Network Working Group P. Hoffman
Internet-Draft ICANN
Intended status: Standards Track 25 November 2020
Expires: 29 May 2021

Recursive to Authoritative DNS with Opportunistic Encryption
draft-pp-recursive-authoritative-opportunistic-03

Abstract

This document describes a use case and a method for a DNS recursive resolver to use opportunistic encryption (that is, encryption with optional authentication) when communicating with authoritative servers. The 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.

IMPORTANT NOTE: This version of the document describes discovery whether an authoritative server supports encryption using port-checking. This restriction is based on the request of the DPRIVE WG during its meeting at IETF 109. It is quite likely that the final protocol will include a better set of methods for such discovery.

Table of Contents

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] (DoT) with authoritative servers in an efficient manner; it is called "ADoT", as described in [I-D.ietf-dnsop-rfc8499bis]. ((A later version of this document might also describe the use of DNS-over-QUIC [I-D.ietf-dprive-dnsoquic] (DoQ).))

Because opportunistic encryption means encryption with optional authentication, a resolver using the mechanism described here will likely achieve authenticated encryption with some authoritative

servers. The resolver can then take advantage of DNS features that require authentication of authoritative servers; such features will be described elsewhere.

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. Summary of Protocol

This protocol has four main parts. This summary gives an overview of how the work together.

* A resolver that uses this protocol has a cache that it uses to know whether to attempt using ADoT with a particular authoritative server, as described in Section 4.

* A resolver fills its transport cache by discovering whether any authoritative server of interest uses encrypted DNS, as described in Section 3.

* If there is no entry for that server in the cache, or the cache says that the authoritative server doesn't support encrypted transport, the resolver uses classic DNS; otherwise, the resolver attempts to connect to the authoritative server with ADoT, as described in Section 2.

* If the TLS session is authenticated and the resolver has use for this authentication, the resolver can mark responses it gets as authenticated, as described in Section 5.

* If the TLS session is not authenticated, the resolver treats the answers it receives as if they were received over classic DNS.

1.3. Definitions

The terms "recursive resolver", "authoritative server", "ADoT", and "classic DNS" are defined in [I-D.ietf-dnsop-rfc8499bis].

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. 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. See Section 5 for a fuller description of the use of authentication.

2.1. Resolvers

A resolver following this protocol uses its transport cache (described in Section 4 to decide whether to use classic DNS or this protocol to contact authoritative servers. If the transport cache indicates that the authoritative server is known to support encrypted DNS, the resolver attempts to connect to in on port 853.

The resolver is configured with a set of timeouts that it uses when it is setting up ADoT. This document does not yet suggest values for those timeouts; they are marked here with ((timeout_)).

The resolver MUST fall back to using classic DNS with a server if any of the following happens when using ADoT:

* The resolver receives a TCP RST response

* The resolver does not receive a reply to the TCP SYN message within timeout ((timeout_syn))

* The resolver does not receive a reply to its first TLS message within timeout ((timeout_tls_start))

* The TLS handshake gets a definitive failure

* The TLS session is set up, but the resolver does not receive a response to its first DNS query in the TLS session within timeout ((timeout_dns_answ))

In any of those cases, the resolver needs to update its transport cache to indicate that the server is not currently available. The time-to-live value for that entry could be ((some value goes here)).

Failures in TLS other than for authentication, such as incorrect algorithm choices or TLS record failures, MUST cause the TLS session from being set up.

A resolver can keep a TLS session to a particular server open if it expects to send additional queries to that server in a short period of time ((timeout_additional)). If the server closes the TLS session, the resolver can re-establish a TLS session of the version of TLS in use allows for session resumption.

2.2. Authoritative Servers

An authoritative server following this protocol establishes an ADoT service at port 853 for each IP address on which it offers service for classic DNS on port 53. The server's TLS certificate MUST have a subject identifier that matches the IP addresses or the domain names it is known by.

A server MAY close the TLS connection at any time. For example, it can close the TLS session if it has not received a DNS query by ((timeout_dns_query)). It can also close the TLS session after it sends a DNS response; however, it might also want to keep the TLS session open waiting for another DNS query from the resolver.

3. Discovering Whether an Authoritative Server Uses Encryption

A recursive resolver can discover whether an authoritative server supports DNS-over-TLS by attempting to open a TLS session on port 853. If the server completes the TLS handshake, the resolver can be fairly confident that the server supports ADoT.

((Note that there are likely better ways to do discovery. The DPRIVE WG requested that this version of this draft only specify port-probing. Future drafts will describe other methods, and how to use multiple methods at the same time for discovery.))

The following are indications of failure for the ability to use ADoT with the server:

* The resolver receives a TCP RST response

* The resolver does not receive a reply to the TCP SYN message within timeout ((timeout_syn))

* The resolver does not receive a reply to its first TLS message within timeout ((timeout_tls_start))

* The TLS handshake gets a definitive failure

((Clearly, further research is needed to determine good timeouts to use here.))

4. The Transport Cache

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, expected round-trip times, authentication mechanisms, and so on. The transport cache is likely to store both positive and negative information about a server's ability to support encrypted DNS.

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 classic 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 ADoT", "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.

5. Authentication

In the opportunistic encryption described here, there is no requirement for the recursive resolver to authenticate the authoritative server because any certificate authentication failure does not 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 being set up.

This document does not describe what to do with successful authentication of a ADoT TLS session. Some suggestions have been floated in the DPRIVE WG, but none have been written into drafts. ((Change this paragraph when that sentence becomes outdated.)) When there are reasons to note authentication of the server, resolvers following this protocol MAY use that authenticated data.

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.

6. 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.

7. Acknowledgements

Puneet Sood and Peter van Dijk contributed many ideas to early drafts of this document.

8. References

8.1. Normative References

[I-D.ietf-dnsop-rfc8499bis]
Hoffman, P. and K. Fujiwara, "DNS Terminology", Work in Progress, Internet-Draft, draft-ietf-dnsop-rfc8499bis-01, 20 November 2020, <http://www.ietf.org/internet-drafts/draft-ietf-dnsop-rfc8499bis-01.txt>.

[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>.

8.2. Informative References

[I-D.ietf-dprive-dnsoquic]
Huitema, C., Mankin, A., and S. Dickinson, "Specification of DNS over Dedicated QUIC Connections", Work in Progress, Internet-Draft, draft-ietf-dprive-dnsoquic-01, 20 October 2020, <http://www.ietf.org/internet-drafts/draft-ietf-dprive-dnsoquic-01.txt>.

[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