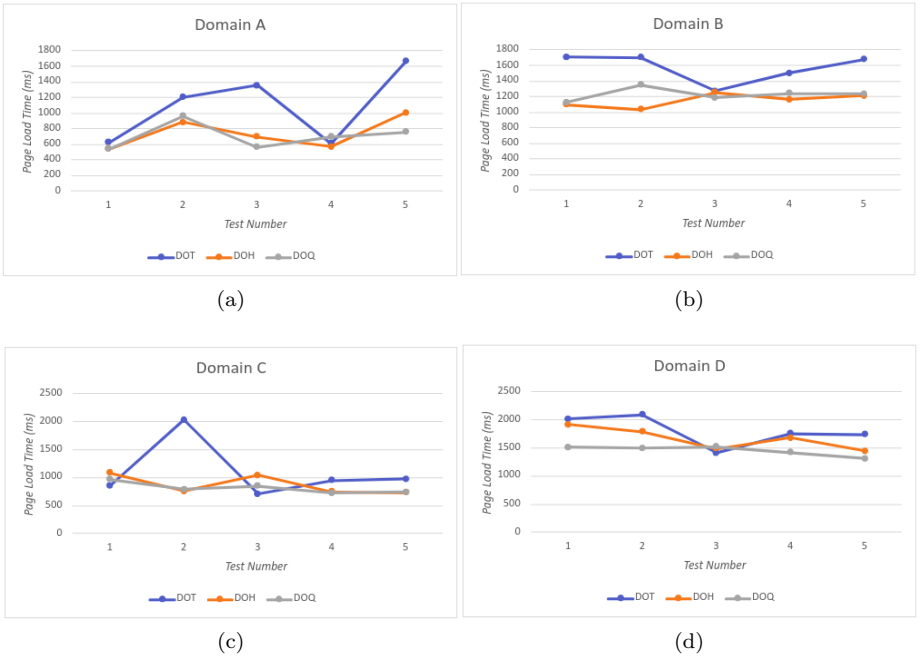

Figure 18: Page load time test for domains A to D

Following up on the handshake duration observations above, figure 18 presents outcomes of the page load time tests collected across the selected four domains A to D. These results do not take into account the handshake times, all collected data points where made after a secure handshake had been established to allow the tests to focus on the ability of each protocol to resolve all domain names within a web page. Across all domains, the results indicate that when DoQ is implemented, web page load times are generally similar to using DoH. The only notable difference is with domain D where load times with DoQ performs considerably better than with DoH. Alternatively page load times while using DoT appear to be significantly slower, This is evident across all four domains where with DoT page load times can be up to 700ms slower portraying a significant difference in performance.

6.2.3 Lookup Time

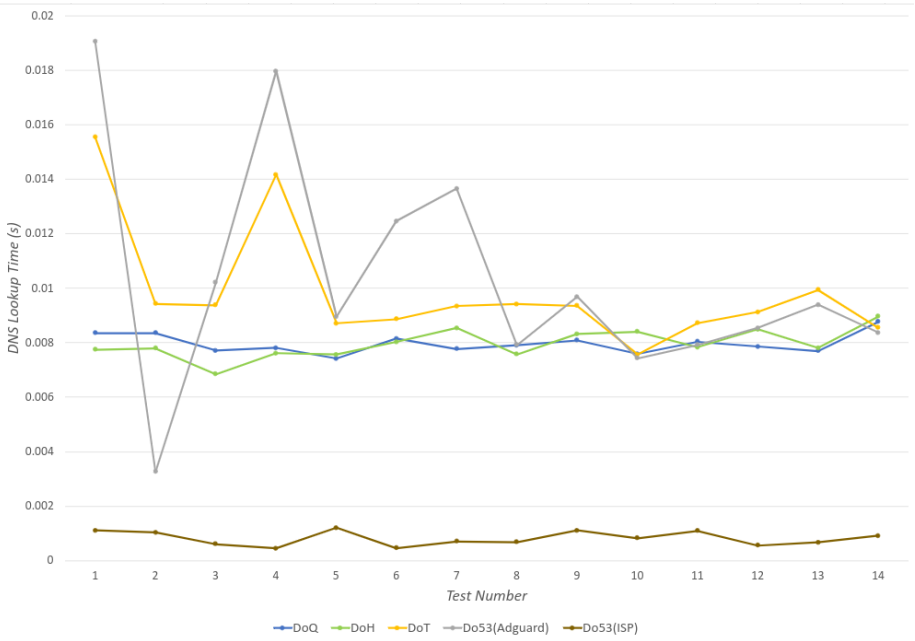

Figure 19: DNS lookup time for `aston.ac.uk` across DoQ, DoT, DoH and Do53

As for DNS lookup times, figure 19 above illustrates the differences among the proposed encrypted DNS solutions and traditional DNS. In this case two different Do53 setups are recorded. First Do53(Adguard) which simply using Adguards DNS server to send unencrypted DNS traffic. Secondly Do53(ISP) which refers to the use of the local default DNS resolver to the network used. This is done to investigate the difference in lookup time when using a closer DNS resolver.

The results above have indicated a major difference in DNS lookup times between encrypted implementations and the default Do53 which is traditionally the ISP's DNS server. Given that the ISP's servers are significantly closer this observation comes with no surprise. Surprisingly however, unencrypted DNS lookups to Adguards DNS server at times actually produced slower results than the encrypted implementations. With regards to DoQ and DoH, similar trends are observed indicating that the two protocols are of similar performance when it comes to resolving DNS queries after a handshake has been established. Overall consistency again appears as a reoccurring feature of DoQ and DoH. Alternatively, results from implementing DoT show larger inconsistencies among different tests at the beginning, later on DoT's inconsistencies reduce considerably but the lookup times are still noticeably slower than DoQ and DoH.

6.2.4 Total DNS Lookup Time

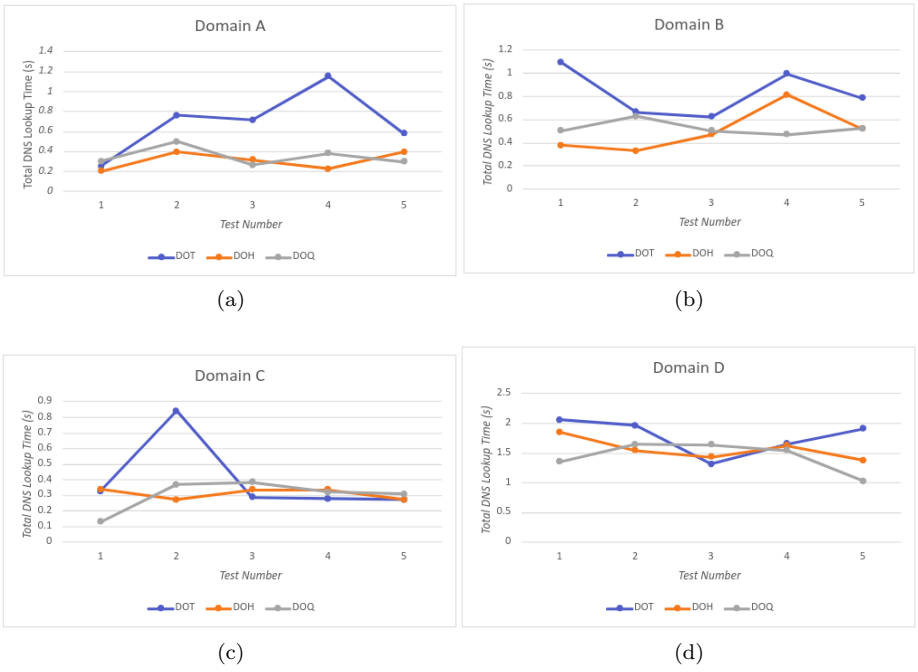

Figure 20: Comparison of DNS lookup time

As section 5.3 outlines, this area looks at the time taken for all DNS queries to be resolved within each of the domains selected. The trends show similar indications to previous investigations of earlier components, DoQ and DoH's performances are quite similar while DoT's results are considerably varying and inconsistent. From the trends in figure 20a DoH and DoQ results are somewhat interchangeable and lie within similar ranges (0.2 to 0.4/0.5 seconds), DoT on the other hand is in most data points significantly slower. Similar trends are portrayed in 20b and d. Nevertheless in 20c, DoT performance matches DoQ and DoH excluding the outlier on the 2nd data point.

7 Discussion and Further Research

Elaborating on the investigation findings in section 6, the following discussions are interpreted with regards to the available academic literature regarding the investigated protocols.

Firstly, with regards to the impact of the investigated encrypted protocols on the privacy of users, the results in this report indicate that DoH is better at enhancing the privacy of users when resolving domain names through an established https connection. This partially due to the fact that DoH traffic is hard to distinguish from normal web traffic on port 443 [30]. The only way to differentiate between DoH and traditional HTTPS traffic is through IP filters targeted at known DoH servers. However as DoH implementations grow and more unknown or private DoH servers exist the complexity of being able to distinguish between the two becomes harder to achieve. Having said that, it is important to note that the HTTPS protocol itself has privacy concerns due to varying implementations. Moreover, as the *Encrypted DNS Deployment Initiative* (EDDI) mentioned in their talk in March of 2022 that there is a possibility that browsers implementing DoH could add identifiers to the HTTP layer in order to allow DNS filtering and other security features to be able to deal with encrypted DNS traffic [8]. This ultimately indicates that although DoH traffic may appear to be the most privacy enhancing among the rest, future developments to DoH and potential changes to HTTP by browsers may increase the privacy risk of using DoH as a private DNS implementation.

Secondly QUIC has showed promising performance results being used as a encrypted transport to DNS. Using Adguard's dnsproxy that bases it's implementation of DoQ on the latest IETF draft [15], DoQ traffic has showed to be similarly susceptible to passive monitoring to DoH and DoT however is notably better performance wise. DoQ has generally maintained consistent performance results unlike DoT and DoH that between subsequent tests resulted in inconsistent performances. In terms of the handshake duration examined in figure 17, DoQ's 1-RTT handshake was notably faster than both DoT and DoH. As QUIC's design allows for 0-RTT handshakes to known servers, this can further enhance the performance of the protocol as DNS clients will only be required to establish the handshake at the first instance [15]. QUIC's connection migration feature could also benefit DNS clients that typically change their IP addresses frequently and will in this case benefit from not needing to establish a new QUIC connection. This however poses serious privacy concern that could allow DNS servers to track clients across different networks [17, 28].

Although encrypted solutions are not enough to prevent pervasive monitoring attacks to uncover visited domains, this is a step in the right direction and the accuracy of the attack can be affected by many changing variables (browser used , location of attacker at the network topology, encrypted DNS implementations used and more). Regardless of the concerning observation that such pervasive monitoring attacks can be applicable without the need for large data samples as presented in section 6.1.2, the fact of the matter is that the success of such attempts can easily be outdated and no longer accurate with changes to the web contents of the domain names, location of hosting (moving the domain to a different provider) or even more simply changing the choice of recursive re-

solver. All of the cases earlier will affect some of the observed packet features and ultimately affect the accuracy of the attack described in section 5.2.6.

With regards to the future plans to implement encrypted DNS solutions on the recursive to the authoritative end, the performance findings of section 6.2 indicate that TCP based solutions do not seem very promising but will have to wait for further research to confirm. This is because even though in the performance investigations in section 6.2 show in some cases that DoT is only marginally slower, the margin of difference is expected to multiply due to the increasing number of hops in the recursive to authoritative end. Additionally, given the larger geographic network scope involved in the recursive to authoritative side (communications will likely cross international boundaries) there will be more handshakes required and more chances of packet loss, congestion control issues and more. All of which will consequently further impact the performance of the protocols.

Finally, commenting on the overall scope of privacy leaks in network communications discussed in section 5.2, sufficing with encrypting DNS traffic will be useless as long as the SNI variable is not encrypted. To gain the privacy benefits the proposed encrypted solutions, both eSNI and encrypted DNS need to be implemented. In this case if the IP address of the destination is not unique to the domain visited then the true privacy benefits can be experienced.

8 Evaluation

In attempt to draw understanding from the obtained results from section 6, this section evaluates next the methods used to achieve the overall goal of the investigations.

8.1 Investigation Limitations

After conducting the various investigations for privacy and performance testing described in section 5, an evaluation of the limitations of the methodology is discussed below.

8.1.1 Client to Recursive Scope of Investigation

The first notable limitation is involved with the scope of research, as described earlier, the current research around DNS privacy has mainly focused the efforts downstream of the recursive. The reasons for this are understandable as mentioned in earlier sections however, when it comes to investigating the performance of DNS, testing the protocols on the recursive to authoritative end is a better alternative to learn about the true performance potential of the proposed protocols given the complexity of communications and scale of operation.

Nevertheless given the context of this problem the performance investigations of this report takes into account this limitation and instead uses what is observed within the client to recursive end to discuss the potential in the recursive to authoritative end as will be presented later.

8.1.2 Selection of Packet Features

The set of selected features listed in section 5.2 have a significant impact on the outcomes of the investigations. Based on the initial set, the outcomes have presented as described in the findings section that classification model performs notably worse with DoH than DoQ and DoT. As a result, the outcomes suggest that using DoH is a better option to enhance the privacy of users. This perspective however may actually not provide a full or fair perspective. The selection of features that the classification model relied on are simply size and time related features of individual DNS transactions. No sequence based features were included in the privacy tests which after detailed study of traffic analysis has appeared to be a considerable limitation to this report's investigation.

For instance when triggering `www.bewater.la` through the browser, three other domains are subsequently triggered `api.mapbox.com`,
`backend.bewater.la` and `events.mapbox.com`. This sequence of DNS requests provide uniqueness of behaviour among different domains that is not examined when looking at single DNS transactions and ultimately the way in which HTTPS, QUIC or TLS connections deal with multiple queries in parallel can uncover another area of weaknesses or strengths of the protocols used.

8.1.3 Test Sample Sizes

Another limitation with the report's investigations is regarding the sample sizes used in both the privacy and performance areas. Given the amount of research required for the report and the time constraints associated with this piece of work, the amount of data samples used to classify different domains using the classification model can be improved to establish a better picture of the actual attributes of each domain across the different protocols. Additionally the privacy tests worked around a set of four domains to achieve the classification report in figure 16. Although the outcomes were sufficient to portray the risk of pervasive monitoring on encrypted traffic using simple features, the investigation does not take into account how the effectiveness of the attack will change in a more realistic scenario with a larger set of domains.

Nevertheless, such limited sample sizes similar to the case in this report show that such de-anonymization attempts do not require a lot of data to produce high levels of accuracy. More so, in Bushart and Rossow [6] investigations, they used larger data sets (18 domains and 51 samples for each) however the results similarly showed high accuracy scores indicating the same idea that encrypting DNS alone will not prevent pervasive monitoring attacks from uncovering the

visited domains.

9 Conclusion

All things considered, DNS privacy concerns have never gained more attention than in recent years [30, 26]. Although the recently adopted protocols DoT, DoH and DoQ provide confidentiality to DNS exchanges, a number of studies including this investigation show that the use of such implementations can be de-anonymised by classification algorithms that mainly use time and size packet features to identify domain names. The investigation evaluated that DoH has considerably been the hardest protocol to deanonymize however, the F1 score of 0.7325 and accuracy of 0.75 using a 3-NN classifier was indicative of the fact that the protocol remains highly susceptible to such de-anonymization attempts. With regards to the performance of the adapted protocols, DoQ's latency enhancing features embedded into it's design generally showed only minor improvements compared to TCP based solutions and especially DoH that in some cases performed better, this however is understandable since the limited investigation scope that focuses solely on the stub to recursive does not test the full performance potential for DoQ. On that note, DoQ's margin of improvement with regards to the quicker handshake design (evident in section 6.2.1) is expected to increase significantly in the more complex recursive to authoritative end.

That being said, from a broad perspective, attaining the full privacy benefits of encrypting DNS will require the other elements such as eSNI and shared web hosting to be present. However, even when such assumptions can be made as done in this report, the DNS privacy concerns are not completely addressed by the proposed protocols. After all, such protocols remain relatively immature as Yan and Lee [30] describe and consequently future improvements will need to consider the uniqueness of time and size features to mitigate de-anonymization threats as presented in this report.

References

[1] 2022. URL: https://datatracker.ietf.org/wg/dprive/about/.

[2] URL: https://publib.boulder.ibm.com/tividd/td/ITWSA/ITWSA_info45/en_US/HTML/guide/r-dnsinfo.html#:~:text=Before%5C%20DNS%5C%2C%5C%20every%5C%20machine%5C%20on,The%5C%20HOSTS..

[3] URL: https://wiki.archlinux.org/title/Systemd-resolved.

[4] Stéphane Bortzmeyer. *DNS Query Name Minimisation to Improve Privacy*. RFC 7816. Mar. 2016. DOI: 10.17487/RFC7816. URL: https://www.rfc-editor.org/info/rfc7816.

[5] Jason Brownlee. *What is a Confusion Matrix in Machine Learning*. 2020. URL: https://machinelearningmastery.com/confusion-matrix-machine-learning/#:~:text=A%5C%20confusion%5C%20matrix%5C%20is%5C%20a%5C%20summary%5C%20of%5C%20prediction%5C%20results%5C%20on,key%5C%20to%5C%20the%5C%20confusion%5C%20matrix..

[6] Jonas Bushart and Christian Rossow. "Padding Ain't Enough: Assessingthe Privacy Guaranties of Encrypted DNS". In: *CISP Helmholtz Center of Information Security* (2019).

[7] Sara Dickinson. *DNSPRIV Tutorial*. 2017. URL: https://datatracker.ietf.org/meeting/99/materials/slides-99-edu-sessh-dns-privacy-tutorial-00. (accessed: 07.07.2021).

[8] Sara Dickinson and Allison Mankin. *Encrypted DNS Policy and Technical Considerations*. 419 Consulting LTD. 2022. URL: https://www.youtube.com/watch?v=F1P5WNsZjL0.

[9] Nguyen Phong Hoang et al. "Assessing the Privacy Benefits of Domain Name Encryption". In: *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*. ACM, Oct. 2020. DOI: 10.1145/3320269.3384728. URL: https://doi.org/10.1145%2F3320269.3384728.

[10] Nguyen Phong Hoang et al. "Assessing the Privacy Benefits of Domain Name Encryption". In: *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*. ACM, Oct. 2020. DOI: 10.1145/3320269.3384728. URL: https://doi.org/10.1145%5C%2F3320269.3384728.

[11] Paul E. Hoffman and Patrick McManus. *DNS Queries over HTTPS (DoH)*. RFC 8484. Oct. 2018. DOI: 10.17487/RFC8484. URL: https://www.rfc-editor.org/info/rfc8484.

[12] Rebekah Houser et al. “An Investigation on Information Leakage of DNS over TLS”. In: *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*. CoNEXT ’19. Orlando, Florida: Association for Computing Machinery, 2019, pp. 123–137. ISBN: 9781450369985. DOI: 10.1145/3359989.3365429. URL: https://doi.org/10.1145/3359989.3365429.

[13] Zi Hu et al. *Specification for DNS over Transport Layer Security (TLS)*. RFC 7858. May 2016. DOI: 10.17487/RFC7858. URL: https://www.rfc-editor.org/info/rfc7858.

[14] Bert Hubert. *NLNOG 2019 - DNS over HTTPS considerations*. NLNOG. 2019. URL: https://www.youtube.com/watch?v=pjin3nv8jAo&list=WL&index=19.

[15] Christian Huitema, Sara Dickinson, and Allison Mankin. *DNS over Dedicated QUIC Connections*. Internet-Draft draft-ietf-dprive-dnsoquic-11. Work in Progress. Internet Engineering Task Force, Mar. 2022. 34 pp. URL: https://datatracker.ietf.org/doc/html/draft-ietf-dprive-dnsoquic-11.

[16] Christian Huitema and Eric Rescorla. *Issues and Requirements for Server Name Identification (SNI) Encryption in TLS*. RFC 8744. 2020. DOI: 10.17487/RFC8744. URL: https://www.rfc-editor.org/info/rfc8744.

[17] Jana Iyengar and Martin Thomson. *QUIC: A UDP-Based Multiplexed and Secure Transport*. RFC 9000. May 2021. DOI: 10.17487/RFC9000. URL: https://www.rfc-editor.org/info/rfc9000.

[18] Lin Jin et al. “Understanding the Impact of Encrypted DNS on Internet Censorship”. In: *Proceedings of the Web Conference 2021*. WWW ’21. Ljubljana, Slovenia: Association for Computing Machinery, 2021, pp. 484–495. ISBN: 9781450383127. DOI: 10.1145/3442381.3450084. URL: https://doi.org/10.1145/3442381.3450084.

[19] Vijay K.Gurbani et al. “When DNS Goes Dark: Understanding privacy and shaping policy of an evolving protocol”. In: *TPRC48: The 48th Research Conference on Communication, Information and Internet Policy* (2020). DOI: http://dx.doi.org/10.2139/ssrn.3749764.

[20] Adam Langley et al. “The QUIC Transport Protocol: Design and Internet-Scale Deployment”. In: *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. SIGCOMM ’17. Los Angeles, CA, USA: Association for Computing Machinery, 2017, pp. 183–196. ISBN: 9781450346535. DOI: 10.1145/3098822.3098842. URL: https://doi.org/10.1145/3098822.3098842.

[21] Andrey Meshkov. *DNS Proxy*. https://github.com/AdguardTeam/dnsproxy. 2021.

[22] Aziz Mohaisen et al. “Look-Aside at Your Own Risk: Privacy Implications of DNSSEC Look-Aside Validation”. In: *IEEE Transactions on Dependable and Secure Computing* 17.4 (2020), pp. 745–759. DOI: 10.1109/TDSC.2018.2816026.

[23] Martin Pramatarov. *DNS history. when and why was DNS created?* Aug. 2021. URL: https://www.cloudns.net/blog/dns-history-creation-first/.

[24] Sandra Deepthy Siby et al. “DNS Privacy not so private: the traffic analysis perspective”. In: 2018.

[25] Sandra Deepthy Siby et al. “Encrypted DNS -¿ Privacy? A Traffic Analysis Perspective”. In: *ArXiv* abs/1906.09682 (2020).

[26] Martino Trevisan et al. “Does domain name encryption increase users’ privacy?” In: *ACM SIGCOMM Computer Communication Review* 50 (July 2020), pp. 16–22. DOI: 10.1145/3411740.3411743.

[27] Jeffrey Walton. *Re: SNI is a security vulnerability all by itself*. 2021. URL: https://seclists.org/oss-sec/2021/q3/77.

[28] Tim Wicinski. *DNS Privacy Concerns*. RFC 9076. RFC Editor, July 2021. URL: https://datatracker.ietf.org/doc/rfc9076/.

[29] Jared Wilber. *Precision and Recall*. 2022. URL: https://mlu-explain.github.io/precision-recall/. (accessed: 14.01.2021).

[30] Zhiwei Yan and Jong-Hyouk Lee. “The road to DNS privacy”. In: *Future Generation Computer Systems* 112 (2020), pp. 604–611. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2020.06.012. URL: https://www.sciencedirect.com/science/article/pii/S0167739X20307068.

A K-Nearest Neighbour Classifier

```
import pandas as py
import numpy as np
import math

# K-NN Import
from sklearn.neighbors import KNeighborsClassifier

# Imported Metrics
from sklearn.metrics import f1_score, confusion_matrix, accuracy_score,
classification_report

#Import the dataset csv file (DoT, DoH and DoQ have individual files)
dataset = py.read_csv('Dataset.csv')

x = dataset.iloc[:, 0:7].values      #Testing Features
y = dataset.iloc[:, 7].values        #Ground Truth Values

# Square root of the total data samples (TO LATER USE AS K VALUE)
# If the result is an even number we subtract 1 and round to the nearest
integer

math.sqrt(len(y))

# Initialising the classifier model; K-NN and assigning a value for K
classifier = KNeighborsClassifier(n_neighbors=3)

# Fitting the model
# K-NN Model is fit using the dataset values (x) and the ground truth values (y)

classifier.fit(x, y)

# To predict an unlabelled set of features:
#
# Example:
# > y_pred = classifier.predict([[95, 112, 23, 2400, 1435, 0.007, 2.2]])
```

[CONTINUES IN THE FOLLOWING PAGE]

```
# Attempting to test the predictions of the classifier based on the model:
# The (x) variable which consists of the entire data set (excluding the ground
    truth values) is given as parameter to the predict() function.

y_pred = classifier.predict(x)

# Print model predictions
print(y_pred)

# Print results
print(classification_report(y, y_pred))     #Classification Report
print(confusion_matrix(y, y_pred))          #Confusion Matrix
print(accuracy_score(y, y_pred))            #Overall Accuracy Score
```