DNS Privacy in Practice and Preparation

Casey Deccio
Brigham Young University
Provo, UT
casey@byu.edu

Jacob Davis*
Sandia National Laboratories
Livermore, CA
jacdavi@sandia.gov

ABSTRACT

An increased demand for privacy in Internet communications has resulted in privacy-centric enhancements to the Domain Name System (DNS), including the use of Transport Layer Security (TLS) and Hypertext Transfer Protocol Secure (HTTPS) for DNS queries. In this paper, we seek to answer questions about their deployment, including their prevalence and their characteristics. Our work includes an analysis of DNS-over-TLS (DoT) and DNS-over-HTTPS (DoH) availability at open resolvers and authoritative DNS servers. We find that DoT and DoH services exist on just a fraction of open resolvers, but among them are the major vendors of public DNS services. We also analyze the state of TCP Fast Open (TFO), which is considered key to reducing the latency associated with TCP-based DNS queries, required by DoT and DoH. The uptake of TFO is extremely low, both on the server side and the client side, and it must be improved to avoid performance degradation with continued adoption of DNS Privacy enhancements.

CCS CONCEPTS

• **Networks → Network privacy and anonymity**; **Naming and addressing**.

KEYWORDS

DNS, Privacy, Measurement

ACM Reference Format:
Casey Deccio and Jacob Davis. 2019. DNS Privacy in Practice and Preparation. In *The 15th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '19), December 9–12, 2019, Orlando, FL, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3359989.3365435

1 INTRODUCTION

An increased demand for privacy in Internet communications has driven the creation of several privacy-centric extensions to the Domain Name System (DNS). Two important enhancements are the use of Transport Layer Security (TLS) and Hypertext Transfer Protocol Secure (HTTPS) for DNS queries. DNS over TLS (DoT) and DNS over HTTPS (DoH) have recently been standardized [25, 26] and have been deployed by several large public DNS services, including Cloudflare [23], Google [21], and Quad9 [36]. With all the momentum surrounding the deployment of these relatively new protocols, we seek to answer questions about their deployment in the wild.

All new Internet protocols require some amount of retrofit because of the many existing protocols already in wide deployment. Both DoT and DoH require that DNS queries be performed over the Transmission Control Protocol (TCP), rather than the User Datagram Protocol (UDP), which has been the primary transport of the DNS. Also, instead of using the standard port (53) that has been in use since the inception of the DNS, DoT uses a newly allocated port (853).

With over 35 years of traditional DNS entrenched in Internet infrastructure, change can be hard. Both DoT and DoH come with additional costs in terms of reduced performance and availability. For example, the use of TCP requires maintaining connection state at the client and the server, and TCP connection establishment can double the delay for a single DNS query-response transaction—not to mention the round-trip time (RTT) associated with the TLS handshake for both DoT and DoH. Additionally, proper usage of DoT and DoH requires support on both client and server platforms, and permissive infrastructure on the network path between. Many administrators are expecting DNS traffic over port 53 only—and possibly only over UDP (despite the fact that TCP support for the DNS has been specified for 30 years [5, 14])—and have configured their networks accordingly.

Several techniques have been introduced to reduce the cost associated with these DNS privacy extensions. TCP Fast Open (TFO) was introduced to eliminate the delay associated with connection establishment when communicating with a known host [9, 37]. TLS 1.3 introduces 0-RTT, with which data can be sent before the TLS handshake is complete [39]. This also reduces delay, albeit with weaker security assurances than are had otherwise. Also, the practices of query pipelining and out-of-order responses over a TCP connection were specified to make the best use of long-lived TCP connection with a server [14].

In this paper we study the deployment of DNS privacy—and the underlying protocols upon which its successful deployment depends. First, we quantify the existence of publicly accessible DoT and DoH resolvers in the wild and assess their deployment configurations, including the certificate authorities (CAs) and TLS versions in use. Next, we investigate the deployment of TFO, both for DoT servers and generally across DNS infrastructure worldwide.

Our measurements show that only a relatively small number of DNS servers support DNS privacy—0.15% of the open resolvers and a non-zero but negligible number of authoritative servers that we queried. However, several popular DNS providers are among those offering this support, including Cloudflare, Facebook, Google,

*This work was performed while the author was a student at Brigham Young University.

CoNEXT '19, December 9–12, 2019, Orlando, FL, USA

ACM ISBN 978-1-4503-6998-5/19/12…$15.00
https://doi.org/10.1145/3359989.3365435

and Quad9. We also find that fewer than half of the DoT resolvers, grouped by IP prefix, support TLS 1.3 for possible 0-RTT usage. Additionally, fewer than 10% of DoT resolvers, grouped by IP prefix, fully support TFO. General TFO support across DNS servers is also low, with less than 1% of open resolvers, Alexa 5k domains, and top-level domains (TLDs) supporting TFO in full.

2 BACKGROUND AND PREVIOUS WORK

The DNS [31, 32] is the system responsible for the translation of domain names to resources such as IP addresses. DNS name resolution involves interactions between several parties. A *stub* resolver asks questions of a *recursive* resolver, and the recursive resolver finds the answers by asking one or more *authoritative* DNS servers.

DoT and DoH are privacy extensions to the DNS, both of which prevent an eavesdropper from manipulating or introspecting a DNS query or its response, by encrypting client-server communications. DoT [26] involves establishing a TCP connection to a remote host, performing a TLS handshake [39] over that connection, and then issuing a DNS query as if it were over plain TCP [31, 32]. With DoH [25], the client issues a DNS query in an HTTPS request [18, 38] and receives the DNS response in the corresponding HTTPS response. DoT and DoH differ from the DNS Security Extensions (DNSSEC) [2–4], in that DNSSEC is used to authenticate the contents of a DNS response, *independent* of the server it came from, and it does not provide privacy.

In comparison with traditional DNS queries, which primarily use UDP, DoT and DoH queries incur additional delay because of TCP connection establishment and the TLS handshake. However, several protocols help reduce this overhead. With TFO, TCP connection establishment delay is effectively eliminated between a client and server after an initial TCP connection is established between the two. During this initial connection, the client includes the TFO option in the TCP header of the SYN packet. The server generates a cookie specific to the client and returns it in the TFO option of the SYN-ACK. The client stores the cookie, and the next time it initiates a TCP connection to that server, it includes in the SYN packet 1) the cookie it received previously in the TFO option and 2) data associated with the SYN. The server authenticates the TFO cookie against the IP address of the client and, if valid, acknowledges with the SYN-ACK any data sent in the SYN.

TLS 1.3 introduces 0-RTT, in which data is sent in the first part of the TLS handshake, encrypted with a pre-shared key that is known in advance to both client and server [39]. This inclusion effectively saves one RTT from a given TLS transaction. One drawback to 0-RTT is the absence of *perfect forward secrecy*, in which the encryption key used to encrypt a session is not revealed in the decrypted communications associated with session initiation [24].

Zhu, et al. [41], demonstrated that connection-oriented DNS, i.e., using TCP, can improve the security posture of the DNS. This is partly because of the exchange of sequence numbers in the TCP handshake, which makes it not as susceptible to spoofing as UDP communications are, and partly because it can support privacy mechanisms such as DoT and DoH. They conclude that performance of DNS over TCP (and, consequently DoT and DoH) depends on TFO.

	Total	16-bit Prefixes
DoT Open Resolvers	1,747	149
Support TLS 1.3	79 (4.5%)	57 (38%)
Support TLS 1.2	1,701 (97%)	145 (97%)
No Support for TLS 1, 1.1	80 (4.6%)	63 (42%)
Use self-signed cert	11 (0.63%)	21 (14%)
Use GoDaddy as CA	1,572 (90%)	28 (19%)
Use Let's Encrypt as CA	90 (5.2%)	72 (48%)
Returns TFO cookie	25 (1.4%)	11 (7.4%)
Acks SYN data	25 (1.4%)	11 (7.4%)
DoH Open Resolvers	9	5
TCP Responsive	557,969	-
Returns TFO cookie	10,851 (1.9%)	-
Acks SYN data	1,257 (0.23%)	-

Table 1: Summary of DoT, DoH, and TFO measurements against open DNS resolvers.

Other studies have been conducted to identify and characterize open DNS resolvers [29, 35] and to measure the availability of DNS authoritative servers [12, 13, 33, 34]. However, the first study that we know of related to the deployment of DNS privacy was released more recently [30]. Our results both confirm and complement their work. The authors used ZMap [17] to perform a scan of port 853 on all IPv4 addresses, and they tried both the `/resolve` and `/dns-query` URIs to test for potential DoH servers. Ultimately, the numbers of resolvers supporting DoT and DoH are on the same order as what we found in our study (section 3). Our study complements that of [30] by including TFO measurements and DNS privacy measurements between recursive and authoritative DNS servers.

3 DOT AND DOH MEASUREMENT

We begin our study by measuring the prevalence of DoT and DoH, first in open DNS resolvers and then in authoritative DNS servers. All of our measurements were performed from Brigham Young University's network (Autonomous System Number 6510). Our analysis of open resolvers (subsection 3.1 and subsection 3.2) only targeted IPv4 addresses, but the analysis of DNS authoritative servers (subsection 3.3) included both IPv4 and IPv6 measurements. We note that our methodology only considers DoT and DoH deployments that are offered on IP addresses that also publicly service traditional DNS queries (i.e., unencrypted, over UDP port 53), whether recursive or authoritative. It is possible, therefore, that public DoT or DoH services are offered on IP addresses that we did not identify or analyze in this study. Our findings are summarized in Table 1.

To generate the set of open resolvers, we issued a single DNS query to every IP address in the public IPv4 address space. In a more in-depth study on the subject, open resolvers might be identified and classified based on how they respond to recursive queries (i.e., with the recursion desired (RD) flag set) [35]. For example, considerations might include whether or not they returned the *expected* answer or response code or whether or not recursive queries yielded a query to the appropriate authoritative servers. However, our interest in this

paper is the *appearance* of available recursive service, specifically to filter out responses from DNS authoritative servers. Thus, we classified an IP address as an open resolver if it responded to our query with the recursion available (RA) flag set and a response code of either NOERROR or NXDOMAIN (query name does not exist). If the server queried was an authoritative server, then the response would typically not have the RA flag set, and the response code would be REFUSED. We identified a total of 1,197,794 open resolvers.

3.1 DoT in Open Resolvers

To measure DoT, we attempted to connect to each of the open resolvers over TCP port 853, establish a TLS connection, and issue a DNS query. We successfully connected and issued queries using DoT to 1,747 (0.15%) of the open resolvers. This number included IP addresses associated with some of the well-known public DNS services, including Cloudflare (1.1.1.1), Google (8.8.8.8), and Quad9 (9.9.9.9).

Rather than simply considering each IP address as an individual entity, we aggregate the resolvers supporting DoT by network—as identified by autonomous system number (ASN) and IP prefix. This serves as a heuristic to enumerate the organizational entities that have deployed DoT, not simply the IP addresses. We used Team Cymru's IP-to-ASN mapping service [11] to learn that the 1,747 DoT-supporting open resolvers originate from 87 unique autonomous systems (ASes). When grouped into prefixes of length 16, only 149 network prefixes are represented. Of the 149 prefixes with resolvers supporting DoT, 109 (73%) correspond to only a single resolver. Another 38 (26%) of the prefixes have between two and six open resolvers that support DoT. The last two prefixes (1.3% of the total) include 509 and 1,020 resolvers that support DoT. Performing a reverse DNS lookup (i.e., in the arpa domain) of each of the relevant IP addresses in these two prefixes revealed that all 1,529 constituent IP addresses are mapped to the domain cleanbrowsing.org; Clean Browsing is a company that provides a DNS-based content filtering service [7]. Outside of these two subnets, an additional 24 DoT resolvers addresses are associated with Clean Browsing, for a total of 1,553 DoT-supporting open resolvers made available by the company.

We analyzed the support for the various versions of TLS. Of the open resolvers that support DoT, only 79 (4.5%) support the newest TLS standard, TLS 1.3, which includes the 0-RTT feature and which was only recently standardized in 2018 [39]. This number represents 57 (38%) of the 16-bit network prefixes. TLS 1.2, which was standardized in 2008 and has no deprecation date at the time of this writing, is supported by 1,701 (97%) of the DoT resolvers, representing 145 (97%) of the network prefixes. This number includes all of the resolvers that support TLS 1.3. The 3% of resolvers that don't support TLS 1.2 are all associated with Clean Browsing [7]. Finally, support for TLS versions 1 and 1.1, which are planned to be deprecated in 2020 [6], has been dropped by 80 (4.6%) of DoT resolvers, representing 63 (42%) of the network prefixes.

From the X.509 certificates that were returned by DoT resolvers in the TLS handshake, a total of 22 unique issuers were identified. In 11 cases, the subject matched the issuer (either by Common Name, Organization, or both), indicating a self-signed certificate. These self-signed certificates account for 22 (1.3%) of the total DoT resolvers and represented 21 (14%) of the 16-bit network prefixes. The most-used CA is GoDaddy [20], which is used by 1,572 (90%) of the DoT resolvers, including all of the Clean Browsing resolvers. This accounted for 28 (19%) of the IP prefixes. The next most prevalent CA was Let's Encrypt [28], which is used by 90 (5.2%) of the DoT resolvers, accounting for 72 (48%) of the network prefixes.

3.2 DoH in Open Resolvers

We measured DoH by attempting to connect to each open resolver over TCP port 443, establish a TLS connection, and issue a DNS query over HTTPS. We used both the GET and POST methods with the template `https://x.x.x.x/dns-query{?dns}` (where `x.x.x.x` is the resolver's IP address) [25]. Only nine of the IP addresses we queried allowed queries in the specified manner. All nine supporting "standard" DoH were associated with one of two ASNs: 13335 (Quad9) and 19281 (Cloudflare).

We note that this study does not identify DNS services that employ alternate DNS privacy solutions in lieu of DoH. This includes Google's 8.8.8.8, which supported DNS over HTTPS using a non-standard protocol [21][1], as well as OpenDNS and Clean Browsing, which both support DNSCrypt [7, 10]. DNSCrypt is an alternate DNS privacy protocol that is supported by several DNS software implementations but lacks IETF standardization [16].

3.3 DoT in Authoritative Servers

While confidentiality for recursive-to-authoritative DNS queries was not one of the initial goals for the DNS community, it has since gained some attention [19]. We scanned two categories of authoritative DNS servers to discover what authoritative-side DoT support might exist: the servers authoritative for the Alexa Top 5k domains [1] and servers authoritative for the 1,530 TLDs (this also includes the root servers, which are authoritative for the arpa TLD). Both sets of servers are considered critical Internet infrastructure and are therefore relevant to our study.

The Alexa Top Domains were downloaded on June 17, 2019, and the TLDs were extracted from a root zone file downloaded on June 19, 2019 [27]. The names and IP addresses (IPv4 and IPv6) for each domain in the collective lists was determined through 1) a lookup of type NS (name server) for the domain and 2) a lookup of type A and AAAA (IPv4 and IPv6 address, respectively) for each name returned in the NS query response. In all, there were 3,592 unique IPv4 and 3,225 unique IPv6 addresses authoritative for TLDs. The authoritative servers for the Alexa Top 5k consisted of 6,708 unique IPv4 addresses and 3,506 unique IPv6 addresses.

In our scan of TLD authoritative servers, not a single IP address served DoT queries. Our scan of the Alexa domains showed only 12 IP addresses supporting DoT, which is 0.012% of the total. These IP addresses are collectively authoritative for 5 (0.10%) of the Alexa Top 5k domains, and all are associated with Facebook (i.e., Facebook, Instagram, and WhatsApp). This is consistent with Cloudflare's declaration that its recursive DNS service began using DoT for any queries to Facebook servers [8].

[1]Since our study was performed, Google's server has transitioned their service to support the DoH standard documented in RFC 8484 [25]

4 MEASURING TFO SUPPORT

Because TFO is an important feature to reduce the protocol overhead associated with TCP generally—and DoT and DoH specifically—our study includes a measure of TFO deployment in the wild. We begin by measuring TFO support at open resolvers—including public DNS services—after which we assess TFO support in authoritative DNS servers.

We begin with a discussion of the requirements for TFO at the operating system and software levels. Full support for TFO between a client and server requires that: 1) the kernels on both client and server support TFO; 2) TFO support is enabled on both client and server; and 3) both server- and client-side applications use the appropriate system calls in preparing the socket and sending the data, respectively.

For example, for properly functioning TFO between a Linux client and a Linux server, the client and server would need to be running kernels with version 3.6 or 3.7 (or higher), respectively. Additionally, TFO should be enabled in the kernel via the `net.ipv4.tcp_fastopen` kernel parameter, by setting it to a value with the least significant bit (LSB) set for clients and the second LSB set for servers. Finally, the client should use the `MSG_FASTOPEN` flag with `sendto()` in lieu of calling `connect()` and `send()`, and the server should have the `TCP_FASTOPEN` option set on the listening socket.

4.1 TFO in Open Resolvers

We begin our study of server-side support of TFO by studying the open resolvers identified in section 3. To detect TFO on the server side of a recursive resolver, we issued two back-to-back queries to that resolver over TCP, using the proper client-side TFO setup. This would allow our client to have at least one prior TCP communication with the resolver to cache the cookie that the server sends. We focus on two questions regarding TCP communications from the server: 1) whether or not the server returns a TFO option with a cookie in the response; and 2) whether or not the server acknowledges data (i.e., the DNS query) sent in the SYN packet, when the request uses that cookie.

Of the 1,197,793 open DNS resolvers responsive over UDP, only 557,969 (47%) were responsive to our TCP queries when testing for TFO support. Our experimental queries were issued within two days of each other, but because of potential churn in open resolver IP addresses [29]—even within that small window of time—we don't expect all the open resolvers responsive during our first set of queries to be responsive for our second set of queries. The TFO TCP option was returned by only 10,851 (1.9%) of DNS resolvers that responded to our queries over TCP. Of those, only 1,257 acknowledged data sent in a SYN, which represents only 12% of the DNS servers that return the TFO option and only 0.23% of all DNS servers responsive over TCP. Thus, just a fraction of a percent of open resolvers fully support TFO, and just over 10% seem to have only partial support.

Among the 9,594 open resolvers that returned the TFO option but did not acknowledge SYN data were 8.8.8.8 and 8.8.4.4, which are part of Google's public DNS service. Further inspection of the TFO behavior showed that the resolvers returned TFO cookies but that the cookie returned in the SYN-ACK of one TCP connection was rarely the same as the cookie returned in the SYN-ACK of the previous connection. Thus, the TFO cookie and associated SYN data that our client sent were rarely accepted.

	Domains	IP Addresses IPv4	IP Addresses IPv6
Alexa Domains	5,000	6,708	3,506
Responsive	4,966 (99%)	6,140 (92%)	3,418 (97%)
Sends TFO cookie	726 (15%)	606 (9.9%)	120 (3.5%)
Acks SYN data	52 (1.0%)	13 (0.21%)	5 (0.15%)
TLDs	1,530	3,592	3,225
Responsive	1,530 (100%)	3,566 (99%)	3,177 (99%)
Sends TFO cookie	47 (3.1%)	6 (0.17%)	5 (0.16%)
Acks SYN data	47 (3.1%)	4 (0.11%)	1 (0.03%)

Table 2: Summary of authoritative TFO measurements against authoritative DNS servers.

To further analyze this anomalous behavior, we issued 1,000 DNS queries to 8.8.8.8 over TCP, all with a TFO option. All queries were sent from a single client IP address. These queries yielded 80 distinct TCP cookies, distributed uniformly across SYN-ACKs. This behavior indicated load balancing of TCP connections from 8.8.8.8 to 80 back-end resolvers and that back-end selection was not based on any previous TCP communication.

We note that, for most clients, the impact of the anomalous TFO behavior on Google's resolvers is relatively small. The first reason is that the efficiencies offered by TFO are more necessary for DoT (and DoH) than for unencrypted DNS, and we did not observe this anomalous TFO behavior on port 853. Secondly, for clients using long-lived TCP connections to issue multiple queries on port 53, the impact of the anomalous TFO behavior exhibited by Google's DNS service is relatively small. This is because the TCP connection establishment is only performed once, so the overhead associated with connection setup is amortized over all the queries. Because 8.8.8.8 offers a recursive DNS service, a Google client will likely use the same IP address repeatedly for unrelated queries, so long-lived TCP connections are feasible.

We now consider just the 1,747 open resolvers that support DoT (see subsection 3.1). When queried over TCP port 853 with the TFO option, 38 (2.2%) of the resolvers were unresponsive. Another 1,684 resolvers (96%) had no server-side support for TFO, i.e., no TFO option was returned. That leaves only 25 (1.4%) of the DoT resolvers that returned the TFO option, and all 25 correctly acknowledged SYN data. These represented 11 (7.4%) of the 16-bit IP prefixes (see subsection 3.1). Seven of the addresses (28%) that returned the TFO option were associated with Cloudflare, and two (8%) were associated with Google.

4.2 TFO in Authoritative Servers

Next we analyzed server-side TFO support on the servers authoritative for the Alexa Top 5k domains and TLDs (see subsection 3.3), using the same methodology as we used in subsection 4.1. Our results are summarized in Table 2.

Of the servers authoritative for TLDs, 3,566 were responsive over IPv4 and 3,177 over IPv6, a 99% response rate for both IP versions. The IPv4 and IPv6 response rates for the Alexa servers were 92% and

97%, respectively, composed of 6,140 and 3,418 respective servers. These servers were authoritative for 4,966 domains (a small fraction of the Alexa Top 5k were offline, or the IP addresses of their servers had changed between the time the addresses were gathered and the data was collected).

A TFO cookie was returned by 726 (7.1%) of the responsive Alexa servers: 606 (10%) of the IPv4 servers and only 120 (3.5%) of the IPv6 servers. The number of Alexa domains with at least one server using TFO was 726, which is about 15% of the total responsive domains. However, only 18 of the Alexa servers, authoritative for 52 domains acknowledged data sent in a SYN; these corresponded to 13 IPv4 addresses and five IPv6 addresses. Thus, full TFO support only exists with 0.19% of servers and 1.0% of domains, according to our study.

The TFO adoption for the TLDs was significantly lower, with only 11 servers returning a TFO option: six IPv4 addresses and five IPv6 addresses. All IP addresses but one were associated with Google's registry [22]. Only five of the 11 servers acknowledged data sent in a SYN (see also subsection 4.1). These 11 corresponded to 47 TLDs, indicating that only 3.1% of TLDs have servers that fully support TFO.

4.3 TFO in Root Server Clients

Up to this point, we have analyzed *server-side* TFO behaviors in DNS servers. Next we investigate the presence of *client-side* TFO support in the wild. We examined queries destined for the DNS root servers [40] captured in the 2018 Day in the Life (DITL) data collection [15]. The collection consisted of 48 hours of queries received at various anycast locations of the root servers themselves—minus G-root, for which there was no data. Our analysis includes only queries made over IPv4.

We observed 3,769,471 unique clients—identified by IP address—that initiated a TCP connection to the collective root servers during the capture period. Of those nearly 4 million clients, only 89 (0.002%) included a TFO cookie option in their TCP SYN. The TFO option for 32 (36%) of the clients included a cookie value, as opposed to a blank value, in which they were requesting a cookie. However, not a single client included any data in their SYN! Thus far, we have been unable to reproduce this behavior in our lab environment to determine what specific system and/or configuration might cause it.

Based on our observation of queries to the root servers, we conclude that minimal TFO support by DNS resolvers is negligible and full TFO support in DNS resolvers is practically non-existent.

5 ARTIFACTS

The code used in this study and the resulting dataset can be found at: https://imaal.byu.edu/papers/2019_conext_dns_privacy/.

6 FUTURE WORK

This paper has consisted of active measurements to quantify the server-side *availability* of DoT and DoH, but there has been no attempt in this work to quantify the *usage* of DNS privacy behind recursive resolvers, i.e., how many recursive queries and/or users are using DoT or DoH. This would be a valuable addition to the results herein presented, as it would help us understand the *demand* for DNS privacy.

Our client-side TFO analysis is based on DITL 2018 data, in part because the DITL 2019 data set has not yet been made available. It would be valuable to compare the 2019 results with those from 2018 and earlier to measure uptake and identify deployment trends.

There is evidence, both on the server side (see subsection 4.1 and subsection 4.2) and on the client side (see subsection 4.3) of systems exhibiting partial TFO support. We would like to identify and further understand the root behaviors to see if they are likely to be a problem based on how prevalent they are now or will be.

Finally, as mentioned in section 3, our study only considered DoT and DoH in open DNS resolvers— identified by our scan of the IPv4 space—and known DNS authoritative servers. A more comprehensive study of DNS privacy would include a scan for DoT, DoH—standard and non-standard—and DNSCrypt. We plan to augment the current study with this data.

7 CONCLUSION

In this paper, we have evaluated the deployment of DNS privacy extensions, namely DoT and DoH. Our measurements show that a fraction of open resolvers have deployed DoT and DoH, and among those are popular DNS vendors. Relatively few of those DoT-capable resolvers have deployed server-side TFO, which is seen as necessary to bridge the performance gap incurred with DoT and DoH.

A broader study of TFO deployment also indicates that its adoption is low, in terms of both client- and server-side support. Lack of TFO support will lead to increased delays in the DNS as the deployment of connection-oriented DNS grows, including the use of DoT and DoH. Client- and server-side deployment of TFO must be improved for DNS privacy efforts to progress without a degradation of service.

We are hopeful that this paper will serve as a baseline for continued and improved deployment of DNS privacy practices, for a more secure DNS and Internet.

ACKNOWLEDGMENTS

We gratefully acknowledge DNS-OARC, who provided us access to the 2018 DITL data. We also thank the CoNEXT 2019 reviewers for their helpful comments.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

REFERENCES

[1] Amazon. 2019. Alexa Top Sites. https://aws.amazon.com/alexa-top-sites/
[2] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. 2005. RFC 4033: DNS Security Introduction and Requirements.
[3] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. 2005. RFC 4034: Resource Records for the DNS Security Extensions.
[4] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. 2005. RFC 4035: Protocol Modifications for the DNS Security Extensions.
[5] R. Braden. 1989. RFC 7766: Requirements for Internet Hosts – Application and Support.
[6] Peter Bright. 2018. Apple, Google, Microsoft, and Mozilla come together to end TLS 1.0. https://arstechnica.com/gadgets/2018/10/browser-vendors-unite-to-end-support-for-20-year-old-tls-1-0/
[7] Clean Browsing. 2019. Clean Browsing. https://cleanbrowsing.org/

[8] Manu Chantra. 2018. DNS over TLS: Encrypting DNS end-to-end. https://code.fb.com/security/dns-over-tls/

[9] Y. Cheng, J. Chu, S. Radhakrishnan, and A. Jain. 2014. RFC 7413: TCP Fast Open.

[10] Cisco. 2019. OpenDNS. https://www.opendns.com/

[11] Team Cymru. 2019. IP-to-ASN Mapping. http://www.team-cymru.com/IP-ASN-mapping.html

[12] Casey Deccio, Chao-Chih Chen, Prasant Mohapatra, Jeff Sedayao, and Krishna Kant. 2009. Quality of name resolution in the Domain Name System. In *Network Protocols, 2009. ICNP 2009. 17th IEEE International Conference on*. IEEE, Princeton, NJ, USA, 113 – 122.

[13] Casey Deccio, Jeff Sedayao, Krishna Kant, and Prasant Mohapatra. 2010. Measuring Availability in the Domain Name System. In *INFOCOM 2010 Proceedings IEEE*. IEEE, San Diego, CA, USA, 1 – 5.

[14] J. Dickinson, S. Dickinson, R. Bellis, A. Mankin, and D. Wessels. 2016. RFC 7766: DNS Transport over TCP - Implementation Requirements.

[15] DNS-OARC. 2018. 2018 DITL Data. https://www.dns-oarc.net/oarc/data/ditl/2018

[16] DNSCrypt. 2019. DNSCrypt. https://dnscrypt.info/

[17] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. 2013. ZMap: Fast Internet-wide Scanning and Its Security Applications. In *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)*. USENIX, Washington, D.C., 605–620.

[18] R. Fielding and J. Reschke. 2014. RFC 7230: Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing.

[19] Internet Engineering Task Force. 2019. DNS PRIVate Exchange (dprive). https://datatracker.ietf.org/wg/dprive/about/

[20] GoDaddy. 2019. GoDaddy: Domain Names, Websites, Hosting & Online Marketing Tools. https://www.godaddy.com/

[21] Google. 2019. Google Public DNS. https://developers.google.com/speed/public-dns/

[22] Google. 2019. Google Registry. https://www.registry.google/

[23] Olafur Gudmundsson. 2019. Introducing DNS Resolver, 1.1.1.1 (not a joke). https://blog.cloudflare.com/dns-resolver-1-1-1-1/

[24] C.G. Günther. 1989. An Identity-Based Key-Exchange Protocol. In *Advances in Cryptology – EUROCRYPT '89. Lecture Notes in Computer Science*, JJ. Quisquater and J. Vandewalle (Eds.), Vol. 434. Springer, Berlin, Heidelberg, 29 – 37.

[25] P. Hoffman and P. McManus. 2018. RFC 8484: DNS Queries over HTTPS (DoH).

[26] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. Hoffman. 2016. RFC 7858: Specification for DNS over Transport Layer Security (TLS).

[27] Internet Assigned Numbers Authority. 2019. Root Files. https://www.iana.org/domains/root/files

[28] Internet Security Research Group. 2019. Let's Encrypt - Free SSL/TLS Certificates. https://letsencrypt.org/

[29] Marc Kührer, Thomas Hupperich, Jonas Bushart, Christian Rossow, and Thorsten Holz. 2015. Going Wild: Large-Scale Classification of Open DNS Resolvers. In *Proceedings of the 2015 Internet Measurement Conference (IMC '15)*. ACM, New York, NY, USA, 355–368. https://doi.org/10.1145/2815675.2815683

[30] Chaoyi Lu, Baojun Liu, Zhou Li, Shuang Hao, Haixin Duan, Mingming Zhang, Chunying Leng, Ying Liu, Zaifeng Zhang, and Jianping Wu. 2019. An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come?. In *IMC '19: Proceedings of the Internet Measurement Conference 2019*. ACM, New York, NY, USA, 14.

[31] P. Mockapetris. 1987. RFC 1034: DOMAIN NAMES - CONCEPTS AND FACILITIES.

[32] P. Mockapetris. 1987. RFC 1035: DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION.

[33] Jeffrey Pang, James Hendricks, Aditya Akella, Roberto De Prisco, Bruce Maggs, and Srinivasan Seshan. 2004. Availability, Usage, and Deployment Characteristics of the Domain Name System. In *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement (IMC '04)*. ACM, New York, NY, USA, 1–14. https://doi.org/10.1145/1028788.1028790

[34] Vasileios Pappas, Zhiguo Xu, Songwu Lu, Daniel Massey, Andreas Terzis, and Lixia Zhang. 2004. Impact of Configuration Errors on DNS Robustness. *SIGCOMM Comput. Commun. Rev.* 34, 4 (Aug. 2004), 319–330. https://doi.org/10.1145/1030194.1015503

[35] Jeman Park, Aminollah Khormali, Manar Mohaisen, and Aziz Mohaisen. 2019. Where Are You Taking Me? Behavioral Analysis of Open DNS Resolvers. In *The 49th IEEE/IFIP International Conference on Dependable Systems and Networks*. IEEE, Portland, OR, USA.

[36] Quad9. 2019. Quad9. https://www.quad9.net/

[37] Sivasankar Radhakrishnan, Yuchung Cheng, Jerry Chu, Arvind Jain, and Barath Raghavan. 2011. TCP Fast Open. In *CoNEXT '11 Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*. ACM, New York, NY, USA, Article 21, 12 pages.

[38] E. Rescorla. 2000. RFC 2818: HTTP Over TLS.

[39] E. Rescorla. 2018. RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.3.

[40] Root Server Operators. 2019. Root DNS. http://root-servers.org/

[41] Liang Zhu, Zi Hu, John Heidemann, Duane Wessels, Allison Mankin, and Nikita Somaiya. 2015. Connection-Oriented DNS to Improve Privacy and Security. In *2015 IEEE Symposium on Security and Privacy*. IEEE, San Jose, CA, USA, 171 – 186.