

dprive
Internet-Draft
Updates: [1995](#) (if approved)
Intended status: Standards Track
Expires: January 9, 2020

H. Zhang
P. Aras
Salesforce
W. Toorop
NLnet Labs
S. Dickinson
Sinodun IT
A. Mankin
Salesforce
July 8, 2019

DNS Zone Transfer-over-TLS
draft-hzpa-dprive-xfr-over-tls-02

Abstract

DNS zone transfers are transmitted in clear text, which gives attackers the opportunity to collect the content of a zone by eavesdropping on network connections. The DNS Transaction Signature (TSIG) mechanism is specified to restrict direct zone transfer to authorized clients only, but it does not add confidentiality. This document specifies use of DNS-over-TLS to prevent zone contents collection via passive monitoring of zone transfers.

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 <http://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 January 9, 2020.

Copyright Notice

Copyright (c) 2019 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 (<http://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	3
2.	Terminology	4
3.	Use Cases for XFR-over-TLS	4
4.	Connection and Data Flows in Existing XFR Mechanisms	5
4.1.	AXFR Mechanism	5
4.2.	IXFR Mechanism	6
4.3.	Data Leakage of NOTIFY and SOA Message Exchanges	7
4.3.1.	NOTIFY	7
4.3.2.	SOA	8
5.	Connection and Data Flows in XoT	8
5.1.	Performance Considerations	8
5.2.	AXoT mechanism	8
5.3.	IXoT mechanism	9
5.3.1.	Fallback to AXFR	10
6.	Zone Transfer with DoT - Authentication	10
6.1.	TSIG	10
6.2.	TLS	10
6.2.1.	Opportunistic	10
6.2.2.	Strict	10
6.2.3.	Mutual	10
6.3.	IP Based ACL on the Primary	11
6.4.	ZONEMD	11
6.5.	Comparison of Authentication Methods	11
7.	Policies for Both AXFR and IXFR	12
8.	Multi-primary Configurations	13
9.	Implementation Considerations	13
10.	Implementation Status	14
11.	IANA Considerations	14
12.	Security Considerations	14
13.	Acknowledgements	14
14.	Changelog	14
15.	References	15
15.1.	Normative References	15
15.2.	Informative References	16
15.3.	URIs	17
	Authors' Addresses	17

1. Introduction

DNS has a number of privacy vulnerabilities, as discussed in detail in [[I-D.bortzmeyer-dprive-rfc7626-bis](#)]. Stub client to recursive resolver query privacy has received the most attention to date. There are now standards track documents for three encryption capabilities for stub to recursive queries and more work going on to guide deployment of specifically DNS-over-TLS (DoT) [[RFC7858](#)] and DNS-over-HTTPS (DoH) [[RFC8484](#)].

[[I-D.bortzmeyer-dprive-rfc7626-bis](#)] established that stub client DNS query transactions are not public and needed protection, but on zone transfer [[RFC1995](#)] [[RFC5936](#)] it says only:

"Privacy risks for the holder of a zone (the risk that someone gets the data) are discussed in [[RFC5936](#)] and [[RFC5155](#)]."

In what way is exposing the full contents of a zone a privacy risk? The contents of the zone could include information such as names of persons used in names of hosts. Best practice is not to use personal information for domain names, but many such domain names exist. There may also be regulatory, policy or other reasons why the zone contents in full must be treated as private.

Neither of the RFCs mentioned in [[I-D.bortzmeyer-dprive-rfc7626-bis](#)] contemplates the risk that someone gets the data through eavesdropping on network connections, only via enumeration or unauthorized transfer as described in the following paragraphs.

[[RFC5155](#)] specifies NSEC3 to prevent zone enumeration, which is when queries for the authenticated denial of existences records of DNSSEC allow a client to walk through the entire zone. Note that the need for this protection also motivates NSEC5 [[I-D.vcelak-nsec5](#)]; zone walking is now possible with NSEC3 due to crypto-breaking advances, and NSEC5 is a response to this problem.

[[RFC5155](#)] does not address data obtained outside zone enumeration (nor does [[I-D.vcelak-nsec5](#)]). Preventing eavesdropping of zone transfers (this draft) is orthogonal to preventing zone enumeration, though they aim to protect the same information.

[[RFC5936](#)] specifies using TSIG [[RFC2845](#)] for authorization of the clients of a zone transfer and for data integrity, but does not express any need for confidentiality, and TSIG does not offer encryption. Some operators use SSH tunneling or IPsec to encrypt the transfer data.

Because the AXFR zone transfer is typically carried out-over-TCP from authoritative DNS protocol implementations, encrypting AXFR using DNS-over-TLS [[RFC7858](#)] seems like a simple step forward. This document specifies how to use DoT to prevent zone collection from zone transfers, including discussion of approaches for IXFR, which uses UDP or TCP.

2. 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](#)] and [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

Privacy terminology is as described in [Section 3 of \[RFC6973\]](#).

Note that in this document we choose to use the terms 'primary' and 'secondary' for two servers engaged in zone transfers.

DNS terminology is as described in [[RFC8499](#)].

DoT: DNS-over-TLS as specified in [[RFC7858](#)]

DoH: DNS-over-HTTPS as specified in [[RFC8484](#)]

XoT: Generic XFR-over-TLS mechanisms as specified in this document

AXoT: AXFR-over-TLS

IXoT: IXFR over-TLS

3. Use Cases for XFR-over-TLS

- o Confidentiality. Clearly using an encrypted transport for zone transfers will defeat zone content leakage that can occur via passive surveillance.
- o Authentication. Use of single or mutual TLS authentication (in combination with ACLs) can complement and potentially be an alternative to TSIG.
- o Performance. Existing AXFR and IXFR mechanisms have the burden of backwards compatibility with older implementations based on the original specifications in [[RFC1034](#)] and [[RFC1035](#)]. For example, some older AXFR servers don't support using a TCP connection for multiple AXFR sessions or XFRs of different zones because they have not been updated to follow the guidance in [[RFC5836](#)]. Any

implementation of XFR-over-TLS would obviously be required to implement optimized and interoperable transfers as described in [RFC5936] e.g. transfer of multiple zones-over-one connection.

- o Performance. Current usage of TCP for IXFR is sub-optimal in some cases i.e. connections are frequently closed after a single IXFR.

4. Connection and Data Flows in Existing XFR Mechanisms

The original specification for zone transfers in [RFC1034] and [RFC1035] was based on a polling mechanism: a secondary performed a periodic SOA query (based on the refresh timer) to determine if an AXFR was required.

[RFC1995] and [RFC1996] introduced the concepts of IXFR and NOTIFY respectively, to provide for prompt propagation of zone updates. This has largely replaced AXFR where possible, particularly for dynamically updated zones.

[RFC5936] subsequently redefined the specification of AXFR to improve performance and interoperability.

In this document we use the phrase "XFR mechanism" to describe the entire set of message exchanges between a secondary and a primary that concludes in a successful AXFR or IXFR request/response. This set may or may not include

- o NOTIFY messages
- o SOA queries
- o Fallback from IXFR to AXFR
- o Fallback from IXFR-over-UDP to IXFR-over-TCP

The term is used to encompass the range of permutations that are possible and is useful to distinguish the 'XFR mechanism' from a single XFR request/response exchange.

4.1. AXFR Mechanism

The figure below provides an outline of an AXFR mechanism including NOTIFYS.

Figure 1. AXFR Mechanism [1]

1. An AXFR is often (but not always) preceded by a NOTIFY (over UDP) from the primary to the secondary. A secondary may also initiate

an AXFR based on a refresh timer or scheduled/triggered zone maintenance.

2. The secondary will normally (but not always) make a SOA query to the primary to obtain the serial number of the zone held by the primary.
3. If the primary serial is higher than the secondaries serial (using Serial Number Arithmetic [[RFC1982](#)]), the secondary makes an AXFR request (over TCP) to the primary after which the AXFR data flows in one or more AXFR responses on the TCP connection.

[RFC5936] specifies that AXFR must use TCP as the transport protocol but details that there is no restriction in the protocol that a single TCP session must be used only for a single AXFR exchange, or even solely for XFRs. For example, it outlines that the SOA query can also happen on this connection. However, this can cause interoperability problems with older implementations that support only the trivial case of one AXFR per connection.

Further details of the limitations in existing AXFR implementations are outlined in [[RFC5936](#)].

It is noted that unless the NOTIFY is sent over a trusted communication channel and/or signed by TSIG is can be spoofed causing unnecessary zone transfer attempts.

Similarly unless the SOA query is sent over a trusted communication channel and/or signed by TSIG the response can, in principle, be spoofed causing a secondary to incorrectly believe its version of the zone is update to date. Repeated successful attacks on the SOA could result in a secondary serving stale zone data.

[4.2.](#) IXFR Mechanism

The figure below provides an outline of the IXFR mechanism including NOTIFYs.

Figure 1. IXFR Mechanism [[2](#)]

1. An IXFR is normally (but not always) preceded by a NOTIFY (over UDP) from the primary to the secondary. A secondary may also initiate an IXFR based on a refresh timer or scheduled/triggered zone maintenance.
2. The secondary will normally (but not always) make a SOA query to the primary to obtain the serial number of the zone held by the primary.

3. If the primary serial is higher than the secondaries serial (using Serial Number Arithmetic [[RFC1982](#)]), the secondary makes an IXFR request to the primary after the primary sends an IXFR response.

[RFC1995] specifies that Incremental Transfer may use UDP if the entire IXFR response can be contained in a single DNS packet, otherwise, TCP is used. In fact it says in non-normative language:

"Thus, a client should first make an IXFR query using UDP."

So there may be a forth step above where the client falls back to IXFR-over-TCP. There may also be a forth step where the secondary must fall back to AXFR because e.g. the primary does not support IXFR.

However it is noted that at least two widely used open source authoritative nameserver implementations (BIND [[3](#)] and NSD [[4](#)]) do IXFR using TCP by default in their latest releases. For BIND TCP connections are sometimes used for SOA queries but in general they are not used persistently and close after an IXFR is completed.

It is noted that the specification for IXFR was published well before TCP was considered a first class transport for DNS. This document therefore updates [[RFC1995](#)] to state that DNS implementations that support IXFR-over-TCP MUST use [[RFC7766](#)] to optimise the use of TCP connections and SHOULD use [[RFC7858](#)] to manage persistent connections.

[4.3.](#) Data Leakage of NOTIFY and SOA Message Exchanges

This section attempts to presents a rationale for also encrypting the other messages in the XFR mechanism.

Since the SOA of the published zone can be trivially discovered by simply querying the publicly available authoritative servers leakage RR of this is not discussed in the following sections.

[4.3.1.](#) NOTIFY

Unencrypted NOTIFY messages identify configured secondaries on the primary.

[RFC1996] also states:

"If ANCOUNT>0, then the answer section represents an unsecure hint at the new RRset for this .

But since the only supported QTYPE for NOTIFY is SOA, this does not pose a potential leak.

4.3.2. SOA

For hidden primaries or secondaries the SOA response leaks the degree of lag of any downstream secondary.

5. Connection and Data Flows in XoT

5.1. Performance Considerations

The details in [[RFC7766](#)], [[RFC7858](#)] and [[RFC8310](#)] about e.g. using persistent connections and TLS Session Resumption [[RFC5077](#)] are fully applicable to XFR-over-TLS as well.

It is RECOMMENDED that clients and servers that support XoT also implement EDNS0 Keepalive [[RFC7828](#)].

5.2. AXoT mechanism

The figure below provides an outline of the AXoT mechanism including NOTIFYs.

Figure 3: AXoT mechanism [[5](#)]

All implementations that support XoT MUST fully implement [[RFC5953](#)] behavior on TLS connections.

Sections [4.1](#), [4.1.1](#) and [4.1.2](#) of [[RFC5936](#)] describe guidance for AXFR clients and servers with regard to re-use of sessions for multiple AXFRs, AXFRs of different zones and using TCP session for other queries including SOA.

For clarity we restate here that an AXoT client MAY use an already opened TLS connection to send a AXFR request. Using an existing open connection is RECOMMENDED over opening a new connection. (Non-AXoT session traffic can also use an open connection.)

For clarity we additionally state here that an AXoT client MAY use an already opened TLS connection to send a SOA request. Using an existing open connection is RECOMMENDED over opening a new connection.

The connection for AXFR-over-TLS SHOULD be established using port 853, as specified in [[RFC7858](#)], unless there is mutual agreement between the secondary and primary to use a port other than port 853 for XFR-over-TLS.

QUESTION: Should there be a requirement that the SOA is always done on a TLS connection if the XFR is? For the case when no transfer is required this could be unnecessary overhead.

5.3. IXoT mechanism

The figure below provides an outline of the IXoT mechanism including NOTIFYs.

Figure 4: IXoT mechanism [6]

The connection for IXFR-over-TLS SHOULD be established using port 853, as specified in [RFC7858], unless there is mutual agreement between the secondary and primary to use a port other than port 853 for XFR-over-TLS.

[RFC1995] says nothing with respect to optimizing IXFRs over TCP or re-using already open TCP connections to perform IXFRs or other queries. We provide guidance here that aligns with the guidance in [RFC5936] for AXFR and with that for performant TCP/TLS usage in [RFC7766] and [RFC7858].

An IXoT client MAY use an already opened TLS connection to send a IXFR request. Using an existing open connection is RECOMMENDED over opening a new connection. (Non-IXoT session traffic can also use an open connection.)

An IXoT client MAY use an already open TLS connection to send an SOA query. Using an existing open connection is RECOMMENDED over opening a new connection.

An IXoT server MUST be able to handle multiple IXoT requests on a single TLS connection, as well as to handle other query/response transactions over it.

An IXoT client MAY keep an existing TLS session open in the expectation it is likely to need to perform an IXFR in the near future. The client may use the frequency of recent IXFRs to calculate an average update rate and then use EDNS0 Keepalive to request an appropriate timeout from the server (if the server supports EDNS0 Keepalive). If the server does not support EDNS0 Keepalive the client MAY keep the connection open for a few seconds ([RFC7766] recommends that servers use timeouts of at least a few seconds).

An IXoT client MAY pipeline IXFR requests for different zones on a single TLS connection. AN IXoT server MAY respond to those requests out of order.

5.3.1. Fallback to AXFR

Fallback to AXFR can happen, for example, if the server is not able to provide an IXFR for the requested SOA. Implementations differ in how long they store zone deltas and how many may be stored at any one time.

After a failed IXFR a IXoT client SHOULD request the AXFR on the already open TLS connection.

6. Zone Transfer with DoT - Authentication

6.1. TSIG

TSIG [[RFC2845](#)] provides a mechanism for two parties to exchange secret keys which can then be used to create a message digest to protect individual DNS messages. This allows each party to authenticate that a request or response (and the data in it) came from the other party, even if it was transmitted-over-an unsecured channel or via a proxy. It provides party-to-party data authentication, but not hop-to-hop channel authentication or confidentiality.

6.2. TLS

6.2.1. Opportunistic

Opportunistic TLS [[RFC8310](#)] provides a defence against passive surveillance, providing on-the-wire confidentiality.

6.2.2. Strict

Strict TLS [[RFC8310](#)] requires that a client is configured with an authentication domain name (and/or SPKI pinset) that should be used to authenticate the TLS handshake with the server. This additionally provides a defense for the client against active surveillance, providing client-to-server authentication and end-to-end channel confidentiality.

6.2.3. Mutual

This is an extension to Strict TLS [[RFC8310](#)] which requires that a client is configured with an authentication domain name (and/or SPKI pinset) and a client certificate. The client offers the certificate for authentication by the server and the client can authentic the server the same way as in Strict TLS. This provides a defense for both parties against active surveillance, providing bi-directional authentication and end-to-end channel confidentiality.

6.3. IP Based ACL on the Primary

Most DNS server implementations offer an option to configure an IP based Access Control List (ACL), which is often used in combination with TSIG based ACLs to restrict access to zone transfers on primary servers.

This is also possible with XoT but it must be noted that as with TCP the implementation of such and ACL cannot be enforced on the primary until a XFR request is received on an established connection.

If control were to be any more fine-grained than this then a separate port would be required for XoT such that implementations would be able to refuse connections on that port to all clients except those configured as secondaries.

6.4. ZONEMD

Message Digest for DNS Zones (ZONEMD)

[[I-D.ietf-dnsop-dns-zone-digest](#)] digest is a mechanism that can be used to verify the content of a standalone zone. It is designed to be independent of the transmission channel or mechanism, allowing a general consumer of a zone to do origin authentication of the entire zone contents. It is not considered suitable for highly dynamic zones. It is complementary the above mechanisms and can be used in conjunction with XFR-over-TLS but is not considered further.

6.5. Comparison of Authentication Methods

The Table below compares the properties of each of the above methods in terms of what protection they provide to the secondary and primary servers during XoT in terms of:

- o 'Data Auth': Authentication that the DNS message data is signed by the party with whom credentials were shared (the signing party may or may not be party operating the far end of a TCP/TLS connection in a 'proxy' scenario). For the primary the TSIG on the XFR request confirms that the requesting party is authorized to request zone data, for the secondary it authenticates the zone data that is received.
- o 'Channel Conf': Confidentiality of the communication channel between the client and server (i.e. the two end points of a TCP/TLS connection).
- o Channel Auth: Authentication of the identity of party to whom a TCP/TLS connection is made (this might not be a direct connection between the primary and secondary in a proxy scenario).

It is noted that zone transfer scenarios can vary from a simple single primary/secondary relationship where both servers are under the control of a single operator to a complex hierarchical structure which includes proxies and multiple operators. Each deployment scenario will require specific analysis to determine which authentication methods are best suited to the deployment model in question.

Table 1: Properties of Authentication methods for XoT [7]

Based on this analysis it can be seen that:

- o A combination of Opportunistic TLS and TSIG provides both data authentication and channel confidentiality for both parties. However this does not stop a MitM attack on the channel which could be used to gather zone data.
- o Using just mutual TLS can be considered a standalone solution if the secondary has reason to place equivalent trust in channel authentication as data authentication e.g. the same operator runs both the primary and secondary.
- o Using TSIG, Strict TLS and an ACL on the primary provides all 3 properties for both parties with probably the lowest operational overhead.

7. Policies for Both AXFR and IXFR

We call the entire group of servers involved in XFR (all the primaries and all the secondaries) the 'transfer group'.

Within any transfer group both AXFRs and IXFRs for a zone SHOULD all use the same policy e.g. if AXFRs use AXoT all IXFRs SHOULD use IXoT.

In order to assure the confidentiality of the zone information, the entire transfer group MUST have a consistent policy of requiring confidentiality. If any do not, this is a weak link for attackers to exploit.

A XoT policy should specify

- o If TSIG is required
- o What kind of TLS is required (Opportunistic, Strict or mTLS)
- o If IP based ACLs should also be used.

Since this may require configuration of a number of servers who may be under the control of different operators the desired consistency could be hard to enforce and audit in practice.

Certain aspects of the Policies can be relatively easily tested independently e.g. by requesting zone transfers without TSIG, from unauthorized IP addresses or over cleartext DNS. Other aspects such as if a secondary will accept data without a TSIG digest or if secondaries are using Strict as opposed to Opportunistic TLS are more challenging.

NOTE: The authors request feedback on this challenge and welcome suggestions of how to practically manage this.

8. Multi-primary Configurations

Also known as multi-master configurations this model can provide flexibility and redundancy particularly for IXFR. A secondary will receive one or more NOTIFY messages and can send an SOA to all of the configured primaries. It can then choose to send an IXFR request to the primary with the highest SOA (or other criteria e.g. RTT).

When using persistent connections the secondary may have a TLS connection already open to one or more primaries. Should a secondary preferentially request an IXFR from a primary to which it already has an open TLS connection or the one with the highest SOA (assuming it doesn't have a connection open to it already)?

Two extremes can be envisaged here. In the first case the secondary continues to use one persistent connection to a single primary until it has reason not to. Reasons not to might include the primary repeatedly closing the connection, long RTTs on transfers or the SOA of the primary being an unacceptable lag behind the SOA of an alternative primary.

At the other extreme a primary could keep multiple persistent connections open to all available primaries and only request IXFRs from the primary with the highest serial number. Since normally the number of secondaries and primaries in direct contact in a transfer group is reasonably low this might be feasible if latency is the most significant concern.

9. Implementation Considerations

TBD

10. Implementation Status

The 1.9.2 version of Unbound [8] includes an option to perform AXFR-over-TLS (instead of TCP). This requires the client (secondary) to authenticate the server (primary) using a configured authentication domain name.

It is noted that use of a TLS proxy in front of the primary server is a simple deployment solution that can enable server side XoT.

11. IANA Considerations

TBD

12. Security Considerations

This document specifies a security measure against a DNS risk: the risk that an attacker collects entire DNS zones through eavesdropping on clear text DNS zone transfers. It presents a new Security Consideration for DNS. Some questions to discuss are:

- o Should DoT in this new case be required to use only TLS 1.3 and higher to avoid residual exposure?
- o How should padding be used in IXFR?
- o Should there be an option to 'pad' an AXFR response (i.e. a set of AXFR responses on a given connection) to hide the zone size?

13. Acknowledgements

The authors thank Benno Overeinder, Shumon Huque and Tim Wicinski for review and discussions.

14. Changelog

[draft-hzpa-dprive-xfr-over-tls-01](#)

- o Substantial re-work of the document.

[draft-hzpa-dprive-xfr-over-tls-01](#)

- o Editorial changes, updates to references.

[draft-hzpa-dprive-xfr-over-tls-00](#)

- o Initial commit

15. References

15.1. Normative References

- [I-D.bortzmeyer-dprive-rfc7626-bis]
Bortzmeyer, S. and S. Dickinson, "DNS Privacy Considerations", [draft-bortzmeyer-dprive-rfc7626-bis-02](#) (work in progress), January 2019.
- [I-D.vcelak-nsec5]
Vcelak, J., Goldberg, S., Papadopoulos, D., Huque, S., and D. Lawrence, "NSEC5, DNSSEC Authenticated Denial of Existence", [draft-vcelak-nsec5-08](#) (work in progress), December 2018.
- [RFC1995] Ohta, M., "Incremental Zone Transfer in DNS", [RFC 1995](#), DOI 10.17487/RFC1995, August 1996, <<https://www.rfc-editor.org/info/rfc1995>>.
- [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>>.
- [RFC2845] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", [RFC 2845](#), DOI 10.17487/RFC2845, May 2000, <<https://www.rfc-editor.org/info/rfc2845>>.
- [RFC5077] Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State", [RFC 5077](#), DOI 10.17487/RFC5077, January 2008, <<https://www.rfc-editor.org/info/rfc5077>>.
- [RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", [RFC 5155](#), DOI 10.17487/RFC5155, March 2008, <<https://www.rfc-editor.org/info/rfc5155>>.
- [RFC5936] Lewis, E. and A. Hoenes, Ed., "DNS Zone Transfer Protocol (AXFR)", [RFC 5936](#), DOI 10.17487/RFC5936, June 2010, <<https://www.rfc-editor.org/info/rfc5936>>.
- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", [RFC 6973](#), DOI 10.17487/RFC6973, July 2013, <<https://www.rfc-editor.org/info/rfc6973>>.

- [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>>.
- [RFC8310] Dickinson, S., Gillmor, D., and T. Reddy, "Usage Profiles for DNS over TLS and DNS over DTLS", [RFC 8310](#), DOI 10.17487/RFC8310, March 2018, <<https://www.rfc-editor.org/info/rfc8310>>.
- [RFC8484] 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>>.
- [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>>.

15.2. Informative References

- [I-D.ietf-dnsop-dns-zone-digest]
Wessels, D., Barber, P., Weinberg, M., Kumari, W., and W. Hardaker, "Message Digest for DNS Zones", [draft-ietf-dnsop-dns-zone-digest-00](#) (work in progress), June 2019.
- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, [RFC 1034](#), DOI 10.17487/RFC1034, November 1987, <<https://www.rfc-editor.org/info/rfc1034>>.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, [RFC 1035](#), DOI 10.17487/RFC1035, November 1987, <<https://www.rfc-editor.org/info/rfc1035>>.
- [RFC1982] Elz, R. and R. Bush, "Serial Number Arithmetic", [RFC 1982](#), DOI 10.17487/RFC1982, August 1996, <<https://www.rfc-editor.org/info/rfc1982>>.
- [RFC1996] Vixie, P., "A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)", [RFC 1996](#), DOI 10.17487/RFC1996, August 1996, <<https://www.rfc-editor.org/info/rfc1996>>.

- [RFC5953] Hardaker, W., "Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP)", [RFC 5953](#), DOI 10.17487/RFC5953, August 2010, <<https://www.rfc-editor.org/info/rfc5953>>.
- [RFC7766] 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>>.

15.3. URIs

- [1] https://github.com/hanzhang0116/hzpa-dprive-xfr-over-tls/blob/02_updates/02-draft-svg/AXFR_mechanism.svg
- [2] https://github.com/hanzhang0116/hzpa-dprive-xfr-over-tls/blob/02_updates/02-draft-svg/IXFR%20mechanism.svg
- [3] <https://www.isc.org/bind/>
- [4] <https://www.nlnetlabs.nl/projects/nsd/about/>
- [5] https://github.com/hanzhang0116/hzpa-dprive-xfr-over-tls/blob/02_updates/02-draft-svg/AXoT_mechanism_1.svg
- [6] https://github.com/hanzhang0116/hzpa-dprive-xfr-over-tls/blob/02_updates/02-draft-svg/IXoT_mechanism_1.svg
- [7] https://github.com/hanzhang0116/hzpa-dprive-xfr-over-tls/blob/02_updates/02-draft-svg/Properties_of_Authentication_methods_for_XoT.svg
- [8] <https://github.com/NLnetLabs/unbound/blob/release-1.9.2/doc/Changelog>

Authors' Addresses

Han Zhang
Salesforce
San Francisco, CA
United States

Email: h Zhang@salesforce.com

Pallavi Aras
Salesforce
Herndon, VA
United States

Email: paras@salesforce.com

Willem Toorop
NLnet Labs
Science Park 400
Amsterdam 1098 XH
The Netherlands

Email: willem@nlnetlabs.nl

Sara Dickinson
Sinodun IT
Magdalen Centre
Oxford Science Park
Oxford OX4 4GA
United Kingdom

Email: sara@sinodun.com

Allison Mankin
Salesforce
Herndon, VA
United States

Email: allison.mankin@gmail.com

