Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 23 May 2021

P. Hoffman
ICANN

19 November 2020

Recursive to Authoritative DNS with Opportunistic Encryption
draft-pp-recursive-authoritative-opportunistic-02

Abstract

This document describes a use case and a method for a DNS recursive resolver to use opportunistic encryption when communicating with authoritative servers. A motivating use case for this method is that more encryption on the Internet is better, and opportunistic encryption is better than no encryption at all. The method here is optional for both the recursive resolver and the authoritative server. Nothing in this method prevents use cases and methods that require authenticated encryption.

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 23 May 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 2
1.1. Use Case 2
1.2. Definitions 3
2. Method for Opportunistic Encryption 3
3. Transport Caches 4
4. Security Considerations 5
5. Acknowledgements 6
6. References 6
6.1. Normative References 6
6.2. Informative References 6
Author's Address 6

1. Introduction

A recursive resolver using traditional DNS over port 53 may wish instead to use encrypted communication with authoritative servers in order to prevent passive snooping of its DNS traffic. The recursive resolver can use opportunistic encryption (defined in [RFC7435] to achieve this goal.

This document describes a use case and a method for recursive resolvers to use opportunistic encryption. The use case is described in Section 1.1. The method uses DNS-over-TLS [RFC7858] with authoritative servers in an efficient manner.

1.1. Use Case

The use case in this document is recursive resolver operators who are happy to use TLS [RFC8446] encryption with authoritative servers if doing so doesn't significantly slow down getting answers, and authoritative server operators that are happy to use encryption with recursive resolvers if it doesn't cost much.

Both parties understand that using encryption costs something, but are willing to absorb the costs for the benefit of more Internet traffic being encrypted. The extra costs (compared to using traditional DNS on port 53) include:

* Extra round trips to establish TCP for every session

* Extra round trips for TLS establishment

* Greater CPU use for TLS establishment

* Greater CPU use for encryption after TLS establishment

* Greater memory use for holding TLS state

1.2. Definitions

The terms "recursive resolver" and "authoritative server" are defined in [RFC8499].

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.

2. Method for Opportunistic Encryption

[RFC7435] defines opportunistic encryption. The method described here uses DNS-over-TLS [RFC7858] between resolvers and authoritative servers.

In this document, the only difference between normal TLS session establishment and opportunistic encryption is that the the TLS client (the recursive resolver) optionally authenticates the server. In normal TLS, the client is required to authenticate the server and the TLS connection fails if the authentication is not successful.

[[The following might be changed. A possible use case to be added, depending on WG discussion, would be optional authentication in order for caches to promote parent-side NS records to a higher standing than they currently have in the caches.]]

In the opportunistic encryption described here, there is no need for the recursive resolver to authenticate the authoritative server because any certificate authentication failure does not cause the TLS session from being set up. Note that other authentication failures in TLS, such as incorrect algorithm choices or TLS record failures, MUST cause the TLS session from being set up. If it is easier programmatically for the recursive resolver to authenticate the authoritative server and then ignore the negative result for certificate authentication, than to just not authenticate, the recursive resolver MAY do that. The recursive resolver MAY note a

certificate authentication failure and act on it (such as by logging it or noting it in the cache), as long as the failure does not prevent the TLS session from completing.

Note that later protocols for encrypted resolver-to-authoritative communication might to require normal TLS authentication. Because of this, authoritative servers SHOULD use TLS certificates that can be used in authenticated TLS authentication, such as those issued by trusted third parties or self-issued certificates that can be authenticated with DANE [RFC6698] records. However, if an authoritative server does not care about the use cases for such future protocols, it MAY use self-issued certificates that cannot be authenticated.

3. Transport Caches

[[Because this section probably applies to use cases for authenticated encryption as well, it should probably be moved to a separate document.]]

A recursive resolver that attempted to use encrypted transport every time it connected to any authoritative server would inherently be slower than one that did not. Similarly, a recursive resolver that made an external lookup of what secure transports every authoritative server supports each time it connected would also inherently be slower than one that did not. Recursive resolver operators desire to give answers to stub resolvers as quickly as possible, so neither of these two strategies would make sense.

Instead, recursive resolvers following the method described in this document MUST keep a cache of relevant information about how DNS-over-TLS is supported by authoritative servers. This is called a "transport cache" in this document. The relevant information could include things such as support for encryption, authentication information such as public keys discovered from TLSA records, which records in a server's NS RRset support DNS-over-TLS, and so on. Future specifications might describe how to use other secure DNS transports for encryption, and thus would also have to describe ways that a resolver could discover whether an authoritative server supports them.

This document only DNS-over-TLS for encryption. Thus, a recursive resolver can test whether an authoritative server supports DNS-over-TLS by attempting to open a TLS session on port 853 or through discovery of appropriate TLSA records (to be described later and/or in different documents). The transport cache should store both positive and negative information about whether the authoritative server supports DNS-over-TLS and about authentication uses by particular name servers.

The recursive resolver MUST look in its transport cache before sending DNS queries to an authoritative server. If there is no entry for an authoritative server in its transport cache, the recursive resolver MUST use plain, unencrypted DNS over port 53. It MAY then probe for encrypted transports, and cache that information for later connections.

This document explicitly does not mandate the contents of the transport cache. Different recursive resolver implementers are likely to have different cache structures based on many factors, such as research results, active measurements, secure protocols supported, and customer feedback, There will likely be different strategies for the time-to-live for parts of the transport cache, such as how often to refresh the data in the cache, how often to refresh negative data, whether to prioritize refreshing certain zones or types of zones, and so on.

This document also explicitly doesn't mandate how the strategy for filling transport caches. Some strategies might include one or more of "try to send a refresh query over DoT", "use external data", "trust a third-party service for filling the transport cache", and so on.

There are no interoperability issues with different implementors making different choices for the contents and fill strategies of their transport caches, and having many different options available will likely cause the cache designs to get better over time.

4. Security Considerations

The method described in this document explicitly allows a stub to perform DNS communications over traditional unencrypted, unauthenticated DNS on port 53.

The method described in this document explicitly allows a stub to choose to allow unauthenticated TLS. In this case, the resulting communication will be susceptible to obvious and well-understood attacks from an attacker in the path of the communications.

5. Acknowledgements

Puneet Sood contributed many ideas to early drafts of this document.

6. References

6.1. 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>.

[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>.

[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>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[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>.

6.2. 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>.

Author's Address

Paul Hoffman
ICANN

Email: paul.hoffman@icann.org