DPRIVE
Internet-Draft
Intended status: Informational
Expires: May 7, 2020

J. Livingood
Comcast
A. Mayrhofer
nic.at GmbH
B. Overeinder
NLnet Labs
November 04, 2019

DNS Privacy Requirements for Exchanges between Recursive Resolvers and Authoritative Servers
draft-lmo-dprive-phase2-requirements-01

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 May 7, 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Develop requirements for adding confidentiality to DNS exchanges between recursive resolvers and authoritative servers (unpublished document).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 Development

TEMPORARY SECTION - WILL BE REMOVED BEFORE PUBLISHING 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. The authors plan to continue developing the document in the lead up to IETF-106, after the draft cut-off date.

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, potentially privacy-protective protocols such as DoT provide encryption between the user's stub resolver and a recursive resolver. This provides (1) protection from observation of end user DNS queries and responses as well as (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. 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

5.1. 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:

5.2. 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"?

5.3. 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?

6. Preliminary Requirements

The requirements of different interested stakeholders are outlined below. The parenthetical risks and priority levels are intended only to spur discussion. But at a high level the requirements may be summarized as follows:

6.1. Mandatory Requirements (Proposed)

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) (low risk, high priority)
2. Implement DoT between a recursive resolver and single level domain authoritative servers (high risk, high priority)
3. Implement DNS privacy protections between a recursive resolver and TLD servers (low risk, low priority)
4. Implement DNS privacy protections between a recursive resolver and the root servers (low risk, low priority)
5. Implement DoT or other DNS privacy protections in a manner that enables operators to perform appropriate performance and security

monitoring, conduct relevant research, etc. (high risk, high priority)

6. Implement QNAME minimisation in all steps of recursion (medium risk, medium priority)

7. The legacy unencrypted DNS protocol (e.g. UDP/TCP port 53) MUST be supported in parallel to DoT (high risk, high priority)

8. Recursive resolvers SHOULD opportunistically upgrade recursive query transmissions to DoT when an authoritative server is detected to support DoT (high risk, high priority)

9. TLS 1.3 (or later versions) MUST be supported and downgrades from TLS 1.3 to prior versions MUST not occur.

6.2. Optional Requirements (Proposed)

1. Implement DoT between a recursive resolver and TLD servers (low risk, low priority)

2. Implement DoT between a recursive resolver and the root servers (low risk, low priority)

3. DNSSEC validation SHOULD be performed

4. Users SHOULD have a method for signaling their preferences for (1) exclusively using DNS privacy & encryption, (2) preferring DNS privacy & encryption but falling back to un-encrypted DNS as needed, (3) exclusively using un-encrypted DNS, or other preferences. (Possible reference to DNSSEC DO bit?)

5. Authoritative domain administrators SHOULD have a method for signaling their preferences for (1) exclusively using DNS privacy & encryption, (2) preferring DNS privacy & encryption but falling back to un-encrypted DNS as needed, (3) exclusively using un-encrypted DNS, or other preferences. (Possible reference to DNSSEC DO bit?)

6.3. Working Group Discussion Needed

- Provisioning impacts - operators and vendors say implementation must be zero-provisioning. What does that mean and how should that be articulated as a requirement?

- Signaling: Provide some method to signal not just binary support DoT / do not support to allow for certain QTYPES or whatever to use DoT while others may not (e.g. an auth server may want to say

in high load that some low risk or low priority queries fallback to unencrypted comms). Is this signaling or negotiation? Perhaps the requirement is ultimately about "Load Shedding" or "Load Management".

- Trust anchor/authority: Should this depend only on the DNS, such as DANE, or Certification Authorities? See discussion at https://github.com/alex-nicat/ietf-dprive-phase2-requirements/issues/13

- Rather than say DNS privacy methods should we specifically say no ECS (or not fine-grained ECS), and to do QNAME minimization?

- There is a new signaling draft at https://tools.ietf.org/html/draft-levine-dprive-signal-00 and a prior one at https://tools.ietf.org/html/draft-bortzmeyer-dprive-step-2-05 - are these informative for our requirements?

- Is signaling good and/or necessary.

6.4. Prioritization of Requirements

The preliminary requirements above each have varying levels of risk and so can be prioritized based on that risk. As a result, the highest risk area is the one that involves the greatest potential for surveillance and modification based on the details of the specific step of recursion. This suggests the highest risk and thus highest priority is between a recursive server and first level authoritative server. Lower risks are to TLDs and root servers, with correspondingly lower priority. Support for monitoring and compliance are also high risk since this is operationally critical, and thus should also be considered high priority.

6.5. Opportunistic Upgrade to Encryption

Opportunistically upgrading to use encryption may be the most viable path to deploy new DNS encryption protocols. This may enable deployment to occur incrementally and without tightly coupled coordination across a diverse global group of very different potential implementors.

EDITORIAL NOTE: This paragraph may be unnecessary and could be cut. The exact method by which a recursive resolver determines whether an authoritative server supports DoT has not been specified in this document. But it seems reasonable to imagine that a recursive server might be able to probe authoritative servers on TCP/853 using the DoT protocol and then build a cached list of servers that support DoT so that subsequent queries will upgrade to use DoT (and can fallback if

DoT connections subsequently fail). It seems also possible to imagine a method might exist for an authoritative domain to use a TBD resource record or other method to specify whether DoT is supported.

6.6. Detection of Availability

EDITORIAL NOTE: This section was just moved up. May need some better integration later on.

Recursive resolvers typically communicate with many authoritative nameservers. Not every authoritative nameserver will support DoT and not every recursive resolver will support every requirement. How should a recursive resolver determine whether DoT is supported for example? (There may be multiple ways, or none)

What scope/granularity should such an availability marker have?

- by zone ("all authoritative nameservers in the "example.net" zone support private queries from resolvers")
- by identified nameserver ("the nameserver "a.ns.example.net" supports private queries from resolvers")
- by IP address ("any nameservers that resolve to 192.0.2.13 support private queries from resolvers")

Note that if there is no signal for availability, recursors could still opportunistically try the DNS privacy mechanism, as this is employed by some stub resolvers when they contact their designated recursors.

Should a signal of availability also indicate a preference for privacy over availability? i.e., are there distinct ways to signal "DNS-privacy is available" separately from "Only contact this server via DNS-privacy if you understand this signal (though we may continue to support non-private DNS queries for clients that don't understand it)".

6.7. Resistance to Downgrade Attack

When a connection is opportunistically upgraded to DoT, if a fallback to unencrypted DNS can be possible via a downgrade attack by blocking or modifying TCP/853 communications. In such cases, it may be best to establish a mechanism whereby the authoritative domain can specify their preferred behaviour. This may range from only use DoT and do not fallback to unencrypted DNS, to opportunistically use DoT but fallback in failure, to do not use DoT. The email application layer protocols have similar methods for asserting how email from a

particular domain should be treated, so following some of the lessons learned there is likely a good idea. Compare HSTS [RFC6797]?

6.8. End-User Policy Propagation

EDITORIAL NOTE: This section was just moved up. May need some better integration later on.

Like any multi-party protocol (e.g. SMTP), the end user's preferences or policies might or might not be respected by later hops in the chain. But if we have a way to express those preferences, we offer cooperating resolvers at least an opportunity to respect them.

WG DISCUSS: Is it better to let auth domains assert whether fallback should be permitted or is that an end user preference or both? The email world might suggest the former while the DNSSEC world the latter. Or specify the standardization of the preferences and their communication and leave it to implementors to decide whether or how to treat those signals?

What sorts of preferences or policy might an end-user want to express? for example:

- do not identify my general location (e.g. don't send my subnet information (ECS) [RFC7871] data about me when talking to authoritative servers), accepting that reduced localization may result in less localized responses from authoritative Content Delivery Network (CDN) servers and thus slower access to content

- prefer DNS privacy over reduced latency (i.e., do not try to do speedups - try opportunistic privacy first and fall back to cleartext only if that fails)

- never do non-private authoritative queries on my behalf (for any external queries you need to do to resolve this request, require strict, well-authenticated DNS privacy)

How specifically are these preferences be expressed by the client? (e.g. new EDNS0 [RFC6891] options?) Should the recursor have a way to indicate whether:

- they are capable of honoring them?

- they intend to honor them?

- they _did_ honor them over the course of a specific lookup?

If a resolver merely forwards a request to another recursor, should it also propagate those preferences/policy? if so, how?

This seems similar to [I-D.ietf-uta-smtp-require-tls].

To implement end-user policies, support for signaling of DNS server capabilities is helpful, see for example [I-D.edmonds-dnsop-capabilities].

6.9. Performance and Efficiency

- Can authoritative server operators limit resource-exhaustion attacks against private DNS mechanisms from having an impact on traditional (non-private) authoritative DNS availability? (JL: seems easy to implement per host connection limits and implement other standard DDoS protections - again for a later BCP doc)

- What are best practices for authoritative server operators that can minimize latency and unavailability?

- What are best practices for recursors?

7. Security Considerations

At this point of the document, the authors have not yet discussed security considerations in detail, as the whole document tends to deal with user privacy, which can be considered part of security. :)

8. IANA Considerations

This document has no actions for IANA.

9. Changelog

Note to RFC editor: Remove this entire section before publication.

9.1. lmo-dprive-phase2-requirements-00

Initial version

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

[I-D.edmonds-dnsop-capabilities]
Edmonds, R., "Signaling DNS Capabilities", draft-edmonds-dnsop-capabilities-00 (work in progress), July 2017.

[I-D.ietf-uta-smtp-require-tls]
Fenton, J., "SMTP Require TLS Option", draft-ietf-uta-smtp-require-tls-09 (work in progress), August 2019.

[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>.

[RFC6797] Hodges, J., Jackson, C., and A. Barth, "HTTP Strict Transport Security (HSTS)", RFC 6797, DOI 10.17487/RFC6797, November 2012, <https://www.rfc-editor.org/info/rfc6797>.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <https://www.rfc-editor.org/info/rfc6891>.

[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>.

[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>.

[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>.

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