DNS Privacy:
Concerns, Issues and Technologies

David Roth, John Worley, Jose Castejon-Amenedo, Ian Cohee
Secure64 Software

The Domain Name Service (DNS) is *the* critical Internet service, without which the modern Internet and the information economy would not exist. It is a public, distributed, highly-redundant, highly-scalable distributed database of translations between domain names, e.g., `www.google.com`, and the information needed to access resources and services, such as web pages, mail servers, files, time of day, etc.

The most common and familiar function is mapping a domain name to an IPv4 or IPv6 address. The mapping is required in order for an application to open a network connection to a remote service. For example, when `www.google.com` is typed into a browser, the domain name portion of the URL is submitted to the system's *resolver* library, which in turn initiates a domain name lookup to the system's configured DNS server, usually an external *caching recursive resolver* configured by the network provider. The web browser then awaits a valid response containing the IP address to connect to the web site over HTTP or HTTP/S. Critical system applications such as NTP function in a similar manner.

DNS Privacy

The domain name system is considered public knowledge and as such DNS queries and responses are sent over the network in the clear. Because DNS data is unencrypted, it is easily collected. Of particular interest is who sent the query and what domain they are visiting. This information may be used to compile end user profiles, which in turn may be used for targeted advertising, or competitive analysis[1]. While DNS data does not reveal the full extent of a user's activities, it does provide valuable metadata for big-data analysis.

To address DNS privacy issues, there are a number of active initiatives to secure DNS requests on the wire, led by `dnsprivacy.org` and within the IETF. This paper provides a detailed look at the problem space, emerging standards, and services available today to protect client DNS privacy. In particular we'll cover the IETF standards for DNS over HTTPS (hereafter DoH) and DNS over TLS (hereafter DoT). The focus will be on stub-to-resolver privacy issues. These approaches to DNS privacy build upon the use of TCP and cryptographic protocols.

Security and Privacy

In many discussions about DNS, the terms security and privacy are used interchangeably; however, they have very different meanings and distinct implications.

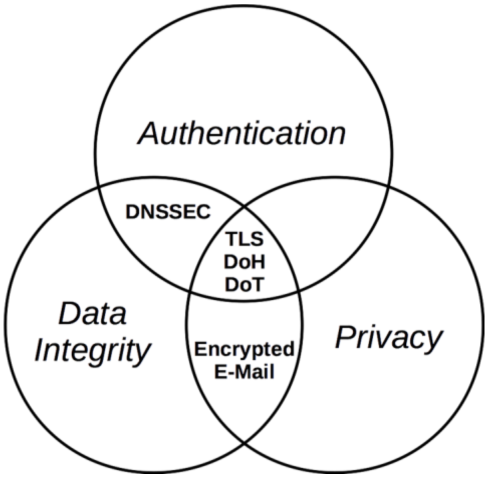
Privacy only means that the contents of a DNS query cannot be viewed in transit by a third party, often referred to as a "man in the middle", or MITM. One simple mechanism to ensure privacy would be to have a point-to-point connection between the client and the recursive server. In the Internet environment, where packets will traverse many hops (routers, proxies, etc.) between the client and server, privacy requires some form of encryption.

[1][1]POWERDNS Technical Blog, "On Firefox moving DNS to a third party"., https://blog.powerdns.com/2018/09/04/on-firefox-moving-dns-to-a-third-party/.

Security, on the other hand, is composed of three different attributes, each of which may be independently present. Privacy is one attribute, but security also includes:

- *Authentication*, a process by which a client knows, with a high degree of certainty, it is communicating with the expected server, and
- *Data Integrity*, a process by which the client knows, with a high degree of certainty, data sent and received cannot be altered or changed without being detected.

As with privacy, both authentication and data integrity use cryptographic methods. However, it is possible to be secure, but not private. An important example of this is DNS Security (or DNSSEC), which protects communication between recursive resolvers and the authoritative servers.



In DNSSEC, authoritative responses are sent in the clear (without encryption) but accompanied by special DNS records called Resource Record Signatures (RRSIG) and DNSSEC Keys (DNSKEY). Together, these records allow the recursive resolver to validate the response, which prevents the response from being altered by a MITM (data integrity). Additionally, the DNSKEY records can be validated against another DNS record, the Delegation of Signing (DS), giving a high degree of confidence the authority is returning the expected data (authentication).

DNS Privacy Considerations

RFC 7626 covers DNS privacy considerations in detail. What follows is a summary of the salient points.

1. A DNS resolver query is typically sent over the wire as an unencrypted UDP packet. This traffic is easily decoded using packet analysis tools such as WIRESHARK, TCPDUMP or deep packet inspection applications. The particular information of interest is the IP source address and the query name. The source port number is also of interest as it may differentiate clients behind a NAT'ed network.

2. Web browsers when connecting to web sites may trigger a number of primary, secondary, and even tertiary requests for name resolution:

    - The primary request is for the website's domain name.
    - Secondary requests are from embedded links in the downloaded HTML page.
    - Tertiary requests which result when a caching server does not have the domain name in its cache and consults the authoritative servers to find it.

    As well, some browsers may prefetch domain names for future use or to autocomplete the URL in the address bar.

3. DNS data is considered public as noted above. However, RFC 7626 argues (§2.1) the need to differentiate between the DNS data and the transaction:

    DNS data and the results of a DNS query are [publicly available] and may not have any confidentiality requirements. However, the same is not true of a single transaction or a sequence of transactions; that transaction is not / should not be public. A typical example from outside the DNS world is: the web site of Alcoholics Anonymous is public; the fact that you visit it should not be.

The DNS attack surface is broad and encompasses the system which issues the DNS lookup request, the caching server which processes the request, and authority servers which assist in the domain name lookup. The attack surface will be further examined in subsequent sections.

DNS Privacy Standards and Industry Efforts

A number of technical solutions have been proposed over the last 10 years intended to address DNS privacy: DNSCrypt[2], DNSCurve[3], Namecoin[4], Confidential DNS[5], the GNU name system[6], and Google DNS over HTTPS. All of these designs take different approaches in reducing the role of DNS as the ultimate source of meta data in the digital panopticon known as the Internet.

These approaches to DNS privacy have yet to see the same level of acceptance as DoH and DoT. For completeness, the following sections provide an overview of each.

DNSCurve

Having been first put forth in 2011, this DNS privacy proposal is one of the oldest available. Details about this solution can be found **here**.

This protocol establishes a secure channel between a client and a server by means of cryptographic techniques based on the Curve25519 elliptic curve[7]. It is meant to be used by caching servers to exchange DNS information with other caching or authoritative servers.

DNSCrypt

DNSCrypt is a formal proposal for the use of DNSCurve.

Namecoin

Namecoin is an approach to DNS confidentiality that uses blockchain techniques. This proposal requires a major overhaul of the DNS infrastructure.

Confidential DNS (IETF Draft)

This proposal introduces a new DNS record `ENCRYPT`. This record is returned by a DNS server when queried and contains a public key for the server. The protocol uses the public key to securely transfer a small amount of data between the client and server, allowing them to settle on specific key material that they can use in order to establish a secure channel for exchanging encrypted queries, and their associated responses.

GNU name system

The GNU Name System (GNS) is part of the GNU Net[8] effort to provide a framework for secure peer-to-peer networking.

[22]https://en.wikipedia.org/wiki/DNSCurve

[33]DNSCrypt, https://github.com/DNSCrypt/dnscrypt-protocol/blob/master/DNSCRYPT-V2-PROTOCOL.txt

[44]https://namecoin.org/

[55]https://tools.ietf.org/html/draft-wijngaards-dnsop-confidentialdns-02

[66]https://gnunet.org/taxonomy/term/34

[77] https://en.wikipedia.org/wiki/Curve2551

[88] https://gnu.net

Like Namecoin, GNS constitutes a complete overhaul of the DNS infrastructure. It relies on the GNU peer-to-peer network, plus a distributed hash table (DHT). The name resolution itself is carried out by means of DHT, whereas the queries and responses are exchanged over the GNS peer-to-peer network. Such data exchange is encrypted and signed, using capabilities provided by the GNU peer-to-peer network.

Google DNS over HTTPS Proposal [9]

This specification defines a web friendly JSON ReST API for the DNS request and response. As such it diverges from RFC 8484, which defines a binary response. A request is sent as follows:

```
https://dns.google.com/resolve?
```

To which the following parameters may be passed: *name*, *type*, *cd*, *edns_client_subnet*, and *random_padding* may be passed as keys with assigned values to further refine the query.

A response is received in the following JSON format:

```
{
  "Status": 0,  // NOERROR - Standard DNS response code (32 bit integer).
  "TC": false,  // Whether the response is truncated
  "RD": true,   // Always true for Google Public DNS
  "RA": true,   // Always true for Google Public DNS
  "AD": false,  // Whether all response data was validated with DNSSEC
  "CD": false,  // Whether the client asked to disable DNSSEC
  "Question":
  [
    {
      "name": "apple.com.",  // FQDN with trailing dot
      "type": 1              // A - Standard DNS RR type
    }
  ],
  "Answer":
  [
    {
      "name": "apple.com.",   // Always matches name in the Question section
      "type": 1,              // A - Standard DNS RR type
      "TTL": 3599,            // Record's time-to-live in seconds
      "data": "17.178.96.59"  // Data for A - IP address as text
    },
    {
      "name": "apple.com.",
      "type": 1,
      "TTL": 3599,
      "data": "17.172.224.47"
    },
    {
      "name": "apple.com.",
      "type": 1,
      "TTL": 3599,
      "data": "17.142.160.59"
    }
  ],
```

[9]9 https://developers.google.com/speed/public-dns/docs/dns-over-https

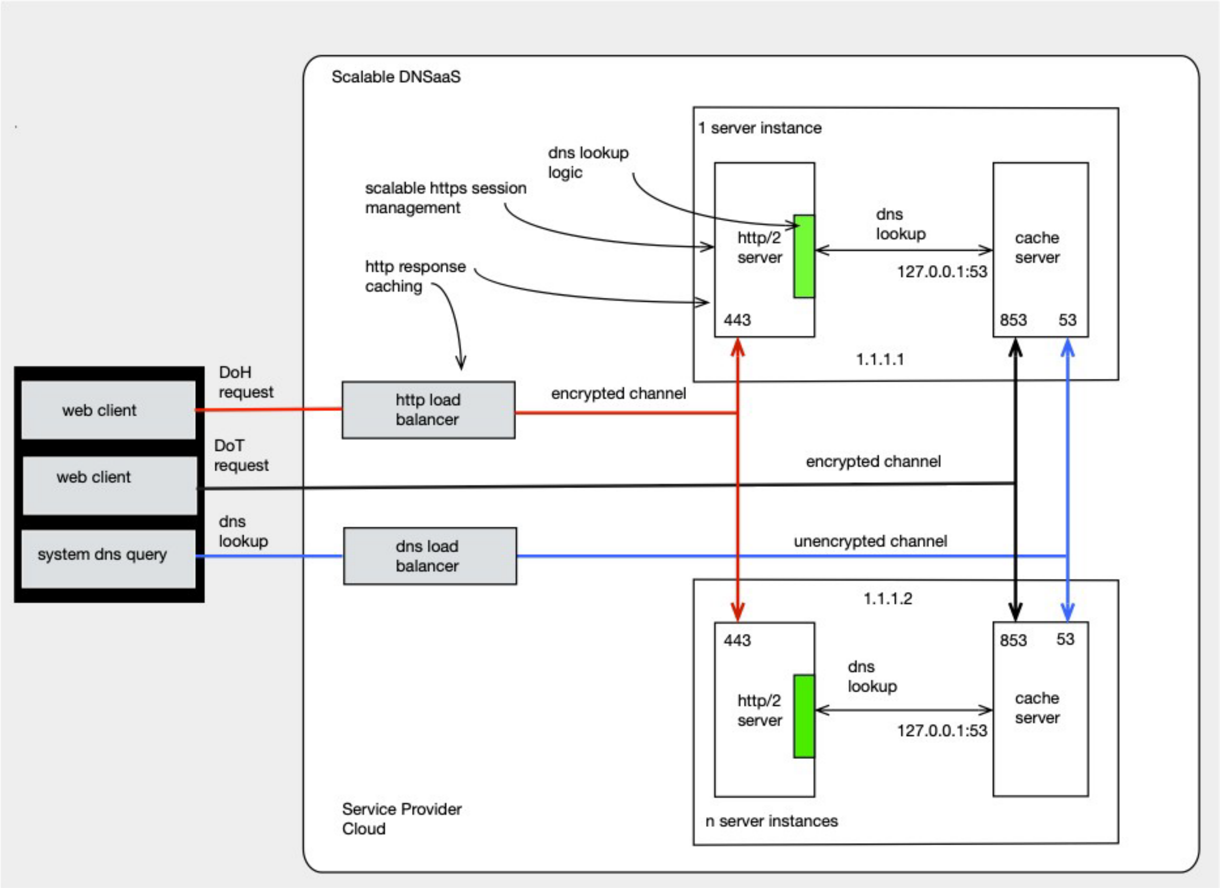
```
          "Additional": [ ],
          "edns_client_subnet": "12.34.56.78/0"  // IP address / scope prefix-length
        }
```

Query Minimization[10] (IETF Draft)

Query minimization is a non-cryptographic technique to reduce the number of authoritative servers that can see the entire client query during recursive resolution. Instead of sending the full query to each authoritative server, only enough of the right-hand portion of the query is sent to get the information for the next step.

For example, the client queries for `system.somename.someregion.com`. When the resolver sends a query to a nameserver for `.com`, it only sends `someregion.com.`, rather than the entire query. Unfortunately, this technique complicates resolution and will slow down resolution in cases where multiple domain-name labels are present in the same zone.

DoH and DoT



The above diagram shows a side by side comparison of DoT, DoH, and unencrypted DNS. It highlights that DoH and DoT are application specific technologies. The communication between a DoH or DoT aware web client protects the end user from exposing the contents of the DNS query. In the case of system services, the query is not protected until such time as operating system resolver libraries support DNS privacy extensions.

[10] 10https://tools.ietf.org/html/draft-bortzmeyer-dns-qname-minimisation-02

DNS over TLS (DoT)

The DoT proposal is detailed in RFC 7858, along with RFC 8310. It proposes a method of sending and receiving DNS queries over a secure TLS session. The RFC dictates a server open a TLS aware connection on the well-known port 853. Using a standard port allows DNS clients and applications to probe the known port to determine if DoT is supported. The RFC calls out the following implementation guidelines:

1. DNS clients should remember servers which do not support DoT and in the TLS session establishment, the client should verify the server to which it is connecting.
2. DNS clients should support a pipeline in which multiple outstanding queries can be submitted to the server. The client is also responsible for matching a response to the DNS request using the message ID as the responses may be out of order.
3. DNS clients should maintain a reusable TCP/TLS session to the server for the sake of efficiency. Its interaction should comply with RFC 7766 which provides guidelines for implementing DNS over TCP. To facilitate TCP session establishment TCP fast open should be used.

DoT supports two security profiles:

1. opportunistic privacy in which a client may learn of a TLS-enabled resolver from an untrusted resource. In this profile the client has the option to verify or not verify the DoT server
2. out of band, key pinned, cryptographically-verified authentication of the DoT server guaranteeing the trust relationship between client and server

Of the two profiles the second offers the highest security.

DNS over HTTPS (DoH)

The DNS over HTTP/S (RFC 8484) has the following objectives:

1. Preventing on path devices from interfering with or ease dropping on DNS operations
2. Providing web applications with an API which allows them to access DNS data in a safe manner

DoH outlines a method for sending and receiving DNS queries over HTTP using HTTPS (a secure TLS session). The DNS queries are formulated as HTTP GET or POST operations and the responses are encoded as a binary DNS packet. The following provides examples of the requests (Section 4.1.1) and answers (4.2.2):

HTTP Get Request:

These examples use HTTP/2-style formatting from [RFC7540]. These examples use a DoH service with a URI Template of `https://dnsserver.example.net/dns-query{?dns}` to resolve IN A records. The requests are represented as bodies with media type “application dns-message". The first example request uses GET to request "www.example.com".

```
:method = GET
:scheme = https
:authority = dnsserver.example.net
:path = /dns-query?dns=AAABAAABAAAAAAAAA3d3dwdleGFtcGxlA2NvbQAAAQAB
accept = application/dns-message
```

HTTP Post Request:

The same DNS query for "www.example.com", using the POST method would be:

```
:method = POST
:scheme = https
:authority = dnsserver.example.net
:path = /dns-query
accept = application/dns-message
content-type = application/dns-message
content-length = 33
<33 bytes represented by the following hex encoding>
00 00 01 00 00 01 00 00  00 00 00 00 03 77 77 77
07 65 78 61 6d 70 6c 65  03 63 6f 6d 00 00 01 00
01
```

HTTP DNS Response Example:

This is an example response for a query for the IN AAAA records for `www.example.com` with recursion turned on. The response bears one answer record with an address of 2001:db8:abcd:12:1:2:3:4 and a TTL of 3709 seconds

```
:status = 200
content-type = application/dns-message
content-length = 61
cache-control = max-age=3709
<61 bytes represented by the following hex encoding>
00 00 81 80 00 01 00 01  00 00 00 00 03 77 77 77
07 65 78 61 6d 70 6c 65  03 63 6f 6d 00 00 1c 00
01 c0 0c 00 1c 00 01 00  00 0e 7d 00 10
```

A notable advantage in the use of HTTP is its support for caching such that the DNS response can be reissued for a similar query. When caching at the HTTP layer it is important it observe the DNS response TTL.

DoH/DoT DNS Proxy Servers

There are DNS proxy servers[11] and forwarding resolvers[12] available today which sit between a trusted and an untrusted domain. In this deployment model, the DNS proxy creates the secure connection to the DoH or DoT cloud resolver service thus protecting the query from being intercepted and collected as it transits the untrusted domain.

The advantages of implementing DNS privacy in this manner are:

1. The proxy handles all DNS lookups within the trusted domain without having to configure a particular application to use DoH or DoT. All systems are configured to use the proxy resolver as its default.

2. The proxy is under the control of the trusted domain reducing the concern of leaking information via query logging or network data collection.

The disadvantages are:

1. The proxy must be configured and deployed by a knowledgeable admin

[11][11]Cloudflare proxy https://developers.cloudflare.com/1.1.1.1/dns-over-https/cloudflared-proxy/

[12][12]Unbound

2. The proxy represents a single point of failure
3. The proxy must be maintained
4. Performance and scaling may be issues

In the end, for knowledgeable individuals familiar with DNS, configuring and maintaining a proxy DNS service seems viable; however, the necessary expertise to install, configure, and manage such a service is not the norm and is sure to hinder its adoption. An additional hurdle is that DoH and DoT do not guarantee an end user complete browsing anonymity. If this is the objective other technologies such as VPNs and TOR offer a more complete privacy solution.

DoH/DoT Cloud Based Resolver Initiatives

Today Google, IBM (Quad9), and Cloudflare offer cloud-based DNS open resolvers which support DoH and DoT. In addition to these mainstream open resolvers, there are fringe DNS resolvers such as BlahDNS[13]. Cloudflare has aligned itself with Firefox (Mozilla) to offer DoH support. While Google supports DoH in Chrome and Android Pie supports DoT as an OS level configuration option.

The Mozilla Firefox DoH configuration is easily enabled in the `Preferences->Network Settings` dialog.

Google Android Pie Private DNS mode comes with the following caveat[14]:

> *The one caveat to this is that some apps have their own, built-in DNS querying mechanisms, which means they will not honor the Private DNS setting. Hopefully Android developers will retool their apps to allow for the use of global DNS. What that means is you cannot be certain if an app is using your newly configured DNS over TLS or their own, insecure, DNS mechanisms. Hopefully Android developers will make this change soon.*

Cloudflare Application on iOS (1⁴ application)

While the Android API is fairly open to developers, more constraints are placed on iOS application developers. For example, Android application developers are more or less allowed to make arbitrary network connections, allowing applications to have one-off name resolution implementations.

While Android 9 ("Pie") allows for the configuration of "Private DNS" at the device level, iPhone users can only configure a DNS resolver for each WiFi network known to the device. This is where iPhone VPN configuration enters: the developers at Cloudflare have utilized the pushing of network configurations via VPN profiles to realize a DNS-over-HTTPS solution, for iPhone users.

Initial observations indicate that the Cloudflare VPN profile does not alter routes nor add tunnels other than a "push-dhcp" dns configuration

TCP and TLS Performance Considerations

Both DoH and DoT are DNS TCP based solutions which assume the use of the TLS protocol to establish a secure channel between the client and the server. TCP session establish and session maintenance result in additional overhead. Once the TCP session is established, TLS performs cryptographic operations for key negotiation between client and server to establish the secure connection.

The following performance considerations apply equally to DoH and DoT:

1. latency: the time to establish the TCP session
2. latency: the time to complete the TLS handshake
3. system resources: TCP session management within the kernel
4. compute resources: TLS session cryptographic operations

[13][13] https://blahdns.com/

[14][14] https://www.techrepublic.com/article/how-to-enable-dns-over-tls-in-android-pie/

TLS Attack Surface

SSL and TLS have a long history of vulnerability to attack. A complete handling of successful attacks on TLS since 2009 can be found in Bulletproof SSL and TLS, chapter 7. In broad strokes, just to name a few categories, TLS and SSL are vulnerable to:

1. Insecure renegotiation
2. Side channel attacks to retrieve cryptographic materials
3. Compression side channel attacks
4. Key and cryptographic cypher weaknesses
5. Traffic analysis
6. TLS attacks such as down-grade and man in the middle

And the following specific attacks:

1. Low Slow DDoS attack
2. R.U.D.Y DDoS attack

Recursive Resolver Attack Surface

The resolver attack surface is detailed in RFC 7626. Of primary concern are the following: The processing of the client request by the resolver exposes the src address of the client and the domain name queried. This information can be captured on the recursive resolver through application level query logging or an instance of TCPDUMP or DNSTAP. Using any of these technologies a complete accounting of DNS queries and all resulting interaction with the upstream authoritative servers is available.

HTTP/S Attack Surface

HTTP has a broad and deep attack surface. A complete handling is outside the scope of this paper. The vulnerabilities discussed below are restricted to TLS in the DoH context.

1. HTTP/S connection to web site is exposed in the TLS certificate handshake revealing the hostname of the server to which the session connects via the SNI certificate field.
2. As well the IP destination address of the site visited can be easily tracked and mapped to the site.

DoH/DoT Adoption Challenges

1. There is broad concern in the technical community that DNS name resolution will be controlled by just a few companies, namely Google, Cloudflare, and IBM.
2. Because these privacy services are TCP/TLS based, meeting the needs of a large user base will require a significant amount of computing resources to handle the potential high number of concurrent TCP connections.
3. Today the use of DoH and DoT is supported within targeted applications such as web browsers. Windows, Linux, BSD kernels do not support the DNS privacy extensions in their system resolver libraries. As a result, the DNS configuration is fragmented between applications which support DoH and DoT, applications which do not.
4. Supporting a fragmented DNS resolver configuration may lead to support issues. For example, my web server is able to access web sites on the Internet, but my email client is unable to connect to the email server.
5. When applications are configured to use DoH and DoT and the resolution service is unavailable, the application behavior is undefined and may result in a DNS request being sent in the clear.
6. TLS 1.2 has an inherent security leak in that the TLS handshake does not encrypt the server name to which the HTTPS client is connecting thus revealing end user activity. The failure to encrypt the SNI field is addressed in TLS 1.3.
7. Many web servers support TLS version 1.3; but web browsers have been slow to adopt it.
8. While traffic between the web browser flows over a secure channel, the HTTP/S IP destination address is always visible to an attacker and can be mapped to the site.
9. While DoH appears to have more momentum in terms of adoption, there is no clear Industry consensus on which approach should be the standard.
10. Some browsers such as Chrome and Firefox offer DNS privacy configuration options, but others have yet to follow suit.
11. TCP and TLS offer a broad attack surface particularly for DDoS attacks. To defend against these attacks a scalable cloud-based solution in addition to network based DDoS mitigation strategies are required.
12. Neither DoH nor DoT completely guarantee privacy on the wire or privacy of HTTP browsing.
13. There exist better Internet browsing technologies for those wishing to remain anonymous, namely a VPN connection to the Internet or the use of TOR.
14. Should Service Providers elect to deploy DoH or DoT, they will still have the means to collect DNS data and build profiles.
15. All Service Providers are subject to Government lawful intercept requests. In fact in the Telco industry lawful intercept support is built into some standards such ETSI TS 103 523-3 V1.1.1 (2018-10), CYBER; Middlebox Security Protocol; Part 3: Profile for enterprise network and data centre access control**.**
16. The widespread adoption of DNS privacy only makes sense if there is a strong consensus among end users for its need. As evidenced by Internet users lack of concern for privacy violations today

by companies such as Google and Facebook, it is hard to imagine DNS privacy will be of broad concern.

Conclusions

1. DoH and DoT offer viable solutions for securing DNS queries over the network between the client and the recursive resolver.
2. End-to-end DNS privacy cannot be guaranteed, however. When the DNS query is processed by the recursive resolver or other query handling technology (e. g. a proxy) the source address of the query and requested domain are in the clear and easily logged.
3. Because DNS data is easily obtained, in order to be credible, service providers will have to state their data retention policies as exemplified by Cloudfare which retains DNS data for 24 hours.
4. Because DoH and DoT build on TCP and TLS, they introduce a broader attack surface making them vulnerable to attack.
5. Until there is broad adoption of TLS 1.3 by the client applications, the SNI field which carries the name of the host to which the HTTPS connection is established will remain in the clear and leak DNS data.
6. DoH and DoT services are gaining some traction as they are configurable in a few popular browsers and mobile phones.
7. DoH and DoT present a challenge to Service Providers who offer services to safe guard web browsing for their customers because the DNS query is hidden from them. To ensure DNS queries are processed by their infrastructure, they may need to block access to 1.1.1.1 or other DoH or DoT based open resolvers.
8. Because DoH and DoT are TCP based, they necessarily require cloud service scale to address the demands of handling a large volume of connections. Those who wish to deploy the service are left to decide the cost benefit ratio.
9. Cloudflare’s 1^4 application for iOS and Android delegates all DNS queries on the device to their service. This obviates the need to modify the OS resolver libraries and may address the DNS configuration fragmentation stated earlier in this paper.
10. Since the role of DNS is largely unknown to the general public, it is hard to predict 1^4 adoption or willingness to configure DNS privacy on Mozilla or Chrome. In the end, it may be only a small percentage of users who elect to use the service.
11. It is difficult to assess customer demand for DNS privacy or privacy in general.
12. DNS privacy is offered today by VPN services and to some degree the TOR network. So for customers seeking complete anonymity, these solutions obviate the need for DNS privacy.
13. Because of their deployment complexity, the broad adoption of DoH proxies in the home seems unlikely. That said, it may be an option within the enterprise.
14. In general, however, most large companies are less concerned with DNS and more concerned with monitoring and blocking HTTP traffic which they often do by requiring all HTTP access to the web be through a proxy.

DoH and DoT both offer viable solutions for securing DNS queries over the network between the client and the recursive resolver. However, end-to-end DNS privacy cannot be guaranteed with these technologies in place. When the DNS query is processed by the recursive resolver or other query handling technology (e. g. a proxy), the source address of the query and requested domain are in the clear and can easily be logged, thus negating the privacy of the transaction. Because this DNS data is easily obtained, credible service providers implementing DoH or DoT and promoting privacy will have to state their data retention policies, as exemplified by Cloudfare which states they retain DNS data for only 24 hours. So, while the end client data is not retained long term, the concept of data privacy is compromised. Until there is broad adoption of TLS 1.3 by the client applications, the SNI field which carries the name of the host to which the HTTPS connection is established will remain in the clear and leak DNS data. It is difficult to assess customer demand for DNS privacy or privacy in general, so the importance of the above points is also difficult to assess.

The network protocols behind DoH and DoT implementations also raise potential issues. Because DoH and DoT build upon TCP and TLS, they introduce a broader attack surface, making them more vulnerable to attack. Furthermore, since DoH and DoT are TCP based, they necessarily require cloud services scale to address the demands of handling a large volume of connections. Those who wish to deploy the service are left to decide the cost benefit ratio.

DoH and DoT services are gaining some traction as they are configurable in a few popular browsers and mobile phones. However, there is a natural conflict between the control the browser providers have and Service Providers who offer content-based services to safe guard web browsing for their customers. This is because with DoH or DoT implemented in the browser, the DNS query is hidden from the service provider and they are unable to provide content-based blocking. To ensure DNS queries are processed by their infrastructure, they may need to block access to 1.1.1.1 or other DoH or DoT based open resolvers.

Cloudflare's 1^4 application for iOS and Android is one such application and it delegates all DNS queries on the device to their service. This may address the DNS configuration fragmentation concerns where some traffic is redirected by the browser and bypasses the Service Provider configurations while other traffic is subject to DNS configuration-based restrictions. Since the role of DNS is largely unknown to the general public, it is hard to predict Cloudflare's 1^4 application adoption or consumer willingness to configure DNS privacy on Mozilla or Chrome. In the end, it may be only a small percentage of users who elect to use the service. Because of their deployment complexity, the broad adoption of DoH proxies in the home seems unlikely. That said, it may be an option within the enterprise where the complexities have a limited impact. In general, however, most large companies are less concerned with DNS and more concerned with monitoring and blocking HTTP traffic which they often do by requiring all HTTP access to the web be through a proxy.

It is difficult to assess customer demand for DNS privacy, or privacy in general, so the importance of the privacy concerns is also difficult to assess. DNS privacy is offered today by VPN services and to some degree the TOR network. For customers seeking complete anonymity, these solutions obviate the need for DNS privacy and therefore, the need for DoH or DoT for those consumers.