Workgroup: dprive
Internet-Draft:
draft-dkgjsal-dprive-unilateral-probing-01
Published: 9 December 2021
Intended Status: Informational
Expires: 12 June 2022
Authors: D.K. Gillmor   J. Salazar
ACLU   A19

Unilateral Opportunistic Deployment of Encrypted Recursive-to-Authoritative DNS

Abstract

This draft sets out steps that DNS servers (recursive resolvers and authoritative servers) can take unilaterally (without any coordination with other peers) to defend DNS query privacy against a passive network monitor. The steps in this draft can be defeated by an active attacker, but should be simpler and less risky to deploy than more powerful defenses. The draft also introduces (but does not try to specify) the semantics of signalling that would permit defense against an active attacker.

The goal of this draft is to simplify and speed deployment of opportunistic encrypted transport in the recursive-to-authoritative hop of the DNS ecosystem. With wider easy deployment of the underlying transport on an opportunistic basis, we hope to facilitate the future specification of stronger cryptographic protections against more powerful attacks.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 ([RFC2119] and [RFC8174]) when, and only when, they appear in all capitals, as shown here.

1.2. Terminology

*"unilateral" means capable of opportunistic probing deployment without external coordination with any of the other parties

*Do53 refers to traditional cleartext DNS over port 53 ([RFC1035])

*DoQ refers to DNS-over-QUIC ([I-D.ietf-dprive-dnsoquic])

*DoT refers to DNS-over-TLS ([RFC7858])

*DoH refers to DNS-over-HTTPS ([RFC8484])

*Encrypted transports refers to DoQ, DoT, and DoH collectively

2. Priorities

This document aims to provide guidance to implementers who want to simply enable protection against passive network observers.

In particular, it focuses on mechanisms that can be adopted unilaterally by recursive resolvers and authoritative servers, without any explicit coordination with the other parties. This guidance provides opportunistic security (see [RFC7435]) --

encrypting things that would otherwise be in the clear, without interfering with or weakening stronger forms of security.

2.1. Minimizing Negative Impacts

It also aims to minimize potentially negative impacts caused by the probing of encrypted transports -- for the systems that adopt these guidelines, for the parties that they communicate with in the "second hump" of the DNS camel, and for uninvolved third parties. The negative impacts that we specifically try to minimize are:

* excessive bandwidth use

* excessive computational resources (CPU and memory in particular)

* amplification attacks (where DNS resolution infrastructure is wielded as part of a DoS attack)

2.2. Protocol Choices

While this document focuses specifically on strategies used by DNS servers, it does not go into detail on the specific protocols used, as those protocols --- in particular, DoT and DoQ --- are described in other documents.

This document does not pursue the use of DoH in this context, because a DoH client needs to know the path part of a DoH endpoint URL, and there are currently no mechanisms for a DNS resolver to predict the path on its own, in an opportunistic or unilateral fashion, without incurring in excessive use of resources. For instance, a recursive resolver in theory could guess the full path to a queried IP address by trying all the URL paths that the client has in records and see if one of those works, but even though it can be expected that this would work 99% of the time with fewer than 100 probes, this technique would likely incur in excessive resource consumption potentially leading to vulnerabilities and amplification attacks. The authors of this draft particularly welcome ideas and contributions from the community that lead to a suitable mechanism for unilaterally probing for DoH-capable authoritative servers, for later consideration in this or other drafts.

3. Guidance for Authoritative Servers

An authoritative server SHOULD implement and deploy DNS-over-TLS (DoT) on TCP port 853.

An authoritative server MAY implement and deploy DNS-over-QUIC (DoQ) on UDP port 853.

3.1. Pooled Authoritative Servers Behind a Single IP Address

Some authoritative DNS servers are structured as a pool of authoritatives standing behind a load-balancer that runs on a single IP address, forwarding queries to members of the pool.

In such a deployment, individual members of the pool typically get updated independently from each other.

A recursive resolver following the guidance in Section 4 that interacts with such a pool likely does not know that it is a pool. If some members of the pool are updated to follow this guidance while others are not, the recursive client might see the pool as a single authoritative server that sometimes offers and sometimes refuses encrypted transport.

To avoid incurring additional minor timeouts for such a recursive resolver, the pool operator SHOULD either:

* ensure that all members of the pool enable the same encrypted transport(s) simultaneously, or

* ensure that the load balancer maps client requests to pool members based on client IP addresses.

Similar concerns apply to authoritative servers responding from an anycast IP address. As long as the pool of servers is in a heterogenous state, any flapping route that switches a given client IP address to a different responder risks incurring an additional timeout. Frequent changes of routing for anycast listening IP addresses are also likely to cause problems for TLS, TCP, or QUIC connection state as well, so stable routes are important to ensure that the service remains available and responsive.

3.2. Authentication

For unilateral deployment, an authoritative server does not need to offer any particular form of authentication.

The simplest deployment would simply provide a self-issued, regularly-updated X.509 certificate. This mechanism is supported by many TLS and QUIC clients, and will be acceptable for any opportunistic connection.

Possible alternate forms of server authentication include:

* an X.509 Certificate issued by a widely-known certification authority associated with the common NS names used for this authoritative server

* DANE authentication (potentially including the TLS handshake)

3.3. Server Name Indication

An authoritative DNS server that wants to handle unilateral queries MAY rely on Server Name Indication (SNI) to select alternate server credentials. However, such a server MUST NOT serve resource records that differ based on SNI (or on the lack of SNI) provided by the client.

3.4. Resource Exhaustion

A well-behaved recursive resolver may keep an encrypted connection open to an authoritative server, to amortize the costs of connection setup for both parties.

However, some authoritative servers may have insufficient resources available to keep many connections open concurrently.

An authoritative server facing resource exhaustion SHOULD cleanly close open connections from recursive resolvers based on the authoritative's preferred prioritization.

A reasonable prioritization scheme would be to close connections in this order, until resources are back in control:

* connections with no outstanding queries, ordered by idle time (longest idle time gets closed first)

* connections with outstanding queries, ordered by age of outstanding query (oldest outstanding query gets closed first)

When resources are especially tight, the authoritative server may also decline to accept new connections over encrypted transport.

4. Guidance for recursive resolvers

This section outlines a probing policy suitable for unilateral adoption by any recursive resolver. Following this policy should not result in failed resolutions or significant delay.

4.1. Overall recursive resolver Settings

A recursive resolver implementing this draft must set system-wide values for some default parameters. These parameters may be set independently for each supported encrypted transport, though a simple implementation may keep the parameters constant across encrypted transports.

Name	Description	Suggested Default
persistence	How long should the recursive resolver remember successful encrypted transport connections?	3 days (259200 seconds)
damping	How long should the recursive resolver remember unsuccessful encrypted transport connections?	1 day (86400 seconds)
timeout	How long should the recursive resolver wait for an initiated encrypted connection to complete?	4 seconds

Table 1: recursive resolver system parameters per encrypted transport

This document uses the notation E-foo to refer to the foo parameter for the encrypted transport E.

For example DoT-persistence would indicate the length of time that the recursive resolver will remember that an authoritative server had a successful connection over DoT.

This document also assumes that the resolver maintains a list of outstanding cleartext queries destined for the authoritative resolver's IP address X. This list is referred to as Do53-queries[X]. This document does not attempt to describe the specific operation of sending and receiving cleartext DNS queries (Do53) for a recursive resolver. Instead it describes a "bolt-on" mechanism that extends the recursive resolver's operation on a few simple hooks into the recursive resolver's existing handling of Do53.

Implementers or deployers of DNS recursive resolvers that follow the strategies in this document are encouraged to report their preferred values of these parameters.

4.2. Recursive Resolver Requirements

To follow this guidance, a recursive resolver MUST implement at least one of either DoT or DoQ in its capacity as a client of authoritative nameservers.

A recursive resolver SHOULD implement the client side of DNS-over-TLS (DoT). A recursive resolver MAY implement the client side of DNS-over-QUIC (DoQ).

DoT queries from the recursive resolver MUST target TCP port 853, with an ALPN of dot. DoQ queries from the recursive resolver MUST target UDP port 853, with an ALPN of doq.

While this document focuses on the recursive-to-authoritative hop, a recursive resolver implementing these strategies SHOULD also accept queries from its clients over some encrypted transport (current common transports are DoH or DoT).

4.3. Authoritative Server Encrypted Transport Connection State

The recursive resolver SHOULD keep a record of the state for each authoritative server it contacts, indexed by the IP address of the authoritative server and the encrypted transports supported by the recursive resolver.

Each record should contain the following fields for each supported encrypted transport, each of which would initially be null:

Name	**Description**	**Retain Across Reset**
session	The associated state of any existing, established session (the structure of this value is dependent on the encrypted transport implementation). If session is not null, it may be in one of two states: pending or established	N
initiated	Timestamp of most recent connection attempt	Y
completed	Timestamp of most recent completed handshake	Y
status	Enumerated value of success or fail or timeout, associated with the completed handshake	Y
resumptions	A stack of resumption tickets (and associated parameters) that could be used to resume a prior successful connection	Y
queries	A queue of queries intended for this authoritative server, each of which has additional status early, unsent, or sent	N
last-activity	A timestamp of the most recent activity on the connection	N

Table 2: recursive resolver state per authoritative IP, per encrypted transport

Note that the session fields in aggregate constitute a pool of open connections to different servers.

With the exception of the session, queries, and last-activity fields, this cache information should be kept across restart of the server unless explicitly cleared by administrative action.

This document uses the notation E-foo[X] to indicate the value of field foo for encrypted transport E to IP address X.

For example, DoT-initiated[192.0.2.4] represents the timestamp when the most recent DoT connection packet was sent to IP address 192.0.2.4.

4.3.1. Separate State for Each of the Recursive Resolver's Own IP Addresses

Note that the recursive resolver should record this per-authoritative-IP state for each IP address it uses as it sends its queries. For example, if a recursive resolver can send a packet to authoritative servers from IP addresses 192.0.2.100 and 192.0.2.200, it should keep two distinct sets of per-authoritative-IP state, one for each source address it uses. Keeping these state tables distinct for each source address makes it possible for a pooled authoritative server behind a load balancer to do a partial rollout while minimizing accidental timeouts (see Section 3.1).

4.4. Maintaining Authoritative State by IP Address

In designing a probing strategy, the recursive resolver could record its knowledge about any given authoritative server with different strategies, including at least:

* the authoritative server's IP address,

* the authoritative server's name (the NS record used), or

* the zone that contains the record being looked up.

This draft encourages the first strategy, to minimize timeouts or accidental delays.

A timeout (accidental delay) is most likely to happen when the recursive client believes that the authoritative server offers encrypted transport, but the actual server reached declines encrypted transport (or worse, filters the incoming traffic and does not even respond with an ICMP port closed message).

By associating state with the IP address, the recursive client is most able to avoid reaching a heterogenous deployment.

For example, consider an authoritative server named ns0.example.com that is served by two installations (with two A records), one at 192.0.2.7 that follows this guidance, and one at 192.0.2.8 that is a legacy (cleartext port 53-only) deployment. A recursive client who associates state with the NS name and reaches .7 first will "learn" that ns0.example.com supports encrypted transport. A subsequent query over encrypted transport dispatched to .8 would fail, potentially delaying the response.

By associating the state with the authoritative IP address, the client can minimize the number of accidental delays introduced (see also Section 4.3.1 and Section 3.1).

4.5. Probing Policy

When a recursive resolver discovers the need for an authoritative lookup to an authoritative DNS server using IP address X, it retrieves the records associated with X from its cache.

The following sections presume that the time of the discovery of the need for lookup is time T0.

If any of the records discussed here are absent, they are treated as null.

The recursive resolver must know to decide whether to initially send a query over Do53, or over any of the supported encrypted transports (DoT or DoQ).

Note that a resolver might initiate this query via any or all of the known transports. When multiple queries are sent, the initial packets for each connection can be sent concurrently, similar to "Happy Eyeballs" ([RFC8305]). However, unlike Happy Eyeballs, when one transport succeeds, the other connections do not need to be terminated, but can instead be continued to establish whether the IP address X is capable of corresponding on the relevant transport.

4.5.1. Sending a query over Do53

For any of the supported encrypted transports E, if either of the following holds true, the resolver SHOULD NOT send a query to X over Do53:

- E-session[X] is in the established state, or

- E-status[X] is success, and (T - E-completed[X]) < persistence

Otherwise, if there is no outstanding session for any encrypted transport, and the last successful encrypted transport connection was long ago, the resolver sends a query to X over Do53. When it does so, it inserts a handle for the query in Do53-queries[X].

4.5.2. Receiving a response over Do53

When a successful response R is received in cleartext from authoritative server X for a query Q that was sent over Do53, the recursive resolver should:

```
*If Q is in Do53-queries[X]:
    -Return R to the requesting client
*Remove Q from Do53-queries[X]
*For each supported encrypted transport E:
    -If Q is in E-queries[X]:
        oRemove Q from E-queries[X]
```

But if R is unsuccessful (e.g. SERVFAIL):

```
*If Q is in Do53-queries[X]:
    -Remove Q from Do53-queries[X]
*if Q is not in any of *-queries[X]:
    -Return SERVFAIL to the client
```

4.5.3. Initiating a connection over encrypted transport

If any E-session[X] is in the established, the recursive resolver SHOULD NOT initiate a new connection to X over any other transport, but should instead send a query through the existing session (see Section 4.5.8). FIXME: What if there's a preferred transport, but the established session does not correspond to that preferred transport?

Otherwise, the timer should examine and possibly refresh its state for encrypted transport E to authoritative IP address X:

```
*if E-session[X] is in state pending, and
*T - E-initiated[X] > E-timeout, then
    -set E-session[X] to null and
    -set E-status[X] to timeout
```

When resources are available to attempt a new encrypted transport, the resolver should only initiate a new connection to X over E as long as one of the following holds true:

* E-status[X] is success, or

* E-status[X] is fail or timeout and (T - E-completed[X]) > damping, or

* E-status[X] is null and E-initiated[X] is null

When initiating a session to X over encrypted transport E, if E-resumptions[X] is not empty, one ticket should be popped off the stack and used to try to resume a previous session. Otherwise, the initial Client Hello handshake should not try to resume any session.

When initiating a connection, the resolver should take the following steps:

* set E-initiated[X] to T0

* store a handle for the new session (which should have pending state) in E-session[X]

* insert a handle for the query that prompted this connection in E-queries[X], with status unsent or early, as appropriate (see below).

4.5.3.1. Early Data

Modern encrypted transports like TLS 1.3 offer the chance to store "early data" from the client into the initial Client Hello in some contexts. A resolver that initiates a connection over a encrypted transport according to this guidance in a context where early data is possible SHOULD send the DNS query that prompted the connection in the early data, according to the sending guidance in Section 4.5.8.

If it does so, the status of Q in E-queries[X] should be set to early instead of unsent.

4.5.3.2. Resumption Tickets

When initiating a new connection (whether by resuming an old session or not), the recursive resolver SHOULD request a session resumption ticket from the authoritative server. If the authoritative server supplies a resumption ticket, the recursive resolver pushes it into the stack at E-resumptions[X].

4.5.3.3. Server Name Indication

For modern encrypted transports like TLS 1.3, most client implementations expect to send a Server Name Indication (SNI) in the Client Hello.

There are two complications with selecting or sending SNI in this unilateral probing:

* Some authoritative servers are known by more than one name; selecting a single name to use for a given connection may be difficult or impossible.

* In most configurations, the contents of the SNI field is exposed on the wire to a passive adversary. This potentially reveals additional information about which query is being made, based on the NS of the query itself.

To avoid additional leakage and complexity, a recursive resolver following this guidance SHOULD NOT send SNI to the authoritative when attempting encrypted transport.

If the recursive resolver needs to send SNI to the authoritative for some reason not found in this document, it is RECOMMENDED that it implements Encrypted Client Hello ([I-D.ietf-tls-esni] to reduce leakage.

4.5.3.4. Authoritative Server Authentication

A recursive resolver following this guidance MAY attempt to verify the server's identity by X.509 certificate or DANE. When doing so, the identity would presumably be based on the NS name used for a given query.

However, since this probing policy is unilateral and opportunistic, the client SHOULD NOT consider it a failure if an encrypted transport handshake that does not authenticate to any particular expected name.

To avoid the complexity of authoritative servers with multiple simultaneous names, or multiple names over time, this draft does not attempt to describe what name a recursive resolver should use when validating an authoritative server, or what the recursive resolver should do with an authentication success.

4.5.4. Establishing an encrypted transport connection

When an encrypted transport connection actually completes (e.g., the TLS handshake completes) at time T1, the resolver sets E-completed[X] to T1 and does the following:

If the handshake completed successfully:

- update E-session[X] so that it is in state established
- set E-status[X] to success
- for each query Q in E-queries[X]:
  - if early data was accepted and Q is early,
    - set the status of Q to sent
  - otherwise:
    - send Q through the session (see Section 4.5.8), and set the status of Q to sent
- set E-last-activity[X] to T1

4.5.5. Failing to establish an encrypted transport connection

If, at time T2 an encrypted transport handshake completes with a failure (e.g. a TLS alert),

- set E-session[X] to null
- set E-status[X] to fail
- set E-completed[X] to T2
- for each query Q in E-queries[X]:
  - if Q is not present in any other *-queries[X] or in Do53-queries[X], add Q to Do53-queries[X] and send query Q to X over Do53.

Note that this failure will trigger the recursive resolver to fall back to cleartext queries to the authoritative server at IP address X. It will retry encrypted transport to X once the damping timer has elapsed.

4.5.6. Encrypted transport failure

Once established, an encrypted transport might fail for a number of reasons (e.g., decryption failure, or improper protocol sequence).

If this happens:

- set E-session[X] to null
- set E-status[X] to fail

* for each query Q in E-queries[X]:

  - if Q is not present in any other *-queries[X] or in Do53-queries[X], add Q to Do53-queries[X] and send query Q to X over Do53. FIXME: should a resumption ticket be used here for this previously successful connection?

Note that this failure will trigger the recursive resolver to fall back to cleartext queries to the authoritative server at IP address X. It will retry encrypted transport to X once the damping timer has elapsed.

FIXME: are there specific forms of failure that we might handle differently? For example, What if a TCP timeout closes an idle DoT connection? What if a QUIC stream ends up timing out but other streams on the same QUIC connection are going through? Do the described scenarios cover the case when an encrypted transport's port is made unavailable/closed?

4.5.7. Handling clean shutdown of encrypted transport connection

At time T3, the recursive resolver may find that authoritative server X cleanly closes an existing outstanding connection (most likely due to resource exhaustion, see Section 3.4).

When this happens:

* set E-session[X] to null

* for each query Q in E-queries[X]:

  - if Q is not present in any other *-queries[X] or in Do53-queries[X], add Q to Do53-queries[X] and send query Q to X over Do53.

Note that this premature shutdown will trigger the recursive resolver to fall back to cleartext queries to the authoritative server at IP address X. Any subsequent query to X will retry the encrypted connection promptly.

4.5.8. Sending a query over encrypted transport

When sending a query to an authoritative server over encrypted transport at time T4, the recursive resolver should take a few reasonable steps to ensure privacy and efficiency.

When sending query Q, the recursive resolver should ensure that its state in E-queries[X] is set to sent.

The recursive resolver also sets E-last-activity[X] to T4.

In addition, the recursive resolver should consider the following guidance:

4.5.8.1. Avoid EDNS client subnet

To protect the privacy of the client, the recursive resolver SHOULD NOT send EDNS(0) Client Subnet information to the authoritative server ([RFC7871]) unless explicitly authorized to do so by the client.

4.5.8.2. Pad to standard policy

To increase the anonymity set for each query, the recursive resolver SHOULD use EDNS(0) padding according to policies described in [RFC8467].

4.5.8.3. Send queries in separate channels

When multiple queries are multiplexed on a single encrypted transport to a single authoritative server, the recursive resolver MUST offer distinct query ID fields for every outstanding query on a connection, and MUST be capable of receiving responses out of order.

To the extent that the encrypted transport can avoid head-of-line blocking (e.g. QUIC can use a separate stream per query) the recursive resolver SHOULD avoid head-of-line blocking.

4.5.9. Receiving a response over encrypted transport

When a response R for query Q arrives at the recursive resolver over encrypted transport E from authoritative server with IP address X at time T5, if Q is in E-queries[X], the recursive resolver takes the following steps:

- Remove R from E-queries[X]

- Set E-last-activity[X] to T5

- If R is successful:

  - send R to the requesting client

  - For each supported encrypted transport N other than E:

    - If Q is in N-queries[X]:

      - Remove Q from N-queries[X]

      - If Q is in Do53-queries[X]:

        - Remove Q from Do53-queries[X]

  * Otherwise (R is unsuccessful, e.g., SERVFAIL):

      - If Q is not in Do53-queries[X] or any other *-queries[X]:

        - Return SERVFAIL to the requesting client FIXME: What response should be sent to the clients in the case that extended DNS errors are used in an authoritative's response?

4.5.10. Resource Exhaustion

Over time, a recursive resolver following this policy may find that it is limited in resources, and may prefer to close some outstanding connections.

This could be done by checking available/consumed resources on a fixed schedule, by having a policy of only keeping a fixed number of connections open, by checking resources when activity occurs, or by some other cadence.

When existing connections should be closed, the recursive resolver should use a reasonable prioritization scheme to close outstanding connections.

One reasonable prioritization scheme would be:

* close outstanding established sessions based on E-last-activity[X] (oldest timestamp gets closed first)

Note that when resources are limited, a recursive resolver following this guidance may also choose not to initiate new connections for encrypted transport.

4.5.11. Maintaining connections

Some recursive resolvers looking to amortize connection costs, and to minimize latency MAY choose to synthesize queries to a particular resolver to keep a encrypted transport session active.

A recursive resolver that adopts this approach should try to align the synthesized queries with other optimizations. For example, a recursive resolver that "pre-fetches" a particular resource record to keep its cache "hot" can send that query over an established encrypted transport session.

5. Signalling for Stronger Defense

This draft *does not* contemplate the specification of any form of coordinated signalling between authoritative servers and recursive resolvers, as such measures would not be unilateral.

However, the draft highlights the needs of a signaling mechanism for stronger defense.

We highlight the following questions for other specifications to solve:

* What does the signal need to contain?
  - type of transport? (DoQ? DoT? DoH?)
  - error reporting if secure, authenticated connection fails (how to report? similar to TLSRPT?)
  - whether to hard-fail if encrypted communication isn't available
  - cryptographic authentication of authoritative server (e.g. pubkeys) vs. names vs. domain?
* How should the signal be presented?
  - SVCB RR or "surprising" DS RR
* How should the signal be scoped?
  - per-nameserver (by NS), per-nameserver (by IP address, via in-addr.arpa), or per-domain?

5.1. Combining Signals with Opportunistic Probing

FIXME: How do the signals get combined with the above opportunistic probing policy? Can we specify that without needing to specify the signalling mechanism itself?

6. IANA Considerations

IANA does not need to do anything for implementers to adopt the guidance found in this draft.

7. Privacy Considerations

7.1. Server Name Indication

A recursive resolver querying an authoritative server over DoT or DoQ that sends Server Name Indication (SNI) in the clear in the cryptographic handshake leaks information about the intended query to a passive network observer.

In particular, if two different zones refer to the same nameserver IP addresses via differently-named NS records, a passive network observer can distinguish queries to one zone from the queries to the other.

Omitting SNI entirely, or using ECH to hide the intended SNI, avoids this additional leakage. However, a series of queries that leak this information is still an improvement over the all-cleartext status quo at the time of this document.

8. Security Considerations

The guidance in this draft provides defense against passive network monitors for most queries. It does not defend against active attackers. It can also leak some queries and their responses due to "happy eyeballs" optimizations when the resolver's cache is cold.

Implementation of the guidance in this draft should increase deployment of opportunistic encrypted DNS transport between recursive resolvers and authoritative servers at little operational risk.

However, implementers should not rely on the guidance in this draft for robust defense against active attackers, but should treat it as a stepping stone en route to stronger defense.

In particular, a recursive resolver following this guidance can easily be forced by an active attacker to fall back to cleartext DNS queries. Or, an active attacker could position itself as a machine-in-the-middle, which the recursive resolver would not defend against or detect due to lack of server authentication. Defending against these attacks without risking additional unexpected protocol failures would require signalling and coordination that are out of scope for this draft.

This guidance is only one part of operating a privacy-preserving DNS ecosystem. A privacy-preserving recursive resolver should adopt other practices as well, such as QNAME minimization, local root zone, etc, to reduce the overall leakage of query information that could infringe on the client's privacy.

9. Acknowledgements

Many people contributed to the development of this draft beyond the authors, including Brian Dickson, Christian Huitema, Eric Nygren, Jim Reid, Kris Shrishak, Paul Hoffman, Ralf Weber, and the DPRIVE working group.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

10.2. Informative References

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[I-D.ietf-dprive-dnsoquic] Huitema, C., Dickinson, S., and A. Mankin, "DNS over Dedicated QUIC Connections", Work in Progress, Internet-Draft, draft-ietf-dprive-dnsoquic-07, 1 December 2021, <https://www.ietf.org/archive/id/draft-ietf-dprive-dnsoquic-07.txt>.

[I-D.ietf-tls-esni] Rescorla, E., Oku, K., Sullivan, N., and C. A. Wood, "TLS Encrypted Client Hello", Work in Progress, Internet-Draft, draft-ietf-tls-esni-13, 12 August 2021, <https://www.ietf.org/archive/id/draft-ietf-tls-esni-13.txt>.

[RFC7435] Dukhovni, V., "Opportunistic Security: Some Protection Most of the Time", RFC 7435, DOI 10.17487/RFC7435, December 2014, <https://www.rfc-editor.org/info/rfc7435>.

[RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <https://www.rfc-editor.org/info/rfc7858>.

[RFC7871] Contavalli, C., van der Gaast, W., Lawrence, D., and W. Kumari, "Client Subnet in DNS Queries", RFC 7871, DOI

10.17487/RFC7871, May 2016, <https://www.rfc-editor.org/info/rfc7871>.

[RFC8305] Schinazi, D. and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency", RFC 8305, DOI 10.17487/RFC8305, December 2017, <https://www.rfc-editor.org/info/rfc8305>.

[RFC8467] Mayrhofer, A., "Padding Policies for Extension Mechanisms for DNS (EDNS(0))", RFC 8467, DOI 10.17487/RFC8467, October 2018, <https://www.rfc-editor.org/info/rfc8467>.

[RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://www.rfc-editor.org/info/rfc8484>.

Appendix A. Document Considerations

[RFC Editor: please remove this section before publication]

This document is currently edited as markdown. Minor editorial changes can be suggested via merge requests at https://gitlab.com/dkg/dprive-unilateral-probing or by e-mail to the editor. Please direct all significant commentary to the public IETF DPRIVE mailing list: dprive@ietf.org

The authors' latest draft can be read online in html or pdf or text formats.

A.1. Document History

A.1.1. Substantive changes from -00 to -01

* Fallback to cleartext when encrypted transport fails.

* Reduce default timeout to 4s

* Clarify SNI guidance: OK for selecting server credentials, not OK for changing answers

* Document ALPN and port numbers

* Justify sorting recursive resolver state by authoritative IP address

Authors' Addresses

Daniel Kahn Gillmor
American Civil Liberties Union
125 Broad St.

New York, NY, 10004
United States of America

Email: dkg@fifthhorseman.net

Joey Salazar
ARTICLE 19
108-114 Golden Lane
London
EC1Y 0TL
United Kingdom

Email: joeygsal@gmail.com