             Usage Profiles for DNS over TLS and DNS over DTLS

Abstract

   This document discusses usage profiles, based on one or more
   authentication mechanisms, which can be used for DNS over Transport
   Layer Security (TLS) or Datagram TLS (DTLS). These profiles can
   increase the privacy of DNS transactions compared to using only
   cleartext DNS. This document also specifies new authentication
   mechanisms -- it describes several ways that a DNS client can use an
   authentication domain name to authenticate a (D)TLS connection to a
   DNS server. Additionally, it defines (D)TLS protocol profiles for
   DNS clients and servers implementing DNS over (D)TLS. This document
   updates RFC 7858.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 7841.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   https://www.rfc-editor.org/info/rfc8310.

Copyright Notice

Table of Contents

1. Introduction

   DNS privacy issues are discussed in [RFC7626]. The specific issues
   described in [RFC7626] that are most relevant to this document are

   o Passive attacks that eavesdrop on cleartext DNS transactions on
      the wire (Section 2.4 of [RFC7626]) and

   o Active attacks that redirect clients to rogue servers to monitor
      DNS traffic (Section 2.5.3 of [RFC7626]).

   Mitigating these attacks increases the privacy of DNS transactions;
   however, many of the other issues raised in [RFC7626] still apply.

   Two documents that provide ways to increase DNS privacy between DNS
   clients and DNS servers are

   o "Specification for DNS over Transport Layer Security (TLS)"
      [RFC7858], referred to here as simply "DNS over TLS".

   o "DNS over Datagram Transport Layer Security (DTLS)" [RFC8094],
      referred to here as simply "DNS over DTLS". Note that [RFC8094]
      is an Experimental specification.

   Both documents are limited in scope to communications between stub
   clients and recursive resolvers, and the same scope is applied to
   this document (see Sections 2 and 3). The proposals here might be
   adapted or extended in future to be used for recursive clients and
   authoritative servers, but this application was out of scope for the
   DNS PRIVate Exchange (dprive) Working Group charter at the time this
   document was published.

   This document specifies two usage profiles (Strict Privacy and
   Opportunistic Privacy) for DTLS [RFC6347] and TLS [RFC5246] that
   provide improved levels of mitigation for the attacks described above
   compared to using only cleartext DNS.

   Section 5 presents a generalized discussion of usage profiles by
   separating the usage profile, which is based purely on the security
   properties it offers the user, from the specific mechanism or
   mechanisms that are used for DNS server authentication. The profiles
   described are

   o A Strict Privacy profile, which requires an encrypted connection
      and successful authentication of the DNS server; this mitigates
      both passive eavesdropping and client redirection (at the expense
      of providing no DNS service if an encrypted, authenticated
      connection is not available).

   o An Opportunistic Privacy profile, which will attempt, but does not
      require, encryption and successful authentication; it therefore
      provides limited or no mitigation for such attacks but maximizes
      the chance of DNS service.

   The above usage profiles attempt authentication of the server using
   at least one authentication mechanism. Section 6.4 discusses how to
   combine authentication mechanisms to determine the overall
   authentication result. Depending on that overall authentication
   result (and whether encryption is available), the usage profile will
   determine if the connection should proceed, fall back, or fail.

   One authentication mechanism is already described in [RFC7858].
   [RFC7858] specifies an authentication mechanism for DNS over TLS that
   is based on Subject Public Key Info (SPKI) in the context of a
   specific case of a Strict Privacy profile using that single
   authentication mechanism. Therefore, the "out-of-band key-pinned
   privacy profile" described in [RFC7858] would qualify as a "Strict
   Privacy profile" that used SPKI pinning for authentication.

   This document extends the use of authentication based on SPKI
   pin sets, so that it is considered a general authentication mechanism
   that can be used with either DNS-over-(D)TLS usage profile. That is,
   the mechanism for SPKI pin sets as described in [RFC7858] MAY be used
   with DNS over (D)TLS.

   This document also describes a number of additional authentication
   mechanisms, all of which specify how a DNS client should authenticate
   a DNS server based on an "authentication domain name". In
   particular, the following topics are described:

   o How a DNS client can obtain the combination of an authentication
      domain name and IP address for a DNS server. See Section 7.

   o What acceptable credentials a DNS server can present to prove its
      identity for (D)TLS authentication based on a given authentication
      domain name. See Section 8.

   o How a DNS client can verify that any given credential matches the
      authentication domain name obtained for a DNS server. See
      Section 8.

   This document defines a (D)TLS protocol profile for use with DNS; see
   Section 9. This profile defines the configuration options and
   protocol extensions required of both parties to (1) optimize
   connection establishment and session resumption for transporting DNS
   and (2) support all currently specified authentication mechanisms.

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

   Several terms are used specifically in the context of this document:

   o DNS client: A DNS stub resolver or forwarder. In the case of a
      forwarder, the term "DNS client" is used to discuss the side that
      sends queries.

   o DNS server: A DNS recursive resolver or forwarder. In the case of
      a forwarder, the term "DNS server" is used to discuss the side
      that responds to queries. Note that, as used in this document,
      this term does not apply to authoritative servers.

   o Privacy-enabling DNS server: A DNS server that implements
      DNS over TLS [RFC7858] and may optionally implement DNS over DTLS
      [RFC8094]. The server should also offer at least one of the
      credentials described in Section 8 and implement the (D)TLS
      profile described in Section 9.

   o (D)TLS: Used for brevity; refers to both Transport Layer Security
      [RFC5246] and Datagram Transport Layer Security [RFC6347].
      Specific terms will be used for any text that applies to either
      protocol alone.

   o DNS over (D)TLS: Used for brevity; refers to both DNS over TLS
      [RFC7858] and DNS over DTLS [RFC8094]. Specific terms will be
      used for any text that applies to either protocol alone.

   o Authentication domain name: A domain name that can be used to
      authenticate a privacy-enabling DNS server. Sources of
      authentication domain names are discussed in Section 7.

   o SPKI pin sets: [RFC7858] describes the use of cryptographic
      digests to "pin" public key information in a manner similar to
      HTTP Public Key Pinning (HPKP) [RFC7469]. An SPKI pin set is a
      collection of these pins that constrains a DNS server.

   o Authentication information: Information a DNS client may use as
      the basis of an authentication mechanism. In this context, this
      information can be either

      * an SPKI pin set or

      * an authentication domain name

   o Reference identifier: A reference identifier as described in
      [RFC6125], constructed by the DNS client when performing TLS
      authentication of a DNS server.

   o Credential: Information available for a DNS server that proves its
      identity for authentication purposes. Credentials discussed here
      include

      * a PKIX certificate

      * a DNSSEC-validated chain to a TLSA record

      but may also include SPKI pin sets.

3. Scope

   This document is limited to describing

   o Usage profiles based on general authentication mechanisms.

   o The details of domain-name-based authentication of DNS servers by
      DNS clients (as defined in Section 2).

   o The (D)TLS profiles needed to support authentication in
      DNS over (D)TLS.

   As such, the following topics are out of scope for this document:

   o Authentication of authoritative servers by recursive resolvers.

   o Authentication of DNS clients by DNS servers.

   o The details of how to perform authentication based on SPKI
      pin sets. This is defined in [RFC7858].

   o Any server identifier other than domain names, including IP
      addresses, organizational names, country of origin, etc.

4. Discussion

   One way to mitigate eavesdropping on cleartext DNS transactions by
   passive attackers is to encrypt the query (and response). Such
   encryption typically provides integrity protection as a side effect;
   this means that on-path attackers cannot simply inject bogus DNS
   responses. To also mitigate active attackers pretending to be the
   server, the client must authenticate the (D)TLS connection to the
   server.

   This document discusses usage profiles, which provide differing
   levels of attack mitigation to DNS clients, based on the requirements
   for authentication and encryption, regardless of the context (for
   example, which network the client is connected to). A usage profile
   is a concept distinct from a usage policy or usage model; a usage
   policy or usage model might dictate which profile should be used in a
   particular context (enterprise vs. coffee shop), with a particular
   set of DNS servers or with reference to other external factors. A
   description of the variety of usage policies is out of scope for this
   document but may be the subject of future work.

   The term "privacy-enabling DNS server" is used throughout this
   document. This is a DNS server that

   o MUST implement DNS over TLS [RFC7858].

   o MAY implement DNS over DTLS [RFC8094].

   o SHOULD offer at least one of the credentials described in
      Section 8.

   o Implements the (D)TLS profile described in Section 9.

5. Usage Profiles

   A DNS client has a choice of usage profiles available to increase the
   privacy of DNS transactions. This choice is briefly discussed in
   both [RFC7858] and [RFC8094]. These usage profiles are

   o Strict Privacy profile: The DNS client requires both an encrypted
      and authenticated connection to a privacy-enabling DNS server. A
      hard failure occurs if this is not available. This requires the
      client to securely obtain authentication information it can use to
      authenticate the server. This profile mitigates both passive and
      active attacks, thereby providing the client with the best
      available privacy for DNS. This profile is discussed in detail in
      Section 6.6.

    o Opportunistic Privacy profile: The DNS client uses Opportunistic
       Security as described in [RFC7435].

       * "... the use of cleartext as the baseline communication
          security policy, with encryption and authentication negotiated
          and applied to the communication when available."

       As described in [RFC7435], it might result in

       * an encrypted and authenticated connection

       * an encrypted connection

       * a cleartext connection

       depending on the fallback logic of the client, the available
       authentication information, and the capabilities of the DNS
       server. In all these cases, the DNS client is willing to continue
       with a connection to the DNS server and perform resolution of
       queries. The use of Opportunistic Privacy is intended to support
       incremental deployment of increased privacy with a view to
       widespread adoption of the Strict Privacy profile. It should be
       employed when the DNS client might otherwise settle for cleartext;
       it provides the maximum protection available, depending on the
       combination of factors described above. If all the configured DNS
       servers are DNS privacy servers, then it can provide protection
       against passive attacks and might protect against active ones.

   Both profiles can include an initial meta-query (performed using
   Opportunistic Privacy) to obtain the IP address for the privacy-
   enabling DNS server to which the DNS client will subsequently
   connect. The rationale for permitting this for the Strict Privacy
   profile is that requiring such meta-queries to also be performed
   using the Strict Privacy profile would introduce significant
   deployment obstacles. However, it should be noted that in this
   scenario an active attack on the meta-query is possible. Such an
   attack could result in a Strict Privacy profile client connecting to
   a server it cannot authenticate (and therefore not obtaining DNS
   service) or an Opportunistic Privacy client connecting to a server
   controlled by the attacker. DNSSEC validation can detect the attack
   on the meta-query, which may result in the client not obtaining DNS
   service (for both usage profiles), depending on its DNSSEC validation
   policy. See Section 7.2 for more discussion.

   To compare the two usage profiles, Table 1 below shows a successful
   Strict Privacy profile alongside the three possible outcomes of an
   Opportunistic Privacy profile. In the best-case scenario for the
   Opportunistic Privacy profile (an authenticated and encrypted

connection), it is equivalent to the Strict Privacy profile. In the
worst-case scenario, it is equivalent to cleartext. Clients using
the Opportunistic Privacy profile SHOULD try for the best case but
MAY fall back to the intermediate case and, eventually, the worst-
case scenario, in order to obtain a response. One reason to fall
back without trying every available privacy-enabling DNS server is if
latency is more important than attack mitigation; see Appendix A.
The Opportunistic Privacy profile therefore provides varying
protection, depending on what kind of connection is actually used,
including no attack mitigation at all.

Note that there is no requirement in Opportunistic Security to notify
the user regarding what type of connection is actually used; the
"detection" described below is only possible if such connection
information is available. However, if it is available and the user
is informed that an unencrypted connection was used to connect to a
server, then the user should assume (detect) that the connection is
subject to both active and passive attacks, since the DNS queries are
sent in cleartext. This might be particularly useful if a new
connection to a certain server is unencrypted when all previous
connections were encrypted. Similarly, if the user is informed that
an encrypted but unauthenticated connection was used, then the user
can detect that the connection may be subject to active attacks. In
other words, for the cases where no protection is provided against an
attacker (N), it is possible to detect that an attack might be
happening (D). This is discussed in Section 6.5.

Usage Profile	Connection	Passive Attacker	Active Attacker
Strict	A, E	P	P
Opportunistic	A, E	P	P
Opportunistic	E	P	N, D
Opportunistic		N, D	N, D

P == Protection; N == No protection; D == Detection is possible;
A == Authenticated connection; E == Encrypted connection

Table 1: Attack Protection by Usage Profile and Type of Attacker

The Strict Privacy profile provides the best attack mitigation and
therefore SHOULD always be implemented in DNS clients that implement
the Opportunistic Privacy profile.

A DNS client that implements DNS over (D)TLS SHOULD NOT be configured
by default to use only cleartext.

The choice between the two profiles depends on a number of factors,
including which is more important to the particular client:

o DNS service, at the cost of no attack mitigation (Opportunistic
   Privacy) or

o Best available attack mitigation, at the potential cost of no DNS
   service (Strict Privacy).

Additionally, the two profiles require varying levels of
configuration (or a trusted relationship with a provider) and DNS
server capabilities; therefore, DNS clients will need to carefully
select which profile to use based on their communication needs.

A DNS server that implements DNS over (D)TLS SHOULD provide at least
one credential (Section 2) so that those DNS clients that wish to use
the Strict Privacy profile are able to do so.

5.1. DNS Resolution

A DNS client SHOULD select a particular usage profile when resolving
a query. A DNS client MUST NOT fall back from Strict Privacy to
Opportunistic Privacy during the resolution of a given query, as this
could invalidate the protection offered against attackers. It is
anticipated that DNS clients will use a particular usage profile for
all queries to all configured servers until an operational issue or
policy update dictates a change in the profile used.

6. Authentication in DNS over (D)TLS

This section describes authentication mechanisms and how they can be
used in either Strict or Opportunistic Privacy for DNS over (D)TLS.

6.1. DNS-over-(D)TLS Startup Configuration Problems

Many (D)TLS clients use PKIX authentication [RFC6125] based on an
authentication domain name for the server they are contacting. These
clients typically first look up the server's network address in the
DNS before making this connection. Such a DNS client therefore has a
bootstrap problem, as it will typically only know the IP address of
its DNS server.

In this case, before connecting to a DNS server, a DNS client needs
to learn the authentication domain name it should associate with the
IP address of a DNS server for authentication purposes. Sources of
authentication domain names are discussed in Section 7.

One advantage of this domain-name-based approach is that it
encourages the association of stable, human-recognizable identifiers
with secure DNS service providers.

6.2. Credential Verification

Verification of SPKI pin sets is discussed in [RFC7858].

In terms of domain-name-based verification, once an authentication
domain name is known for a DNS server, a choice of authentication
mechanisms can be used for credential verification. Section 8
discusses these mechanisms -- namely, PKIX certificate-based
authentication and DNS-Based Authentication of Named Entities (DANE)
-- in detail.

Note that the use of DANE adds requirements on the ability of the
client to get validated DNSSEC results. This is discussed in more
detail in Section 8.2.

6.3. Summary of Authentication Mechanisms

This section provides an overview of the various authentication
mechanisms. Table 2 below indicates how the DNS client obtains
information to use for authentication for each option: either
statically via direct configuration or dynamically. Of course, the
Opportunistic Privacy profile does not require authentication, and so
a client using that profile may choose to connect to a
privacy-enabling DNS server on the basis of just an IP address.

```
+---+-----------+------------+----------------------------------+
| # | Static    | Dynamically| Short name: Description           |
|   | Config    | Obtained   |                                  |
+---+-----------+------------+----------------------------------+
| 1 | SPKI + IP |            | SPKI: SPKI pin set(s) and IP     |
|   |           |            | address obtained out of band     |
|   |           |            | [RFC7858]                        |
|   |           |            |                                  |
| 2 | ADN + IP  |            | ADN: ADN and IP address obtained |
|   |           |            | out of band (see Section 7.1)    |
|   |           |            |                                  |
| 3 | ADN       | IP         | ADN only: Opportunistic Privacy  |
|   |           |            | meta-queries to a NP DNS server  |
|   |           |            | for A/AAAA (see Section 7.2)     |
|   |           |            |                                  |
| 4 |           | ADN + IP   | DHCP: DHCP configuration only (see|
|   |           |            | Section 7.3.1)                   |
|   |           |            |                                  |
| 5 | [ADN + IP]| [ADN + IP] | DANE: DNSSEC chain obtained via  |
|   |           | TLSA record| Opportunistic Privacy meta-queries|
|   |           |            | to NP DNS server (see Section    |
|   |           |            | 8.2.1)                           |
|   |           |            |                                  |
| 6 | [ADN + IP]| [ADN + IP] | TLS extension: DNSSEC chain      |
|   |           | TLSA record| provided by PE DNS server in TLS |
|   |           |            | DNSSEC chain extension (see      |
|   |           |            | Section 8.2.2)                   |
+---+-----------+------------+----------------------------------+
```

```
           SPKI == SPKI pin set(s); IP == IP Address;
       ADN == Authentication Domain Name; NP == Network-Provided;
      PE == Privacy-Enabling; [ ] == Data may be obtained either
                       statically or dynamically
```

Table 2: Overview of Authentication Mechanisms

The following summary attempts to present some key attributes of each
of the mechanisms (using the "Short name" from Table 2), indicating
attractive attributes with a "+" and undesirable attributes
with a "-".

1. SPKI

    + Minimal leakage (note that the ADN is always leaked in the
      Server Name Indication (SNI) field in the ClientHello in TLS
      when communicating with a privacy-enabling DNS server)

    - Overhead of ongoing key management required

2. ADN

    + Minimal leakage

    + One-off direct configuration only

3. ADN only

    + Minimal one-off direct configuration; only a human-recognizable
      domain name needed

    - A/AAAA meta-queries leaked to network-provided DNS server that
      may be subject to active attack (attack can be mitigated by
      DNSSEC validation)

4. DHCP

    + No static config

    - Requires a non-standard or future DHCP option in order to
      provide the ADN

    - Requires secure and trustworthy connection to DHCP server if
      used with a Strict Privacy profile

5. DANE

    The ADN and/or IP may be obtained statically or dynamically, and
    the relevant attributes of that method apply.

    + DANE options (e.g., matching on entire certificate)

    - Requires a DNSSEC-validating stub implementation (the
      deployment of which is limited at the time of this writing)

    - DNSSEC chain meta-queries leaked to network-provided DNS server
      that may be subject to active attack

6. TLS extension

    The ADN and/or IP may be obtained statically or dynamically, and
    the relevant attributes of that method apply.

    + Reduced latency compared with DANE

    + No network-provided DNS server required if ADN and IP
      statically configured

           + DANE options (e.g., matching on entire certificate)

           - Requires a DNSSEC-validating stub implementation

6.4. Combining Authentication Mechanisms

   This document does not make explicit recommendations about how an
   authentication mechanism based on SPKI pin sets should be combined
   with a domain-based mechanism from an operator perspective. However,
   it can be envisaged that a DNS server operator may wish to make both
   an SPKI pin set and an authentication domain name available to allow
   clients to choose which mechanism to use. Therefore, the following
   text provides guidance on how clients ought to behave if they choose
   to configure both, as is possible in HPKP [RFC7469].

   A DNS client that is configured with both an authentication domain
   name and an SPKI pin set for a DNS server SHOULD match on both a
   valid credential for the authentication domain name and a valid SPKI
   pin set (if both are available) when connecting to that DNS server.
   In this case, the client SHOULD treat individual SPKI pins as
   specified in Section 2.6 of [RFC7469] with regard to user-defined
   trust anchors. The overall authentication result SHOULD only be
   considered successful if both authentication mechanisms are
   successful.

6.5. Authentication in Opportunistic Privacy

   An Opportunistic Privacy Profile (based on Opportunistic Security
   [RFC7435]) that MAY be used for DNS over (D)TLS is described in
   [RFC7858] and is further specified in this document.

   DNS clients that issue queries under an Opportunistic Privacy profile
   and that know authentication information for a given privacy-enabling
   DNS server SHOULD try to authenticate the server using the mechanisms
   described here. This is useful for detecting (but not preventing)
   active attacks, since the fact that authentication information is
   available indicates that the server in question is a privacy-enabling
   DNS server to which it should be possible to establish an
   authenticated and encrypted connection. In this case, whilst a
   client cannot know the reason for an authentication failure, from a
   security standpoint the client should consider an active attack in
   progress and proceed under that assumption. For example, a client
   that implements a nameserver selection algorithm that preferentially
   uses nameservers that successfully authenticated (see Section 5)
   might not continue to use the failing server if there were
   alternative servers available.

Attempting authentication is also useful for debugging or diagnostic
purposes if there are means to report the result. This information
can provide a basis for a DNS client to switch to (preferred) Strict
Privacy where it is viable, e.g., where all the configured servers
support DNS over (D)TLS and successfully authenticate.

6.6. Authentication in Strict Privacy

To authenticate a privacy-enabling DNS server, a DNS client needs to
know authentication information for each server it is willing to
contact. This is necessary to protect against active attacks that
attempt to redirect clients to rogue DNS servers.

A DNS client requiring Strict Privacy MUST use either (1) one of the
sources listed in Section 7, to obtain an authentication domain name
for the server it contacts or (2) an SPKI pin set as described in
[RFC7858].

A DNS client requiring Strict Privacy MUST only attempt to connect to
DNS servers for which at least one piece of authentication
information is known. The client MUST use the available verification
mechanisms described in Section 8 to authenticate the server and MUST
abort connections to a server when no verification mechanism
succeeds.

With Strict Privacy, the DNS client MUST NOT commence sending DNS
queries until at least one of the privacy-enabling DNS servers
becomes available.

A privacy-enabling DNS server may be temporarily unavailable when
configuring a network. For example, for clients on networks that
require registration through web-based login (a.k.a. "captive
portals"), such registration may rely on DNS interception and
spoofing. Techniques such as those used by dnssec-trigger
[dnssec-trigger] MAY be used during network configuration, with the
intent to transition to the designated privacy-enabling DNS servers
after captive-portal registration. If using a Strict Privacy
profile, the system MUST alert by some means that the DNS is not
private during such a bootstrap operation.

6.7. Implementation Guidance

Section 9 describes the (D)TLS profile for DNS over (D)TLS.
Additional considerations relating to general implementation
guidelines are discussed in both Section 11 and Appendix A.

7. Sources of Authentication Domain Names

7.1. Full Direct Configuration

   DNS clients may be directly and securely provisioned with the
   authentication domain name of each privacy-enabling DNS server -- for
   example, using a client-specific configuration file or API.

   In this case, direct configuration for a DNS client would consist of
   both an IP address and an authentication domain name for each DNS
   server that were obtained via an out-of-band mechanism.

7.2. Direct Configuration of ADN Only

   A DNS client may be configured directly and securely with only the
   authentication domain name of each of its privacy-enabling DNS
   servers -- for example, using a client-specific configuration file
   or API.

   A DNS client might learn of a default recursive DNS resolver from an
   untrusted source (such as DHCP's DNS Recursive Name Server option
   [RFC3646]). It can then use meta-queries performed using an
   Opportunistic Privacy profile to an untrusted recursive DNS resolver
   to establish the IP address of the intended privacy-enabling DNS
   resolver by doing a lookup of A/AAAA records. A DNSSEC-validating
   client SHOULD apply the same validation policy to the A/AAAA
   meta-queries as it does to other queries. A client that does not
   validate DNSSEC SHOULD apply the same policy (if any) to the A/AAAA
   meta-queries as it does to other queries. Private DNS resolution can
   now be done by the DNS client against the pre-configured privacy-
   enabling DNS resolver, using the IP address obtained from the
   untrusted DNS resolver.

   A DNS client so configured that successfully connects to a privacy-
   enabling DNS server MAY choose to locally cache the server host IP
   addresses in order to not have to repeat the meta-query.

7.3. Dynamic Discovery of ADN

   This section discusses the general case of a DNS client discovering
   both the authentication domain name and IP address dynamically. At
   the time of this writing, this is not possible by any standard means.
   However, since, for example, a future DHCP extension could (in
   principle) provide this mechanism, the required security properties
   of such mechanisms are outlined here.

   When using a Strict Privacy profile, the dynamic discovery technique
   used as a source of authentication domain names MUST be considered
   secure and trustworthy. This requirement does not apply when using
   an Opportunistic Privacy profile, given the security expectation of
   that profile.

7.3.1. DHCP

   In the typical case today, a DHCP server [RFC2131] [RFC3315] provides
   a list of IP addresses for DNS resolvers (see Section 3.8 of
   [RFC2132]) but does not provide an authentication domain name for the
   DNS resolver, thus preventing the use of most of the authentication
   methods described here (all of those that are based on a mechanism
   with ADN; see Table 2).

   This document does not specify or request any DHCP extension to
   provide authentication domain names. However, if one is developed in
   future work, the issues outlined in Section 8 of [RFC7227] should be
   taken into account, as should the security considerations discussed
   in Section 23 of [RFC3315].

   This document does not attempt to describe secured and trusted
   relationships to DHCP servers, as this is purely a DHCP issue (and
   still open, at the time of this writing). Whilst some implementation
   work is in progress to secure IPv6 connections for DHCP, IPv4
   connections have received little or no implementation attention in
   this area.

8. Credential Verification Based on Authentication Domain Name

8.1. Authentication Based on PKIX Certificate

   When a DNS client configured with an authentication domain name
   connects to its configured DNS server over (D)TLS, the server may
   present it with a PKIX certificate. In order to ensure proper
   authentication, DNS clients MUST verify the entire certification path
   per [RFC5280]. The DNS client additionally uses validation
   techniques as described in [RFC6125] to compare the domain name to
   the certificate provided.

   A DNS client constructs one reference identifier for the server based
   on the authentication domain name: a DNS-ID, which is simply the
   authentication domain name itself.

   If the reference identifier is found (as described in Section 6 of
   [RFC6125]) in the PKIX certificate's subjectAltName extension, the
   DNS client should accept the certificate for the server.

A compliant DNS client MUST only inspect the certificate's
subjectAltName extension for the reference identifier. In
particular, it MUST NOT inspect the Subject field itself.

8.2. DANE

DANE [RFC6698] provides various mechanisms using DNSSEC to anchor
trust for certificates and raw public keys. However, this requires
the DNS client to have an authentication domain name (which must be
obtained via a trusted source) for the DNS privacy server.

This section assumes a solid understanding of both DANE [RFC6698] and
DANE operations [RFC7671]. A few pertinent issues covered in these
documents are outlined here as useful pointers, but familiarity with
both of these documents in their entirety is expected.

Note that [RFC6698] says

   Clients that validate the DNSSEC signatures themselves MUST use
   standard DNSSEC validation procedures. Clients that rely on
   another entity to perform the DNSSEC signature validation MUST use
   a secure mechanism between themselves and the validator.

Note that [RFC7671] covers the following topics:

o Sections 4.1 ("Opportunistic Security and PKIX Usages") and 14
   ("Security Considerations") of [RFC7671], which both discuss the
   use of schemes based on trust anchors and end entities (PKIX-TA(0)
   and PKIX-EE(1), respectively) for Opportunistic Security.

o Section 5 ("Certificate-Usage-Specific DANE Updates and
   Guidelines") of [RFC7671] -- specifically, Section 5.1 of
   [RFC7671], which outlines the combination of certificate usage
   DANE-EE(3) and selector SPKI(1) with raw public keys [RFC7250].
   Section 5.1 of [RFC7671] also discusses the security implications
   of this mode; for example, it discusses key lifetimes and
   specifies that validity period enforcement is based solely on the
   TLSA RRset properties for this case.

o Section 13 ("Operational Considerations") of [RFC7671], which
   discusses TLSA TTLs and signature validity periods.

The specific DANE record for a DNS privacy server would take the form

   _853._tcp.[authentication-domain-name] for TLS

   _853._udp.[authentication-domain-name] for DTLS

8.2.1. Direct DNS Meta-Queries

   The DNS client MAY choose to perform the DNS meta-queries to retrieve
   the required DANE records itself. The DNS meta-queries for such DANE
   records MAY use the Opportunistic Privacy profile or be in the clear
   to avoid trust recursion. The records MUST be validated using DNSSEC
   as described in [RFC6698].

8.2.2. TLS DNSSEC Chain Extension

   The DNS client MAY offer the TLS extension described in
   [TLS-DNSSEC-Chain-Ext]. If the DNS server supports this extension,
   it can provide the full chain to the client in the handshake.

   If the DNS client offers the TLS DNSSEC chain extension, it MUST be
   capable of validating the full DNSSEC authentication chain down to
   the leaf. If the supplied DNSSEC chain does not validate, the client
   MUST ignore the DNSSEC chain and validate only via other supplied
   credentials.

9. (D)TLS Protocol Profile

   This section defines the (D)TLS protocol profile of DNS over (D)TLS.

   Clients and servers MUST adhere to the (D)TLS implementation
   recommendations and security considerations of [RFC7525], except with
   respect to the (D)TLS version.

   Since encryption of DNS using (D)TLS is a greenfield deployment, DNS
   clients and servers MUST implement only (D)TLS 1.2 or later. For
   example, implementing (D)TLS 1.3 [TLS-1.3] [DTLS-1.3] is also an
   option.

   Implementations MUST NOT offer or provide TLS compression, since
   compression can leak significant amounts of information, especially
   to a network observer capable of forcing the user to do an arbitrary
   DNS lookup in the style of the Compression Ratio Info-leak Made Easy
   (CRIME) attacks [CRIME].

Implementations compliant with this profile MUST implement the
following items:

o TLS session resumption without server-side state [RFC5077], which
   eliminates the need for the server to retain cryptographic state
   for longer than necessary. (This statement updates [RFC7858].)

o Raw public keys [RFC7250], which reduce the size of the
   ServerHello and can be used by servers that cannot obtain
   certificates (e.g., DNS servers on private networks). A client
   MUST only indicate support for raw public keys if it has an SPKI
   pin set pre-configured (for interoperability reasons).

Implementations compliant with this profile SHOULD implement the
following items:

o TLS False Start [RFC7918], which reduces round trips by allowing
   the TLS second flight of messages (ChangeCipherSpec) to also
   contain the (encrypted) DNS query.

o The Cached Information Extension [RFC7924], which avoids
   transmitting the server's certificate and certificate chain if the
   client has cached that information from a previous TLS handshake.

Guidance specific to TLS is provided in [RFC7858], and guidance
specific to DTLS is provided in [RFC8094].

10. IANA Considerations

   This document does not require any IANA actions.

11. Security Considerations

   Security considerations discussed in [RFC7525], [RFC8094], and
   [RFC7858] apply to this document.

   DNS clients SHOULD implement (1) support for the mechanisms described
   in Section 8.2 and (2) offering a configuration option that limits
   authentication to using only those mechanisms (i.e., with no fallback
   to pure PKIX-based authentication) such that authenticating solely
   via the PKIX infrastructure can be avoided.

11.1. Countermeasures to DNS Traffic Analysis

   This section makes suggestions for measures that can reduce the
   ability of attackers to infer information pertaining to encrypted
   client queries by other means (e.g., via an analysis of encrypted
   traffic size or via monitoring of the unencrypted traffic from a DNS
   recursive resolver to an authoritative server).

   DNS-over-(D)TLS clients and servers SHOULD implement the following
   relevant DNS extensions:

   o Extension Mechanisms for DNS (EDNS(0)) padding [RFC7830], which
      allows encrypted queries and responses to hide their size, making
      analysis of encrypted traffic harder.

   Guidance on padding policies for EDNS(0) is provided in
   [EDNS0-Pad-Policies].

   DNS-over-(D)TLS clients SHOULD implement the following relevant DNS
   extensions:

   o Privacy election per [RFC7871] ("Client Subnet in DNS Queries").
      If a DNS client does not include an edns-client-subnet EDNS0
      option with SOURCE PREFIX-LENGTH set to 0 in a query, the DNS
      server may potentially leak client address information to the
      upstream authoritative DNS servers. A DNS client ought to be able
      to inform the DNS resolver that it does not want any address
      information leaked, and the DNS resolver should honor that
      request.

12. References

12.1. Normative References

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119,
              DOI 10.17487/RFC2119, March 1997,
              <https://www.rfc-editor.org/info/rfc2119>.

   [RFC5077] Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig,
              "Transport Layer Security (TLS) Session Resumption without
              Server-Side State", RFC 5077, DOI 10.17487/RFC5077,
              January 2008, <https://www.rfc-editor.org/info/rfc5077>.

   [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security
              (TLS) Protocol Version 1.2", RFC 5246,
              DOI 10.17487/RFC5246, August 2008,
              <https://www.rfc-editor.org/info/rfc5246>.

   [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S.,
              Housley, R., and W. Polk, "Internet X.509 Public Key
              Infrastructure Certificate and Certificate Revocation List
              (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008,
              <https://www.rfc-editor.org/info/rfc5280>.

   [RFC6125] Saint-Andre, P. and J. Hodges, "Representation and
              Verification of Domain-Based Application Service Identity
              within Internet Public Key Infrastructure Using X.509
              (PKIX) Certificates in the Context of Transport Layer
              Security (TLS)", RFC 6125, DOI 10.17487/RFC6125,
              March 2011, <https://www.rfc-editor.org/info/rfc6125>.

   [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer
              Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347,
              January 2012, <https://www.rfc-editor.org/info/rfc6347>.

   [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication
              of Named Entities (DANE) Transport Layer Security (TLS)
              Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698,
              August 2012, <https://www.rfc-editor.org/info/rfc6698>.

   [RFC7250] Wouters, P., Ed., Tschofenig, H., Ed., Gilmore, J.,
              Weiler, S., and T. Kivinen, "Using Raw Public Keys in
              Transport Layer Security (TLS) and Datagram Transport
              Layer Security (DTLS)", RFC 7250, DOI 10.17487/RFC7250,
              June 2014, <https://www.rfc-editor.org/info/rfc7250>.

   [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre,
              "Recommendations for Secure Use of Transport Layer
              Security (TLS) and Datagram Transport Layer Security
              (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525,
              May 2015, <https://www.rfc-editor.org/info/rfc7525>.

   [RFC7671] Dukhovni, V. and W. Hardaker, "The DNS-Based
              Authentication of Named Entities (DANE) Protocol: Updates
              and Operational Guidance", RFC 7671, DOI 10.17487/RFC7671,
              October 2015, <https://www.rfc-editor.org/info/rfc7671>.

   [RFC7830] Mayrhofer, A., "The EDNS(0) Padding Option", RFC 7830,
              DOI 10.17487/RFC7830, May 2016,
              <https://www.rfc-editor.org/info/rfc7830>.

   [RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D.,
              and P. Hoffman, "Specification for DNS over Transport
              Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858,
              May 2016, <https://www.rfc-editor.org/info/rfc7858>.

   [RFC7918] Langley, A., Modadugu, N., and B. Moeller, "Transport
              Layer Security (TLS) False Start", RFC 7918,
              DOI 10.17487/RFC7918, August 2016,
              <https://www.rfc-editor.org/info/rfc7918>.

   [RFC7924] Santesson, S. and H. Tschofenig, "Transport Layer Security
              (TLS) Cached Information Extension", RFC 7924,
              DOI 10.17487/RFC7924, July 2016,
              <https://www.rfc-editor.org/info/rfc7924>.

   [RFC8094] Reddy, T., Wing, D., and P. Patil, "DNS over Datagram
              Transport Layer Security (DTLS)", RFC 8094,
              DOI 10.17487/RFC8094, February 2017,
              <https://www.rfc-editor.org/info/rfc8094>.

   [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in
              RFC 2119 Key Words", BCP 14, RFC 8174,
              DOI 10.17487/RFC8174, May 2017,
              <https://www.rfc-editor.org/info/rfc8174>.

12.2. Informative References

   [CRIME]   Rizzo, J. and T. Duong, "The CRIME Attack", Ekoparty
              Security Conference, 2012,
              <https://www.ekoparty.org/archivo/2012/eko8-CRIME.pdf>.

   [dnssec-trigger]
              NLnetLabs, "Dnssec-Trigger", December 2017,
              <https://www.nlnetlabs.nl/projects/dnssec-trigger/>.

   [DTLS-1.3]
              Rescorla, E., Tschofenig, H., and N. Modadugu, "The
              Datagram Transport Layer Security (DTLS) Protocol
              Version 1.3", Work in Progress, draft-ietf-tls-dtls13-26,
              March 2018.

   [EDNS0-Pad-Policies]
              Mayrhofer, A., "Padding Policy for EDNS(0)", Work in
              Progress, draft-ietf-dprive-padding-policy-04,
              February 2018.

   [RFC2131] Droms, R., "Dynamic Host Configuration Protocol",
              RFC 2131, DOI 10.17487/RFC2131, March 1997,
              <https://www.rfc-editor.org/info/rfc2131>.

   [RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor
              Extensions", RFC 2132, DOI 10.17487/RFC2132, March 1997,
              <https://www.rfc-editor.org/info/rfc2132>.

   [RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins,
              C., and M. Carney, "Dynamic Host Configuration Protocol
              for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315,
              July 2003, <https://www.rfc-editor.org/info/rfc3315>.

   [RFC3646] Droms, R., Ed., "DNS Configuration options for Dynamic
              Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646,
              DOI 10.17487/RFC3646, December 2003,
              <https://www.rfc-editor.org/info/rfc3646>.

   [RFC7227] Hankins, D., Mrugalski, T., Siodelski, M., Jiang, S., and
              S. Krishnan, "Guidelines for Creating New DHCPv6 Options",
              BCP 187, RFC 7227, DOI 10.17487/RFC7227, May 2014,
              <https://www.rfc-editor.org/info/rfc7227>.

   [RFC7435] Dukhovni, V., "Opportunistic Security: Some Protection
              Most of the Time", RFC 7435, DOI 10.17487/RFC7435,
              December 2014, <https://www.rfc-editor.org/info/rfc7435>.

   [RFC7469] Evans, C., Palmer, C., and R. Sleevi, "Public Key Pinning
              Extension for HTTP", RFC 7469, DOI 10.17487/RFC7469,
              April 2015, <https://www.rfc-editor.org/info/rfc7469>.

   [RFC7626] Bortzmeyer, S., "DNS Privacy Considerations", RFC 7626,
              DOI 10.17487/RFC7626, August 2015,
              <https://www.rfc-editor.org/info/rfc7626>.

   [RFC7871] Contavalli, C., van der Gaast, W., Lawrence, D., and
              W. Kumari, "Client Subnet in DNS Queries", RFC 7871,
              DOI 10.17487/RFC7871, May 2016,
              <https://www.rfc-editor.org/info/rfc7871>.

   [TLS-1.3] Rescorla, E., "The Transport Layer Security (TLS) Protocol
              Version 1.3", Work in Progress, draft-ietf-tls-tls13-27,
              March 2018.

   [TLS-DNSSEC-Chain-Ext]
              Shore, M., Barnes, R., Huque, S., and W. Toorop, "A DANE
              Record and DNSSEC Authentication Chain Extension for TLS",
              Work in Progress, draft-ietf-tls-dnssec-chain-
              extension-06, January 2018.

Appendix A. Server Capability Probing and Caching by DNS Clients

   This section presents a non-normative discussion of how DNS clients
   might probe for, and cache capabilities of, privacy-enabling DNS
   servers.

   Deployment of both DNS over TLS and DNS over DTLS will be gradual.
   Not all servers will support one or both of these protocols, and the
   well-known port might be blocked by some middleboxes. Clients will
   be expected to keep track of servers that support DNS over TLS and/or
   DNS over DTLS, as well as those that have been previously
   authenticated.

   If no server capability information is available, then (unless
   otherwise specified by the configuration of the DNS client) DNS
   clients that implement both TLS and DTLS should try to authenticate
   using both protocols before failing or falling back to an
   unauthenticated or cleartext connection. DNS clients using an
   Opportunistic Privacy profile should try all available servers
   (possibly in parallel) in order to obtain an authenticated and
   encrypted connection before falling back. (RATIONALE: This approach
   can increase latency while discovering server capabilities but
   maximizes the chance of sending the query over an authenticated and
   encrypted connection.)

Authors' Addresses

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

   Email: sara@sinodun.com
   URI: https://www.sinodun.com/

   Daniel Kahn Gillmor
   ACLU
   125 Broad Street, 18th Floor
   New York, NY 10004
   United States of America

   Email: dkg@fifthhorseman.net

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

   Email: TirumaleswarReddy_Konda@McAfee.com