Network Working Group
Internet-Draft
Intended status: Experimental
Expires: 26 August 2021

P. Hoffman
ICANN
P. van Dijk
PowerDNS

22 February 2021

Recursive to Authoritative DNS with Encryption
draft-ietf-dprive-opportunistic-adotq-01

Abstract

This document describes a use case and a method for a DNS recursive resolver to use either opportunistic encryption (that is, encryption with optional authentication) or fully-authenticated encryption when communicating with authoritative servers. The motivating use case for this method is that more encryption on the Internet is better, some resolver operators will only want to offer fully-authenticated encryption when encryption is available, and some resolver operators believe that opportunistic encryption is better than no encryption at all. The method described here is optional for both the recursive resolver and the authoritative server. This method supports both fully-authenticate encryption and opportunistic encryption using the same mechanism for discovery of encryption support and discovery of authenticated public keys for the server.

IMPORTANT NOTE: This version of the document is completely different than the earlier version. It now covers both opportunistic and fully-authenticated encryption. It is in a very rough state, and there are many holes in the description.

Copyright Notice

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 limit snooping of its DNS traffic by passive or on-path attackers. The recursive resolver can use opportunistic encryption (defined in [RFC7435] or fully-authenticated encryption to achieve this goal.

This document describes two use cases for recursive resolvers: opportunistic encryption (described in Section 1.1) and fully-authenticated encryption described in Section 1.2). The encryption 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]. The document also describes a single discovery method that both shows if an authoritative server supports ADoT, and also supports fully-authenticated encryption by authenticating the allowed public keys for the server.

((A later version of this document will probably also describe the use of DNS-over-QUIC [I-D.ietf-dprive-dnsoquic] (DoQ).))

Resolvers and authoritative servers 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 (but not necessarily for every query)

* 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

1.1. Use Case for Opportunistic Encryption

The use case in this document for opportunistic encryption is recursive resolver operators who are happy to use 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. In this use case, resolvers do not want to return an error for requests that were sent over an encrypted channel if they would have been able to give a correct answer using unencrypted transport.

1.2. Use Case for Fully-Authenticated Encryption

The use case in this document for fully-authenticated encryption is recursive resolver operators who want to prevent on-path attackers from impersonating authoritative servers for zones for which the resolver is sending queries. The result of using fully-authenticated encryption, when possible, is that resolvers will know that either the authoritative server they are communicating with is in fact the one they expect, or they will know that the responses they get will be as untrusted as if the response came over unencrypted DNS.

1.3. Summary of Protocol

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

* The resolver discovers whether any authoritative server of interest supports encrypted DNS with ADoT by querying for the TLSA records [RFC6698].

* Information from the TLSA queries is stored in the resolver's normal cache of resource records; this protocol does not require a new cache for resolvers.

* When a resolver of either type is about to resolve a name in a zone, it uses the TLSA records in its cache to determine which authoritative servers support ADoT.

* A resolver uses any authoritative server with a positive TLSA record to perform unauthenticated ADoT.

* If the resolver is using fully-authenticated encryption, it tries each indicated authoritative server until it sets up an authenticated TLS connection. If there was at least one positive TLSA record, but none of the servers contacted could be authenticated during TLS setup, the resolver responds to the original query with a SERVFAIL response code.

* There should be a way for resolver operators to tell authoritative server operators when failures occur.

1.4. 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. Discovering Whether an Authoritative Server Uses ADoT

A recursive resolver discovers whether an authoritative server supports ADoT by looking for a cached TLSA record of the name of the authenticated server for TCP port 853 with a positive answer. A cached TLSA record with a negative answer indicates that the authoritative server does not support ADoT. Positive and negative responses for TLSA records are cached the same way as for all other DNS resource records.

For example, if the nameservers for example.com are "ns1.example.net" and "ns2.example.net", before connecting to either nameserver, the resolver checks its cache for "_853._tcp.ns1.example.net" and "_853._tcp.ns2.example.net." If either or both of these records have positive answers, those authoritative servers indicate that they support ADoT.

If the cache has no positive or negative answers for any TLSA record for any of a zone's authoritative servers, the resolver MUST send queries for the TLSA records for at least some of the zone's authoritative servers, and SHOULD send send queries for the TLSA records for all of the zone's authoritative servers.

Discovery using TLSA records differs between resolvers using opportunistic encryption and those using fully-authenticated encryption:

* If the resolver is using opportunistic encryption, the TLSA records do not need to be DNSSEC signed.

* If the resolver is using fully-authenticated encryption, the TLSA records must be signed by DNSSEC, and the signatures must verify. Thus, queries for these TLSA records MUST include the DO bit. These resolvers also use the servers' public keys from the TLSA records for authentication of the TLS session.

For either type of resolver, if there is DNSSEC information for the TLSA record set, it MUST be validated according to normal DNSSEC semantics [RFC4035]. If verification fails, the TLSA records MUST NOT be used as either a positive or negative indicator for ADoT service. ((Some resolvers may have a way to ignore the validation status on some records based on configured context; if so, maybe we can add text here allowing opportunistic resolvers to do so.))

TLSA records are defined in RFC 6698, although this protocol does not use the protocol defined in RFC 6698. Specifically, the protocol defined here does not require that the TLSA records be signed with DNSSEC in all cases. ((To do: fill this out more to make the differences between this document an RFC 6698 explicit.))

3. Resolving with ADoT

A resolver following this protocol MUST use TLSA records in its cache to decide whether to use classic DNS or ADoT to contact authoritative servers for a zone. If any of the TLSA records in the cache for the authoritative servers for a zone are positive responses, the resolver uses any of those servers for ADoT. A resolver MUST NOT attempt ADoT for a server that has a negative response in its cache for the associated TLSA record.

If all of the TLSA records in the cache for the authoritative servers for a zone are negative responses, the resolver MUST use classic (unencrypted) DNS instead of ADoT.

If there are any TLSA records in the cache for the authoritative servers for a zone with a postive response, the resolver MUST try each indicated authoritative server until it successfully sets up a TLS connection. Reasons for TLS failures are listed in Section 3.1.

If a TLS session is set up, the resolver uses that authoritative server for whatever query about the zone it was going to send. If a resolver is using opportunistic encryption, and it cannot set up a TLS session with any of the authoritative servers, it MUST attempt to perform the resolution over classic (unencrypted) DNS as it would have without ADoT. If a resolver is using fully-authenticated encryption, and it cannot set up a TLS session with any of the authoritative servers, it MUST respond to the original query with a SERVFAIL response code.

A resolver SHOULD 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. If the server closes the TLS session, the resolver can re-establish a TLS session if the version of TLS in use allows for session resumption.

3.1. Resolver Session Failures

The resolver is configured with a set of timeouts that it uses when it is setting up ADoT. This document does has suggested values for those timeouts; they are marked here with ((timeout_)). Resolver software might use these suggested values for defaults, or might choose their own default values.

((The proposed default values here are based on research that I have done but not published. The research is expected to be published before IETF 110.))

The following are the reasons that TLS might not be set up in 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 suggested default is 1.3 seconds

* The resolver does not receive a reply to its first TLS message within timeout "timeout_tls_start"; the suggested default (which includes the TCP startup time) is 2.4 seconds

* The TLS handshake gets a definitive failure the suggested default is 5 seconds (which includes the TCP and TLS startup times)

* The TLS session fails for reasons other than for authentication, such as incorrect algorithm choices or TLS record failures

* If the resolver is using fully-authenticated encryption, the the TLS session cannot be authenticated against the public key indicated in a TLSA record for the authoritative server

4. Serving with ADoT

An operator of authoritative service for a zone that is following this protocol MAY support an ADoT service for any IP address on which it offers service for classic DNS on port 53. It is acceptable for such an operator to only offer ADoT on some of the named authoritative servers, such as when the operator is determining how far to roll out ADoT service.

A server MAY close a TLS connection at any time. For example, it can close the TLS session if it has not received a DNS query in a defined length of time; the suggested default for this timeout, called "timeout_dns_query", is 20 seconds. The server MAY 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.

5. Resolvers Reporting Errors to Authoritative Servers

((TBD))

6. IANA Considerations

((Update registration for TCP/853 to also include ADoT }}

7. Security Considerations

The method described in this document explicitly allows a resolver to perform DNS communications over traditional unencrypted, unauthenticated DNS on port 53, if it cannot find an authoritative server that advertises that it supports ADoT. The method described in this document explicitly allows a resolver using opportunistic ADoT to choose to allow unauthenticated TLS. In either of these cases, the resulting communication will be susceptible to obvious and well-understood attacks from an attacker in the path of the communications.

8. Acknowledgements

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

Comments on early versions of this document led the authors to change it from being just about opportunistic resolution to both opportunistic and fully-authenticated resolution.

9. References

9.1. 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>.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <https://www.rfc-editor.org/info/rfc4035>.

[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>.

[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>.

9.2. 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>.

Authors' Addresses

Paul Hoffman
ICANN

Email: paul.hoffman@icann.org

Peter van Dijk
PowerDNS

Email: peter.van.dijk@powerdns.com