DPRIVE
Internet-Draft
Intended status: Informational
Expires: June 16, 2020

J. Livingood
Comcast
A. Mayrhofer
nic.at GmbH
B. Overeinder
NLnet Labs
December 14, 2019

DNS Privacy Requirements for Exchanges between Recursive Resolvers and Authoritative Servers
draft-ietf-dprive-phase2-requirements-00

Abstract

This document provides requirements for adding confidentiality to DNS exchanges between recursive resolvers and authoritative servers.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on June 16, 2020.

Table of Contents

1. Introduction & Scope

The 2018 approved charter of the IETF DPRIVE Working Group [1] contains milestones related to confidentiality aspects of DNS transactions between the iterative resolver and authoritative name servers.

This is also reflected in the DPRIVE milestones [2], which (as of October 2019) contains two relevant milestones:

> Develop requirements for adding confidentiality to DNS exchanges between recursive resolvers and authoritative servers (unpublished document).
>
> Investigate potential solutions for adding confidentiality to DNS exchanges involving authoritative servers (Experimental).

This document intends to cover the first milestone for defining requirements for adding confidentiality to DNS exchanges between recursive resolvers and authoritative servers. This may in turn lead

to progress in investigating, developing and standardizing potential experimental methods of meeting those requirements.

The motivation for this work is to extend the confidentiality methods used between a user's stub resolver and a recursive resolver to the recursive queries sent by recursive resolvers in response to a DNS lookup (when a cache miss occurs and the server must perform recursion to obtain a response to the query). A recursive resolver will send queries to root servers, to Top Level Domain (TLD) servers, to authoritative second level domain servers and potentially to other authoritative DNS servers and each of these query/response transactions presents an opportunity to extend the confidentiality of user DNS queries.

2. Document Work Via GitHub

The authors are working on this document via GitHub at https://github.com/alex-nicat/ietf-dprive-phase2-requirements. Feedback via pull requests and issues are invited there.

3. Terminology

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] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document also makes use of DNS Terminology defined in [RFC8499]

4. Threat Model and Problem Statement

Currently, protocols such as DoT provide encryption between the user's stub resolver and a recursive resolver. This potentially provides (1) protection from observation of end user DNS queries and responses, (2) protection from on-the-wire modification DNS queries or responses (including potentially forcing a downgrade to an unencrypted communication). Of course, observation and modification are still possible when performed by the recursive resolver, which decrypts queries, serves a response from cache or performs recursion to obtain a response (or synthesizes a response), and then encrypts the response and sends it back to the user's stub resolver.

But observation and modification threats still exist when a recursive resolver must perform DNS recursion, from the root to TLD to authoritative servers. This document specifies requirements for filling those gaps.

5. Requirements

The requirements of different interested stakeholders are outlined below.

5.1. Mandatory Requirements

1. Each implementing party should be able to independently take incremental steps to meet requirements without the need for close coordination (e.g. loosely coupled)

2. Use a secure transport protocol between a recursive resolver and authoritative servers

3. Use a secure transport protocol between a recursive resolver and TLD servers

4. Use a secure transport protocol between a recursive resolver and the root servers

5. The secure transport MUST only be established when referential integrity can be verified, MUST NOT have circular dependencies, and MUST be easily analyzed for diagnostic purposes.

6. Use a secure transport protocol or other DNS privacy protections in a manner that enables operators to perform appropriate performance and security monitoring, conduct relevant research, etc.

7. The authoritative domain owner or their administrator MUST have the option to specify their secure transport preferences (e.g. what specific protocols are supported). This SHALL include a method to publish a list of secure transport protocols (e.g. DoH, DoT and other future protocols not yet developed). In addition this SHALL include whether a secure transport protocol MUST always be used (non-downgradable) or whether a secure transport protocol MAY be used on an opportunistic (not strict) basis.

8. The authoritative domain owner or their administrator MUST have the option to vary their preferences on an authoritative nameserver to nameserver basis, due to the fact that administration of a particular DNS zone may be delegated to multiple parties (such as several CDNs), each of which may have different technical capabilities.

9. The specification of secure transport preferences MUST be performed using the DNS and MUST NOT depend on non-DNS protocols.

10. For the secure transport, TLS 1.3 (or later versions) MUST be supported and downgrades from TLS 1.3 to prior versions MUST not occur.

5.2. Optional Requirements

1. QNAME minimisation SHOULD be implemented in all steps of recursion

2. DNSSEC validation SHOULD be performed

3. If an authoritative domain owner or their administrator indicates that (1) multiple secure transport protocols are available or that (2) a secure transport and insecure transport are available, then per the recommendations in [RFC8305] (aka Happy Eyeballs) a recursive server SHOULD initiate concurrent connections to available protocols. Consistent with Section 2 of [RFC8305] this would be: (1) Initiation of asynchronous DNS queries to determine what transport protocols are supported, (2) Sorting of resolved destination transport protocols, (3) Initiation of asynchronous connection attempts, and (4) Establishment of one connection, which cancels all other attempts.

6. Security Considerations

This entire document concerns the security of DNS traffic, so a specific section on security is superfluous.

7. IANA Considerations

This document has no actions for IANA.

8. Changelog

Version 00: Updated prior individual draft following IETF-106 feedback

9. APPENDIX: Perspectives and Use Cases

The DNS resolving process involves several entities. These entities have different interests/requirements, and hence it does make sense to examine the interests of those entities separately - though in many cases their interests are aligned. Four different entities can

be identified, and their interests are described in the following sections:

- Users
- Operators
- Implementors / Software Developers
- Researchers

9.1. The User Perspective and Use Cases

The privacy and confidentiality of Users (that is, users as in clients of recursive resolvers, which in turn forward/resolve the user's DNS requests by contacting authoritative servers) can be improved in several ways. We call this "minimisation of exposure", and there are currently three ways to reduce that exposure:

- Qname minimisation [RFC7816], reducing the amount of information which is absolutely necessary to resolve a query
- Aggressive NSEC/local auth cache [RFC8198], reducing the amount of outgoing queries in the first place
- Encryption, removing exposure of information while in transit

As recursors typically forwards queries received from the user to authoritative servers. This creates a transitive trust between the user and the recursor, as well as the authoritative server, since information created by the user is exposed to the authoritative server. However, the user has never a chance to identify which data was exposed to which authoritative party (via which path).

Also, Users would want to be informed about the status of the connections which were made on their behalf, which adds a fourth point

Encryption/privacy status signaling

TODO: Actual requirements - what do users "want"? Start below:

9.2. The Operator Perspective and Use Cases

Operators of authoritative services have to provide stable and fast DNS services, and interact with a wide range of clients, not all of them authoritative servers. The operator side actually consists of two sides:

- The "upstream" facing side of recursive resolvers
- The "downstream" side of authoritative servers

Those two sides are typically operated by different entities, but many entities operate "both sides". Even though that is discouraged (*TODO* source), the two sides might even be operated on the same nameserver.

- Maybe different technical perspectives for operators
  - Intelligence (sharing information)
  - SLD popularity for marketing
- Focus initially on Second Level Domains (SLDs) initially
  - Is there a difference for TLDs vs. SLDs from a "protocol" perspective?
- Monitoring and aggregated data analysis
- Signaling provisioning information
  - New record type for finding authoritative server key and authentication? Use SRV? (Being able to use different servers for serving up DNS-over-{TCP,UDP} vs DNS-over-TLS responses may be valuable.
  - Signal secure transport details (DNS-over-TLS, DNS-over-QUIC, EncryptedSNI, connectionless, etc.), perhaps in an extensible manner? Minimize RTTs and reduce need for trials.
  - Large provider use cases where the NS names are out of bailiwick for the zone (e.g. small number of distinct NS records serving 100k+ zones)
- EDNS client subnet (JL: Not sure ECS crosses the cost/benefit threshold to be included as a requirement and many CDNs that run auth servers will likely say ECS is quite operationally important)
- Decide between TLS and connectionless (such as COSE-based messages)
- Costs of TLS connection vs. connectionless
  - Technical solution, e.g. encryption of the DNS query, shouldn't enable an attack vector for DDoS or resource exhaustion. For

example, only if the client uses DNS-over-TLS, the upstream query to the authoritative will be over DNS-over-TLS also. If the client uses UDP, the resolver won't invest resources in DNS-over-TLS to prevent a potential resource exhaustion attack.

* Reuse connection state (if any) and examine resumption considerations

* Minimize server-side state (eg, with session tickets)

* Need empirical studies on capacity, traffic, attack vectors

* Evaluate impact on architecture and footprint expansion

* Analyze optimal persistent connection time/time-out

* Analyze optimal number of persistent connections recursive resolvers should maintain

* Consider operational concerns with respect to capabilities signaling

* Develop a profile that has operational advantages for operators

TODO: Actual requirements - what do operators "want"?

9.3. The Implementor / Software Vendor Perspective and Use Cases

Implementer requirements follows requirements from user and operator perspectives:

- Non-functional requirements, e.g. diversity of implementations

- Horizontal vs. vertical scaling, for example similar to http servers

- Use of DANE [RFC6698] for authentication: strict vs. opportunistic

- Incremental deployment

- Cache reuse vs. downgrade? Does the cache need to be partitioned? When can an in-cache answer retrieved via cleartext be served encrypted to a recursive query?

- (Use of TCP fast open) - but this might be a requirement for the actual encryption protocol

TODO: Actual requirements of implementors - essentially, they follow what Operators need?

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>.

[RFC8499] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <https://www.rfc-editor.org/info/rfc8499>.

10.2. Informative References

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <https://www.rfc-editor.org/info/rfc6698>.

[RFC7816] Bortzmeyer, S., "DNS Query Name Minimisation to Improve Privacy", RFC 7816, DOI 10.17487/RFC7816, March 2016, <https://www.rfc-editor.org/info/rfc7816>.

[RFC8198] Fujiwara, K., Kato, A., and W. Kumari, "Aggressive Use of DNSSEC-Validated Cache", RFC 8198, DOI 10.17487/RFC8198, July 2017, <https://www.rfc-editor.org/info/rfc8198>.

[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>.

10.3. URIs

[1] https://datatracker.ietf.org/doc/charter-ietf-dprive/

[2] https://datatracker.ietf.org/wg/dprive/about/

Acknowledgments

TODO

Authors' Addresses

Jason Livingood
Comcast

Email: Jason_Livingood@comcast.com

Alexander Mayrhofer
nic.at GmbH

Email: alex.mayrhofer.ietf@gmail.com

Benno Overeinder
NLnet Labs

Email: benno@NLnetLabs.nl