Using Early Data in DNS over TLS
draft-ghedini-dprive-early-data-01

Abstract

This document illustrates the risks of using TLS 1.3 early data with DNS over TLS, and specifies behaviors that can be adopted by clients and servers to reduce those risks.

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 January 7, 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 (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.
# DNS Early Data

## 1. Introduction

TLS 1.3 [TLS13] defines a mechanism, called 0-RTT session resumption or early data, that allows clients to send data to servers in the first round-trip of a resumed connection without having to wait for the TLS handshake to complete.

This can be used to send DNS queries to DNS over TLS [DOT] servers without incurring in the cost of the additional round-trip required by the TLS handshake. This can provide significant performance improvements in cases where new DNS over TLS connections need to be established often such as on mobile clients where the network might not be stable, or on resolvers where keeping an open connection to many authoritative servers might not be practical.

However the use of early data allows an attacker to capture and replay the encrypted DNS queries carried on the TLS connection. This can have unwanted consequences and help in recovering information about those queries. While [TLS13] describes techniques to reduce the likelihood of a replay attack, they are not perfect and still leave some potential for exploitation.

## 2. Notational Conventions

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.
3. Early Data in DNS over TLS

Early data forms a single stream of data along with other application data, meaning that one or more DNS queries can either be partially or fully contained within early data. Once the TLS handshake has completed, the early data is known to not be a replayed copy of that data, but this doesn’t mean that it can’t be replayed, or that it hasn’t already been replayed, in another connection.

A server can signal to clients whether it is willing to accept early data in future connections by providing the "early_data" TLS extension as part of a TLS session ticket, as well as limit the amount of early data it is willing to accept using the "max_early_data_size" field of the "early_data" extension.

In addition to the mitigation mechanisms mandated in [TLS13] that reduce the ability of an attacker to replay early data, but may not completely eliminate it, a server that decided to offer early data to clients MAY reject early data at the TLS layer, or delay the processing of early data after the handshake is completed.

If the server rejects early data at the TLS layer, a client MUST forget information it optimistically assumed about the connection when sending early data, such as the negotiated protocol [ALPN]. Any DNS queries sent in early data will need to be sent again, unless the client decides to abandon them.

Not all types of DNS queries are safe to be sent as early data. Clients MUST NOT use early data to send DNS Updates ([RFC2136]) or Zone Transfers ([RFC5936]) messages. Servers receiving any of those messages MUST reply with a "FormErr" response code.

[[TODO: forbid other types? use a different status code? should we define a whitelist instead of a blacklist?]]

4. Security Considerations

4.1. Information Exposure

By replaying DNS queries that were captured when transmitted over early data, an attacker might be able to expose information about those queries, even if encrypted.

For example, it’s a common behavior for DNS servers to statefully rotate the order of RRs when replying to DNS queries for an RRSet that contains multiple RRs. If the order of rotation is predictable, replaying a captured early data DNS query and observing the order of RRs in DNS responses before and after the replayed query, might allow
the attacker to confirm whether the query targeted a specific name that was suspected of being queried.

Servers SHOULD either use fixed ordering for multiple RRs in the same DNS response or shuffle the RRs at random, but MUST NOT use stateful and deterministic ordering across multiple queries.

4.2. Denial of Service

Accepting early data exposes a server to potential denial of service through the replay of queries that might be expensive to handle.

When under load, a server MAY reject TLS early data such that the client is forced to retry them after the handshake is completed.

4.3. Privacy

TBD

[[TODO: linkability (e.g. clients changing network, ...) and more?]]

5. IANA Considerations

This document has no actions for IANA.

6. Acknowledgments

Thanks to Martin Thomson, Mark Nottingham and Willy Tarreau for writing [RFC8470] which heavily inspired this document, and to Daniel Kahn Gillmor and Colm MacCarthaigh who also provided important ideas and contributions.

7. References

7.1. Normative References


7.2. Informative References


Author’s Address

Alessandro Ghedini
Cloudflare, Inc.

Email: alessandro@cloudflare.com
Abstract

DNS over TLS (DoT) has been gaining attention, primarily as a means of communication between stub resolvers and recursive resolvers. There have also been discussions and experiments involving the use of DoT to communicate with authoritative nameservers (Authoritative DNS over TLS or "ADoT"), including communication between recursive and authoritative resolvers. However, we have identified a number of operational concerns with ADoT that have arisen as DNS operators have begun to experiment with and prepare for deploying DoT. These operational concerns need to be addressed prior to ADoT’s deployment at scale by DNS operators in order to maintain the stability and resilience of the global DNS. The document also provides some suggested next steps to advance the operator community’s understanding of ADoT’s operational impact.

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 February 14, 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 (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 ................................................. 3
   1.1. Background and Motivation ............................. 3
       1.1.1. Why operational considerations are so important for
              ADoT ........................................ 3
       1.1.2. Other considerations related to ADoT ............... 4
2. Terminology .................................................. 5
   2.1. Requirements Language .................................. 5
   2.2. Definitions ............................................. 5
3. Key Issues and Questions .................................... 6
   3.1. Signaling Support for ADoT ............................. 6
   3.2. Port number ............................................ 6
   3.3. TLS version ............................................. 7
   3.4. Resumptions ............................................. 7
   3.5. Operational Monitoring ................................. 8
   3.6. Architecture ........................................... 8
   3.7. Socket efficiency/tuning considerations ............... 8
   3.8. Post-Quantum Security .................................. 9
4. Suggestions for further research and development .......... 9
   4.1. Required studies and analysis ......................... 9
   4.2. Authoritative DNS over TLS (ADoT) Profile ............ 10
5. Security Considerations ..................................... 10
6. References ................................................... 10
   6.1. Informative References ............................... 10
   6.2. URIs ................................................... 13
Appendix A. Acknowledgements ................................... 13
Appendix B. Change Log ......................................... 13
Authors’ Addresses ........................................... 14
1. Introduction

This is an operational considerations document that focuses on the factors operators need to consider when implementing Authoritative DNS over TLS. An evaluation of the merits of DNS over TLS are beyond the scope and intent of this document.

Typically, DNS communication between stub resolvers, recursive resolvers, and authoritative servers is not encrypted. Some argue that this can pose a privacy challenge for Internet users, because their access to named network resources can potentially be tracked through their DNS communication. In principle, any network element along the path between the user and resolving or authoritative nameservers could observe this unencrypted traffic. Many of these concerns are addressed in [RFC7626].

[RFC8310] proposes using DNS over TLS (DoT) in order to encrypt DNS traffic between hosts.

Historically, much of the work on DNS encryption has focused on the stub-to-recursive path, as the recursive-to-authoritative server path did not leak user specific information, other than if identifying information was encoded in the QName. However, with the increased deployment of EDNS0 Client Subnet [RFC7871], recursive-to-authoritative encryption is becoming an area of interest. Therefore, this document’s scope is the recursive-to-authoritative aspect of DoT, or Authoritative DNS over TLS (ADoT), in order to differentiate it from the stub-to-recursive path.

The addition of ADoT, while providing encryption for DNS communication, also introduces other factors that might impact the stability and resiliency of authoritative nameserver operations which may have been optimized for unencrypted DNS, often focusing on UDP transport.

The objective of this document is to try to describe the problem space, make suggestions about solutions, and propose next steps that can help inform both recursive and authoritative operators on how to assess and address the challenges posed by ADoT deployment.

1.1. Background and Motivation

1.1.1. Why operational considerations are so important for ADoT

The main concerns for most authoritative operators are the stability, resiliency, scalability, and performance of their platforms. These concerns need to be weighed against the benefits, provided to the end user, by encrypting DNS queries to the authoritative servers.
As a result of caching, the recursive-to-authoritative server communication is less attributable to a particular user than information communicated along the stub-to-recursive path. In cases where the recursive is shared and using QName minimization, some user privacy concerns may be addressed, but cases exist where QName Minimization may not be used, or users may be running their own resolver, defeating the shared service protection.

Initial deployments of ADoT may offer an immediate expansion of the attack surface (additional port, transport protocol, and computationally expensive crypto operations for an attacker to exploit) while, in some cases, providing limited protection to end users.

1.1.2. Other considerations related to ADoT

As resolvers add encryption on the client-to-recursive path, they may also change the way they handle data on the recursive-to-authoritative path. This is expressed in Mozilla Trusted Recursive Resolver (TRR) requirements [1], for example, which require participating resolvers to perform QNAME minimization [RFC7816], and TRR requirement #6, which forbids the EDNS0 Client subnet (ECS) from being propagated unless the recursive-to-authoritative path is encrypted.

The latter requirement may have the possible unintended consequence of reducing the authoritative name servers’ ability to provide a best response to DNS queries, until such time as they deploy DNS encryption.

Given that recursive resolvers should be configured to prevent ECS transmission to root, top-level, and effective top-level domain (TLD) servers [RFC7871] section 12.1 [2] - the ECS encryption requirement motivates consideration of authoritative DNS encryption below these levels.

At the higher levels, techniques such as QNAME minimization and Aggressive Use of DNSSEC-Validated Cache [RFC8198] arguably provide an alternate path toward mitigating the risk of disclosure of sensitive information without the operational risk of DNS encryption.

Resolver requirements may change as the understanding of DNS encryption options evolve, but in the meantime, they provide motivation for authoritative name server operators to weigh the risks and benefits of DNS encryption, hence the importance of understanding these operational considerations.
2. Terminology

2.1. Requirements Language

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.

2.2. Definitions

Authoritative DoT (ADoT): [I-D.draft-hoffman-dns-terminology-ter-01]

Attack Surface: The sum of attack vectors where an unauthorized user (attacker) can try to enter or extract data from the environment or compromise a service via resource starvation.

Authoritative Operator: An operator of an authoritative DNS server.

CDN: Content Delivery Network - distributed network of servers which proxy traffic between content providers and end users in order to provide high availability and high performance.

ECS: EDNS0 Client Subnet [RFC7871] - an extension to EDNS0 where the client’s subnet is included in the DNS query, intended to provide a hint to authoritative servers who may wish to provide different answers in an attempt to provide higher performance for end users based on their network location.

EPSK: External Pre-Shared Key - TLS 1.3 [RFC8446] uses the same PSK extension for keys established both during handshake (resumption PSK) and keys established externally. The EPSK acronym was introduced in draft [I-D.draft-wood-tls-external-psk-importer-00] in order to disambiguate External vs Resumption PSKs.

Performance:

- QRTT: Query Round-Trip Time - the time it takes between sending a query and receiving a response.

- Best Response - whether or not the authoritative server, if dynamic responses are used as they are in CDNs, are able to determine or infer location and provide the most local response. It is a key part of the end-to-end performance for end users to get not just _an_ answer quickly but to get the best and most local answer.
3. Key Issues and Questions

3.1. Signaling Support for ADoT

[ RFC8310 ] does not define a method for a nameserver to advertise its support of DoT other than to have the client make a connection attempt to the default port of 853. The extra round-trip to check for ADoT support imposes a penalty for clients and resolvers that either do not remember the nameserver or have not communicated with that nameserver before. The extra round-trip required may lead some implementers down a similar path to happy eyeballs [ RFC8305 ] which, in the case of DNS, would send the same query over both encrypted and un-encrypted channels at the same time. A happy eyeballs type approach, which we’ll call "leaky resolvers", would defeat the purpose of the encryption protection for the testing query, but may enable subsequent queries to be sent over a private channel with the first query being subject to on-path adversaries. An implementation could use some constant query string as a test query. However any query included in the set of queries comprising the iterative resolution for a QNAME first sent over an encrypted channel that leaks the original stub QNAME, SHOULD NOT be used.

3.2. Port number

[ RFC7858 ] section 3 [ 3 ] indicates that port 853 MUST be used for session establishment unless otherwise negotiated and configured by both the client and server. In the stub-to-recursive connection, changing the port is something that can be done at stub configuration time however, managing this negotiation between the recursive-to-authoritative server is not scalable or standardized. The scalability problem is due to the fact that recursive resolvers...
communicate with thousands of authoritative servers, therefore port/ service discovery for each of these authoritatives becomes difficult.

Static use of a pre-defined port provides on-path adversaries the ability to more easily drop or manipulate traffic intended for that port, possibly triggering resolvers to downgrade a connection back to a traditional DNS query, eliminating the encryption protections. This attack is more likely to happen on the stub-to-recursive connection but is also a possible threat for recursive-to-authoritative connections.

3.3. TLS version

Implementers of ADoT should read, understand, and follow the guidance provided in BCP195 [4], also known as [RFC7525], when deploying DoT on their platforms. At the time of writing, [RFC7525] did not include coverage for TLS 1.3. However, TLS 1.3 should be included in the document that obsoletes this BCP. Until this happens, TLS 1.3 SHOULD be preferred over TLS 1.2, as 1.3 offers both security and performance enhancements. Additionally, operators should monitor TLS version issues and cipher suite vulnerabilities for the version of TLS that their platforms offer.

In the absence of any widespread ADoT deployments, it is easier to limit TLS version 1.3 or greater. The absence of widespread adoption also allows the IETF to create and enforce standards/policies that ensure TLS versions are kept current going forward.

3.4. Resumptions

TLS resumption allows clients and servers to use information from a previously established session in order to bootstrap the cryptographic state while avoiding a full handshake. The resumption mechanism is redesigned in TLS 1.3 [RFC8446] section 2.2 [5] and section 2.3 [6], eliminating both [RFC5077] session tickets and session ID resumption.

Resumption improves both connection and resource (socket and CPU) efficiency, therefore operators SHOULD allow for TLS resumption. However, special consideration should be given to 0-RTT resumption as it is vulnerable to replay attacks [RFC3552] see Section 3.3.1 [7]. The replay attack may not be as important for DNS, as DNS queries are generally idempotent. However consideration should be given to possible side-channel attacks [8].
3.5. Operational Monitoring

Many operators use external passive monitors in order to understand the health and performance of their infrastructure. Infrastructure monitoring is also often done to retain a copy of traffic for forensic purposes - such as the BIND "packet of death" [9] scenario. These legacy monitoring systems may break with the use of TLS 1.3. Therefore alternatives may need to be deployed/developed in order to maintain effective operational performance and security monitoring functionality.

A number of solutions have been suggested:

- TLS Security and Data Center Monitoring: Searching for a Path Forward [10]

3.6. Architecture

Operators often reconfigure their architectural designs to best deliver a new product offering or service. Operators should consider the following design alternatives for the new ADoT service:

- Operators should consider segregating ADoT addresses from traditional DNS over UDP/TCP to enable better attack mitigation, better service monitoring, less service interference, and more stability.

- Operators should weigh the pros/cons of using a TLS proxy vs direct client-to-host connection. In case of ADoT, the client is most likely a recursive resolver and the host is the authoritative host server.

3.7. Socket efficiency/tuning considerations

Operators can realize substantial gains in client session establishment and improve overall RTT by tuning sockets setting for best use-case efficiency.

For the ADoT use case, operators should consult [RFC7766] section 6.2 [12] and minimally consider the following:

- Optimal number of persistent connections - consideration should be given to the number of persistent connections maintained for both the recursive resolvers and authoritative servers

- Optimal read/write buffer size
3.8. Post-Quantum Security

Given that ADoT deployments will likely have a long lifetime and are being introduced in an era where post-quantum security is now an important design consideration, it is prudent to consider how protections against quantum computers might be integrated into the deployments.

[I-D.draft-hoffman-c2pq-05] outlines the threat quantum computing presents to classical cryptographic algorithms.

External Pre-Shared Keys (EPSKs) may be less vulnerable to quantum attacks. A proposed approach to combining EPSKs and certificates in TLS is described in [I-D.draft-housley-tls-tls13-cert-with-extern-psk-03].

4. Suggestions for further research and development

4.1. Required studies and analysis

Unlike stub-to-recursive DNS communication, authoritative nameservers affect users in ways that end users cannot avoid or work around. In the event that all authoritative servers for a zone are unreachable, the zone becomes globally unavailable. Hence, in order to preserve stability and resiliency of authoritative nameservers when deploying ADoT, more empirical studies and analysis MUST be conducted. The following list is a minimal set of studies and considerations that need to be conducted/addressed in order to maintain authoritative stability and resilience.

- Attack vectors and mitigation: consider the new adversarial powers enabled by ADoT - types of attacks and denial of service, or other security challenges that are created with the addition of ADoT to authoritative nameservers.

- Traffic: consider how traffic patterns to authoritative nameservers change with the introduction of ADoT and how these traffic patterns change when the parameters of the service are changed; e.g. persistent connection lifetime, TLS connection parameters, use of TLS session tickets [RFC5077] or Pre-Shared Key extension in TLS 1.3 [RFC8446] section 2.2 [13]. Consider how
these traffic pattern changes will affect the architecture and infrastructure for authoritative operators.

- ADoT capacity and footprint expansion: consider how common scaling techniques impact authoritative operators; e.g. anycast, load balancing, custom hardware.

- DTLS/UDP - consider if there is any reason to implement DTLS given that we lose the benefit of pipelining requests and must drop back to TLS/TCP in the case of fragmentation.

It is critical to conduct large-scale measurements of DNS infrastructure in order to quantify some of the scalability issues. While these tests may be performed initially in a controlled lab environment, the public Internet is fundamentally more variable. Therefore, global testing at scale on the Internet MUST also be conducted in order to understand and measure potential issues which must be overcome before full global deployment can occur.

4.2. Authoritative DNS over TLS (ADoT) Profile

Profiles can be used as a mechanism to help mitigate operational concerns over increased attack surface by restricting features such as computationally expensive processes, insecure ciphers, general starvation vectors, or other features that may limit operational performance.

Therefore, an ADoT application profile draft, taking into account the conclusions of required studies and analysis, may help assuage some of the concerns raised in this document.

5. Security Considerations

In addition to the applicable security considerations described in RFCs [RFC7626] and [RFC8310], considerations focused on future deployment of quantum computers are described in Post-Quantum Security (Section 3.8). Additional considerations associated with ADoT are TBD based on working group discussions.

6. References

6.1. Informative References

[I-D.draft-hoffman-c2pq-05]

[I-D.draft-hoffman-dns-terminology-ter-01]

[I-D.draft-housley-tls-tls13-cert-with-extern-psk-03]

[I-D.draft-wood-tls-external-psk-importer-00]
Wood, C., "Importing External PSKs for TLS 1.3", draft-wood-tls-external-psk-importer-00 (work in progress), October 2018.


6.2. URIs


Appendix A. Acknowledgements

Thanks to those that provided usage data, reviewed and/or improved this document, including: Piet Barber, Michael Bentkofsky, David Blacka, Florent Guiliani, Scott Hollenbeck, Burt Kaliski, Glen Wiley, and Richard Wilhelm.

Appendix B. Change Log

RFC EDITOR: PLEASE REMOVE THE THIS SECTION PRIOR TO PUBLICATION.

TODO: Zero this change log out when -00 is submitted to IETF.

From -01 to -02:

  o Updates to the introduction framing
Rewording the "Why operational considerations are so important for ADoT" section

From -00 to -01:

- Removing static-dh reference
- Further clarifying the scope of the document

pre-00

- Initial draft.

```
01234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234
```

Authors’ Addresses

Karl Henderson
Verisign
Email: khenderson@verisign.com

Tim April
Akamai
Email: ietf@tapril.net

Jason Livingood
Comcast
Email: jason_livingood@comcast.com
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

- Confidentiality. Clearly using an encrypted transport for zone transfers will defeat zone content leakage that can occur via passive surveillance.

- Authentication. Use of single or mutual TLS authentication (in combination with ACLs) can complement and potentially be an alternative to TSIG.

- 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.

- 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

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

The term is used to encompasses 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 is 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 an 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 authenticate 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 an 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:

- **‘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.

- **‘Channel Conf’**: Confidentiality of the communication channel between the client and server (i.e. the two end points of a TCP/TLS connection).

- **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:

- 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.

- 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.

- 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

- If TSIG is required
- What kind of TLS is required (Opportunistic, Strict or mTLS)
- 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:

- Should DoT in this new case be required to use only TLS 1.3 and higher to avoid residual exposure?
- How should padding be used in IXFR?
- 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-dprime-xfr-over-tls-01
  - Substantial re-work of the document.

- draft-hzpa-dprime-xfr-over-tls-01
  - Editorial changes, updates to references.

- draft-hzpa-dprime-xfr-over-tls-00
  - Initial commit
15. References

15.1. Normative References

[I-D.bortzmeyer-dprive-rfc7626-bis]

[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.


15.2. Informative References

[I-D.ietf-dnsop-dns-zone-digest]


15.3. URIs

[1] https://github.com/hanzhang0116/hzpa-dprive-xfr-over-tls/blob/02_updates/02-draft-svg/AXFR_mechanism.svg


Authors' Addresses

Han Zhang
Salesforce
San Francisco, CA
United States

Email: hzhang@salesforce.com
Abstract

This document presents operational, policy and security considerations for DNS operators who choose to offer DNS Privacy services. With these recommendations, the operator can make deliberate decisions regarding which services to provide, and how the decisions and alternatives impact the privacy of users.

This document also presents a framework to assist writers of DNS Privacy Policy and Practices Statements (analogous to DNS Security Extensions (DNSSEC) Policies and DNSSEC Practice Statements described in [RFC6841]).

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.
Table of Contents

1. Introduction ................................................. 3
2. Scope ......................................................... 5
3. Privacy related documents ................................... 5
4. Terminology .................................................. 5
5. Recommendations for DNS privacy services .................. 6
   5.1. On the wire between client and server ................ 7
   5.1.1. Transport recommendations .......................... 7
   5.1.2. Authentication of DNS privacy services ............ 8
   5.1.3. Protocol recommendations ............................ 9
   5.1.4. DNSSEC ................................................. 11
   5.1.5. Availability .......................................... 11
   5.1.6. Service options ....................................... 12
   5.1.7. Impact on Operators .................................. 12
   5.1.8. Limitations of using a pure TLS proxy .............. 13
   5.2. Data at rest on the server ............................. 13
   5.2.1. Data handling ......................................... 13
   5.2.2. Data minimization of network traffic ............... 14
   5.2.3. IP address pseudonymization and anonymization methods 15
   5.2.4. Pseudonymization, anonymization or discarding of other correlation data 16
   5.2.5. Cache snooping ........................................ 17
   5.3. Data sent onwards from the server ..................... 17
   5.3.1. Protocol recommendations ............................ 17
   5.3.2. Client query obfuscation ............................ 18
   5.3.3. Data sharing .......................................... 19
6. DNS privacy policy and practice statement ............... 19
   6.1. Recommended contents of a DPPPS ....................... 20
   6.1.1. Policy ................................................ 20
   6.1.2. Practice .............................................. 21
   6.2. Current policy and privacy statements ............... 22
   6.3. Enforcement/accountability ............................ 22
7. IANA considerations .......................................... 23
8. Security considerations ..................................... 23
9. Acknowledgements ........................................... 23
10. Contributors ............................................... 23
11. Changelog .................................................. 23
1. Introduction

The Domain Name System (DNS) is at the core of the Internet; almost every activity on the Internet starts with a DNS query (and often several). However, the DNS was not originally designed with strong security or privacy mechanisms. A number of developments have taken place in recent years which aim to increase the privacy of the DNS system and these are now seeing some deployment. This latest evolution of the DNS presents new challenges to operators and this document attempts to provide an overview of considerations for privacy-focused DNS services.

In recent years, there has also been an increase in the availability of "public resolvers" [I-D.ietf-dnsop-terminology-bis] which users may prefer to use instead of the default network resolver because they offer a specific feature (e.g., good reachability, encrypted transport, strong privacy policy, filtering (or lack of), etc.). These open resolvers have tended to be at the forefront of adoption of privacy-related enhancements but it is anticipated that operators of other resolver services will follow.

Whilst protocols that encrypt DNS messages on the wire provide protection against certain attacks, the resolver operator still has (in principle) full visibility of the query data and transport identifiers for each user. Therefore, a trust relationship exists. The ability of the operator to provide a transparent, well-documented, and secure privacy service will likely serve as a major differentiating factor for privacy-conscious users if they make an active selection of which resolver to use.
It should also be noted that the choice of a user to configure a single resolver (or a fixed set of resolvers) and an encrypted transport to use in all network environments has both advantages and disadvantages. For example the user has a clear expectation of which resolvers have visibility of their query data however this resolver/transport selection may provide an added mechanism to track them as they move across network environments. Commitments from operators to minimize such tracking are also likely to play a role in user selection of resolvers.

More recently the global legislative landscape with regard to personal data collection, retention, and pseudonymization has seen significant activity. It is an untested area that simply using a DNS resolution service constitutes consent from the user for the operator to process their query data. The impact of recent legislative changes on data pertaining to the users of both Internet Service Providers and public DNS resolvers is not fully understood at the time of writing.

This document has two main goals:

- To provide operational and policy guidance related to DNS over encrypted transports and to outline recommendations for data handling for operators of DNS privacy services.

- To introduce the DNS Privacy Policy and Practice Statement (DPPPS) and present a framework to assist writers of this document. A DPPPS is a document that an operator can publish outlining their operational practices and commitments with regard to privacy thereby providing a means for clients to evaluate the privacy properties of a given DNS privacy service. In particular, the framework identifies the elements that should be considered in formulating a DPPPS. This document does not, however, define a particular Policy or Practice Statement, nor does it seek to provide legal advice or recommendations as to the contents.

A desired operational impact is that all operators (both those providing resolvers within networks and those operating large anycast services) can demonstrate their commitment to user privacy thereby driving all DNS resolution services to a more equitable footing. Choices for users would (in this ideal world) be driven by other factors e.g. differing security policies or minor difference in operator policy rather than gross disparities in privacy concerns.

Community insight [or judgment?] about operational practices can change quickly, and experience shows that a Best Current Practice (BCP) document about privacy and security is a point-in-time
2. Scope

"DNS Privacy Considerations" [I-D.bortzmeyer-dprive-rfc7626-bis] describes the general privacy issues and threats associated with the use of the DNS by Internet users and much of the threat analysis here is lifted from that document and from [RFC6973]. However this document is limited in scope to best practice considerations for the provision of DNS privacy services by servers (recursive resolvers) to clients (stub resolvers or forwarders). Privacy considerations specifically from the perspective of an end user, or those for operators of authoritative nameservers are out of scope.

This document includes (but is not limited to) considerations in the following areas (taken from [I-D.bortzmeyer-dprive-rfc7626-bis]):

1. Data "on the wire" between a client and a server
2. Data "at rest" on a server (e.g. in logs)
3. Data "sent onwards" from the server (either on the wire or shared with a third party)

Whilst the issues raised here are targeted at those operators who choose to offer a DNS privacy service, considerations for areas 2 and 3 could equally apply to operators who only offer DNS over unencrypted transports but who would like to align with privacy best practice.

3. Privacy related documents

There are various documents that describe protocol changes that have the potential to either increase or decrease the privacy of the DNS. Note this does not imply that some documents are good or bad, better or worse, just that (for example) some features may bring functional benefits at the price of a reduction in privacy and conversely some features increase privacy with an accompanying increase in complexity. A selection of the most relevant documents are listed in Appendix A for reference.

4. 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.

DNS terminology is as described in [I-D.ietf-dnsop-terminology-bis] with one modification: we restate the clause in the original definition of Privacy-enabling DNS server in [RFC8310] to include the requirement that a DNS over (D)TLS server should also offer at least one of the credentials described in Section 8 and implement the (D)TLS profile described in Section 9 of [RFC8310].

Other Terms:

- DPPPS: DNS Privacy Policy and Practice Statement, see Section 6.
- DNS privacy service: The service that is offered via a privacy-enabling DNS server and is documented either in an informal statement of policy and practice with regard to users privacy or a formal DPPPS.

5. Recommendations for DNS privacy services

We describe two classes of threats:

- ‘Privacy Considerations for Internet Protocols’ [RFC6973] Threats
  * Privacy terminology, threats to privacy and mitigations as described in Sections 3, 5 and 6 of [RFC6973].

- DNS Privacy Threats
  * These are threats to the users and operators of DNS privacy services that are not directly covered by [RFC6973]. These may be more operational in nature such as certificate management or service availability issues.

We describe three classes of actions that operators of DNS privacy services can take:

- Threat mitigation for well understood and documented privacy threats to the users of the service and in some cases to the operators of the service.
- Optimization of privacy services from an operational or management perspective
- Additional options that could further enhance the privacy and usability of the service
This document does not specify policy only best practice, however for DNS Privacy services to be considered compliant with these best practice guidelines they SHOULD implement (where appropriate) all:

- Threat mitigations to be minimally compliant
- Optimizations to be moderately compliant
- Additional options to be maximally compliant

5.1. On the wire between client and server

In this section we consider both data on the wire and the service provided to the client.

5.1.1. Transport recommendations

[RFC6973] Threats:

- Surveillance:
  - Passive surveillance of traffic on the wire
    [I-D.bortzmeyer-dprive-rfc7626-bis] Section 2.4.2.

DNS Privacy Threats:

- Active injection of spurious data or traffic

Mitigations:

A DNS privacy service can mitigate these threats by providing service over one or more of the following transports

- DNS-over-TLS [RFC7858] and [RFC8310]
- DoH [RFC8484]

It is noted that a DNS privacy service can also be provided over DNS-over-DTLS [RFC8094], however this is an Experimental specification and there are no known implementations at the time of writing.

It is also noted that DNS privacy service might be provided over IPSec, DNSCrypt or VPNs. However, use of these transports for DNS are not standardized and any discussion of best practice for providing such a service is out of scope for this document.

Whilst encryption of DNS traffic can protect against active injection this does not diminish the need for DNSSEC, see Section 5.1.4.
5.1.2. Authentication of DNS privacy services

[RFC6973] Threats:

- Surveillance:
  * Active attacks that can redirect traffic to rogue servers

[I-D.bortzmeyer-dprive-rfc7626-bis] Section 2.5.3.

Mitigations:

DNS privacy services should ensure clients can authenticate the server. Note that this, in effect, commits the DNS privacy service to a public identity users will trust.

When using DNS-over-TLS clients that select a ‘Strict Privacy’ usage profile [RFC8310] (to mitigate the threat of active attack on the client) require the ability to authenticate the DNS server. To enable this, DNS privacy services that offer DNS-over-TLS should provide credentials in the form of either X.509 certificates or SPKI pinsets.

When offering DoH [RFC8484], HTTPS requires authentication of the server as part of the protocol.

NOTE: At this time the reference to the TLS DNSSEC chain extension draft has been removed as it is no longer considered an active TLS WG document.

Optimizations:

DNS privacy services can also consider the following capabilities/options:

- As recommended in [RFC8310] providing DANE TLSA records for the nameserver
  * In particular, the service could provide TLSA records such that authenticating solely via the PKIX infrastructure can be avoided.

5.1.2.1. Certificate management

Anecdotal evidence to date highlights the management of certificates as one of the more challenging aspects for operators of traditional DNS resolvers that choose to additionally provide a DNS privacy service as management of such credentials is new to those DNS operators.
It is noted that SPKI pinset management is described in [RFC7858] but that key pinning mechanisms in general have fallen out of favor operationally for various reasons such as the logistical overhead of rolling keys.

DNS Privacy Threats:
- Invalid certificates, resulting in an unavailable service.
- Mis-identification of a server by a client e.g. typos in URLs or authentication domain names

Mitigations:
It is recommended that operators:
- Follow the guidance in Section 6.5 of [RFC7525] with regards to certificate revocation
- Choose a short, memorable authentication name for the service
- Automate the generation and publication of certificates
- Monitor certificates to prevent accidental expiration of certificates

5.1.3. Protocol recommendations

5.1.3.1. DNS-over-TLS

DNS Privacy Threats:
- Known attacks on TLS such as those described in [RFC7457]
- Traffic analysis, for example: Pitfalls of DNS Encryption [1]
- Potential for client tracking via transport identifiers
- Blocking of well known ports (e.g. 853 for DNS-over-TLS)

Mitigations:
In the case of DNS-over-TLS, TLS profiles from Section 9 and the Countermeasures to DNS Traffic Analysis from section 11.1 of [RFC8310] provide strong mitigations. This includes but is not limited to:
- Adhering to [RFC7525]
Implementing only (D)TLS 1.2 or later as specified in [RFC8310]

Implementing EDNS(0) Padding [RFC7830] using the guidelines in [RFC8467]

Clients should not be required to use TLS session resumption [RFC5077] or Domain Name System (DNS) Cookies [RFC7873].

A DNS-over-TLS privacy service on both port 853 and 443. This practice may not be possible if e.g. the operator deploys DoH on the same IP address.

Optimizations:

Concurrent processing of pipelined queries, returning responses as soon as available, potentially out of order as specified in [RFC7766]. This is often called ‘OOOR’ – out-of-order responses. (Providing processing performance similar to HTTP multiplexing)

Management of TLS connections to optimize performance for clients using either

  * [RFC7766] and EDNS(0) Keepalive [RFC7828] and/or
  * DNS Stateful Operations [I-D.ietf-dnsop-session-signal]

Additional options that providers may consider:

Offer a .onion [RFC7686] service endpoint

5.1.3.2. DoH

DNS Privacy Threats:

Known attacks on TLS such as those described in [RFC7457]

Traffic analysis, for example: DNS Privacy not so private: the traffic analysis perspective [2]

Potential for client tracking via transport identifiers

Mitigations:

Clients must be able to forego the use of HTTP Cookies [RFC6265] and still use the service
Clients should not be required to include any headers beyond the absolute minimum to obtain service from a DoH server. (See Section 6.1 of [I-D.ietf-httpbis-bcp56bis].)

5.1.4. DNSSEC

DNS Privacy Threats:

- Users may be directed to bogus IP addresses for e.g. websites where they might reveal personal information to attackers.

Mitigations:

- All DNS privacy services must offer a DNS privacy service that performs DNSSEC validation. In addition they must be able to provide the DNSSEC RRs to the client so that it can perform its own validation.

The addition of encryption to DNS does not remove the need for DNSSEC [RFC4033] - they are independent and fully compatible protocols, each solving different problems. The use of one does not diminish the need nor the usefulness of the other.

While the use of an authenticated and encrypted transport protects origin authentication and data integrity between a client and a DNS privacy service it provides no proof (for a non-validating client) that the data provided by the DNS privacy service was actually DNSSEC authenticated. As with cleartext DNS the user is still solely trusting the AD bit (if present) set by the resolver.

It should also be noted that the use of an encrypted transport for DNS actually solves many of the practical issues encountered by DNS validating clients e.g. interference by middleboxes with cleartext DNS payloads is completely avoided. In this sense a validating client that uses a DNS privacy service which supports DNSSEC has a far simpler task in terms of DNS Roadblock avoidance.

5.1.5. Availability

DNS Privacy Threats:

- A failed DNS privacy service could force the user to switch providers, fallback to cleartext or accept no DNS service for the outage.

Mitigations:
A DNS privacy service must be engineered for high availability. Particular care should be taken to protect DNS privacy services against denial-of-service attacks, as experience has shown that unavailability of DNS resolving because of attacks is a significant motivation for users to switch services. See, for example Section IV-C of Passive Observations of a Large DNS Service: 2.5 Years in the Life of Google [3].

Techniques such as those described in Section 10 of [RFC7766] can be of use to operators to defend against such attacks.

5.1.6. Service options

DNS Privacy Threats:

- Unfairly disadvantaging users of the privacy service with respect to the services available. This could force the user to switch providers, fallback to cleartext or accept no DNS service for the outage.

Mitigations:

A DNS privacy service should deliver the same level of service as offered on un-encrypted channels in terms of such options as filtering (or lack thereof), DNSSEC validation, etc.

5.1.7. Impact on Operators

DNS Privacy Threats:

- Increased use of encryption impacts operator ability to manage their network [RFC8404]

Many monitoring solutions for DNS traffic rely on the plain text nature of this traffic and work by intercepting traffic on the wire, either using a separate view on the connection between clients and the resolver, or as a separate process on the resolver system that inspects network traffic. Such solutions will no longer function when traffic between clients and resolvers is encrypted. There are, however, legitimate reasons for operators to inspect DNS traffic, e.g. to monitor for network security threats. Operators may therefore need to invest in alternative means of monitoring that relies on either the resolver software directly, or exporting DNS traffic from the resolver using e.g. dnstap [4].

Optimization:
When implementing alternative means for traffic monitoring, operators of a DNS privacy service should consider using privacy conscious means to do so (see, for example, the discussion on the use of Bloom Filters in the #documents appendix in this document).

5.1.8. Limitations of using a pure TLS proxy

DNS Privacy Threats:

- Limited ability to manage or monitor incoming connections using DNS specific techniques
- Misconfiguration of the target server could lead to data leakage if the proxy to target server path is not encrypted.

Optimization:

Some operators may choose to implement DNS-over-TLS using a TLS proxy (e.g. nginx [5], haproxy [6] or stunnel [7]) in front of a DNS nameserver because of proven robustness and capacity when handling large numbers of client connections, load balancing capabilities and good tooling. Currently, however, because such proxies typically have no specific handling of DNS as a protocol over TLS or DTLS using them can restrict traffic management at the proxy layer and at the DNS server. For example, all traffic received by a nameserver behind such a proxy will appear to originate from the proxy and DNS techniques such as ACLs, RRL or DNS64 will be hard or impossible to implement in the nameserver.

Operators may choose to use a DNS aware proxy such as dnsdist [8] which offer custom options (similar to that proposed in [I-D.bellis-dnsop-xpf]) to add source information to packets to address this shortcoming. It should be noted that such options potentially significantly increase the leaked information in the event of a misconfiguration.

5.2. Data at rest on the server

5.2.1. Data handling

[RFC6973] Threats:

- Surveillance
- Stored data compromise
- Correlation
o Identification
o Secondary use
o Disclosure

Other Threats

o Contravention of legal requirements not to process user data?

Mitigations:

The following are common activities for DNS service operators and in all cases should be minimized or completely avoided if possible for DNS privacy services. If data is retained it should be encrypted and either aggregated, pseudonymized or anonymized whenever possible. In general the principle of data minimization described in [RFC6973] should be applied.

- Transient data (e.g. that is used for real time monitoring and threat analysis which might be held only memory) should be retained for the shortest possible period deemed operationally feasible.
- The retention period of DNS traffic logs should be only those required to sustain operation of the service and, to the extent that such exists, meet regulatory requirements.
- DNS privacy services should not track users except for the particular purpose of detecting and remedying technically malicious (e.g. DoS) or anomalous use of the service.
- Data access should be minimized to only those personnel who require access to perform operational duties.

Optimizations:

- Consider use of full disk encryption for logs and data capture storage.

5.2.2. Data minimization of network traffic

Data minimization refers to collecting, using, disclosing, and storing the minimal data necessary to perform a task, and this can be achieved by removing or obfuscating privacy-sensitive information in network traffic logs. This is typically personal data, or data that can be used to link a record to an individual, but may also include
revealing other confidential information, for example on the structure of an internal corporate network.

The problem of effectively ensuring that DNS traffic logs contain no or minimal privacy-sensitive information is not one that currently has a generally agreed solution or any Standards to inform this discussion. This section presents and overview of current techniques to simply provide reference on the current status of this work.

Research into data minimization techniques (and particularly IP address pseudonymization/anonymization) was sparked in the late 1990s/early 2000s, partly driven by the desire to share significant corpuses of traffic captures for research purposes. Several techniques reflecting different requirements in this area and different performance/resource tradeoffs emerged over the course of the decade. Developments over the last decade have been both a blessing and a curse; the large increase in size between an IPv4 and an IPv6 address, for example, renders some techniques impractical, but also makes available a much larger amount of input entropy, the better to resist brute force re-identification attacks that have grown in practicality over the period.

Techniques employed may be broadly categorized as either anonymization or pseudonymization. The following discussion uses the definitions from [RFC6973] Section 3, with additional observations from van Dijkhuizen et al. [9]

- **Anonymization.** To enable anonymity of an individual, there must exist a set of individuals that appear to have the same attribute(s) as the individual. To the attacker or the observer, these individuals must appear indistinguishable from each other.

- **Pseudonymization.** The true identity is deterministically replaced with an alternate identity (a pseudonym). When the pseudonymization schema is known, the process can be reversed, so the original identity becomes known again.

In practice there is a fine line between the two; for example, how to categorize a deterministic algorithm for data minimization of IP addresses that produces a group of pseudonyms for a single given address.

5.2.3. IP address pseudonymization and anonymization methods

As [I-D.bortzmeyer-dprive-rfc7626-bis] makes clear, the big privacy risk in DNS is connecting DNS queries to an individual and the major vector for this in DNS traffic is the client IP address.
There is active discussion in the space of effective pseudonymization of IP addresses in DNS traffic logs, however there seems to be no single solution that is widely recognized as suitable for all or most use cases. There are also as yet no standards for this that are unencumbered by patents. The following table presents a high level comparison of various techniques employed or under development today and classifies them according to categorization of technique and other properties. The list of techniques includes the main techniques in current use, but does not claim to be comprehensive. Appendix B provides a more detailed survey of these techniques and definitions for the categories and properties listed below.

Figure showing comparison of IP address techniques (SVG) [10]

The choice of which method to use for a particular application will depend on the requirements of that application and consideration of the threat analysis of the particular situation.

For example, a common goal is that distributed packet captures must be in an existing data format such as PCAP [pcap] or C-DNS [I-D.ietf-dnsop-dns-capture-format] that can be used as input to existing analysis tools. In that case, use of a format-preserving technique is essential. This, though, is not cost-free - several authors (e.g. Brenker & Arnes [11]) have observed that, as the entropy in an IPv4 address is limited, given a de-identified log from a target, if an attacker is capable of ensuring packets are captured by the target and the attacker can send forged traffic with arbitrary source and destination addresses to that target, any format-preserving pseudonymization is vulnerable to an attack along the lines of a cryptographic chosen plaintext attack.

5.2.4. Pseudonymization, anonymization or discarding of other correlation data

DNS Privacy Threats:

- IP TTL/Hoplimit can be used to fingerprint client OS
- TLS version/Cipher suite combinations can be used to fingerprint the client application or TLS library
- Tracking of TCP sessions
- Tracking of TLS sessions and session resumption mechanisms
- Resolvers _might_ receive client identifiers e.g. MAC addresses in EDNS(0) options - some CPE devices are known to add them.
5.3. Data sent onwards from the server

In this section we consider both data sent on the wire in upstream queries and data shared with third parties.

5.3.1. Protocol recommendations

As specified in [RFC8310] for DNS-over-TLS but applicable to any DNS Privacy services the server should:

- Implement QNAME minimization [RFC7816]
- Honor a SOURCE PREFIX-LENGTH set to 0 in a query containing the EDNS(0) Client Subnet (ECS) option and not send an ECS option in upstream queries.

Optimizations:

- The server should either
  * not use the ECS option in upstream queries at all, or
offer alternative services, one that sends ECS and one that does not.

If operators do offer a service that sends the ECS options upstream they should use the shortest prefix that is operationally feasible (NOTE: the authors believe they will be able to add a reference for advice here soon) and ideally use a policy of whitelisting upstream servers to send ECS to in order to minimize data leakage. Operators should make clear in any policy statement what prefix length they actually send and the specific policy used.

Whitelisting has the benefit that not only does the operator know which upstream servers can use ECS but also allows the operator to decide which upstream servers apply privacy policies that the operator is happy with. However some operators consider whitelisting to incur significant operational overhead compared to dynamic detection of ECS on authoritative servers.

Additional options:

- Aggressive Use of DNSSEC-Validated Cache [RFC8198] to reduce the number of queries to authoritative servers to increase privacy.
- Run a copy of the root zone on loopback [RFC7706] to avoid making queries to the root servers that might leak information.

5.3.2. Client query obfuscation

Additional options:

Since queries from recursive resolvers to authoritative servers are performed using cleartext (at the time of writing), resolver services need to consider the extent to which they may be directly leaking information about their client community via these upstream queries and what they can do to mitigate this further. Note, that even when all the relevant techniques described above are employed there may still be attacks possible, e.g. [Pitfalls-of-DNS-Encryption]. For example, a resolver with a very small community of users risks exposing data in this way and OUGHT obfuscate this traffic by mixing it with ‘generated’ traffic to make client characterization harder. The resolver could also employ aggressive pre-fetch techniques as a further measure to counter traffic analysis.

At the time of writing there are no standardized or widely recognized techniques to perform such obfuscation or bulk pre-fetches.

Another technique that particularly small operators may consider is forwarding local traffic to a larger resolver (with a privacy policy
that aligns with their own practices) over an encrypted protocol so that the upstream queries are obfuscated among those of the large resolver.

5.3.3. Data sharing

[RFC6973] Threats:

- Surveillance
- Stored data compromise
- Correlation
- Identification
- Secondary use
- Disclosure

DNS Privacy Threats:

- Contravention of legal requirements not to process user data

Mitigations:

Operators should not provide identifiable data to third-parties without explicit consent from clients (we take the stance here that simply using the resolution service itself does not constitute consent).

Even when consent is granted operators should employ data minimization techniques such as those described in Section 5.2.1 if data is shared with third-parties.

Operators should consider including specific guidelines for the collection of aggregated and/or anonymized data for research purposes, within or outside of their own organization. This can benefit not only the operator (through inclusion in novel research) but also the wider Internet community. See SURFnet’s policy [13] on data sharing for research as an example.

6. DNS privacy policy and practice statement
6.1. Recommended contents of a DPPPS

6.1.1. Policy

1. Make an explicit statement that IP addresses are treated as PII

2. State if IP addresses are being logged

3. Specify clearly what data (including whether it is aggregated, pseudonymized or anonymized and the conditions of data transfer) is:
   * Collected and retained by the operator, and for what period it is retained
   * Shared with partners
   * Shared, sold or rented to third-parties

4. Specify any exceptions to the above, for example technically malicious or anomalous behavior

5. Declare any partners, third-party affiliations or sources of funding

6. Whether user DNS data is correlated or combined with any other personal information held by the operator

7. Result filtering. This section should explain whether the operator filters, edits or alters in any way the replies that it receives from the authoritative servers for each DNS zone, before forwarding them to the clients. For each category listed below, the operator should also specify how the filtering lists are created and managed, whether it employs any third-party sources for such lists, and which ones.
   * Specify if any replies are being filtered out or altered for network and computer security reasons (e.g. preventing connections to malware-spreading websites or botnet control servers)
   * Specify if any replies are being filtered out or altered for mandatory legal reasons, due to applicable legislation or binding orders by courts and other public authorities
   * Specify if any replies are being filtered out or altered for voluntary legal reasons, due to an internal policy by the operator aiming at reducing potential legal risks
Specify if any replies are being filtered out or altered for any other reason, including commercial ones

6.1.2. Practice

This section should explain the current operational practices of the service.

1. Specify any temporary or permanent deviations from the policy for operational reasons

2. With reference to section Section 5 provide specific details of which capabilities are provided on which client facing addresses and ports

3. Specify the authentication name to be used (if any) and if TLSA records are published (including options used in the TLSA records)

4. Specify the SPKI pinsets to be used (if any) and policy for rolling keys

5. Provide contact/support information for the service

6. Jurisdiction. This section should communicate the applicable jurisdictions and law enforcement regimes under which the service is being provided.

   * Specify the entity or entities that will control the data and be responsible for their treatment, and their legal place of business

   * Specify, either directly or by pointing to the applicable privacy policy, the relevant privacy laws that apply to the treatment of the data, the rights that users enjoy in regard to their own personal information that is treated by the service, and how they can contact the operator to enforce them

   * Specify the countries in which the servers handling the DNS requests and the data are located (if the operator applies a geolocation policy so that requests from certain countries are only served by certain servers, this should be specified as well)

   * Specify whether the operator has any agreement in place with law enforcement agencies, or other public and private parties dealing with security and intelligence, to give them access to the servers and/or to the data
7. Describe how consent is obtained from the user of the DNS privacy service differentiating
   * Uninformed users for whom this trust relationship is implicit
   * Privacy-conscious users, that make an explicit trust choice
   (this may prove relevant in the context of e.g. the GDPR as it relates to consent)

6.2. Current policy and privacy statements

A tabular comparison of existing policy and privacy statements from various DNS Privacy service operators based on the proposed DPPPS structure can be found on dnsprivacy.org [14].

We note that the existing set of policies vary widely in style, content and detail and it is not uncommon for the full text for a given operator to equate to more than 10 pages of moderate font sized A4 text. It is a non-trivial task today for a user to extract a meaningful overview of the different services on offer.

It is also noted that Mozilla have published a Security/DoH-resolver policy [15], which describes the minimum set of policy requirements that a party must satisfy to be considered as a potential partner for Mozilla’s Trusted Recursive Resolver (TRR) program.

6.3. Enforcement/accountability

Transparency reports may help with building user trust that operators adhere to their policies and practices.

Independent monitoring or analysis could be performed where possible of:
   o ECS, QNAME minimization, EDNS(0) padding, etc.
   o Filtering
   o Uptime

This is by analogy with e.g. several TLS or website analysis tools that are currently available e.g. SSL Labs [16] or Internet.nl [17].

Additionally operators could choose to engage the services of a third party auditor to verify their compliance with their published DPPPS.
7. IANA considerations

None

8. Security considerations

Security considerations for DNS-over-TCP are given in [RFC7766], many of which are generally applicable to session based DNS.

TODO: e.g. New issues for DoS defence, server admin policies

9. Acknowledgements

Many thanks to Amelia Andersdotter for a very thorough review of the first draft of this document. Thanks to John Todd for discussions on this topic, and to Stephane Bortzmeyer, Puneet Sood and Vittorio Bertola for review. Thanks to Daniel Kahn Gillmor, Barry Green, Paul Hoffman, Dan York, John Reed, Lorenzo Colitti for comments at the mic. Thanks to Loganaden Velvindron for useful updates to the text.

Sara Dickinson thanks the Open Technology Fund for a grant to support the work on this document.

10. Contributors

The below individuals contributed significantly to the document:

John Dickinson
Sinodun Internet Technologies
Magdalen Centre
Oxford Science Park
Oxford OX4 4GA
United Kingdom

Jim Hague
Sinodun Internet Technologies
Magdalen Centre
Oxford Science Park
Oxford OX4 4GA
United Kingdom

11. Changelog

draft-ietf-dprive-bcp-op-03

- Add paragraph about operational impact
o Move DNSSEC requirement out of the Appendix into main text as a privacy threat that should be mitigated

o Add TLS version/Cipher suite as tracking threat

o Add reference to Mozilla TRR policy

o Remove several TODOs and QUESTIONS.

draft-ietf-dprive-bcp-op-02

o Change ‘open resolver’ for ‘public resolver’

o Minor editorial changes

o Remove recommendation to run a separate TLS 1.3 service

o Move TLSA to purely a optimisation in Section 5.2.1

o Update reference on minimal DoH headers.

o Add reference on user switching provider after service issues in Section 5.1.4

o Add text in Section 5.1.6 on impact on operators.

o Add text on additional threat to TLS proxy use (Section 5.1.7)

o Add reference in Section 5.3.1 on example policies.

draft-ietf-dprive-bcp-op-01

o Many minor editorial fixes

o Update DoH reference to RFC8484 and add more text on DoH

o Split threat descriptions into ones directly referencing RFC6973 and other DNS Privacy threats

o Improve threat descriptions throughout

o Remove reference to the DNSSEC TLS Chain Extension draft until new version submitted.

o Clarify use of whitelisting for ECS

o Re-structure the DPPPS, add Result filtering section.
Internet-Draft     DNS Privacy Service Recommendations     July 2019

- Remove the direct inclusion of privacy policy comparison, now just reference dnsprivacy.org and an example of such work.

- Add an appendix briefly discussing DNSSEC

- Update affiliation of 1 author
draft-ietf-dprive-bcp-op-00

- Initial commit of re-named document after adoption to replace draft-dickinson-dprive-bcp-op-01

12. References

12.1. Normative References

[I-D.ietf-dnsop-session-signal]


12.2. Informative References

[I-D.bellis-dnsop-xpf]

[I-D.bortzmeyer-dprive-rfc7626-bis]

[I-D.ietf-dnsop-dns-capture-format]

[I-D.ietf-dnsop-dns-tcp-requirements]

[I-D.ietf-dnsop-terminology-bis]

[I-D.ietf-httpbis-bcp56bis]
Nottingham, M., "Building Protocols with HTTP", draft-ietf-httpbis-bcp56bis-08 (work in progress), November 2018.

[pcap]

[ Pitfalls-of-DNS-Encryption]
12.3. URIs


[8] https://dnssdist.org
[9] https://doi.org/10.1145/3182660
   draft-00/ip_techniques_table.svg
[11] https://pdfs.semanticscholar.org/7b34/12c951cebe71cd2cddac5fda16
   4fb2138a44.pdf
[12] https://kb.isc.org/docs/aa-00482
   Comparison+of+policy+and+privacy+statements
[16] https://www.ssllabs.com/ssltest/
[17] https://internet.nl
[18] https://support.google.com/analytics/answer/2763052?hl=en
[19] https://www.conversionworks.co.uk/blog/2017/05/19/anonymize-ip-
   geo-impact-test/
   anon.pdf
[23] https://www.cc.gatech.edu/computing/Telecomm/projects/cryptopan/
Appendix A.  Documents

This section provides an overview of some DNS privacy related
documents, however, this is neither an exhaustive list nor a
definitive statement on the characteristic of the document.

A.1.  Potential increases in DNS privacy

These documents are limited in scope to communications between stub
clients and recursive resolvers:

- 'Specification for DNS over Transport Layer Security (TLS)' [RFC7858],
  referred to here as 'DNS-over-TLS'.

- 'DNS over Datagram Transport Layer Security (DTLS)' [RFC8094],
  referred to here as 'DNS-over-DTLS'. Note that this document has
  the Category of Experimental.

- 'DNS Queries over HTTPS (DoH)' [RFC8484] referred to here as DoH.

- 'Usage Profiles for DNS over TLS and DNS over DTLS' [RFC8310]

- 'The EDNS(0) Padding Option' [RFC7830] and 'Padding Policy for
  EDNS(0)' [RFC8467]

These documents apply to recursive to authoritative DNS but are
relevant when considering the operation of a recursive server:

- 'DNS Query Name minimization to Improve Privacy' [RFC7816]
  referred to here as 'QNAME minimization'

A.2.  Potential decreases in DNS privacy

These documents relate to functionality that could provide increased
tracking of user activity as a side effect:

- 'Client Subnet in DNS Queries' [RFC7871]
A.3. Related operational documents

- ‘DNS Transport over TCP - Implementation Requirements’ [RFC7766]
- ‘Operational requirements for DNS-over-TCP’ [I-D.ietf-dnsop-dns-tcp-requirements]
- ‘The edns-tcp-keepalive EDNS0 Option’ [RFC7828]
- ‘DNS Stateful Operations’ [I-D.ietf-dnsop-session-signal]

Appendix B. IP address techniques

Data minimization methods may be categorized by the processing used and the properties of their outputs. The following builds on the categorization employed in [RFC6235]:

- Format-preserving. Normally when encrypting, the original data length and patterns in the data should be hidden from an attacker. Some applications of de-identification, such as network capture de-identification, require that the de-identified data is of the same form as the original data, to allow the data to be parsed in the same way as the original.

- Prefix preservation. Values such as IP addresses and MAC addresses contain prefix information that can be valuable in analysis, e.g. manufacturer ID in MAC addresses, subnet in IP addresses. Prefix preservation ensures that prefixes are de-identified consistently; e.g. if two IP addresses are from the same subnet, a prefix preserving de-identification will ensure that their de-identified counterparts will also share a subnet. Prefix preservation may be fixed (i.e. based on a user selected prefix length identified in advance to be preserved) or general.
o Replacement. A one-to-one replacement of a field to a new value of the same type, for example using a regular expression.

o Filtering. Removing (and thus truncating) or replacing data in a field. Field data can be overwritten, often with zeros, either partially (grey marking) or completely (black marking).

o Generalization. Data is replaced by more general data with reduced specificity. One example would be to replace all TCP/UDP port numbers with one of two fixed values indicating whether the original port was ephemeral (>=1024) or non-ephemeral (>1024). Another example, precision degradation, reduces the accuracy of e.g. a numeric value or a timestamp.

o Enumeration. With data from a well-ordered set, replace the first data item data using a random initial value and then allocate ordered values for subsequent data items. When used with timestamp data, this preserves ordering but loses precision and distance.

o Reordering/shuffling. Preserving the original data, but rearranging its order, often in a random manner.

o Random substitution. As replacement, but using randomly generated replacement values.

o Cryptographic permutation. Using a permutation function, such as a hash function or cryptographic block cipher, to generate a replacement de-identified value.

B.1. Google Analytics non-prefix filtering

Since May 2010, Google Analytics has provided a facility [18] that allows website owners to request that all their users IP addresses are anonymized within Google Analytics processing. This very basic anonymization simply sets to zero the least significant 8 bits of IPv4 addresses, and the least significant 80 bits of IPv6 addresses. The level of anonymization this produces is perhaps questionable. There are some analysis results [19] which suggest that the impact of this on reducing the accuracy of determining the user’s location from their IP address is less than might be hoped; the average discrepancy in identification of the user city for UK users is no more than 17%.

Anonymization: Format-preserving, Filtering (grey marking).
B.2. dnswasher

Since 2006, PowerDNS have included a de-identification tool dnswasher [20] with their PowerDNS product. This is a PCAP filter that performs a one-to-one mapping of end user IP addresses with an anonymized address. A table of user IP addresses and their de-identified counterparts is kept; the first IPv4 user addresses is translated to 0.0.0.1, the second to 0.0.0.2 and so on. The de-identified address therefore depends on the order that addresses arrive in the input, and running over a large amount of data the address translation tables can grow to a significant size.

Anonymization: Format-preserving, Enumeration.

B.3. Prefix-preserving map

Used in TCPdpriv [21], this algorithm stores a set of original and anonymised IP address pairs. When a new IP address arrives, it is compared with previous addresses to determine the longest prefix match. The new address is anonymized by using the same prefix, with the remainder of the address anonymized with a random value. The use of a random value means that TCPdpriv is not deterministic; different anonymized values will be generated on each run. The need to store previous addresses means that TCPdpriv has significant and unbounded memory requirements, and because of the need to allocated anonymized addresses sequentially cannot be used in parallel processing.

Anonymization: Format-preserving, prefix preservation (general).

B.4. Cryptographic Prefix-Preserving Pseudonymisation

Cryptographic prefix-preserving pseudonymisation was originally proposed as an improvement to the prefix-preserving map implemented in TCPdpriv, described in Xu et al. [22] and implemented in the Crypto-PAn tool [23]. Crypto-PAn is now frequently used as an acronym for the algorithm. Initially it was described for IPv4 addresses only; extension for IPv6 addresses was proposed in Harvan & Schoenwaelder [24] and implemented in snmpdump. This uses a cryptographic algorithm rather than a random value, and thus pseudonymity is determined uniquely by the encryption key, and is deterministic. It requires a separate AES encryption for each output bit, so has a non-trivial calculation overhead. This can be mitigated to some extent (for IPv4, at least) by pre-calculating results for some number of prefix bits.

Pseudonymization: Format-preserving, prefix preservation (general).
B.5. Top-hash Subtree-replicated Anonymisation

Proposed in Ramaswamy & Wolf [25], Top-hash Subtree-replicated Anonymisation (TSA) originated in response to the requirement for faster processing than Crypto-PAn. It used hashing for the most significant byte of an IPv4 address, and a pre-calculated binary tree structure for the remainder of the address. To save memory space, replication is used within the tree structure, reducing the size of the pre-calculated structures to a few Mb for IPv4 addresses. Address pseudonymization is done via hash and table lookup, and so requires minimal computation. However, due to the much increased address space for IPv6, TSA is not memory efficient for IPv6.

Pseudonymization: Format-preserving, prefix preservation (general).

B.6. ipcipher

A recently-released proposal from PowerDNS [26], ipcipher [27] is a simple pseudonymization technique for IPv4 and IPv6 addresses. IPv6 addresses are encrypted directly with AES-128 using a key (which may be derived from a passphrase). IPv4 addresses are similarly encrypted, but using a recently proposed encryption ipcrypt [28] suitable for 32bit block lengths. However, the author of ipcrypt has since indicated [29] that it has low security, and further analysis has revealed it is vulnerable to attack.

Pseudonymization: Format-preserving, cryptographic permutation.

B.7. Bloom filters

van Rijswijk-Deij et al. [30] have recently described work using Bloom filters to categorize query traffic and record the traffic as the state of multiple filters. The goal of this work is to allow operators to identify so-called Indicators of Compromise (IOCs) originating from specific subnets without storing information about, or be able to monitor the DNS queries of an individual user. By using a Bloom filter, it is possible to determine with a high probability if, for example, a particular query was made, but the set of queries made cannot be recovered from the filter. Similarly, by mixing queries from a sufficient number of users in a single filter, it becomes practically impossible to determine if a particular user performed a particular query. Large numbers of queries can be tracked in a memory-efficient way. As filter status is stored, this approach cannot be used to regenerate traffic, and so cannot be used with tools used to process live traffic.

Anonymized: Generalization.
Authors’ Addresses

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

Email: sara@sinodun.com

Benno J. Overeinder
NLnet Labs
Science Park 400
Amsterdam 1098 XH
The Netherlands

Email: benno@nlnetLabs.nl

Roland M. van Rijswijk-Deij
NLnet Labs
Science Park 400
Amsterdam 1098 XH
The Netherlands

Email: roland@nlnetLabs.nl

Allison Mankin
Salesforce

Email: allison.mankin@gmail.com
Abstract

This document describes the privacy issues associated with the use of the DNS by Internet users. It is intended to be an analysis of the present situation and does not prescribe solutions. This document obsoletes RFC 7626.

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.
Section 1.  Introduction

This document is an analysis of the DNS privacy issues, in the spirit of Section 8 of [RFC6973].

The Domain Name System is specified in [RFC1034], [RFC1035], and many later RFCs, which have never been consolidated. It is one of the most important infrastructure components of the Internet and often ignored or misunderstood by Internet users (and even by many professionals). Almost every activity on the Internet starts with a DNS query (and often several). Its use has many privacy implications and this is an attempt at a comprehensive and accurate list.

Let us begin with a simplified reminder of how the DNS works. (See also [RFC8499]) A client, the stub resolver, issues a DNS query to a server, called the recursive resolver (also called caching resolver or full resolver or recursive name server). Let’s use the query "What are the AAAA records for www.example.com?" as an example. AAAA

---

<table>
<thead>
<tr>
<th>Table of Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.  Introduction ........................................ 2</td>
</tr>
<tr>
<td>2.  Risks .................................................. 4</td>
</tr>
<tr>
<td>2.1.  The Alleged Public Nature of DNS Data ............ 5</td>
</tr>
<tr>
<td>2.2.  Data in the DNS Request .......................... 5</td>
</tr>
<tr>
<td>2.2.1.  Data in the DNS payload ........................... 7</td>
</tr>
<tr>
<td>2.3.  Cache Snooping ...................................... 7</td>
</tr>
<tr>
<td>2.4.  On the Wire .......................................... 7</td>
</tr>
<tr>
<td>2.4.1.  Unencrypted Transports ........................... 7</td>
</tr>
<tr>
<td>2.4.2.  Encrypted Transports .............................. 9</td>
</tr>
<tr>
<td>2.5.  In the Servers ................................. 10</td>
</tr>
<tr>
<td>2.5.1.  In the Recursive Resolvers ....................... 10</td>
</tr>
<tr>
<td>2.5.2.  In the Authoritative Name Servers ............... 12</td>
</tr>
<tr>
<td>2.5.3.  Rogue Servers .................................... 13</td>
</tr>
<tr>
<td>2.5.4.  Authentication of servers ........................ 13</td>
</tr>
<tr>
<td>2.5.5.  Blocking of services ............................. 14</td>
</tr>
<tr>
<td>2.6.  Re-identification and Other Inferences ............ 14</td>
</tr>
<tr>
<td>2.7.  More Information .................................... 15</td>
</tr>
<tr>
<td>3.  Actual &quot;Attacks&quot; ...................................... 15</td>
</tr>
<tr>
<td>4.  Legalities ............................................ 15</td>
</tr>
<tr>
<td>5.  Security Considerations .............................. 16</td>
</tr>
<tr>
<td>6.  Acknowledgments ........................................ 16</td>
</tr>
<tr>
<td>7.  Changelog ............................................... 16</td>
</tr>
<tr>
<td>8.  References ............................................. 17</td>
</tr>
<tr>
<td>8.1.  Normative References ................................ 17</td>
</tr>
<tr>
<td>8.2.  Informative References ............................. 17</td>
</tr>
<tr>
<td>8.3.  URIs .................................................. 22</td>
</tr>
<tr>
<td>Authors’ Addresses ....................................... 22</td>
</tr>
</tbody>
</table>
is the QTYPE (Query Type), and www.example.com is the QNAME (Query Name). (The description that follows assumes a cold cache, for instance, because the server just started.) The recursive resolver will first query the root name servers. In most cases, the root name servers will send a referral. In this example, the referral will be to the .com name servers. The resolver repeats the query to one of the .com name servers. The .com name servers, in turn, will refer to the example.com name servers. The example.com name server will then return the answer. The root name servers, the name servers of .com, and the name servers of example.com are called authoritative name servers. It is important, when analyzing the privacy issues, to remember that the question asked to all these name servers is always the original question, not a derived question. The question sent to the root name servers is "What are the AAAA records for www.example.com?", not "What are the name servers of .com?". By repeating the full question, instead of just the relevant part of the question to the next in line, the DNS provides more information than necessary to the name server.

Because DNS relies on caching heavily, the algorithm described just above is actually a bit more complicated, and not all questions are sent to the authoritative name servers. If a few seconds later the stub resolver asks the recursive resolver, "What are the SRV records of _xmpp-server._tcp.example.com?", the recursive resolver will remember that it knows the name servers of example.com and will just query them, bypassing the root and .com. Because there is typically no caching in the stub resolver, the recursive resolver, unlike the authoritative servers, sees all the DNS traffic. (Applications, like web browsers, may have some form of caching that does not follow DNS rules, for instance, because it may ignore the TTL. So, the recursive resolver does not see all the name resolution activity.)

It should be noted that DNS recursive resolvers sometimes forward requests to other recursive resolvers, typically bigger machines, with a larger and more shared cache (and the query hierarchy can be even deeper, with more than two levels of recursive resolvers). From the point of view of privacy, these forwarders are like resolvers, except that they do not see all of the requests being made (due to caching in the first resolver).

At the time of writing, almost all this DNS traffic is currently sent in clear (unencrypted). However there is increasing deployment of DNS-over-TLS (DoT) [RFC7858] and DNS-over-HTTPS (DoH) [RFC8484], particularly in mobile devices, browsers and by providers of anycast recursive DNS resolution services. There are a few cases where there is some alternative channel encryption, for instance, in an IPsec VPN, at least between the stub resolver and the resolver.
Today, almost all DNS queries are sent over UDP [thomas-ditl-tcp]. This has practical consequences when considering encryption of the traffic as a possible privacy technique. Some encryption solutions are only designed for TCP, not UDP.

Another important point to keep in mind when analyzing the privacy issues of DNS is the fact that DNS requests received by a server are triggered by different reasons. Let’s assume an eavesdropper wants to know which web page is viewed by a user. For a typical web page, there are three sorts of DNS requests being issued:

- Primary request: this is the domain name in the URL that the user typed, selected from a bookmark, or chose by clicking on an hyperlink. Presumably, this is what is of interest for the eavesdropper.

- Secondary requests: these are the additional requests performed by the user agent (here, the web browser) without any direct involvement or knowledge of the user. For the Web, they are triggered by embedded content, Cascading Style Sheets (CSS), JavaScript code, embedded images, etc. In some cases, there can be dozens of domain names in different contexts on a single web page.

- Tertiary requests: these are the additional requests performed by the DNS system itself. For instance, if the answer to a query is a referral to a set of name servers, and the glue records are not returned, the resolver will have to do additional requests to turn the name servers’ names into IP addresses. Similarly, even if glue records are returned, a careful recursive server will do tertiary requests to verify the IP addresses of those records.

It can be noted also that, in the case of a typical web browser, more DNS requests than strictly necessary are sent, for instance, to prefetch resources that the user may query later or when autocompleting the URL in the address bar. Both are a big privacy concern since they may leak information even about non-explicit actions. For instance, just reading a local HTML page, even without selecting the hyperlinks, may trigger DNS requests.

For privacy-related terms, we will use the terminology from [RFC6973].

2. Risks

This document focuses mostly on the study of privacy risks for the end user (the one performing DNS requests). We consider the risks of pervasive surveillance [RFC7258] as well as risks coming from a more
focused surveillance. Privacy risks for the holder of a zone (the risk that someone gets the data) are discussed in [RFC5936] and [RFC5155]. Non-privacy risks (such as cache poisoning) are out of scope.

2.1. The Alleged Public Nature of DNS Data

It has long been claimed that "the data in the DNS is public". While this sentence makes sense for an Internet-wide lookup system, there are multiple facets to the data and metadata involved that deserve a more detailed look. First, access control lists and private namespaces notwithstanding, the DNS operates under the assumption that public-facing authoritative name servers will respond to "usual" DNS queries for any zone they are authoritative for without further authentication or authorization of the client (resolver). Due to the lack of search capabilities, only a given QNAME will reveal the resource records associated with that name (or that name’s non-existence). In other words: one needs to know what to ask for, in order to receive a response. The zone transfer QTYPE [RFC5936] is often blocked or restricted to authenticated/authorized access to enforce this difference (and maybe for other reasons).

Another differentiation to be considered is between the DNS data itself and a particular transaction (i.e., a DNS name lookup). DNS data and the results of a DNS query are public, within the boundaries described above, and may not have any confidentiality requirements. However, the same is not true of a single transaction or a sequence of transactions; that transaction is not / should not be public. A typical example from outside the DNS world is: the web site of Alcoholics Anonymous is public; the fact that you visit it should not be.

2.2. Data in the DNS Request

The DNS request includes many fields, but two of them seem particularly relevant for the privacy issues: the QNAME and the source IP address. "source IP address" is used in a loose sense of "source IP address + maybe source port", because the port is also in the request and can be used to differentiate between several users sharing an IP address (behind a Carrier-Grade NAT (CGN), for instance [RFC6269]).

The QNAME is the full name sent by the user. It gives information about what the user does ("What are the MX records of example.net?" means he probably wants to send email to someone at example.net, which may be a domain used by only a few persons and is therefore very revealing about communication relationships). Some QNAMEs are more sensitive than others. For instance, querying the A record of a
well-known web statistics domain reveals very little (everybody visits web sites that use this analytics service), but querying the A record of www.verybad.example where verybad.example is the domain of an organization that some people find offensive or objectionable may create more problems for the user. Also, sometimes, the QNAME embeds the software one uses, which could be a privacy issue. For instance, _ldap._tcp.Default-First-Site-Name._sites.gc._msdcs.example.org. There are also some BitTorrent clients that query an SRV record for _bittorrent-tracker._tcp.domain.example.

Another important thing about the privacy of the QNAME is the future usages. Today, the lack of privacy is an obstacle to putting potentially sensitive or personally identifiable data in the DNS. At the moment, your DNS traffic might reveal that you are doing email but not with whom. If your Mail User Agent (MUA) starts looking up Pretty Good Privacy (PGP) keys in the DNS [RFC7929], then privacy becomes a lot more important. And email is just an example; there would be other really interesting uses for a more privacy-friendly DNS.

For the communication between the stub resolver and the recursive resolver, the source IP address is the address of the user’s machine. Therefore, all the issues and warnings about collection of IP addresses apply here. For the communication between the recursive resolver and the authoritative name servers, the source IP address has a different meaning; it does not have the same status as the source address in an HTTP connection. It is now the IP address of the recursive resolver that, in a way, "hides" the real user. However, hiding does not always work. Sometimes EDNS(0) Client subnet [RFC7871] is used (see its privacy analysis in [denis-edns-client-subnet]). Sometimes the end user has a personal recursive resolver on her machine. In both cases, the IP address is as sensitive as it is for HTTP [sidn-entrada].

A note about IP addresses: there is currently no IETF document that describes in detail all the privacy issues around IP addressing. In the meantime, the discussion here is intended to include both IPv4 and IPv6 source addresses. For a number of reasons, their assignment and utilization characteristics are different, which may have implications for details of information leakage associated with the collection of source addresses. (For example, a specific IPv6 source address seen on the public Internet is less likely than an IPv4 address to originate behind a CGN or other NAT.) However, for both IPv4 and IPv6 addresses, it’s important to note that source addresses are propagated with queries and comprise metadata about the host, user, or application that originated them.
2.2.1. Data in the DNS payload

At the time of writing there are no standardized client identifiers contained in the DNS payload itself (ECS [RFC7871] while widely used is only of Category Informational).

DNS Cookies [RFC7873] are a lightweight DNS transaction security mechanism that provides limited protection against a variety of increasingly common denial-of-service and amplification/forgery or cache poisoning attacks by off-path attackers. It is noted, however, that they are designed to just verify IP addresses (and should change once a client’s IP address changes), they are not designed to actively track users (like HTTP cookies).

There are anecdotal accounts of MAC addresses [1] and even user names being inserted in non-standard EDNS(0) options for stub to resolver communications to support proprietary functionality implemented at the resolver (e.g. parental filtering).

2.3. Cache Snooping

The content of recursive resolvers’ caches can reveal data about the clients using it (the privacy risks depend on the number of clients). This information can sometimes be examined by sending DNS queries with RD=0 to inspect cache content, particularly looking at the DNS TTLs [grangeia.snooping]. Since this also is a reconnaissance technique for subsequent cache poisoning attacks, some counter measures have already been developed and deployed.

2.4. On the Wire

2.4.1. Unencrypted Transports

For unencrypted transports, DNS traffic can be seen by an eavesdropper like any other traffic. (DNSSEC, specified in [RFC4033], explicitly excludes confidentiality from its goals.) So, if an initiator starts an HTTPS communication with a recipient, while the HTTP traffic will be encrypted, the DNS exchange prior to it will not be. When other protocols will become more and more privacy-aware and secured against surveillance (e.g. [RFC8446], [I-D.ietf-quic-transport]), the use of unencrypted transports for DNS may become "the weakest link" in privacy. It is noted that at the time of writing there is on-going work attempting to encrypt the SNI in the TLS handshake [I-D.ietf-tls-sni-encryption].

An important specificity of the DNS traffic is that it may take a different path than the communication between the initiator and the recipient. For instance, an eavesdropper may be unable to tap the
wire between the initiator and the recipient but may have access to
the wire going to the recursive resolver, or to the authoritative
name servers.

The best place to tap, from an eavesdropper’s point of view, is
clearly between the stub resolvers and the recursive resolvers,
because traffic is not limited by DNS caching.

The attack surface between the stub resolver and the rest of the
world can vary widely depending upon how the end user’s computer is
configured. By order of increasing attack surface:

The recursive resolver can be on the end user’s computer. In
(currently) a small number of cases, individuals may choose to
operate their own DNS resolver on their local machine. In this
case, the attack surface for the connection between the stub
resolver and the caching resolver is limited to that single
machine.

The recursive resolver may be at the local network edge. For
many/most enterprise networks and for some residential users, the
caching resolver may exist on a server at the edge of the local
network. In this case, the attack surface is the local network.
Note that in large enterprise networks, the DNS resolver may not
be located at the edge of the local network but rather at the edge
of the overall enterprise network. In this case, the enterprise
network could be thought of as similar to the Internet Access
Provider (IAP) network referenced below.

The recursive resolver can be in the IAP premises. For most
residential users and potentially other networks, the typical case
is for the end user’s computer to be configured (typically
automatically through DHCP) with the addresses of the DNS
recursive resolvers at the IAP. The attack surface for on-the-
wire attacks is therefore from the end-user system across the
local network and across the IAP network to the IAP’s recursive
resolvers.

The recursive resolver can be a public DNS service. Some machines
may be configured to use public DNS resolvers such as those
operated today by Google Public DNS or OpenDNS. The end user may
have configured their machine to use these DNS recursive resolvers
themselves -- or their IAP may have chosen to use the public DNS
resolvers rather than operating their own resolvers. In this
case, the attack surface is the entire public Internet between the
end user’s connection and the public DNS service.
2.4.2. Encrypted Transports

The use of encrypted transports directly mitigates passive surveillance of the DNS payload, however there are still some privacy attacks possible.

These are cases where user identification, fingerprinting or correlations may be possible due to the use of certain transport layers or clear text/observable features. These issues are not specific to DNS, but DNS traffic is susceptible to these attacks when using specific transports.

There are some general examples, for example, certain studies have highlighted that IP TTL or TCP Window sizes os-fingerprint [2] values can be used to fingerprint client OS’s or that various techniques can be used to de-NAT DNS queries dns-de-nat [3].

The use of clear text transport options to decrease latency may also identify a user e.g. using TCP Fast Open [RFC7413].

More specifically, (since the deployment of encrypted transports is not widespread at the time of writing) users wishing to use encrypted transports for DNS may in practice be limited in the resolver services available. Given this, the choice of a user to configure a single resolver (or a fixed set of resolvers) and an encrypted transport to use in all network environments can actually serve to identify the user as one that desires privacy and can provide an added mechanism to track them as they move across network environments.

Users of encrypted transports are also highly likely to re-use sessions for multiple DNS queries to optimize performance (e.g. via DNS pipelining or HTTPS multiplexing). Certain configuration options for encrypted transports could also in principle fingerprint a user, for example session resumption, the maximum number of messages to send or a maximum connection time before closing a connections and re-opening.

Whilst there are known attacks on older versions of TLS the most recent recommendations [RFC7525] and developments [RFC8446] in this area largely mitigate those.

Traffic analysis of unpadded encrypted traffic is also possible [pitfalls-of-dns-encryption] because the sizes and timing of encrypted DNS requests and responses can be correlated to unencrypted DNS requests upstream of a recursive resolver.
2.5. In the Servers

Using the terminology of [RFC6973], the DNS servers (recursive resolvers and authoritative servers) are enablers: they facilitate communication between an initiator and a recipient without being directly in the communications path. As a result, they are often forgotten in risk analysis. But, to quote again [RFC6973], "Although [...] enablers may not generally be considered as attackers, they may all pose privacy threats (depending on the context) because they are able to observe, collect, process, and transfer privacy-relevant data." In [RFC6973] parlance, enablers become observers when they start collecting data.

Many programs exist to collect and analyze DNS data at the servers -- from the "query log" of some programs like BIND to tcpdump and more sophisticated programs like PacketQ [packetq] [packetq-list] and DNSmezzo [dnsmezzo]. The organization managing the DNS server can use this data itself, or it can be part of a surveillance program like PRISM [prism] and pass data to an outside observer.

Sometimes, this data is kept for a long time and/or distributed to third parties for research purposes [ditl] [day-at-root], security analysis, or surveillance tasks. These uses are sometimes under some sort of contract, with various limitations, for instance, on redistribution, given the sensitive nature of the data. Also, there are observation points in the network that gather DNS data and then make it accessible to third parties for research or security purposes ("passive DNS" [passive-dns]).

2.5.1. In the Recursive Resolvers

Recursive Resolvers see all the traffic since there is typically no caching before them. To summarize: your recursive resolver knows a lot about you. The resolver of a large IAP, or a large public resolver, can collect data from many users. You may get an idea of the data collected by reading the privacy policy of a big public resolver, e.g., <https://developers.google.com/speed/public-dns/privacy>.

2.5.1.1. Encrypted transports

Use of encrypted transports does not reduce the data available in the recursive resolver and ironically can actually expose more information about users to operators. As mentioned in Section 2.4 use of session based encrypted transports (TCP/TLS) can expose correlation data about users. Such concerns in the TCP/TLS layers apply equally to DoT and DoH which both use TLS as the underlying transport.
2.5.1.2. DoH vs DoT

The proposed specification for DoH [RFC8484] includes a Privacy Considerations section which highlights some of the differences between HTTP and DNS. As a deliberate design choice DoH inherits the privacy properties of the HTTPS stack and as a consequence introduces new privacy concerns when compared with DNS over UDP, TCP or TLS [RFC7858]. The rationale for this decision is that retaining the ability to leverage the full functionality of the HTTP ecosystem is more important than placing specific constraints on this new protocol based on privacy considerations (modulo limiting the use of HTTP cookies).

In analyzing the new issues introduced by DoH it is helpful to recognize that there exists a natural tension between

- the wide practice in HTTP to use various headers to optimize HTTP connections, functionality and behaviour (which can facilitate user identification and tracking)

- and the fact that the DNS payload is currently very tightly encoded and contains no standardized user identifiers.

DoT, for example, would normally contain no client identifiers above the TLS layer and a resolver would see only a stream of DNS query payloads originating within one or more connections from a client IP address. Whereas if DoH clients commonly include several headers in a DNS message (e.g. user-agent and accept-language) this could lead to the DoH server being able to identify the source of individual DNS requests not only to a specific end user device but to a specific application.

Additionally, depending on the client architecture, isolation of DoH queries from other HTTP traffic may or may not be feasible or desirable. Depending on the use case, isolation of DoH queries from other HTTP traffic may or may not increase privacy.

The picture for privacy considerations and user expectations for DoH with respect to what additional data may be available to the DoH server compared to DNS over UDP, TCP or TLS is complex and requires a detailed analysis for each use case. In particular the choice of HTTPS functionality vs privacy is specifically made an implementation choice in DoH and users may well have differing privacy expectations depending on the DoH use case and implementation.

At the extremes, there may be implementations that attempt to achieve parity with DoT from a privacy perspective at the cost of using no identifiable headers, there might be others that provide feature rich
data flows where the low-level origin of the DNS query is easily identifiable.

Privacy focussed users should be aware of the potential for additional client identifiers in DoH compared to DoT and may want to only use DoH implementations that provide clear guidance on what identifiers they add.

2.5.2. In the Authoritative Name Servers

Unlike what happens for recursive resolvers, observation capabilities of authoritative name servers are limited by caching; they see only the requests for which the answer was not in the cache. For aggregated statistics ("What is the percentage of LOC queries?"), this is sufficient, but it prevents an observer from seeing everything. Still, the authoritative name servers see a part of the traffic, and this subset may be sufficient to violate some privacy expectations.

Also, the end user typically has some legal/contractual link with the recursive resolver (he has chosen the IAP, or he has chosen to use a given public resolver), while having no control and perhaps no awareness of the role of the authoritative name servers and their observation abilities.

As noted before, using a local resolver or a resolver close to the machine decreases the attack surface for an on-the-wire eavesdropper. But it may decrease privacy against an observer located on an authoritative name server. This authoritative name server will see the IP address of the end client instead of the address of a big recursive resolver shared by many users.

This "protection", when using a large resolver with many clients, is no longer present if ECS [RFC7871] is used because, in this case, the authoritative name server sees the original IP address (or prefix, depending on the setup).

As of today, all the instances of one root name server, L-root, receive together around 50,000 queries per second. While most of it is "junk" (errors on the Top-Level Domain (TLD) name), it gives an idea of the amount of big data that pours into name servers. (And even "junk" can leak information; for instance, if there is a typing error in the TLD, the user will send data to a TLD that is not the usual one.)

Many domains, including TLDs, are partially hosted by third-party servers, sometimes in a different country. The contracts between the domain manager and these servers may or may not take privacy into
account. Whatever the contract, the third-party hoster may be honest or not but, in any case, it will have to follow its local laws. So, requests to a given ccTLD may go to servers managed by organizations outside of the ccTLD’s country. End users may not anticipate that, when doing a security analysis.

Also, it seems (see the survey described in [aeris-dns]) that there is a strong concentration of authoritative name servers among "popular" domains (such as the Alexa Top N list). For instance, among the Alexa Top 100K, one DNS provider hosts today 10% of the domains. The ten most important DNS providers host together one third of the domains. With the control (or the ability to sniff the traffic) of a few name servers, you can gather a lot of information.

2.5.3. Rogue Servers

The previous paragraphs discussed DNS privacy, assuming that all the traffic was directed to the intended servers and that the potential attacker was purely passive. But, in reality, we can have active attackers redirecting the traffic, not to change it but just to observe it.

For instance, a rogue DHCP server, or a trusted DHCP server that has had its configuration altered by malicious parties, can direct you to a rogue recursive resolver. Most of the time, it seems to be done to divert traffic by providing lies for some domain names. But it could be used just to capture the traffic and gather information about you. Other attacks, besides using DHCP, are possible. The traffic from a DNS client to a DNS server can be intercepted along its way from originator to intended source, for instance, by transparent DNS proxies in the network that will divert the traffic intended for a legitimate DNS server. This rogue server can masquerade as the intended server and respond with data to the client. (Rogue servers that inject malicious data are possible, but it is a separate problem not relevant to privacy.) A rogue server may respond correctly for a long period of time, thereby foregoing detection. This may be done for what could be claimed to be good reasons, such as optimization or caching, but it leads to a reduction of privacy compared to if there was no attacker present. Also, malware like DNSchanger [dnschanger] can change the recursive resolver in the machine’s configuration, or the routing itself can be subverted (for instance, [ripe-atlas-turkey]).

2.5.4. Authentication of servers

Both DoH and Strict mode for DoT require authentication of the server and therefore as long as the authentication credentials are obtained over a secure channel then using either of these transports defeats
the attack of re-directing traffic to rogue servers. Of course
attacks on these secure channels are also possible, but out of the
scope of this document.

2.5.5. Blocking of services

User privacy can also be at risk if there is blocking (by local
network operators or more general mechanisms) of access to recursive
servers that offer encrypted transports. For example active blocking
of port 853 for DoT or of specific IP addresses (e.g. 1.1.1.1 or
2606:4700:4700::1111) could restrict the resolvers available to the
client. Similarly attacks on such services e.g. DDoS could force
users to switch to other services that do not offer encrypted
transports for DNS.

2.6. Re-identification and Other Inferences

An observer has access not only to the data he/she directly collects
but also to the results of various inferences about this data.

For instance, a user can be re-identified via DNS queries. If the
adversary knows a user’s identity and can watch their DNS queries for
a period, then that same adversary may be able to re-identify the
user solely based on their pattern of DNS queries later on regardless
of the location from which the user makes those queries. For
example, one study [herrmann-reidentification] found that such re-
identification is possible so that "73.1% of all day-to-day links
were correctly established, i.e. user u was either re-identified
unambiguously (1) or the classifier correctly reported that u was not
present on day t+1 any more (2)." While that study related to web
browsing behavior, equally characteristic patterns may be produced
even in machine-to-machine communications or without a user taking
specific actions, e.g., at reboot time if a characteristic set of
services are accessed by the device.

For instance, one could imagine that an intelligence agency
identifies people going to a site by putting in a very long DNS name
and looking for queries of a specific length. Such traffic analysis
could weaken some privacy solutions.

The IAB privacy and security program also have a work in progress
[RFC7624] that considers such inference-based attacks in a more
general framework.
2.7. More Information

Useful background information can also be found in [tor-leak] (about the risk of privacy leak through DNS) and in a few academic papers: [yanbin-tsudik], [castillo-garcia], [fangming-hori-sakurai], and [federrath-fuchs-herrmann-piosecny].

3. Actual "Attacks"

A very quick examination of DNS traffic may lead to the false conclusion that extracting the needle from the haystack is difficult. "Interesting" primary DNS requests are mixed with useless (for the eavesdropper) secondary and tertiary requests (see the terminology in Section 1). But, in this time of "big data" processing, powerful techniques now exist to get from the raw data to what the eavesdropper is actually interested in.

Many research papers about malware detection use DNS traffic to detect "abnormal" behavior that can be traced back to the activity of malware on infected machines. Yes, this research was done for the good, but technically it is a privacy attack and it demonstrates the power of the observation of DNS traffic. See [dns-footprint], [dagon-malware], and [darkreading-dns].

Passive DNS systems [passive-dns] allow reconstruction of the data of sometimes an entire zone. They are used for many reasons -- some good, some bad. Well-known passive DNS systems keep only the DNS responses, and not the source IP address of the client, precisely for privacy reasons. Other passive DNS systems may not be so careful. And there is still the potential problems with revealing QNAMEs.

The revelations (from the Edward Snowden documents, which were leaked from the National Security Agency (NSA)) of the MORECOWBELL surveillance program [morecowbell], which uses the DNS, both passively and actively, to surreptitiously gather information about the users, is another good example showing that the lack of privacy protections in the DNS is actively exploited.

4. Legalities

To our knowledge, there are no specific privacy laws for DNS data, in any country. Interpreting general privacy laws like [data-protection-directive] or GDPR [4] applicable in the European Union in the context of DNS traffic data is not an easy task, and we do not know a court precedent here. See an interesting analysis in [sidn-entrada].
5. Security Considerations

This document is entirely about security, more precisely privacy. It just lays out the problem; it does not try to set requirements (with the choices and compromises they imply), much less define solutions. Possible solutions to the issues described here are discussed in other documents (currently too many to all be mentioned); see, for instance, 'Recommendations for DNS Privacy Operators' [I-D.ietf-dprive-bcp-op].

6. Acknowledgments

Thanks to Nathalie Boulvard and to the CENTR members for the original work that led to this document. Thanks to Ondrej Sury for the interesting discussions. Thanks to Mohsen Souissi and John Heidemann for proofreading and to Paul Hoffman, Matthijs Mekking, Marcos Sanz, Tim Wicinski, Francis Dupont, Allison Mankin, and Warren Kumari for proofreading, providing technical remarks, and making many readability improvements. Thanks to Dan York, Suzanne Woolf, Tony Finch, Stephen Farrell, Peter Koch, Simon Josefsson, and Frank Denis for good written contributions. And thanks to the IESG members for the last remarks.

7. Changelog

draft-ietf-dprive-rfc7627-bis-00
  o Rename after WG adoption
  o Use DoT acronym throughout
  o Minor updates to status of deployment and other drafts
draft-bortzmeyer-dprive-rfc7626-bis-02
  o Update various references and fix some nits.
draft-bortzmeyer-dprive-rfc7626-bis-01
  o Update reference for dickinson-bcp-op to draft-dickinson-dprive-bcp-op
draft-bortzmeyer-dprive-rfc7626-bis-00:
Initial commit. Differences to RFC7626:
  o Update many references
8. References

8.1. Normative References


8.2. Informative References


[darkreading-dns]

[data-protection-directive]

[day-at-root]

[denis-edns-client-subnet]

[ditl]

[dns-footprint]

[dnschanger]

[dnsmezzo]
[fangming-hori-sakurai]

[federrath-fuchs-herrmann-piosecny]

[grangeia.snooping]

[herrmann-reidentification]

[I-D.ietf-dprive-bcp-op]

[I-D.ietf-quic-transport]

[I-D.ietf-tls-sni-encryption]
[morecowbell]

[packetq]

[packetq-list]

[passive-dns]

[pitfalls-of-dns-encryption]

[prism]

[RFC4033]

[RFC5155]

[RFC5936]

[RFC6269]

[RFC7413]


8.3. URIs


Authors’ Addresses
Stephane Bortzmeyer  
AFNIC  
1, rue Stephenson  
Montigny-le-Bretonneux  
France  78180  
Email: bortzmeyer+ietf@nic.fr

Sara Dickinson  
Sinodun IT  
Magdalen Centre  
Oxford Science Park  
Oxford  OX4 4GA  
United Kingdom  
Email: sara@sinodun.com
A Bootstrapping Procedure to Discover and Authenticate DNS-over-(D)TLS and DNS-over-HTTPS Servers
draft-reddy-dprive-bootstrap-dns-server-04

Abstract

This document specifies mechanisms to automatically bootstrap endpoints (e.g., hosts, Customer Equipment) to discover and authenticate DNS-over-(D)TLS and DNS-over-HTTPS servers provided by a local network.

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 December 16, 2019.

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 (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents.
1. Introduction

Traditionally a caching DNS server has been provided by local networks. This provides benefits such as low latency to reach that DNS server (owing to its network proximity to the endpoint). However, if an endpoint is configured to use Internet-hosted or public DNS-over-(D)TLS [RFC7858] [RFC8094] or DNS-over-HTTPS [RFC8484] servers, any available local DNS server cannot serve DNS requests from local endpoints. If public DNS servers are used instead of using local DNS servers, some operational problems can occur such as those listed below:

- "Split DNS" [RFC2775] to use the special internal-only domain names (e.g., "internal.example.com") in enterprise networks will
not work, and ".local" and "home.arpa" names cannot be locally resolved in home networks.

- Content Delivery Networks (CDNs) that map traffic based on DNS may lose the ability to direct end-user traffic to a nearby service-specific cluster in cases where a DNS service is being used that is not affiliated with the local network and which does not send "EDNS Client Subnet" (ECS) information [RFC7871] to the CDN’s DNS authorities [CDN].

If public DNS servers are used instead of using local DNS servers, the following discusses the impact on network-based security:

- Various network security services are provided by Enterprise networks to protect endpoints (e.g., Hosts, IoT devices). [I-D.camwinget-tls-use-cases] discusses some of the network-based service use cases. These network security services act on DNS requests originating from endpoints.

- However, if an endpoint is configured to use public DNS-over-(D)TLS or DNS-over-HTTPS servers, network security services cannot act efficiently on DNS requests from these endpoints.

- In order to act on DNS requests from endpoints, network security services can block DNS-over-(D)TLS traffic by dropping outgoing packets to destination port 853. Identifying DNS-over-HTTPS traffic is far more challenging than DNS-over-(D)TLS traffic. Network security services may try to identify the domains offering DNS-over-HTTPS servers, and DNS-over-HTTPS traffic can be blocked by dropping outgoing packets to these domains. If an endpoint has enabled strict privacy profile (Section 5 of [RFC8310]), and the network security service blocks the traffic to the public DNS server, the DNS service won’t be available to the endpoint and ultimately the endpoint cannot access Internet-reachable services.

- If an endpoint has enabled opportunistic privacy profile (Section 5 of [RFC8310]), and the network security service blocks traffic to the public DNS server, the endpoint will either fallback to an encrypted connection without authenticating the DNS server provided by the local network or fallback to clear text DNS, and cannot exchange encrypted DNS messages.

If the network security service fails to block DNS-over-(D)TLS or DNS-over-HTTPS traffic, this can compromise the endpoint security; some of the potential security threats are listed below:

- The network security service cannot prevent an endpoint from accessing malicious domains.
If the endpoint is an IoT device which is configured to use public DNS-over-(D)TLS or DNS-over-HTTPS servers, and if a policy enforcement point in the local network is programmed using, for example, a Manufacturer Usage Description (MUD) file [RFC8520] by a MUD manager to only allow intended communications to and from the IoT device, the policy enforcement point cannot enforce the network Access Control List (ACL) rules based on domain names (Section 8 of [RFC8520]).

If the network security service successfully blocks DNS-over-(D)TLS and DNS-over-HTTPS traffic, this can still compromise the endpoint security and privacy; some of the potential security threats are listed below:

- Pervasive monitoring of DNS traffic.
- An internal attacker can modify the DNS responses to re-direct the client to malicious servers.

To overcome the above threats, this document specifies a mechanism to automatically bootstrap endpoints to discover and authenticate the DNS-over-(D)TLS and DNS-over-HTTPS servers provided by their local network. The overall procedure can be structured into the following steps:

- **Bootstrapping (Section 4)** is necessary only when connecting to a new network or when the network’s DNS certificate has changed. Bootstrapping authenticates the Enrollment over Secure Transport (EST) [RFC7030] server to the endpoint. After authenticating the EST server, DNS server certificate used by the local network is downloaded to the endpoint. This DNS server certificate enables subsequent authenticated encrypted communication with the local DNS server (e.g., DNS-over-HTTPS) during the connection phase.

- **Discovery (Section 6)** is performed by a previously bootstrapped endpoint whenever connecting to a network. During discovery, the endpoint is instructed which privacy-enabling DNS protocol(s), port number(s), and IP addresses are supported on a local network. This effectively takes the place of DNS server IP address traditionally provided by IPv4 or IPv6 DHCP or by IPv6 Router Advertisement [RFC8106].

- **Connection handshake and service invocation (Section 7):** The DNS client initiates a (D)TLS handshake with the DNS server learned in the discovery phase, and validates the DNS server’s identity using the credentials obtained in the bootstrapping phase.
Note: The strict and opportunistic privacy profiles as defined in [RFC8310] only applies to DNS-over-(D)TLS protocols, there has been no such distinction made for DNS-over-HTTPS protocol.

2. Scope

The problems discussed in Section 1 will be encountered in Enterprise networks. Typically Enterprise networks do not assume that all devices in their network are managed by the IT team or Mobile Device Management (MDM) devices, especially in the quite common BYOD ("Bring Your Own Device") scenario. The mechanisms specified in this document can be used by BYOD devices to discover and authenticate DNS-over-(D)TLS and DNS-over-HTTPS servers provided by the Enterprise network. This mechanism can also be used by IoT devices (managed by IT team) after onboarding to discover and authenticate DNS-over-(D)TLS and DNS-over-HTTPS servers provided by the Enterprise network.

3. 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][RFC8174] when, and only when, they appear in all capitals, as shown here.

(D)TLS is used for statements that apply to both Transport Layer Security [RFC8446] and Datagram Transport Layer Security [RFC6347]. Specific terms are used for any statement that applies to either protocol alone.

This document uses the terms defined in [RFC8499].

4. Bootstrapping Endpoint Devices

The following steps detail the mechanism to automatically bootstrap an endpoint with the local network’s DNS server certificate:

1. The endpoint authenticates to the local network and discovers the Enrollment over Secure Transport (EST) [RFC7030] server using the procedure discussed in Section 8.

2. The endpoint establishes provisional TLS connection with that EST server, i.e., the endpoint provisionally accepts the unverified TLS server certificate. However, the endpoint MUST authenticate the EST server before it accepts the DNS server certificate. The endpoint either uses password-based authenticated key exchange (PAKE) with TLS 1.3 [I-D.barnes-tls-pake] as an authentication
method or uses the mutual authentication protocol for HTTP [RFC8120] to authenticate the discovered EST server.

As a reminder, PAKE is an authentication method that allows the use of usernames and passwords over unencrypted channels without revealing the passwords to an eavesdropper. Similarly, the mutual authentication for HTTP is based on PAKE and provides mutual authentication between an HTTP client and an HTTP server using username and password as credentials. The cryptographic algorithms to use with the mutual authentication protocol for HTTP are defined in [RFC8121].

3. The endpoint needs to use PAKE scheme to perform authentication the first time it connects to an EST server. If the EST server authentication is successful, the server’s identity can be used to authenticate subsequent TLS connections to that EST server. The endpoint configures the reference identifier for the EST server using the DNS-ID identifier type in the EST server certificate. On subsequent connections to the EST server, the endpoint MUST validate the EST server certificate using the Implicit Trust Anchor database (i.e., the EST server certificate must pass PKIX certification path validation) and match the reference identifier against the EST server’s identity according to the rules specified in Section 6.4 of [RFC6125].

4. The endpoint learns the End-Entity certificates [RFC8295] from the EST server. The certificate provisioned to the DNS server in the local network will be treated as an End-Entity certificate. As a reminder, the End-Entity certificates must be validated by the endpoint using an authorized trust anchor (Section 3.2 of [RFC8295]). The endpoint needs to identify the certificate provisioned to the DNS server. The SRV-ID identifier type [RFC6125] within subjectAltName entry MUST be used to identify the DNS server certificate.

For example, DNS server certificate will include SRV-ID "_domain-s.example.net" along with DNS-ID "example.net". The SRV service label "domain-s" is defined in Section 6 of [RFC7858]. As a reminder, the protocol component is not included in the SRV-ID [RFC4985].

5. The endpoint configures the authentication domain name (ADN) (defined in [RFC8310]) for the DNS server from the DNS-ID identifier type within subjectAltName entry in the DNS server certificate. The DNS server certificate is associated with the ADN to be matched with the certificate given by the DNS server in (D)TLS. To some extent, this approach is similar to certificate usage PKIX-EE(1) defined in [RFC7671].
Figure 1 illustrates a sequence diagram for bootstrapping an endpoint with the local network’s DNS server certificate.

```
+----------+                                     +--------+  +--------+
| Endpoint |                                     |  EST   |  |  DNS   |
|          |                                     |  Server |  |  Server |
+----------+                                     +--------+  +--------+

<table>
<thead>
<tr>
<th>DNS-SD query to discover the EST server</th>
</tr>
</thead>
<tbody>
<tr>
<td>optional: mDNS query to discover the EST server</td>
</tr>
<tr>
<td>Establish provisional TLS connection</td>
</tr>
<tr>
<td>PAKE scheme to authenticate the EST server</td>
</tr>
</tbody>
</table>

[Generate reference identifier for the EST server to compare with the EST server certificate in subsequent TLS connections]

| Get EE certificates |

[Identify the DNS server certificate in EE certificates to match with the certificate by the DNS server in (D)TLS handshake]

[Configure ADN and associate DNS server certificate]

Figure 1: Bootstrapping Endpoint Devices

5. Bootstrapping IoT Devices

The following steps explain the mechanism to automatically bootstrap IoT devices with local network’s CA certificates and DNS server certificate:

- Bootstrapping Remote Secure Key Infrastructures (BRSKI) discussed in [I-D.ietf-anima-bootstrapping-keyinfra] provides a solution for secure automated bootstrap of devices. BRSKI specifies means to provision credentials on devices to be used to operationally access networks. In addition, BRSKI provides an automated mechanism for the bootstrap distribution of CA certificates from

the EST server. The IoT device can use BRSKI to automatically bootstrap the IoT device using the IoT manufacturer provisioned X.509 certificate, in combination with a registrar provided by the local network and IoT device manufacturer’s authorizing service (MASA):

1. The IoT device authenticates to the local network using the IoT manufacturer provisioned X.509 certificate. The IoT device can request and get a voucher from the MASA service via the registrar. The voucher is signed by the MASA service and includes the local network’s CA public key.

2. The IoT device validates the signed voucher using the manufacturer installed trust anchor associated with the MASA, stores the CA’s public key and validates the provisional TLS connection to the registrar.

3. The IoT device requests the full EST distribution of current CA certificates (Section 5.9.1 in [I-D.ietf-anima-bootstrapping-keyinfra]) from the registrar operating as a BRSKI-EST server. The IoT devices stores the CA certificates as Explicit Trust Anchor database entries. The IoT device uses the Explicit Trust Anchor database to validate the DNS server certificate.

4. The IoT device learns the End-Entity certificates from the BRSKI-EST server. The certificate provisioned to the DNS server in the local network will be treated as an End-Entity certificate. The IoT device needs to identify the certificate provisioned to the DNS server. The SRV-ID identifier type within subjectAltName entry MUST be used to identify the DNS server certificate.

5. The endpoint configures the ADN for the DNS server from the DNS-ID identifier type within subjectAltName entry in the DNS server certificate. The DNS server certificate is associated with the ADN to be matched with the certificate given by the DNS server in (D)TLS.

6. DNS-over-(D)TLS and DNS-over-HTTPS Server Discovery Procedure

This specification defines "DPRIVE" as the application service tag (Section 12.1.1) and "dns.tls" (Section 12.1.2), "dns.dtls" (Section 12.1.3), and "dns.https" (Section 12.1.4) as application protocol tags. A DNS client discovers the DNS server in the local network supporting DNS-over-TLS, DNS-over-DTLS and DNS-over-HTTPS protocols by using the following discovery mechanism:
The DNS client makes an S-NAPTR [RFC3958] lookup with the authentication domain name and the 'DPRIVE' application service tag to learn the protocols DNS-over-TLS, DNS-over-DTLS, and DNS-over-HTTPS supported by the DNS server and the DNS privacy protocol preferred by the DNS server administrators. The S-NAPTR lookup is performed using an recursive DNS resolver discovered from an untrusted source (such as DHCP).

In the example depicted in Figure 2, for authentication domain name 'example.net', the resolution algorithm will result in the privacy-enabling protocols supported by the DNS server and usable DNS server IP addresses and port numbers.

```
examp1e.net.
IN NAPTR 100 10 "" DPRIVE:dns.tls "" dns1.example.net.
IN NAPTR 200 10 "" DPRIVE:dns.dtls "" dns2.example.net.

dns1.example.net.
IN NAPTR 100 10 S DPRIVE:dns.tls "" _domain-s._tcp.example.net.

dns2.example.net.
IN NAPTR 100 10 S DPRIVE:dns.dtls "" _domain-s._udp.example.net.

_domain-s._tcp.example.net.
IN SRV 0 0 853 a.example.net.

_domain-s._udp.example.net.
IN SRV 0 0 853 a.example.net.

a.example.net.
IN A 192.0.2.1
IN AAAA 2001:db8:8:4::2
```

Figure 2

If DNS-over-HTTPS protocol is supported by the DNS server, the DNS client finds the URI template of the DNS-over-HTTPS server using one of the mechanisms discussed in [I-D.ietf-doh-resolver-associated-doh] to use the https URI scheme (Section 3 of [RFC8484]).

If no DNS-specific S-NAPTR records can be retrieved, the discovery procedure fails for this authentication domain name. However, before retrying a lookup that has failed, a DNS client MUST wait a time period that is appropriate for the encountered error (e.g., NXDOMAIN, timeout, etc.).
7. Connection Handshake and Service Invocation

The DNS client initiates (D)TLS handshake with the DNS server, the DNS server presents its certificate in ServerHello message, and the DNS client MUST match the DNS server certificate downloaded in Step 4 in Section 4 or Section 5 with the certificate provided by the DNS server in (D)TLS handshake. If the match is successful, the DNS client MUST validate the server certificate using the Implicit Trust Anchor database (i.e., the DNS server certificate must pass PKIX certification path validation).

If the match is successful and server certificate is successfully validated, the client continues with the connection as normal. Otherwise, the client MUST treat the server certificate validation failure as a non-recoverable error. If the DNS client cannot reach or establish an authenticated and encrypted connection with the privacy-enabling DNS server provided by the local network, the DNS client can fallback to the privacy-enabling public DNS server.

8. EST Service Discovery Procedure

DNS-based Service Discovery (DNS-SD) [RFC6763] and Multicast DNS (mDNS) [RFC6762] provide generic solutions for discovering services available in a local network. DNS-SD/mDNS define a set of naming rules for certain DNS record types that they use for advertising and discovering services.

Section 4.1 of [RFC6763] specifies that a service instance name in DNS-SD has the following structure:

<Instance> . <Service> . <Domain>

The <Domain> portion specifies the DNS sub-domain where the service instance is registered. It may be "local.", indicating the mDNS local domain, or it may be a conventional domain name such as "example.com.". The <Service> portion of the EST service instance name MUST be "_est._tcp".

8.1. mDNS

A EST client application can proactively discover an EST server being advertised in the site by multicasting a PTR query to the following:

- "_est._tcp.local"

A EST server can send out gratuitous multicast DNS answer packets whenever it starts up, wakes from sleep, or detects a change in EST
server configuration. EST client application can receive these gratuitous packets and cache information contained in them.

9. Network Reattachment

On subsequent attachments to the network, the endpoint discovers the privacy-enabling DNS server using the authentication domain name (configured in Step 5 of Section 4 or Section 5), initiates (D)TLS handshake with the DNS server and follows the mechanism discussed in Section 7 to validate the DNS server certificate.

If the DNS server certificate is invalid (e.g., revoked or expired) or the procedure to discover the privacy-enabling DNS server fails (e.g., the domain name of the privacy-enabling DNS server has changed because the Enterprise network has switched to a public privacy-enabling DNS server capable of blocking access to malicious domains), the endpoint discovers and initiates TLS handshake with the EST server, and uses the validation techniques described in [RFC6125] to compare the reference identifier (created in Step 2 of Section 4 in this document) to the EST server certificate and verifies the entire certification path as per [RFC5280]. The endpoint then gets the DNS server certificate from the EST server. If the DNS-ID identifier type within subjectAltName entry in the DNS server certificate does not match the configured ADN, the ADN is replaced with the DNS-ID identifier type. The DNS server certificate associated with the ADN is replaced with the one provided by the EST server. If the ADN has changed, the endpoint discovers the privacy-enabling DNS server, initiates (D)TLS handshake with the DNS server and follows the mechanism discussed in Section 7 to validate the DNS server certificate.

Figure 3 illustrates a sequence diagram for re-configuring an endpoint with ADN and local network’s DNS server certificate on subsequent attachments to the network.
Figure 3: Bootstrapping Endpoint Devices on subsequent attachments to the network

10. Privacy Considerations

[RFC7626] discusses DNS privacy considerations in both "on the wire" (Section 2.4 of [RFC7626]) and "in the server" (Section 2.5 of [RFC7626]) contexts. The endpoint may not know if the DNS-over-(D)TLS or DNS-over-HTTPS server in the local network has a privacy preserving data policy. A new privacy certificate extension is defined that identifies the privacy preserving data policy of the DNS server.

10.1. Privacy Extension Format

Like all X.509 certificate extensions, the privacy certificate extension is defined using ASN.1 [ASN1-88]. The non-critical privacy extension is identified by id-pe-privacy.
PKIX Object Identifier Registry

id-pkix OBJECT IDENTIFIER ::= { iso(1) identified-organization(3)
dod(6) internet(1) security(5) mechanisms(5) pkix(7) }

PKIX Arcs
id-mod OBJECT IDENTIFIER ::= { id-pkix 0 } -- modules
id-pe OBJECT IDENTIFIER ::= { id-pkix 1 } -- private certificate extensions

PKIX modules
id-mod-privacy-extn OBJECT IDENTIFIER ::= { id-mod TBD2 }
id-pe-privacy OBJECT IDENTIFIER ::= { id-pe TBD1 }

A non-null privacy always includes a base privacy. The privacy extension includes the following information:

- If the client IP address is Personally Identifiable Information (PII) data or non PII-data.
- If the user identity that sent the DNS query is logged or not, and if user identity address is indeed logged, the period for which the user identity is logged. User identity such as username, IP address, MAC address or personally identifiable data. Logging duration is represented in hours. A negative one (-1) of logging duration indicates indefinite duration.
- If the transaction data (e.g., DNS messages) is logged or not, and if transaction data is logged, the period for which the transaction data is stored.
- If the transaction data is logged to notify the user access to certain domains (e.g., malicious domains) is blocked, the period for which the transaction data is stored. If access to malicious domains is logged, the period for which the transaction data is stored. If the transaction data is logged for analytics (e.g. to detect malicious domains), the period for which the transaction data is stored.
- If the transaction data is shared with partners or not, and if the transaction data is shared with partners, the names of the partners. If anonymized data or client identifiable data is shared with partners.
- If the transaction data is shared or sold to third parties.
- If the DNS server will block DNS resolution of certain domains (e.g., malicious domains).
o A URL that points to the privacy preserving data policy, and a URL that points to the security assessment report of the DNS server by a third party auditor.

10.2. Privacy Extension Syntax

The syntax for the privacy extension is as follows:

```
Privacy ::= CHOICE {
    none                 NULL,
    -- No privacy policy provided
    pPolicy              PrivacyPolicy
    -- Privacy preserving data policy  }

PrivacyPolicy ::= SEQUENCE {
    base              PrivacyInfo,
    pURL         [0]  PrivacyURL OPTIONAL,
    aURL         [1]  AuditURL OPTIONAL  }

PrivacyInfo ::= SEQUENCE {
    ipaddresspii      BOOLEAN,
    -- TRUE means client IP address is PII
    log          [0]  Logging,
    sdata        [2]  ShareData,
    transferdata [3]  BOOLEAN,
    -- TRUE means share or sell data to third parties
    blockdomains [4]  BOOLEAN
    -- TRUE means domains will be blocked  }

LoggingTypes ::= BIT STRING {
    none           (0),
    -- No logging
    all            (1),
    -- Log all transaction data
    useridentity   (2),
    -- Log user identity (e.g., username, IP address)
    notifyuser     (3),
    -- Log to notify user access
    -- to certain domains is blocked
    knownmalware   (4),
    -- Log access to malicious domains
    analytics      (5)
    -- Log transaction data for analytics
    -- (e.g. to detect malicious domains)  }

LoggingDuration ::= SEQUENCE {
    all          [0]  INTEGER OPTIONAL,
    }
useridentity [1] INTEGER OPTIONAL,
notifyuser [2] INTEGER OPTIONAL,
knownmalware [3] INTEGER OPTIONAL,
analytics [4] INTEGER OPTIONAL }

Logging ::= SEQUENCE {
loggingTypes LoggingTypes DEFAULT {none},
loggingDuration LoggingDuration OPTIONAL
-- Transaction data is cleared
-- after logging duration,
-- Negative one (-1) indicates indefinite
-- duration }

ShareData ::= SEQUENCE {
sharepartners BOOLEAN,
partners [1] SEQUENCE SIZE (1..MAX) OF UTF8String OPTIONAL,
-- Names of the partners
anonymizeddata [0] BOOLEAN OPTIONAL
-- TRUE means anonymized data
-- is shared with partners }

PrivacyURL ::= IA5String -- MUST use https scheme
AuditURL ::= IA5String -- MUST use https scheme

11. Security Considerations

The bootstrapping procedure to obtain the certificate of the local networks DNS server uses a client identity and password to authenticate the EST server using PAKE schemes. Security considerations such as those discussed in [I-D.barnes-tls-pake] or [RFC8120] and [RFC8121] need to be taken into consideration.

Users cannot be expected to enable or disable the bootstrapping or the discovery procedure as they switch networks. Thus, it is RECOMMENDED that users indicate to their system in some way that they desire bootstrapping to be performed when connecting to a specific network, similar to the way users disable VPN connection in specific network (e.g., Enterprise network) and enable VPN connection by default in other networks.

If an endpoint has enabled strict privacy profile, and the network security service blocks the traffic to the privacy-enabling public DNS server, a hard failure occurs and the user is notified. The user has a choice to switch to another network or if the user trusts the network, the user can enable strict privacy profile with the DNS-
The primary attacks against the methods described in Section 6 are the ones that would lead to impersonation of a DNS server and spoofing the DNS response to indicate that the DNS server does not support any privacy-enabling protocols. To protect against DNS-vectored attacks, secured DNS (DNSSEC) can be used to ensure the validity of the DNS records received. Impersonation of the DNS server is prevented by validating the certificate presented by the DNS server. If the EST server conveys the DNS server certificate, but the S-NAPTR lookup indicates that the DNS server does not support any privacy-enabling protocols, the client can detect the DNS response is spoofed.

Security considerations in [I-D.ietf-anima-bootstrapping-keyinfra] need to be taken into consideration for IoT devices.

12. IANA Considerations

IANA is requested to allocate the SRV service name of "est".

IANA is requested to add the following entry in the "SMI Security for PKIX Certificate Extension" (1.3.6.1.5.5.7.1) registry:

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Description</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD1</td>
<td>id-pe-privacy</td>
<td>this document</td>
</tr>
</tbody>
</table>

IANA is requested to add the following entry in the "SMI Security for PKIX Module Identifier" (1.3.6.1.5.5.7.0) registry:

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Description</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD2</td>
<td>id-mod-privacy-extn</td>
<td>this document</td>
</tr>
</tbody>
</table>

12.1. Application Service & Application Protocol Tags

This document requests IANA to make the following allocations from the registry available at: https://www.iana.org/assignments/s-naptr-parameters/s-naptr-parameters.xhtml.
12.1.1. DNS Application Service Tag Registration
- Application Protocol Tag: DPRIVE
- Intended Usage: See Section 6
- Security Considerations: See Section 11
- Contact Information: <one of the authors>

12.1.2. dns.tls Application Protocol Tag Registration
- Application Protocol Tag: dns.tls
- Intended Usage: See Section 6
- Security Considerations: See Section 11
- Contact Information: <one of the authors>

12.1.3. dns.dtls Application Protocol Tag Registration
- Application Protocol Tag: dns.dtls
- Intended Usage: See Section 6
- Security Considerations: See Section 11
- Contact Information: <one of the authors>

12.1.4. dns.https Application Protocol Tag Registration
- Application Protocol Tag: dns.https
- Intended Usage: See Section 6
- Security Considerations: See Section 11
- Contact Information: <one of the authors>

13. Acknowledgments

Thanks to Joe Hildebrand, Harsha Joshi, Shashank Jain, Patrick McManus, Bob Harold, Livingood Jason, Winfield Alister, Eliot Lear and Sara Dickinson for the discussion and comments.
14. References

14.1. Normative References

[I-D.ietf-doh-resolver-associated-doh]
Hoffman, P., "Associating a DoH Server with a Resolver",
draft-ietf-doh-resolver-associated-doh-03 (work in progress), March 2019.

[RFC2119]  Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,

[RFC3958]  Daigle, L. and A. Newton, "Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)", RFC 3958, DOI 10.17487/RFC3958,

[RFC4985]  Santesson, S., "Internet X.509 Public Key Infrastructure Subject Alternative Name for Expression of Service Name", RFC 4985, DOI 10.17487/RFC4985, August 2007,


[RFC6762]  Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762,
DOI 10.17487/RFC6762, February 2013,

14.2. Informative References


Authors’ Addresses

Tirumaleswar Reddy
McAfee, Inc.
Embassy Golf Link Business Park
Bangalore, Karnataka  560071
India

Email: kondtir@gmail.com

Dan Wing
Citrix Systems, Inc.
USA

Email: dwing-ietf@fuggles.com

Michael C. Richardson
Sandelman Software Works
USA

Email: mcr+ietf@sandelman.ca

Mohamed Boucadair
Orange
Rennes  35000
France

Email: mohamed.boucadair@orange.com
DNS Zone Transfer using DNS Stateful Operations
draft-zatda-dprive-xfr-using-dso-00

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. This document specifies use of DNS Stateful Operations to enable a subscribe/publish mechanism for zone transfers reducing the overhead introduced by NOTITY/SOA interactions prior to zone transfer request. This additionally prevents zone contents collection via passive monitoring of zone transfers by restricting XFR using DSO to require TLS.

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.

1. Introduction

[I-D.hzpa-dprive-xfr-over-tls] enumerates the existing issues with clear text XFR mechanisms, outlines some use cases for using encrypted channels for zone transfer and also describes using TLS for zone transfers. It additionally discusses the various authentication

mechanisms that can be used to provide data and channel authentication, and channel confidentiality.

This draft describes the use of a DSO [RFC8490] based protocol to perform zone transfers. This mechanism is heavily based on an existing use of DSO where DNS clients can subscribe to receive asynchronous notifications of changes to RRSets of interest: DNS PUSH Notifications [I-D.ietf-dnssd-push]. That specification was developed with DNS Service Discovery in mind, this document describes an analogous protocol (XFR-using-DSO) where DNS clients can subscribe to receive asynchronous notifications of changes to zones of interest, it is developed with efficient and confidential zone transfers between primaries and secondaries in mind.

In the XFR-using-DSO model, a DSO connection is first opened between the client and server, the client can then subscribe to one or more zones to be notified of changes and the server can publish changes to the zone over the connection. Clients can choose to unsubscribe from zone updates at any time.

Servers could also use the DSO session to send command-style messages to the client, for example, to instruct a client to stop serving a zone or delete a zone. No such commands are defined in this version of the specification, but will likely be added in a future version.

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].

DNS terminology is as described in [RFC8499].

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

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

XuD: XFR-using-DOS mechanisms as specified in this document
3. Use Cases for XFR-using-DSO

This section includes additional use cases in addition to those specified in [I-D.hzpa-dprive-xfr-over-tls] that XuD can offer.

- **Confidentiality.** Since this mechanism could, in principle, eliminate the need for NOTIFY and SOA queries it can provide complete confidentiality for the entire zone transfer mechanism.

- **Security.** For some network configurations it is not desirable to have port 53 on the secondary open to an untrusted network for the sole purpose of receiving NOTIFYs. NOTIFYs can also be trivially spoofed unless secured with TSIG. For the DSO case, secondaries could initiate DSO connections to the primary and following that server-initiated DSO NOTIFY messages could be sent on that connection which could simultaneously be used for SOA and IXFR requests. This would allow a firewall to be restricted to just allowing outgoing connections from secondary to primary. Note that a similar but more constrained mechanism exists for IXFR whereby a short refresh period can be configured which triggers periodic SOA/IXFR requests from the secondary. TODO: Look at the details of the NSD implementation.

- **Performance.** For the DSO case, a new subscribe/publish mechanism could be envisaged that greatly reducing the number of messages required to perform one transfer.

- **Improved error handling and retries.** In the DSO case new explicit error codes could be defined that allow a server to indicate the reason for a failed or aborted XFR request. Also a new client initiated message could be used to gracefully cancel AXFRs.

- **New command channel.** For the DSO case it would be possible to include new server-initiated ‘control’ commands e.g. ‘stop serving this zone’, ‘delete this zone’.

**QUESTION:** Is there any case where the primary might want to initiate the DSO connection to the secondary?

4. **Overview**

The figure below provides an outline of the XuD protocol.

**Figure 1: XuD protocol [1]**

A DNS XuD client subscribes for zone notifications for a particular zone by connecting to the appropriate authoritative server for that zone, and sending DSO message(s) indicating the zone(s) of interest.
When the client loses interest in receiving further updates to these zones, it unsubscribes.

The authoritative server for a DNS zone is any server capable of generating the correct change notifications for a zone. It may be a primary, secondary, or stealth name server [RFC7719].

Standard DNS Queries MAY be sent over a XuD (i.e., DSO) session. For any zone for which the server is authoritative, it MUST respond authoritatively for queries on names falling within that zone both for normal DNS queries and for XuD subscriptions. For names for which the server is acting as a recursive resolver, e.g. when the server is the local recursive resolver, for any query for which it supports XuD subscriptions, it MUST also support standard queries.

XuD imposes less load on the responding server than rapid polling would, but XuD notifications do still have a cost, so XuD clients MUST only create XuD subscriptions for zones they are authorised to transfer.

Generally, as described in the DNS Stateful Operations specification [RFC8490], a client must not keep a session to a server open indefinitely if it has no subscriptions (or other operations) active on that session. A client MAY close a session as soon as it becomes idle, and then if needed in the future, open a new session when required. Alternatively, a client MAY speculatively keep an idle session open for some time, subject to the constraint that it MUST NOT keep a session open that has been idle for more than the session’s idle timeout (15 seconds by default) [RFC8490].

5. Transport

XuD clients MUST use DNS Stateful Operations [RFC8490] running over TLS over TCP [RFC7858].

The connection for XuD 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 XuD.

QUESTION: Is there a use case to allow XuD over TCP where confidentiality is not an issue e.g when the zone contents are already publicly available?

6. State Considerations

Each XuD server is capable of handling some finite number of XuD subscriptions. This number will vary from server to server and is based on physical machine characteristics, network bandwidth, and
operating system resource allocation. After a client establishes a
session to a DNS server, each subscription is individually accepted
or rejected. Servers may employ various techniques to limit
subscriptions to a manageable level. Correspondingly, the client is
free to establish simultaneous sessions to alternate DNS servers that
support XuDs for the zone and distribute subscriptions at the
client's discretion. In this way, both clients and servers can react
to resource constraints.

7. Protocol Operation

The XuD protocol is a session-oriented protocol, and makes use of DNS
Stateful Operations (DSO) [RFC8490].

For details of the DSO message format refer to the DNS Stateful
Operations specification [RFC8490]. Those details are not repeated
here.

XuD clients and servers MUST support DSO. A single server can
support DNS Queries, DNS Updates, and XuD (using DSO) on the same TCP
port.

A XuD exchange begins with the client making a TLS/TCP connection to
the appropriate server.

A typical XuD client will immediately issue a DSO Keepalive operation
to request a session timeout and/or keepalive interval longer than
the 15-second default values, but this is not required. A XuD
client MAY issue other requests on the session first, and only issue
a DSO Keepalive operation later if it determines that to be
necessary. Sending either a DSO Keepalive operation or a XuD
subscription over the TLS/TCP connection to the server signals the
client’s support of DSO and serves to establish a DSO session.

In accordance with the current set of active subscriptions, the
server sends relevant asynchronous XuD notifications to the client.
Note that a client MUST be prepared to receive (and silently ignore)
XuD notifications for subscriptions it has previously removed, since
there is no way to prevent the situation where a XuD notification is
in flight from server to client while the client’s unsubscribe
message cancelling that subscription is simultaneously in flight from
client to server.

7.1. XuD SUBSCRIBE-XFR

After connecting, and requesting a longer idle timeout and/or
keepalive interval if necessary, a XuD client then indicates its
desire to receive XuD notifications for a given zone by sending a
SUBSCRIBE-XFR request to the server. A SUBSCRIBE-XFR request is encoded in a DSO message [RFC8490]. This specification defines a primary DSO TLV for XuD SUBSCRIBE-XFR Requests (tentatively DSO Type Code 0x50).

DSO messages with the SUBSCRIBE-XFR TLV as the Primary TLV are not permitted in early data.

The entity that initiates a SUBSCRIBE-XFR request is by definition the client. A server MUST NOT send a SUBSCRIBE-XFR request over an existing session from a client. If a server does send a SUBSCRIBE-XFR request over a DSO session initiated by a client, this is a fatal error and the client should immediately abort the connection with a TLS close_notify alert. See Section 6.1 of [RFC8446].

TODO: Need to define a DSO version of TSIG to cover the SUBSCRIBE-XFR and DSO-XFR responses, since the Additional section count in DSO message MUST be zero. Note the client only needs to use TSIG in the SUBSCRIBE-XFR message to prove it is authorised to request zone transfers, but all DSO-XFR messages should be signed if primary TSIG is required for the authentication model in use.

7.1.1. SUBSCRIBE-XFR Request

A SUBSCRIBE-XFR request begins with the standard DSO 12-byte header [RFC8490], followed by the SUBSCRIBE-XFR primary TLV. A SUBSCRIBE-XFR request message is illustrated in Figure 2.

The MESSAGE ID field MUST be set to a unique value, that the client is not using for any other active operation on this DSO session. For the purposes here, a MESSAGE ID is in use on this session if the client has used it in a request for which it has not yet received a response, or if the client has used it for a subscription which it has not yet cancelled using UNSUBSCRIBE-XFR. In the SUBSCRIBE-XFR response the server MUST echo back the MESSAGE ID value unchanged.

The other header fields MUST be set as described in the DSO specification [RFC8490]. The DNS OPCODE field contains the OPCODE value for DNS Stateful Operations (6). The four count fields MUST be zero, and the corresponding four sections MUST be empty (i.e., absent).

The DSO-TYPE is SUBSCRIBE-XFR (tentatively 0x50).

The DSO-LENGTH is the length of the DSO-DATA that follows, which specifies the name and class of the zone and optionally the SOA value of the client’s version of the zone.
If the client has no copy of the zone it MUST omit the SOA value to indicate to the server that a DSO-AXFR is required in response (see the next section).

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|             MESSAGE ID             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|QR| OPCODE(6) | Z | RCODE |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|      QDCOUNT (MUST BE ZERO)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|      ANCOUNT (MUST BE ZERO)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|      NSCOUNT (MUST BE ZERO)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|      ARCOUNT (MUST BE ZERO)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| DSO-TYPE = SUBSCRIBE-XFR (tentatively 0x50) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| DSO-LENGTH (number of octets in DSO-DATA) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|             NAME                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|         CLASS                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|         SOA value                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 2: SUBSCRIBE-XFR Request

The DSO-DATA for a SUBSCRIBE-XFR request MUST contain exactly one NAME, CLASS and SOA value. Since SUBSCRIBE-XFR requests are sent over TCP, multiple SUBSCRIBE-XFR DSO request messages can be concatenated in a single TCP stream and packed efficiently into TCP segments.

If accepted, the subscription will stay in effect until the client cancels the subscription using UNSUBSCRIBE-XFR or until the DSO session between the client and the server is closed.

SUBSCRIBE-XFR requests on a given session MUST be unique. A client MUST NOT send a SUBSCRIBE-XFR message that duplicates the NAME, CLASS and SOA value of an existing active subscription on that DSO session. For the purpose of this matching, the established DNS case-
insensitivity for US-ASCII letters applies (e.g., "example.com" and "Example.com" are the same). If a server receives such a duplicate SUBSCRIBE-XFR message this is an error and the server MUST immediately terminate the connection with a TLS close_notify alert.

QUESTION: Is there a use case where a client may want to signal that the version of the zone it holds has been updated via another mechanism and the zone transfer should restart from a different SOA than that currently exchanged between client and server?

DNS wildcarding is not supported. SUBSCRIBE-XFR requests received for zones containing wildcards are considered an error (see below).

A CLASS of ‘ANY’ (255) is not supported.

7.1.2. SUBSCRIBE-XFR Response

Each SUBSCRIBE-XFR request generates exactly one SUBSCRIBE-XFR response from the server. A SUBSCRIBE-XFR request message is illustrated in Figure 3.

A SUBSCRIBE-XFR response begins with the standard DSO 12-byte header [RFC8490]. The QR bit in the header is set indicating it is a response. The header MAY be followed by one or more optional TLVs, such as a Retry Delay TLV.

The MESSAGE ID field MUST echo the value given in the Message ID field of the SUBSCRIBE-XFR request. This is how the client knows which request is being responded to.

A SUBSCRIBE-XFR response message MUST NOT include a SUBSCRIBE-XFR TLV. If a client receives a SUBSCRIBE-XFR response message containing a SUBSCRIBE-XFR TLV then the response message is processed but the SUBSCRIBE-XFR TLV MUST be silently ignored.
In the SUBSCRIBE-XFR response the RCODE indicates whether or not the subscription was accepted. Supported RCODEs are as follows:

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOERROR</td>
<td>0</td>
<td>SUBSCRIBE-XFR successful.</td>
</tr>
<tr>
<td>FORMERR</td>
<td>1</td>
<td>Server failed to process request due to a malformed request.</td>
</tr>
<tr>
<td>SERVFAIL</td>
<td>2</td>
<td>Server failed to process request due to a problem with the server.</td>
</tr>
<tr>
<td>NOTIMP</td>
<td>4</td>
<td>Server does not implement DSO.</td>
</tr>
<tr>
<td>REFUSED</td>
<td>5</td>
<td>Server refuses to process request for policy or security reasons.</td>
</tr>
<tr>
<td>NOTAUTH</td>
<td>9</td>
<td>Server is not authoritative for the requested name.</td>
</tr>
<tr>
<td>DSOTYPENI</td>
<td>11</td>
<td>SUBSCRIBE-XFR operation not supported.</td>
</tr>
</tbody>
</table>

Table 1: SUBSCRIBE-XFR Response codes

This document specifies only these RCODE values for SUBSCRIBE-XFR Responses. Servers sending SUBSCRIBE-XFR Responses SHOULD use one of these values. Note that NXDOMAIN is not a valid RCODE in response to a SUBSCRIBE-XFR Request. However, future circumstances may create situations where other RCODE values are appropriate in SUBSCRIBE-XFR Responses, so clients MUST be prepared to accept SUBSCRIBE-XFR Responses with any other RCODE value.
If the server sends a nonzero RCODE in the SUBSCRIBE-XFR response, that means:

a) the client is (at least partially) misconfigured,

b) the server resources are exhausted, or

c) there is some other unknown failure on the server.

In any case, the client shouldn’t retry the subscription to this server right away. If a client has other authoritative servers configured for a given zone an alternative server can be tried immediately.

If the client has other successful subscriptions to this server, these subscriptions remain even though additional subscriptions may be refused. Neither the client nor the server are required to close the connection, although, either end may choose to do so.

If the server sends a nonzero RCODE then it SHOULD append a Retry Delay TLV [RFC8490] to the response specifying a delay before the client attempts this operation again. Recommended values for the delay for different RCODE values are given below. These recommended values apply both to the default values a server should place in the Retry Delay TLV, and the default values a client should assume if the server provides no Retry Delay TLV.

For RCODE = 1 (FORMERR) the delay may be any value selected by the implementer. A value of five minutes is RECOMMENDED, to reduce the risk of high load from defective clients.

For RCODE = 2 (SERVFAIL) the delay should be chosen according to the level of server overload and the anticipated duration of that overload. By default, a value of one minute is RECOMMENDED. If a more serious server failure occurs, the delay may be longer in accordance with the specific problem encountered.

For RCODE = 4 (NOTIMP), which occurs on a server that doesn’t implement DNS Stateful Operations [RFC8490], it is unlikely that the server will begin supporting DSO in the next few minutes, so the retry delay SHOULD be one hour. Note that in such a case, a server that doesn’t implement DSO is unlikely to place a Retry Delay TLV in its response, so this recommended value in particular applies to what a client should assume by default.

For RCODE = 5 (REFUSED), which occurs on a server that implements XuDs, but is currently configured to disallow XuDs, the retry delay may be any value selected by the implementer and/or configured by the
operator. Since it is possible that the misconfiguration may be repaired at any time, the retry delay should not be set too high. By default, a value of 5 minutes is RECOMMENDED.

For RCODE = 9 (NOTAUTH), which occurs on a server that implements XuDs, but is not configured to be authoritative for the requested name, the retry delay may be any value selected by the implementer and/or configured by the operator. Since it is possible that the misconfiguration may be repaired at any time, the retry delay should not be set too high. By default, a value of 5 minutes is RECOMMENDED.

For RCODE = 11 (DSOTYPENI), which occurs on a server that implements DSO but doesn’t implement XuD, it is unlikely that the server will begin supporting XuD in the next few minutes, so the retry delay SHOULD be one hour.

For other RCODE values, the retry delay should be set by the server as appropriate for that error condition. By default, a value of 5 minutes is RECOMMENDED.

For RCODE = 9 (NOTAUTH), the time delay applies to requests for other names falling within the same zone. Requests for names falling within other zones are not subject to the delay. For all other RCODEs the time delay applies to all subsequent requests to this server.

After sending an error response the server MAY allow the session to remain open, or MAY send a Retry Delay Operation TLV instructing the client to close the session, as described in the DSO specification [RFC8490]. Clients MUST correctly handle both cases.

7.2. XuD Notifications

Once a subscription has been successfully established, the server generates DSO-IXFR messages to send to the client as appropriate. In the case that the server could not provide a DSO-IXFR message based on the SOA received from the client an initial DSO-AXFR message will be sent immediately following the SUBSCRIBE-XFR Response. Subsequent changes to the zone are then communicated to the client in subsequent DSO-IXFR messages.

Until an UNSUBSCRIBE-XFR message is received the server MUST assume that the client is updating the client’s version of the zone with the notifications sent and can therefore hold state on the SOA version the client holds. It MUST use this to generate the DSO-IXFR messages sent on a XuD session.
7.2.1. DSO-IXFR Message

A DSO-IXFR unidirectional message begins with the standard DSO 12-byte header [RFC8490], followed by the DSO-IXFR primary TLV. A DSO-IXFR message is illustrated in Figure 4.

In accordance with the definition of DSO unidirectional messages, the MESSAGE ID field MUST be zero. There is no client response to a DSO-IXFR message.

The other header fields MUST be set as described in the DSO specification [RFC8490]. The DNS OPCODE field contains the OPCODE value for DNS Stateful Operations (6). The four count fields MUST be zero, and the corresponding four sections MUST be empty (i.e., absent).

The DSO-TYPE is DSO-IXFR (tentatively 0x51).

The DSO-LENGTH is the length of the DSO-DATA that follows, which specifies the changes being communicated.

The DSO-DATA contains one or more change notifications. A DSO-IXFR Message MUST contain at least one change notification. If a DSO-IXFR Message is received that contains no change notifications, this is a fatal error, and the receiver MUST immediately terminate the connection with a TLS close_notify alert.
The DSO-DATA in a DSO-IXFR message is identical to the contents of a [RFC1995] IXFR message that would be sent to communicate the same zone incremental zone transfer over UDP or TCP i.e. the set of one or more difference sequences that follow the DNS Header in an IXFR message.

When processing the records received in a DSO-IXFR Message, the receiving client MUST validate that the zone being updated correspond with at least one currently active subscription on that session. Specifically, the SOA name and CLASS MUST match the SOA name and CLASS given in a SUBSCRIBE-XFR request, subject to the usual established DNS case-insensitivity for US-ASCII letters.

7.2.2. Fallback to AXFR

The format of the DSO-AXFR message is a standard DSO header with DSO-TYPE of DSO-AXFR (tentatively DSO Type Code 0x52) and the body is identical to a [RFC5936] AXFR response body.

TODO: More detail here.
If the SUBSCRIBE-XFR message contained no SOA value, the server MUST send a DSO-AXFR message as its first message on the connection.

Alternatively if incremental zone transfer is not available, the entire zone MAY be returned in a DSO-AXFR message.

QUESTION: Should we bother with a separate DSO-AXFR message or just allow full zone transfer inside the DSO-IXFR message as with [RFC1995] IXFR? A separate message type makes is more explicit and IXFR was constrained by having to respond to a IXFR request.

7.3. XuD UNSUBSCRIBE-XFR

To cancel an individual subscription without closing the entire DSO session, the client sends an UNSUBSCRIBE-XFR message over the established DSO session to the server. The UNSUBSCRIBE-XFR message is encoded as a DSO unidirectional message [RFC8490]. This specification defines a primary unidirectional DSO TLV for XuD UNSUBSCRIBE-XFR Messages (tentatively DSO Type Code 0x53).

A server MUST NOT initiate an UNSUBSCRIBE-XFR message. If a server does send an UNSUBSCRIBE-XFR message over a DSO session initiated by a client, this is a fatal error and the client should immediately abort the connection with a TLS close_notify alert.

7.3.1. UNSUBSCRIBE-XFR Message

An UNSUBSCRIBE-XFR unidirectional message begins with the standard DSO 12-byte header [RFC8490], followed by the UNSUBSCRIBE-XFR primary TLV. An UNSUBSCRIBE-XFR message is illustrated in Figure 5.

In accordance with the definition of DSO unidirectional messages, the MESSAGE ID field MUST be zero. There is no server response to an UNSUBSCRIBE-XFR message.

The other header fields MUST be set as described in the DSO specification [RFC8490]. The DNS OPCODE field contains the OPCODE value for DNS Stateful Operations (6). The four count fields MUST be zero, and the corresponding four sections MUST be empty (i.e., absent).

The DSO-TYPE is UNSUBSCRIBE-XFR (tentatively 0x53).

The DSO-LENGTH field contains the value 2, the length of the 2-octet MESSAGE ID contained in the DSO-DATA.

The DSO-DATA contains the value given in the MESSAGE ID field of an active SUBSCRIBE-XFR request. This is how the server knows which
SUBSCRIBE-XFR request is being cancelled. After receipt of the UNSUBSCRIBE-XFR message, the SUBSCRIBE-XFR request is no longer active.

It is allowable for the client to issue an UNSUBSCRIBE-XFR message for a previous SUBSCRIBE-XFR request for which the client has not yet received a SUBSCRIBE-XFR response. This is to allow for the case where a client starts and stops a subscription in less than the round-trip time to the server. The client is NOT required to wait for the SUBSCRIBE-XFR response before issuing the UNSUBSCRIBE-XFR message.

Consequently, it is possible for a server to receive an UNSUBSCRIBE-XFR message that does not match any currently active subscription. This can occur when a client sends a SUBSCRIBE-XFR request, which subsequently fails and returns an error code, but the client sent an UNSUBSCRIBE-XFR message before it became aware that the SUBSCRIBE-XFR request had failed. Because of this, servers MUST silently ignore UNSUBSCRIBE-XFR messages that do not match any currently active subscription.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+  \
|           MESSAGE ID (MUST BE ZERO)           |   \
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+    |
|QR| OPCODE(6) |         Z          |   RCODE   |    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+    |
|             QDCOUNT (MUST BE ZERO)               |    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+    |
|             ANCOUNT (MUST BE ZERO)               |    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+    |
|             NSCOUNT (MUST BE ZERO)               |    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+    |
|             ARCOUNT (MUST BE ZERO)               |   /
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+   /
| DSO-TYPE = UNSUBSCRIBE-XFR (tentatively 0x53) |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+    |
| DSO-LENGTH (2)                                     |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+    |
|             SUBSCRIBE-XFR MESSAGE ID              |   > DSO-DATA
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+    |
```

Figure 5: UNSUBSCRIBE-XFR Message

QUESTION: Do we need the equivalent of a RECONFIRM message from DNS PUSH Notifications [I-D.ietf-dnssd-push]?
7.4. Authentication

The authentication considerations are largely the same as those presented in [I-D.hzpa-dprive-xfr-over-tls].

7.5. Multi-primary configurations

The multi-primary considerations share some of the same issues as those presented in [I-D.hzpa-dprive-xfr-over-tls] but are different because the client is not performing SOA queries.

TODO: More detail required here.

7.6. DNS Stateful Operations TLV Context Summary

This document defines four new DSO TLVs. As suggested in Section 8.2 of the DNS Stateful Operations specification [RFC8490], the valid contexts of these new TLV types are summarized below.

The client TLV contexts are:

C-P: Client request message, primary TLV
C-U: Client unidirectional message, primary TLV
C-A: Client request or unidirectional message, additional TLV
CRP: Response back to client, primary TLV
CRA: Response back to client, additional TLV

<table>
<thead>
<tr>
<th>TLV Type</th>
<th>C-P</th>
<th>C-U</th>
<th>C-A</th>
<th>CRP</th>
<th>CRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBSCRIBE-XFR</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSO-IXFR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSO-AXFR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNSUBSCRIBE-XFR</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: DSO TLV Client Context Summary

The server TLV contexts are:

S-P: Server request message, primary TLV
S-U: Server unidirectional message, primary TLV
S-A: Server request or unidirectional message, additional TLV
SRP: Response back to server, primary TLV
SRA: Response back to server, additional TLV

+-----------------+-----+-----+-----+-----+-----+
| TLV Type        | S-P | S-U | S-A | SRP | SRA |
+-----------------+-----+-----+-----+-----+-----+
| SUBSCRIBE-XFR   |     |     |     |     |     |
| DSO-IXFR        |     |  X  |     |     |     |
| DSO-AXFR        |     |  X  |     |     |     |
| UNSUBSCRIBE-XFR |     |     |     |     |     |
+-----------------+-----+-----+-----+-----+-----+

Table 3: DSO TLV Server Context Summary

8. IANA Considerations

This document also defines four new DNS Stateful Operation TLV types to be recorded in the IANA DSO Type Code Registry.

+-----------------+----------+----------+--------------+------------+
|       Name      |  Value   |  Early   |    Status    | Definition |
|                 |          |   Data   |              |            |
+-----------------+----------+----------+--------------+------------+
|  SUBSCRIBE-XFR  |   TBA    |    NO    |  Standards   |  Section   |
|                 |  (0x50)  |          |    Track     |    7.1     |
|     DSO-IXFR    |   TBA    |    NA    |  Standards   |  Section   |
|                 |  (0x51)  |          |    Track     |    7.1     |
|     DSO-AXFR    |   TBA    |    NA    |  Standards   |  Section   |
|                 |  (0x51)  |          |    Track     |    7.2     |
| UNSUBSCRIBE-XFR |   TBA    |    NA    |  Standards   |  Section   |
|                 |  (0x52)  |          |    Track     |    7.2     |
+-----------------+----------+----------+--------------+------------+

Table 5: IANA DSO TLV Type Code Assignment

9. Implementation Considerations

TBD

10. Implementation Status

TBD
11. 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:

- Should DoT in this new case be required to use only TLS 1.3 and higher to avoid residual exposure?
- How should padding be used in IXFR?
- 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?

12. Acknowledgements

13. Changelog

draft-zatda-dprive-xfr-using-dso-00

- Initial commit

14. References

14.1. Normative References

[I-D.hzpa-dprive-xfr-over-tls]

[I-D.ietf-dnssd-push]

[RFC2119]

[RFC6973]
14.2. Informative References


14.3. URIs


Authors’ Addresses

Han Zhang
Salesforce
San Francisco, CA
United States

Email: hzhang@salesforce.com