Workgroup: Network Working Group
Internet-Draft:
draft-ietf-dprive-unauth-to-authoritative-04
Published: 28 September 2021
Intended Status: Experimental
Expires: 1 April 2022
Authors: P. Hoffman   P. van Dijk
ICANN   PowerDNS

Recursive to Authoritative DNS with Unauthenticated Encryption

Abstract

This document describes a use case and a method for a DNS recursive resolver to use unauthenticated encryption when communicating with authoritative servers. The motivating use case for this method is that more encryption on the Internet is better, and some resolver operators believe that unauthenticated encryption is better than no encryption at all. The method described here is optional for both the recursive resolver and the authoritative server.

Status of This Memo

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 unauthenticated encryption (defined in [OPPORTUN]) to achieve this goal.

This document describes the use case for unauthenticated encryption in recursive resolvers in Section 1.1. The encryption method with authoritative servers can be DNS-over-TLS [DNS-OVER-TLS] (DoT), DNS-over-HTTPS [DNS-OVER-HTTPS] (DoH), and/or DNS-over-QUIC [DNS-OVER-QUIC] (DoQ).

The document also describes a discovery method that shows if an authoritative server supports encryption in Section 2.

See [FULL-AUTH] for a description of the use case and a proposed mechanism for fully-authenticated encryption.

NOTE: The draft uses the SVCB record as a discovery mechanism for encryption by a particular authoritative server. Any record type that can show multiple types of encryption (currently DoT, DoH, and DoQ) can be used for discovery. Thus, this record type might change in the future, depending on the discussion in the DPRIVE WG.

1.1. Use Case for Unauthenticated Encryption

The use case in this document for unauthenticated 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. Ultimately, this effort has two two goals: to protect queries from failing in case authenticated encryption is not available, and to enable recursive resolver operators to encrypt without server authentication.

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

This use case is not expected to apply to all resolvers or authoritative servers. For example, according to [RSO_STATEMENT], some root server operators do not want to be the early adopters for DNS with encryption. The protocol in this document explicitly allows authoritative servers to signal when they are ready to begin offering DNS with encryption.

1.2. 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 DNS with encryption by querying for the SVCB records [SVCB]. As described in [DNS-SVCB], SVCB records can indicate that a server supports encrypted transport of DNS queries.

NOTE: In this document, the term "SVCB record" is used *only* for SVCB records that indicate encryption as described in [DNS-SVCB]. SVCB records that do not have these indicators in the RDATA are not included in the term "SVCB record" in this document.

* The resolver uses any authoritative server with a SVCB record that indicates encryption to perform unauthenticated encryption.

* The resolver does not fail to set up encryption if server authentication in the TLS session fails.

1.3. Definitions

The terms "recursive resolver", "authoritative server", and "classic DNS" are defined in [DNS-TERM].

"DNS with encryption" means transport of DNS over any of DoT, DoH, or DoQ. A server that supports DNS with encryption supports transport over one or more of DoT, DoH, or DoQ.

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 [MUST-SHOULD-1] [MUST-SHOULD-2] when, and only when, they appear in all capitals, as shown here.

2. Discovery of Authoritative Server Encryption

An authoritative server that supports DNS with encryption makes itself discoverable by publishing one or more DNS SVCB records that contain "alpn" parameter keys. SVCB records are defined in [SVCB], and the DNS extension to those records is defined in [DNS-SVCB].

A recursive resolver discovers whether an authoritative server supports DNS with encryption by looking for cached SVCB records for the name of the authoritative server with a positive answer. A cached DNS SVCB record with a negative answer indicates that the authoritative server does not support any encrypted transport.

A resolver MAY also use port probing, although the mechanism for that is not described here.

If the cache has no positive or negative answers for any SVCB record for any of a zone's authoritative servers, the resolver MAY send queries for the SVCB records (and for the A/AAAA records of names mentioned in those SVCB records) for some or all of the zone's authoritative servers and wait for a positive response so that the resolver can use DNS with encryption for the original query. In this situation, the resolver MAY instead just use classic DNS for the original query but simultaneously queue queries for the SVCB (and subsequent A/AAAA) records for some or all of the zone's authoritative servers so that future queries might be able to use DNS with encryption.

DNSSEC validation of SVCB RRsets used strictly for this discovery mechanism is not mandated.

3. Processing Discovery Responses

After a resolver has DNS SCVB records in its cache (possibly due to having just queried for them), it needs to use those records to try to find an authoritative server that uses DNS with encryption. This section describes how the resolver can make that selection.

A resolver MUST NOT attempt encryption for a server that has a negative response in its cache for the associated DNS SVCB record.

After sending out all requests for SVCB records for the authoritative servers in the NS RRset for a name, if all of the SVCB records for those authoritative servers in the cache are negative responses, the resolver MUST use classic (unencrypted) DNS instead of encryption. Similarly, if none of the DNS SVCB records for the authoritative servers in the cache have supported "alpn" parameters, the resolver MUST use classic (unencrypted) DNS instead of encryption.

If there are any DNS SVCB records in the cache for the authoritative servers for a zone with supported "alpn" parameters, the resolver MUST try each indicated authoritative server using DNS with encryption until it successfully sets up a connection. The resolver attempts to use the encrypted transports that are in the associated SVCB record for the authoritative server.

A resolver SHOULD keep a DNS with encryption session to a particular server open if it expects to send additional queries to that server in a short period of time. [DNS-OVER-TCP] says "both clients and servers SHOULD support connection reuse" for TCP connections, and that advice could apply as well for DNS with encryption, especially as DNS with encryption has far greater overhead for re-establishing a connection. If the server closes the DNS with encryption session, the resolver can possibly re-establish a DNS with encryption session using encrypted session resumption. Configuration for the maximum timeout, minimum timeout, and duration of encrypted sessions should take into consideration the recommendations given in [TCP-TIMEOUT], [EDNS-TCP], and (for DoH) [HTTP-1.1].

For any DNS with encryption protocols, TLS version 1.3 [TLS-13] or later MUST be used.

A resolver following this protocol does not need to authenticate TLS servers. Thus, when setting up a TLS connection, if the server's authentication credentials do not match those expected by the resolver, the resolver continues with the TLS connection. Privacy-oriented resolvers (defined in [PRIVACY-REC]) following this protocol MUST NOT indicate that they are using encryption because this protocol is susceptible to on-path attacks.

If the resolver gets a TLS failure (such as those listed in Section 3.2, the resolver instead uses classic DNS on any of the authoritative servers.

3.1. Resolver Process as Pseudocode

This section is meant as an informal clarification of the protocol, and is not normative. The pseudocode here is designed to show the intent of the protocol, so it is not optimized for things like intersection of sets and other shortcuts.

In this code, signal_rrset(this_name) means an SVCB query for the '_dns' prefix of this_name. The Query over secure transport until successful section ignores differences in name server selection and retry behaviour in different resolvers.

```
# Inputs
ns_names = List of NS Rdatas from the NS RRset for the queried name
can_do_secure = List of secure transports supported by resolver
secure_names_and_transports = Empty list, filled in below

# Fill secure_names_and_transports with (name, transport) tuples
for this_name in ns_names:
  if signal_rrset(this_name) is in the resolver cache:
    if signal_rrset(this_name) positively does not exist:
      continue
    for this_transport in signal_rrset(this_name):
      if this_transport in can_do_secure:
        add (this_name, this_transport) to secure_names_and_transports
  else: # signal_rrset(this_name) is not in the resolver cache
    queue a query for signal_rrset(this_name) for later caching

# Query over secure transport until successful
for (this_name, this_transport) tuple in secure_names_and_transports:
  query using this_transport on this_name
  if successful:
    finished

# Got here if no this_name/this_transport query was successful
#   or if secure_names_and_transports was empty
query using classic DNS; finished
```

3.2. Resolver Session Failures

The following are some of the reasons that a DNS with encryption session might fail to be set up:

* The resolver receives a TCP RST response
* The resolver does not receive replies to TCP or TLS setup (such as getting the TCP SYN message, the first TLS message, or completing TLS handshakes)
* The TLS handshake gets a definitive failure
* The encrypted session fails for reasons other than for authentication, such as incorrect algorithm choices or TLS record failures

4. Serving with Encryption

An operator of an authoritative server following this protocol SHOULD publish SVCB records as described in Section 2. If they cannot publish such records, the security properties of their authoritative servers will not be found. If an operator wants to test serving using encryption, they can publish SVCB records with short TTLs and then stop serving with encryption after removing the SVCB records and waiting for the TTLs to expire.

It is acceptable for an operator of authoritative servers to only offer encryption on some of the named authoritative servers, such as when the operator is determining how far to roll out encrypted service.

A server MAY close an encrypted connection at any time. For example, it can close the session if it has not received a DNS query in a defined length of time. The server MAY close an encrypted session after it sends a DNS response; however, it might also want to keep the session open waiting for another DNS query from the resolver. [DNS-OVER-TCP] says "both clients and servers SHOULD support connection reuse" for TCP connections, and that advice could apply as well for DNS with encryption, especially as DNS with encryption has far greater overhead for re-establishing a connection. If the server closes the DNS with encryption session, the resolver can possibly re-establish a DNS with encryption session using encrypted session resumption.

For any DNS with encryption protocols, TLS version 1.3 [TLS-13] or later MUST be used.

5. IANA Considerations

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

((Maybe other updates for DoH and DoQ))

6. 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 encryption. The method described in this document explicitly allows a resolver using encryption to choose to allow unauthenticated encryption. 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.

[TLS-1.3] specifically warns against anonymous connections because such connections only provide protection against passive eavesdropping while failing to protect against active on-path attacks. Section C.5 of [TLS-1.3] explicitly states applications MUST NOT use TLS with unverifiable server authentication unless there is explicit configuration or a specific application profile to do so. This document is such an application profile.

Encrypting the traffic between resolvers and authoritative servers does not solve all the privacy issues for resolution. See [PRIVACY-REC] and [PRIVACY-CONS] for in-depth discussion of the associated privacy issues.

7. Acknowledgements

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

The DPRIVE Working Group has contributed many ideas that keep shifting the focus and content of this document.

8. References

8.1. Normative References

[DNS-SVCB] Schwartz, B., "Service Binding Mapping for DNS Servers", Work in Progress, Internet-Draft, draft-schwartz-svcb-dns-04, 26 July 2021, <https://www.ietf.org/archive/id/draft-schwartz-svcb-dns-04.txt>.

[DNS-TERM] Hoffman, P. and K. Fujiwara, "DNS Terminology", Work in Progress, Internet-Draft, draft-ietf-dnsop-rfc8499bis-03, 28 September 2021, <https://www.ietf.org/archive/id/draft-ietf-dnsop-rfc8499bis-03.txt>.

[FULL-AUTH] Pauly, T., Rescorla, E., Schinazi, D., and C. A. Wood, "Signaling Authoritative DNS Encryption", Work in Progress, Internet-Draft, draft-rescorla-dprive-adox-latest-00, 26 February 2021, <https://www.ietf.org/archive/id/draft-rescorla-dprive-adox-latest-00.txt>.

[MUST-SHOULD-1] 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>.

[MUST-SHOULD-2] 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>.

[OPPORTUN] 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>.

[SVCB] Schwartz, B., Bishop, M., and E. Nygren, "Service binding and parameter specification via the DNS (DNS SVCB and HTTPS RRs)", Work in Progress, Internet-Draft, draft-ietf-dnsop-svcb-https-07, 5 August 2021, <https://www.ietf.org/archive/id/draft-ietf-dnsop-svcb-https-07.txt>.

[TLS-13] 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

[DNS-OVER-HTTPS] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://www.rfc-editor.org/info/rfc8484>.

[DNS-OVER-QUIC] Huitema, C., Dickinson, S., and A. Mankin, "Specification of DNS over Dedicated QUIC Connections", Work in Progress, Internet-Draft, draft-ietf-dprive-dnsoquic-04, 3 September 2021, <https://www.ietf.org/archive/id/draft-ietf-dprive-dnsoquic-04.txt>.

[DNS-OVER-TCP] Dickinson, J., Dickinson, S., Bellis, R., Mankin, A., and D. Wessels, "DNS Transport over TCP - Implementation Requirements", RFC 7766, DOI 10.17487/RFC7766, March 2016, <https://www.rfc-editor.org/info/rfc7766>.

[DNS-OVER-TLS]
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>.

[EDNS-TCP] Wouters, P., Abley, J., Dickinson, S., and R. Bellis, "The edns-tcp-keepalive EDNS0 Option", RFC 7828, DOI 10.17487/RFC7828, April 2016, <https://www.rfc-editor.org/info/rfc7828>.

[HTTP-1.1] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <https://www.rfc-editor.org/info/rfc7230>.

[PRIVACY-CONS] Wicinski, T., Ed., "DNS Privacy Considerations", RFC 9076, DOI 10.17487/RFC9076, July 2021, <https://www.rfc-editor.org/info/rfc9076>.

[PRIVACY-REC] Dickinson, S., Overeinder, B., van Rijswijk-Deij, R., and A. Mankin, "Recommendations for DNS Privacy Service Operators", BCP 232, RFC 8932, DOI 10.17487/RFC8932, October 2020, <https://www.rfc-editor.org/info/rfc8932>.

[RSO_STATEMENT] "Statement on DNS Encryption", 2021, <https://root-servers.org/media/news/Statement_on_DNS_Encryption.pdf>.

[TCP-TIMEOUT] Kristoff, J. and D. Wessels, "DNS Transport over TCP - Operational Requirements", Work in Progress, Internet-Draft, draft-ietf-dnsop-dns-tcp-requirements-12, 18 August 2021, <https://www.ietf.org/archive/id/draft-ietf-dnsop-dns-tcp-requirements-12.txt>.

[TLS-1.3] 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>.

Authors' Addresses

Paul Hoffman
ICANN

Email: paul.hoffman@icann.org

Peter van Dijk
PowerDNS

Email: peter.van.dijk@powerdns.com