SMTP security via opportunistic DANE TLS
draft-ietf-dane-smtp-with-dane-08

Abstract

   This memo describes a downgrade-resistant protocol for SMTP transport
   security between Mail Transfer Agents (MTAs) based on the DNS-Based
   Authentication of Named Entities (DANE) TLSA DNS record. Adoption of
   this protocol enables an incremental transition of the Internet email
   backbone to one using encrypted and authenticated Transport Layer
   Security (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 October 25, 2014.

Copyright Notice

Table of Contents

1. Introduction

This memo specifies a new connection security model for Message
Transfer Agents (MTAs). This model is motivated by key features of
inter-domain SMTP delivery, in particular the fact that the
destination server is selected indirectly via DNS Mail Exchange (MX)
records and that with MTA to MTA SMTP the use of TLS is generally
opportunistic.

Problems with existing use of TLS in MTA to MTA SMTP that motivate this specification are described in Section 1.3. The specification itself follows in Section 2. Then, in Section 3, we discuss application of DANE TLS to destinations for which channel integrity and confidentiality are mandatory. In Section 4 we briefly comment on potential applicability of this specification to Message User Agents.

1.1. 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 [RFC2119].

The following terms or concepts are used through the document:

Man-in-the-middle or MITM attack: Active modification of network
   traffic by an adversary able to thereby compromise the
   confidentiality or integrity of the data.

secure, bogus, insecure, indeterminate: DNSSEC validation results,
   as defined in Section 4.3 of [RFC4035].

Validating Security-Aware Stub Resolver and   Non-Validating
Security-Aware Stub Resolver:
   Capabilities of the stub resolver in use as defined in [RFC4033];
   note that this specification requires the use of a Security-Aware
   Stub Resolver; Security-Oblivious stub-resolvers MUST NOT be used.

opportunistic DANE TLS: Best-effort use of TLS, resistant to
   downgrade attacks for destinations with DNSSEC-validated TLSA
   records. When opportunistic DANE TLS is determined to be
   unavailable, clients should fall back to opportunistic TLS below.
   Opportunistic DANE TLS requires support for DNSSEC, DANE and
   STARTTLS on the client side and STARTTLS plus a DNSSEC published
   TLSA record on the server side.

(pre-DANE) opportunistic TLS: Best-effort use of TLS that is
   generally vulnerable to DNS forgery and STARTTLS downgrade
   attacks. When a TLS-encrypted communication channel is not
   available, message transmission takes place in the clear. MX
   record indirection generally precludes authentication even when
   TLS is available.

reference identifier: (Special case of [RFC6125] definition). One
   of the domain names associated by the SMTP client with the
   destination SMTP server for performing name checks on the server

certificate. When name checks are applicable, at least one of the reference identifiers MUST match an [RFC6125] DNS-ID (or if none are present the [RFC6125] CN-ID) of the server certificate (see Section 2.3.2.3).

MX hostname: The RRDATA of an MX record consists of a 16 bit preference followed by a Mail Exchange domain name (see [RFC1035], Section 3.3.9). We will use the term "MX hostname" to refer to the latter, that is, the DNS domain name found after the preference value in an MX record. Thus an "MX hostname" is specifically a reference to a DNS domain name, rather than any host that bears that name.

delayed delivery: Email delivery is a multi-hop store & forward process. When an MTA is unable forward a message that may become deliverable later, the message is queued and delivery is retried periodically. Some MTAs may be configured with a fallback next-hop destination that handles messages that the MTA would otherwise queue and retry. In these cases, messages that would otherwise have to be delayed, may be sent to the fallback next-hop destination instead. The fallback destination may itself be subject to opportunistic or mandatory DANE TLS as though it were the original message destination.

original next hop destination: The logical destination for mail delivery. By default this is the domain portion of the recipient address, but MTAs may be configured to forward mail for some or all recipients via designated relays. The original next hop destination is, respectively, either the recipient domain or the associated configured relay.

MTA: Message Transfer Agent ([RFC5598], Section 4.3.2).

MSA: Message Submission Agent ([RFC5598], Section 4.3.1).

MUA: Message User Agent ([RFC5598], Section 4.2.1).

RR: A DNS Resource Record

RRset: A set of DNS Resource Records for a particular class, domain and record type.

1.2. Background

The Domain Name System Security Extensions (DNSSEC) add data origin authentication, data integrity and data non-existence proofs to the Domain Name System (DNS). DNSSEC is defined in [RFC4033], [RFC4034] and [RFC4035].

   As described in the introduction of [RFC6698], TLS authentication via
   the existing public Certification Authority (CA) PKI suffers from an
   over-abundance of trusted parties capable of issuing certificates for
   any domain of their choice. DANE leverages the DNSSEC infrastructure
   to publish trusted public keys and certificates for use with the
   Transport Layer Security (TLS) [RFC5246] protocol via a new "TLSA"
   DNS record type. With DNSSEC each domain can only vouch for the keys
   of its directly delegated sub-domains.

   The TLS protocol enables secure TCP communication. In the context of
   this memo, channel security is assumed to be provided by TLS. Used
   without authentication, TLS provides only privacy protection against
   eavesdropping attacks. With authentication, TLS also provides data
   integrity protection to guard against MITM attacks.

1.3. SMTP channel security

   With HTTPS, Transport Layer Security (TLS) employs X.509 certificates
   [RFC5280] issued by one of the many Certificate Authorities (CAs)
   bundled with popular web browsers to allow users to authenticate
   their "secure" websites. Before we specify a new DANE TLS security
   model for SMTP, we will explain why a new security model is needed.
   In the process, we will explain why the familiar HTTPS security model
   is inadequate to protect inter-domain SMTP traffic.

   The subsections below outline four key problems with applying
   traditional PKI to SMTP that are addressed by this specification.
   Since SMTP channel security policy is not explicitly specified in
   either the recipient address or the MX record, a new signaling
   mechanism is required to indicate when channel security is possible
   and should be used. The publication of TLSA records allows server
   operators to securely signal to SMTP clients that TLS is available
   and should be used. DANE TLSA makes it possible to simultaneously
   discover which destination domains support secure delivery via TLS
   and how to verify the authenticity of the associated SMTP services,
   providing a path forward to ubiquitous SMTP channel security.

1.3.1. STARTTLS downgrade attack

The Simple Mail Transfer Protocol (SMTP) [RFC5321] is a single-hop
protocol in a multi-hop store & forward email delivery process. SMTP
envelope recipient addresses are not transport addresses and are
security-agnostic. Unlike the Hypertext Transfer Protocol (HTTP) and
its corresponding secured version, HTTPS, where the use of TLS is
signalled via the URI scheme, email recipient addresses do not
directly signal transport security policy. Indeed, no such signaling
could work well with SMTP since TLS encryption of SMTP protects email
traffic on a hop-by-hop basis while email addresses could only
express end-to-end policy.

With no mechanism available to signal transport security policy, SMTP
relays employ a best-effort "opportunistic" security model for TLS.
A single SMTP server TCP listening endpoint can serve both TLS and
non-TLS clients; the use of TLS is negotiated via the SMTP STARTTLS
command ([RFC3207]). The server signals TLS support to the client
over a cleartext SMTP connection, and, if the client also supports
TLS, it may negotiate a TLS encrypted channel to use for email
transmission. The server's indication of TLS support can be easily
suppressed by an MITM attacker. Thus pre-DANE SMTP TLS security can
be subverted by simply downgrading a connection to cleartext. No TLS
security feature, such as the use of PKIX, can prevent this. The
attacker can simply disable TLS.

1.3.2. Insecure server name without DNSSEC

With SMTP, DNS Mail Exchange (MX) records abstract the next-hop
transport endpoint and allow administrators to specify a set of
target servers to which SMTP traffic should be directed for a given
domain.

A PKIX TLS client is vulnerable to MITM attacks unless it verifies
that the server's certificate binds the public key to a name that
matches one of the client's reference identifiers. A natural choice
of reference identifier is the server's domain name. However, with
SMTP, server names are obtained indirectly via MX records. Without
DNSSEC, the MX lookup is vulnerable to MITM and DNS cache poisoning
attacks. Active attackers can forge DNS replies with fake MX records
and can redirect email to servers with names of their choice.
Therefore, secure verification of SMTP TLS certificates matching the
server name is not possible without DNSSEC.

One might try to harden TLS for SMTP against DNS attacks by using the
envelope recipient domain as a reference identifier and requiring
each SMTP server to possess a trusted certificate for the envelope
recipient domain rather than the MX hostname. Unfortunately, this is
impractical as email for many domains is handled by third parties
that are not in a position to obtain certificates for all the domains

they serve. Deployment of the Server Name Indication (SNI) extension
to TLS (see [RFC6066] Section 3) is no panacea, since SNI key
management is operationally challenging except when the email service
provider is also the domain's registrar and its certificate issuer;
this is rarely the case for email.

Since the recipient domain name cannot be used as the SMTP server
reference identifier, and neither can the MX hostname without DNSSEC,
large-scale deployment of authenticated TLS for SMTP requires that
the DNS be secure.

Since SMTP security depends critically on DNSSEC, it is important to
point out that consequently SMTP with DANE is the most conservative
possible trust model. It trusts only what must be trusted and no
more. Adding any other trusted actors to the mix can only reduce
SMTP security. A sender may choose to further harden DNSSEC for
selected high-value receiving domains, by configuring explicit trust
anchors for those domains instead of relying on the chain of trust
from the root domain. Detailed discussion of DNSSEC security
practices is out of scope for this document.

1.3.3. Sender policy does not scale

Sending systems are in some cases explicitly configured to use TLS
for mail sent to selected peer domains. This requires sending MTAs
to be configured with appropriate subject names or certificate
content digests to expect in the presented server certificates.
Because of the heavy administrative burden, such statically
configured SMTP secure channels are used rarely (generally only
between domains that make bilateral arrangements with their business
partners). Internet email, on the other hand, requires regularly
contacting new domains for which security configurations cannot be
established in advance.

The abstraction of the SMTP transport endpoint via DNS MX records,
often across organization boundaries, limits the use of public CA PKI
with SMTP to a small set of sender-configured peer domains. With
little opportunity to use TLS authentication, sending MTAs are rarely
configured with a comprehensive list of trusted CAs. SMTP services
that support STARTTLS often deploy X.509 certificates that are self-
signed or issued by a private CA.

1.3.4. Too many certification authorities

Even if it were generally possible to determine a secure server name,
the SMTP client would still need to verify that the server's
certificate chain is issued by a trusted Certification Authority (a
trust anchor). MTAs are not interactive applications where a human

operator can make a decision (wisely or otherwise) to selectively
disable TLS security policy when certificate chain verification
fails. With no user to "click OK", the MTAs list of public CA trust
anchors would need to be comprehensive in order to avoid bouncing
mail addressed to sites that employ unknown Certification
Authorities.

On the other hand, each trusted CA can issue certificates for any
domain. If even one of the configured CAs is compromised or operated
by an adversary, it can subvert TLS security for all destinations.
Any set of CAs is simultaneously both overly inclusive and not
inclusive enough.

2. Opportunistic DANE TLS

Neither email addresses nor MX hostnames signal a requirement for
either secure or cleartext transport. Therefore, aside from a few
manually configured exceptions, SMTP transport security is of
necessity opportunistic.

This specification uses the presence of DANE TLSA records to securely
signal TLS support and to publish the means by which SMTP clients can
successfully authenticate legitimate SMTP servers. This becomes
"opportunistic DANE TLS" and is resistant to downgrade and MITM
attacks, and enables an incremental transition of the email backbone
to authenticated TLS delivery, with increased global protection as
adoption increases.

With opportunistic DANE TLS, traffic from SMTP clients to domains
that publish "usable" DANE TLSA records in accordance with this memo
is authenticated and encrypted. Traffic from non-compliant clients
or to domains that do not publish TLSA records will continue to be
sent in the same manner as before, via manually configured security,
(pre-DANE) opportunistic TLS or just cleartext SMTP.

2.1. DNS errors, bogus and indeterminate responses

An SMTP client that implements opportunistic DANE TLS per this
specification depends critically on the integrity of DNSSEC lookups,
as discussed in Section 1.3. This section lists the DNS resolver
requirements needed to avoid downgrade attacks when using
opportunistic DANE TLS.

A DNS lookup may signal an error or return a definitive answer. A
security-aware resolver must be used for this specification.
Security-aware resolvers will indicate the security status of a DNS
RRset with one of four possible values defined in Section 4.3 of
[RFC4035]: "secure", "insecure", "bogus" and "indeterminate". In
[RFC4035] the meaning of the "indeterminate" security status is:

   An RRset for which the resolver is not able to determine whether
   the RRset should be signed, as the resolver is not able to obtain
   the necessary DNSSEC RRs. This can occur when the security-aware
   resolver is not able to contact security-aware name servers for
   the relevant zones.

Note, the "indeterminate" security status has a conflicting
definition in section 5 of [RFC4033].

   There is no trust anchor that would indicate that a specific
   portion of the tree is secure.

SMTP clients following this specification SHOULD NOT distinguish
between "insecure" and "indeterminate" in the [RFC4033] sense. Both
"insecure" and RFC4033 "indeterminate" are handled identically: in
either case unvalidated data for the query domain is all that is and
can be available, and authentication using the data is impossible.
In what follows, when we say "insecure", we include also DNS results
for domains that lie in a portion of the DNS tree for which there is
no applicable trust anchor. With the DNS root zone signed, we expect
that validating resolvers used by Internet-facing MTAs will be
configured with trust anchor data for the root zone. Therefore,
RFC4033-style "indeterminate" domains should be rare in practice.
From here on, when we say "indeterminate", it is exclusively in the
sense of [RFC4035].

As noted in section 4.3 of [RFC4035], a security-aware DNS resolver
MUST be able to determine whether a given non-error DNS response is
"secure", "insecure", "bogus" or "indeterminate". It is expected
that most security-aware stub resolvers will not signal an
"indeterminate" security status in the RFC4035-sense to the
application, and will signal a "bogus" or error result instead. If a
resolver does signal an RFC4035 "indeterminate" security status, this
MUST be treated by the SMTP client as though a "bogus" or error
result had been returned.

An MTA making use of a non-validating security-aware stub resolver
MAY use the stub resolver's ability, if available, to signal DNSSEC
validation status based on information the stub resolver has learned
from an upstream validating recursive resolver. In accordance with
section 4.9.3 of [RFC4035]:

   ... a security-aware stub resolver MUST NOT place any reliance on
   signature validation allegedly performed on its behalf, except
   when the security-aware stub resolver obtained the data in question
   from a trusted security-aware recursive name server via a secure
   channel.

To avoid much repetition in the text below, we will pause to explain
the handling of "bogus" or "indeterminate" DNSSEC query responses.
These are not necessarily the result of a malicious actor; they can,
for example, occur when network packets are corrupted or lost in
transit. Therefore, "bogus" or "indeterminate" replies are equated
in this memo with lookup failure.

There is an important non-failure condition we need to highlight in
addition to the obvious case of the DNS client obtaining a non-empty
"secure" or "insecure" RRset of the requested type. Namely, it is
not an error when either "secure" or "insecure" non-existence is
determined for the requested data. When a DNSSEC response with a
validation status that is either "secure" or "insecure" reports
either no records of the requested type or non-existence of the query
domain, the response is not a DNS error condition. The DNS client
has not been left without an answer; it has learned that records of
the requested type do not exist.

Security-aware stub resolvers will, of course, also signal DNS lookup
errors in other cases, for example when processing a "ServFail"
RCODE, which will not have an associated DNSSEC status. All lookup
errors are treated the same way by this specification, regardless of
whether they are from a "bogus" or "indeterminate" DNSSEC status or
from a more generic DNS error: the information that was requested can
not be obtained by the security-aware resolver at this time. A
lookup error is thus a failure to obtain the relevant RRset if it
exists, or to determine that no such RRset exists when it does not.

In contrast to a "bogus" or an "indeterminate" response, an
"insecure" DNSSEC response is not an error, rather it indicates that
the target DNS zone is either securely opted out of DNSSEC validation
or is not connected with the DNSSEC trust anchors being used.
Insecure results will leave the SMTP client with degraded channel
security, but do not stand in the way of message delivery. See
section Section 2.2 for further details.

When a stub resolver receives a response containing a CNAME alias, it
will generally restart the query at the target of the alias, and
should do so recursively up to some configured or implementation-
dependent recursion limit. If at any stage of recursive CNAME
expansion a query fails, the stub resolver's lookup of the original
requested records will result in a failure status being returned. If

at any stage of recursive expansion the response is "insecure", then
it and all subsequent results (in particular, the final result) MUST
be considered "insecure" regardless of whether the other responses
received were deemed "secure". If at any stage of recursive
expansion the validation status is "bogus" or "indeterminate" or
associated with another DNS lookup error, the resolution of the
requested records MUST be considered to have failed.

When a DNS lookup failure (error or "bogus" or "indeterminate" as
defined above) prevents an SMTP client from determining which SMTP
server or servers it should connect to, message delivery MUST be
delayed. This naturally includes, for example, the case when a
"bogus" or "indeterminate" response is encountered during MX
resolution. When multiple MX hostnames are obtained from a
successful MX lookup, but a later DNS lookup failure prevents network
address resolution for a given MX hostname, delivery may proceed via
any remaining MX hosts.

When a particular SMTP server is selected as the delivery
destination, a set of DNS lookups must be performed to discover any
related TLSA records. If any DNS queries used to locate TLSA records
fail (be it due to "bogus" or "indeterminate" records, timeouts,
malformed replies, ServFails, etc.), then the SMTP client MUST treat
that server as unreachable and MUST NOT deliver the message via that
server. If no servers are reachable, delivery is delayed.

In what follows, we will only describe what happens when all relevant
DNS queries succeed. If any DNS failure occurs, the SMTP client MUST
behave as described in this section, by skipping the problem SMTP
server, or the problem destination. Queries for candidate TLSA
records are explicitly part of "all relevant DNS queries" and SMTP
clients MUST NOT continue to connect to an SMTP server or destination
whose TLSA record lookup fails.

2.2. TLS discovery

As noted previously (in Section 1.3.1), opportunistic TLS with SMTP
servers that advertise TLS support via STARTTLS is subject to an MITM
downgrade attack. Also some SMTP servers that are not, in fact, TLS
capable erroneously advertise STARTTLS by default and clients need to
be prepared to retry cleartext delivery after STARTTLS fails. In
contrast, DNSSEC validated TLSA records MUST NOT be published for
servers that do not support TLS. Clients can safely interpret their
presence as a commitment by the server operator to implement TLS and
STARTTLS.

This memo defines four actions to be taken after the search for a
TLSA record returns secure usable results, secure unusable results,

   insecure or no results or an error signal. The term "usable" in this
   context is in the sense of Section 4.1 of [RFC6698]. Specifically,
   if the DNS lookup for a TLSA record returns:

   A secure TLSA RRset with at least one usable record: A connection to
      the MTA MUST be made using authenticated and encrypted TLS, using
      the techniques discussed in the rest of this document. Failure to
      establish an authenticated TLS connection MUST result in falling
      back to the next SMTP server or delayed delivery.

   A Secure non-empty TLSA RRset where all the records are unusable: A
      connection to the MTA MUST be made via TLS, but authentication is
      not required. Failure to establish an encrypted TLS connection
      MUST result in falling back to the next SMTP server or delayed
      delivery.

   An insecure TLSA RRset or DNSSEC validated proof-of-non-existent TLSA
    records:
      A connection to the MTA SHOULD be made using (pre-DANE)
      opportunistic TLS, this includes using cleartext delivery when the
      remote SMTP server does not appear to support TLS. The MTA MAY
      retry in cleartext when delivery via TLS fails either during the
      handshake or even during data transfer.

   Any lookup error: Lookup errors, including "bogus" and
      "indeterminate", as explained in Section 2.1 MUST result in
      falling back to the next SMTP server or delayed delivery.

   An SMTP client MAY be configured to require DANE verified delivery
   for some destinations. We will call such a configuration "mandatory
   DANE TLS". With mandatory DANE TLS, delivery proceeds only when
   "secure" TLSA records are used to establish an encrypted and
   authenticated TLS channel with the SMTP server.

   A note about DNAME aliases: a query for a domain name whose ancestor
   domain is a DNAME alias returns the DNAME RR for the ancestor domain,
   along with a CNAME that maps the query domain to the corresponding
   sub-domain of the target domain of the DNAME alias [RFC6672].
   Therefore, whenever we speak of CNAME aliases, we implicitly allow
   for the possibility that the alias in question is the result of an
   ancestor domain DNAME record. Consequently, no explicit support for
   DNAME records is needed in SMTP software, it is sufficient to process
   the resulting CNAME aliases. DNAME records only require special
   processing in the validating stub-resolver library that checks the
   integrity of the combined DNAME + CNAME reply. When DNSSEC
   validation is handled by a local caching resolver, rather than the
   MTA itself, even that part of the DNAME support logic is outside the
   MTA.

When the original next-hop destination is an address literal, rather
than a DNS domain, DANE TLS does not apply. Delivery proceeds using
any relevant security policy configured by the MTA administrator.
Similarly, when an MX RRset incorrectly lists a network address in
lieu of an MX hostname, if the MTA chooses to connect to the network
address DANE TLSA does not apply for such a connection.

In the subsections that follow we explain how to locate the SMTP
servers and the associated TLSA records for a given next-hop
destination domain. We also explain which name or names are to be
used in identity checks of the SMTP server certificate.

2.2.1. MX resolution

In this section we consider next-hop domains that are subject to MX
resolution and have MX records. The TLSA records and the associated
base domain are derived separately for each MX hostname that is used
to attempt message delivery. Clearly, if DANE TLS security is to
apply to message delivery via any of the SMTP servers, the MX records
must be obtained securely via a DNSSEC validated MX lookup.

MX records MUST be sorted by preference; an MX hostname with a worse
(numerically higher) MX preference that has TLSA records MUST NOT
preempt an MX hostname with a better (numerically lower) preference
that has no TLSA records. In other words, prevention of delivery
loops by obeying MX preferences MUST take precedence over channel
security considerations. Even with two equal-preference MX records,
an MTA is not obligated to choose the MX hostname that offers more
security. Domains that want secure inbound mail delivery need to
ensure that all their SMTP servers and MX records are configured
accordingly.

In the language of [RFC5321] Section 5.1, the original next-hop
domain is the "initial name". If the MX lookup of the initial name
results in a CNAME alias, the MTA replaces the initial name with the
resulting name and performs a new lookup with the new name. MTAs
typically support recursion in CNAME expansion, so this replacement
is performed repeatedly until the ultimate non-CNAME domain is found.

If the MX RRset (or any CNAME leading to it) is "insecure" (see
Section 2.1), DANE TLS does not apply, and delivery proceeds via pre-
DANE opportunistic TLS. Otherwise (assuming no DNS errors or "bogus"
/"indeterminate" responses), the MX RRset is "secure", and the SMTP
client MUST treat each MX hostname as a separate non-MX destination
for opportunistic DANE TLS as described in Section 2.2.2. When, for
a given MX hostname, no TLSA records are found, or only "insecure"
TLSA records are found, DANE TLSA is not applicable with the SMTP
server in question and delivery proceeds to that host as with pre-

DANE opportunistic TLS. To avoid downgrade attacks, any errors
during TLSA lookups MUST, as explained in Section 2.1, cause the SMTP
server in question to be treated as unreachable.

2.2.2. Non-MX destinations

This section describes the algorithm used to locate the TLSA records
and associated TLSA base domain for an input domain not subject to MX
resolution. Such domains include:

o Each MX hostname used in a message delivery attempt for an
   original next-hop destination domain subject to MX resolution.
   Note, MTAs are not obligated to support CNAME expansion of MX
   hostnames.

o Any administrator configured relay hostname, not subject to MX
   resolution. This frequently involves configuration set by the MTA
   administrator to handle some or all mail.

o A next-hop destination domain subject to MX resolution that has no
   MX records. In this case the domain's name is implicitly also its
   sole SMTP server name.

Note that DNS queries with type TLSA are mishandled by load balancing
nameservers that serve the MX hostnames of some large email
providers. The DNS zones served by these nameservers are not signed
and contain no TLSA records, but queries for TLSA records fail,
rather than returning the non-existence of the requested TLSA
records.

To avoid problems delivering mail to domains whose SMTP servers are
served by the problem nameservers the SMTP client MUST perform any A
and/or AAAA queries for the destination before attempting to locate
the associated TLSA records. This lookup is needed in any case to
determine whether the destination domain is reachable and the DNSSEC
validation status of the chain of CNAME queries required to reach the
ultimate address records.

If no address records are found, the destination is unreachable. If
address records are found, but the DNSSEC validation status of the
first query response is "insecure" (there may be additional queries
if the initial response is a CNAME alias), the SMTP client SHOULD NOT
proceed to search for any associated TLSA records. With the problem
domains, TLSA queries will lead to DNS lookup errors and cause
messages to be consistently delayed and ultimately returned to the
sender. We don't expect to find any "secure" TLSA records associated
with a TLSA base domain that lies in an unsigned DNS zone.
Therefore, skipping TLSA lookups in this case will also reduce
latency with no detrimental impact on security.

If the A and/or AAAA lookup of the "initial name" yields a CNAME, we
replace it with the resulting name as if it were the initial name and
perform a lookup again using the new name. This replacement is
performed recursively.

We consider the following cases for handling a DNS response for an A
or AAAA DNS lookup:

Not found: When the DNS queries for A and/or AAAA records yield
   neither a list of addresses nor a CNAME (or CNAME expansion is not
   supported) the destination is unreachable.

Non-CNAME: The answer is not a CNAME alias. If the address RRset
   is "secure", TLSA lookups are performed as described in
   Section 2.2.3 with the initial name as the candidate TLSA base
   domain. If no "secure" TLSA records are found, DANE TLS is not
   applicable and mail delivery proceeds with pre-DANE opportunistic
   TLS (which, being best-effort, degrades to cleartext delivery when
   STARTTLS is not available or the TLS handshake fails).

Insecure CNAME: The input domain is a CNAME alias, but the ultimate
   network address RRset is "insecure" (see Section 2.1). If the
   initial CNAME response is also "insecure", DANE TLS does not
   apply. Otherwise, this case is treated just like the non-CNAME
   case above, where a search is performed for a TLSA record with the
   original input domain as the candidate TLSA base domain.

Secure CNAME: The input domain is a CNAME alias, and the ultimate
   network address RRset is "secure" (see Section 2.1). Two
   candidate TLSA base domains are tried: the fully CNAME-expanded
   initial name and, failing that, then the initial name itself.

In summary, if it is possible to securely obtain the full, CNAME-
expanded, DNSSEC-validated address records for the input domain, then
that name is the preferred TLSA base domain. Otherwise, the
unexpanded input-MX domain is the candidate TLSA base domain. When

no "secure" TLSA records are found at either the CNAME-expanded or
unexpanded domain, then DANE TLS does not apply for mail delivery via
the input domain in question. And, as always, errors, bogus or
indeterminate results for any query in the process MUST result in
delaying or abandoning delivery.

2.2.3. TLSA record lookup

Each candidate TLSA base domain (the original or fully CNAME-expanded
name of a non-MX destination or a particular MX hostname of an MX
destination) is in turn prefixed with service labels of the form
"_<port>._tcp". The resulting domain name is used to issue a DNSSEC
query with the query type set to TLSA ([RFC6698] Section 7.1).

For SMTP, the destination TCP port is typically 25, but this may be
different with custom routes specified by the MTA administrator in
which case the SMTP client MUST use the appropriate number in the
"_<port>" prefix in place of "_25". If, for example, the candidate
base domain is "mx.example.com", and the SMTP connection is to port
25, the TLSA RRset is obtained via a DNSSEC query of the form:

_25._tcp.mx.example.com. IN TLSA ?

The query response may be a CNAME, or the actual TLSA RRset. If the
response is a CNAME, the SMTP client (through the use of its
security-aware stub resolver) restarts the TLSA query at the target
domain, following CNAMEs as appropriate and keeping track of whether
the entire chain is "secure". If any "insecure" records are
encountered, or the TLSA records don't exist, the next candidate TLSA
base is tried instead.

If the ultimate response is a "secure" TLSA RRset, then the candidate
TLSA base domain will be the actual TLSA base domain and the TLSA
RRset will constitute the TLSA records for the destination. If none
of the candidate TLSA base domains yield "secure" TLSA records then
delivery should proceed via pre-DANE opportunistic TLS.

TLSA record publishers may leverage CNAMEs to reference a single
authoritative TLSA RRset specifying a common Certification Authority
or a common end entity certificate to be used with multiple TLS
services. Such CNAME expansion does not change the SMTP client's
notion of the TLSA base domain; thus, when _25._tcp.mx.example.com is
a CNAME, the base domain remains mx.example.com and this is still the
reference identifier used together with the next-hop domain in peer
certificate name checks.

Note, shared end entity certificate associations expose the
publishing domain to substitution attacks, where an MITM attacker can

reroute traffic to a different server that shares the same end entity
certificate. Such shared end entity records SHOULD be avoided unless
the servers in question are functionally equivalent (an active
attacker gains nothing by diverting client traffic from one such
server to another).

For example, given the DNSSEC validated records below:

```
example.com.                 IN MX 0 mx1.example.com.
example.com.                 IN MX 0 mx2.example.com.
_25._tcp.mx1.example.com.   IN CNAME tlsa211._dane.example.com.
_25._tcp.mx2.example.com.   IN CNAME tlsa211._dane.example.com.
tlsa211._dane.example.com.  IN TLSA 2 1 1 e3b0c44298fc1c149a...
```

The SMTP servers mx1.example.com and mx2.example.com will be expected
to have certificates issued under a common trust anchor, but each MX
hostname's TLSA base domain remains unchanged despite the above CNAME
records. Correspondingly, each SMTP server will be associated with a
pair of reference identifiers consisting of its hostname plus the
next-hop domain "example.com".

If, during TLSA resolution (including possible CNAME indirection), at
least one "secure" TLSA record is found (even if not usable because
it is unsupported by the implementation or support is
administratively disabled), then the corresponding host has signaled
its commitment to implement TLS. The SMTP client SHOULD NOT deliver
mail via the corresponding host unless a TLS session is negotiated
via STARTTLS. This is required to avoid MITM STARTTLS downgrade
attacks.

As noted previously (in Section [Section 2.2.2](#)), when no "secure" TLSA
records are found at the fully CNAME-expanded name, the original
unexpanded name MUST be tried instead. This supports customers of
hosting providers where the provider's zone cannot be validated with
DNSSEC, but the customer has shared appropriate key material with the
hosting provider to enable TLS via SNI. Intermediate names that
arise during CNAME expansion that are neither the original, nor the
final name, are never candidate TLSA base domains, even if "secure".

[2.3](#). DANE authentication

This section describes which TLSA records are applicable to SMTP
opportunistic DANE TLS and how to apply such records to authenticate
the SMTP server. With opportunistic DANE TLS, both the TLS support
implied by the presence of DANE TLSA records and the verification
parameters necessary to authenticate the TLS peer are obtained
together. In contrast to protocols where channel security policy is
set exclusively by the client, authentication via this protocol is

expected to be less prone to connection failure caused by
incompatible configuration of the client and server.

2.3.1. **TLSA certificate usages**

The DANE TLSA specification [RFC6698] defines multiple TLSA RR types
via combinations of 3 numeric parameters. The numeric values of
these parameters were later given symbolic names in
[I-D.ietf-dane-registry-acronyms]. The rest of the TLSA record is
the "certificate association data field", which specifies the full or
digest value of a certificate or public key. The parameters are:

The TLSA Certificate Usage field: Section 2.1.1 of [RFC6698]
   specifies 4 values: PKIX-TA(0), PKIX-EE(1), DANE-TA(2), and DANE-
   EE(3). There is an additional private-use value: PrivCert(255).
   All other values are reserved for use by future specifications.

The selector field: Section 2.1.2 of [RFC6698] specifies 2 values:
   Cert(0), SPKI(1). There is an additional private-use value:
   PrivSel(255). All other values are reserved for use by future
   specifications.

The matching type field: Section 2.1.3 of [RFC6698] specifies 3
   values: Full(0), SHA2-256(1), SHA2-512(2). There is an additional
   private-use value: PrivMatch(255). All other values are reserved
   for use by future specifications.

We may think of TLSA Certificate Usage values 0 through 3 as a
combination of two one-bit flags. The low bit chooses between trust
anchor (TA) and end entity (EE) certificates. The high bit chooses
between public PKI issued and domain-issued certificates.

The selector field specifies whether the TLSA RR matches the whole
certificate: Cert(0), or just its subjectPublicKeyInfo: SPKI(1). The
subjectPublicKeyInfo is an ASN.1 DER encoding of the certificate's
algorithm id, any parameters and the public key data.

The matching type field specifies how the TLSA RR Certificate
Association Data field is to be compared with the certificate or
public key. A value of Full(0) means an exact match: the full DER
encoding of the certificate or public key is given in the TLSA RR. A
value of SHA2-256(1) means that the association data matches the
SHA2-256 digest of the certificate or public key, and likewise
SHA2-512(2) means a SHA2-512 digest is used.

Since opportunistic DANE TLS will be used by non-interactive MTAs,
with no user to "press OK" when authentication fails, reliability of
peer authentication is paramount. Server operators are advised to

publish TLSA records that are least likely to fail authentication due
to interoperability or operational problems. Because DANE TLS relies
on coordinated changes to DNS and SMTP server settings, the best
choice of records to publish will depend on site-specific practices.

The certificate usage element of a TLSA record plays a critical role
in determining how the corresponding certificate association data
field is used to authenticate server's certificate chain. The next
two subsections explain the process for certificate usages DANE-EE(3)
and DANE-TA(2). The third subsection briefly explains why
certificate usages PKIX-TA(0) and PKIX-EE(1) are not applicable with
opportunistic DANE TLS.

In summary, we recommend the use of either "DANE-EE(3) SPKI(1)
SHA2-256(1)" or "DANE-TA(2) Cert(0) SHA2-256(1)" TLSA records
depending on site needs. Other combinations of TLSA parameters are
either explicitly unsupported, or offer little to recommend them over
these two.

The mandatory to support digest algorithm in [RFC6698] is
SHA2-256(1). When the server's TLSA RRset includes records with a
matching type indicating a digest record (i.e., a value other than
Full(0)), a TLSA record with a SHA2-256(1) matching type SHOULD be
provided along with any other digest published, since some SMTP
clients may support only SHA2-256(1). If at some point the SHA2-256
digest algorithm is tarnished by new cryptanalytic attacks,
publishers will need to include an appropriate stronger digest in
their TLSA records, initially along with, and ultimately in place of,
SHA2-256.

2.3.1.1. Certificate usage DANE-EE(3)

Authentication via certificate usage DANE-EE(3) TLSA records involves
simply checking that the server's leaf certificate matches the TLSA
record. In particular the binding of the server public key to its
name is based entirely on the TLSA record association The server MUST
be considered authenticated even if none of the names in the
certificate match the client's reference identity for the server.

Similarly, the expiration date of the server certificate MUST be
ignored, the validity period of the TLSA record key binding is
determined by the validity interval of the TLSA record DNSSEC
signature.

With DANE-EE(3) servers need not employ SNI (may ignore the client's
SNI message) even when the server is known under independent names
that would otherwise require separate certificates. It is instead
sufficient for the TLSA RRsets for all the domains in question to

match the server's default certificate. Of course with SMTP servers
it is simpler still to publish the same MX hostname for all the
hosted domains.

For domains where it is practical to make coordinated changes in DNS
TLSA records during SMTP server key rotation, it is often best to
publish end-entity DANE-EE(3) certificate associations. DANE-EE(3)
certificates don't suddenly stop working when leaf or intermediate
certificates expire, and don't fail when the server operator neglects
to configure all the required issuer certificates in the server
certificate chain.

TLSA records published for SMTP servers SHOULD, in most cases, be
"DANE-EE(3) SPKI(1) SHA2-256(1)" records. Since all DANE
implementations are required to support SHA2-256, this record type
works for all clients and need not change across certificate renewals
with the same key.

2.3.1.2. Certificate usage DANE-TA(2)

Some domains may prefer to avoid the operational complexity of
publishing unique TLSA RRs for each TLS service. If the domain
employs a common issuing Certification Authority to create
certificates for multiple TLS services, it may be simpler to publish
the issuing authority as a trust anchor (TA) for the certificate
chains of all relevant services. The TLSA query domain (TLSA base
domain with port and protocol prefix labels) for each service issued
by the same TA may then be set to a CNAME alias that points to a
common TLSA RRset that matches the TA. For example:

```
  example.com.                 IN MX 0 mx1.example.com.
  example.com.                 IN MX 0 mx2.example.com.
  _25._tcp.mx1.example.com.   IN CNAME tlsa211._dane.example.com.
  _25._tcp.mx2.example.com.   IN CNAME tlsa211._dane.example.com.
  tlsa211._dane.example.com.  IN TLSA 2 1 1 e3b0c44298fc1c14....
```

With usage DANE-TA(2) the server certificates will need to have names
that match one of the client's reference identifiers (see [RFC6125]).
The server MAY employ SNI to select the appropriate certificate to
present to the client.

SMTP servers that rely on certificate usage DANE-TA(2) TLSA records
for TLS authentication MUST include the TA certificate as part of the
certificate chain presented in the TLS handshake server certificate
message even when it is a self-signed root certificate. At this
time, many SMTP servers are not configured with a comprehensive list
of trust anchors, nor are they expected to at any point in the
future. Some MTAs will ignore all locally trusted certificates when

   processing usage DANE-TA(2) TLSA records. Thus even when the TA
   happens to be a public Certification Authority known to the SMTP
   client, authentication is likely to fail unless the TA certificate is
   included in the TLS server certificate message.

   TLSA records with selector Full(0) are discouraged. While these
   potentially obviate the need to transmit the TA certificate in the
   TLS server certificate message, client implementations may not be
   able to augment the server certificate chain with the data obtained
   from DNS, especially when the TLSA record supplies a bare key
   (selector SPKI(1)). Since the server will need to transmit the TA
   certificate in any case, server operators SHOULD publish TLSA records
   with a selector other than Full(0) and avoid potential
   interoperability issues with large TLSA records containing full
   certificates or keys.

   TLSA Publishers employing DANE-TA(2) records SHOULD publish records
   with a selector of Cert(0). Such TLSA records are associated with
   the whole trust anchor certificate, not just with the trust anchor
   public key. In particular, the SMTP client SHOULD then apply any
   relevant constraints from the trust anchor certificate, such as, for
   example, path length constraints.

   While a selector of SPKI(1) may also be employed, the resulting TLSA
   record will not specify the full trust anchor certificate content,
   and elements of the trust anchor certificate other than the public
   key become mutable. This may, for example, allow a subsidiary CA to
   issue a chain that violates the trust anchor's path length or name
   constraints.

2.3.1.3. Certificate usages PKIX-TA(0) and PKIX-EE(1)

   SMTP servers SHOULD NOT publish TLSA RRs with certificate usage
   "PKIX-TA(0)" or "PKIX-EE(1)". SMTP clients cannot be expected to be
   configured with a suitably complete set of trusted public CAs. Even
   with a full set of public CAs, SMTP clients cannot (without relying
   on DNSSEC for secure MX records and DANE for STARTTLS support
   signaling) perform [RFC6125] server identity verification or prevent
   STARTTLS downgrade attacks. The use of trusted public CAs offers no
   added security since an attacker capable of compromising DNSSEC is
   free to replace any PKIX-TA(0) or PKIX-EE(1) TLSA records with
   records bearing any convenient non-PKIX certificate usage.

   SMTP client treatment of TLSA RRs with certificate usages "PKIX-
   TA(0)" or "PKIX-EE(1)" is undefined. For example, clients MAY (will
   likely) treat such TLSA records as unusable.

2.3.2. Certificate matching

When at least one usable "secure" TLSA record is found, the SMTP
client SHOULD use TLSA records to authenticate the SMTP server.
Messages SHOULD NOT be delivered via the SMTP server if
authentication fails, otherwise the SMTP client is vulnerable to MITM
attacks.

2.3.2.1. DANE-EE(3) name checks

The SMTP client MUST NOT perform certificate name checks with
certificate usage DANE-EE(3), see Section 2.3.1.1 above.

2.3.2.2. DANE-TA(2) name checks

To match a server via a TLSA record with certificate usage DANE-
TA(2), the client MUST perform name checks to ensure that it has
reached the correct server. In all DANE-TA(2) cases the SMTP client
MUST include the TLSA base domain as one of the valid reference
identifiers for matching the server certificate.

TLSA records for MX hostnames: If the TLSA base domain was obtained
   indirectly via an MX lookup (including any CNAME-expanded name of
   an MX hostname), then the original next-hop domain used in the MX
   lookup MUST be included as as a second reference identifier. The
   CNAME-expanded original next-hop domain MUST be included as a
   third reference identifier if different from the original next-hop
   domain.

TLSA records for Non-MX hostnames: If MX records were not used
   (e.g., if none exist) and the TLSA base domain is the CNAME-
   expanded original next-hop domain, then the original next-hop
   domain MUST be included as a second reference identifier.

Accepting certificates with the original next-hop domain in addition
to the MX hostname allows a domain with multiple MX hostnames to
field a single certificate bearing a single domain name (i.e., the
email domain) across all the SMTP servers. This also aids
interoperability with pre-DANE SMTP clients that are configured to
look for the email domain name in server certificates. For example,
with "secure" DNS records as below:

```
   exchange.example.org.              IN CNAME mail.example.org.
   mail.example.org.                  IN CNAME example.com.
   example.com.                       IN MX    10 mx10.example.com.
   example.com.                       IN MX    15 mx15.example.com.
   example.com.                       IN MX    20 mx20.example.com.
   ;
   mx10.example.com.                  IN A     192.0.2.10
   _25._tcp.mx10.example.com.         IN TLSA  2 0 1 ...
   ;
   mx15.example.com.                  IN CNAME mxbackup.example.com.
   mxbackup.example.com.              IN A     192.0.2.15
   ; _25._tcp.mxbackup.example.com. IN TLSA ? (NXDOMAIN)
   _25._tcp.mx15.example.com.         IN TLSA  2 0 1 ...
   ;
   mx20.example.com.                  IN CNAME mxbackup.example.net.
   mxbackup.example.net.              IN A     198.51.100.20
   _25._tcp.mxbackup.example.net.     IN TLSA  2 0 1 ...
```

Certificate name checks for delivery of mail to exchange.example.org
via any of the associated SMTP servers MUST accept at least the names
"exchange.example.org" and "example.com", which are respectively the
original and fully expanded next-hop domain. When the SMTP server is
mx10.example.com, name checks MUST accept the TLSA base domain
"mx10.example.com". If, despite the fact that MX hostnames are
required to not be aliases, the MTA supports delivery via
"mx15.example.com" or "mx20.example.com" then name checks MUST accept
the respective TLSA base domains "mx15.example.com" and
"mxbackup.example.net".

2.3.2.3. Reference identifier matching

When name checks are applicable (certificate usage DANE-TA(2)), if
the server certificate contains a Subject Alternative Name extension
([RFC5280]), with at least one DNS-ID ([RFC6125]) then only the DNS-
IDs are matched against the client's reference identifiers. The CN-
ID ([RFC6125]) is only considered when no DNS-IDs are present. The
server certificate is considered matched when one of its presented
identifiers ([RFC5280]) matches any of the client's reference
identifiers.

Wildcards are valid in either DNS-IDs or the CN-ID when applicable.
The wildcard character must be entire first label of the DNS-ID or
CN-ID. Thus, "*.example.com" is valid, while "smtp*.example.com" and
"*smtp.example.com" are not. SMTP clients MUST support wildcards
that match the first label of the reference identifier, with the
remaining labels matching verbatim. For example, the DNS-ID
"*.example.com" matches the reference identifier "mx1.example.com".
SMTP clients MAY, subject to local policy allow wildcards to match

multiple reference identifier labels, but servers cannot expect broad
support for such a policy. Therefore any wildcards in server
certificates SHOULD match exactly one label in either the TLSA base
domain or the next-hop domain.

2.3.3. Key rotation

Two TLSA records MUST be published before employing a new EE or TA
public key or certificate, one matching the currently deployed key
and the other matching the new key scheduled to replace it. Once
sufficient time has elapsed for all DNS caches to expire the previous
TLSA RRset and related signature RRsets, servers may be configured to
use the new EE private key and associated public key certificate or
may employ certificates signed by the new trust anchor.

Once the new public key or certificate is in use, the TLSA RR that
matches the retired key can be removed from DNS, leaving only RRs
that match keys or certificates in active use.

2.3.4. Digest algorithm agility

While [RFC6698] specifies multiple digest algorithms, it does not
specify a protocol by which the SMTP client and TLSA record publisher
can agree on the strongest shared algorithm. Such a protocol would
allow the client and server to avoid exposure to any deprecated
weaker algorithms that are published for compatibility with less
capable clients, but should be ignored when possible. We specify
such a protocol below.

Suppose that a DANE TLS client authenticating a TLS server considers
digest algorithm BETTER stronger than digest algorithm WORSE.
Suppose further that a server's TLSA RRset contains some records with
BETTER as the digest algorithm. Finally, suppose that for every raw
public key or certificate object that is included in the server's
TLSA RRset in digest form, whenever that object appears with
algorithm WORSE with some usage and selector it also appears with
algorithm BETTER with the same usage and selector. In that case our
client can safely ignore TLSA records with the weaker algorithm
WORSE, because it suffices to check the records with the stronger
algorithm BETTER.

Server operators MUST ensure that for any given usage and selector, each object (certificate or public key), for which a digest association exists in the TLSA RRset, is published with the SAME SET of digest algorithms as all other objects that published with that usage and selector. In other words, for each usage and selector, the records with non-zero matching types will correspond to on a cross-product of a set of underlying objects and a fixed set of digest algorithms that apply uniformly to all the objects.

To achieve digest algorithm agility, all published TLSA RRsets for use with opportunistic DANE TLS for SMTP MUST conform to the above requirements. Then, for each combination of usage and selector, SMTP clients can simply ignore all digest records except those that employ the strongest digest algorithm. The ordering of digest algorithms by strength is not specified in advance, it is entirely up to the SMTP client. SMTP client implementations SHOULD make the digest algorithm preference order configurable. Only the future will tell which algorithms might be weakened by new attacks and when.

Note, TLSA records with a matching type of Full(0), that publish the full value of a certificate or public key object, play no role in digest algorithm agility. They neither trump the processing of records that employ digests, nor are they ignored in the presence of any records with a digest (i.e. non-zero) matching type.

SMTP clients SHOULD use digest algorithm agility when processing the DANE TLSA records of an SMTP server. Algorithm agility is to be applied after first discarding any unusable or malformed records (unsupported digest algorithm, or incorrect digest length). Thus, for each usage and selector, the client SHOULD process only any usable records with a matching type of Full(0) and the usable records whose digest algorithm is believed to be the strongest among usable records with the given usage and selector.

The main impact of this requirement is on key rotation, when the TLSA RRset is pre-populated with digests of new certificates or public keys, before these replace or augment their predecessors. Were the newly introduced RRs to include previously unused digest algorithms, clients that employ this protocol could potentially ignore all the digests corresponding to the current keys or certificates, causing connectivity issues until the new keys or certificates are deployed. Similarly, publishing new records with fewer digests could cause problems for clients using cached TLSA RRsets that list both the old and new objects once the new keys are deployed.

To avoid problems, server operators SHOULD apply the following strategy:

o When changing the set of objects published via the TLSA RRset
   (e.g. during key rotation), DO NOT change the set of digest
   algorithms used; change just the list of objects.

o When changing the set of digest algorithms, change only the set of
   algorithms, and generate a new RRset in which all the current
   objects are re-published with the new set of digest algorithms.

After either of these two changes are made, the new TLSA RRset should
be left in place long enough that the older TLSA RRset can be flushed
from caches before making another change.

3. **Mandatory TLS Security**

An MTA implementing this protocol may require a stronger security
assurance when sending email to selected destinations. The sending
organization may need to send sensitive email and/or may have
regulatory obligations to protect its content. This protocol is not
in conflict with such a requirement, and in fact can often simplify
authenticated delivery to such destinations.

Specifically, with domains that publish DANE TLSA records for their
MX hostnames, a sending MTA can be configured to use the receiving
domains's DANE TLSA records to authenticate the corresponding SMTP
server. Authentication via DANE TLSA records is easier to manage, as
changes in the receiver's expected certificate properties are made on
the receiver end and don't require manually communicated
configuration changes. With mandatory DANE TLS, when no usable TLSA
records are found, message delivery is delayed. Thus, mail is only
sent when an authenticated TLS channel is established to the remote
SMTP server.

Administrators of mail servers that employ mandatory DANE TLS, need
to carefully monitor their mail logs and queues. If a partner domain
unwittingly misconfigures their TLSA records, disables DNSSEC, or
misconfigures SMTP server certificate chains, mail will be delayed
and may bounce if the issue is not resolved in a timely manner.

4. **Note on DANE for Message User Agents**

We note that the SMTP protocol is also used between Message User
Agents (MUAs) and Message Submission Agents (MSAs) [RFC6409]. In
[RFC6186] a protocol is specified that enables an MUA to dynamically
locate the MSA based on the user's email address. SMTP connection
security considerations for MUAs implementing [RFC6186] are largely
analogous to connection security requirements for MTAs, and this
specification could be applied largely verbatim with DNS MX records
replaced by corresponding DNS Service (SRV) records
[I-D.ietf-dane-srv].

However, until MUAs begin to adopt the dynamic configuration
mechanisms of [RFC6186] they are adequately served by more
traditional static TLS security policies. Specification of DANE TLS
for Message User Agent (MUA) to Message Submission Agent (MSA) SMTP
is left to future documents that focus specifically on SMTP security
between MUAs and MSAs.

5. Interoperability considerations

5.1. SNI support

To ensure that the server sends the right certificate chain, the SMTP
client MUST send the TLS SNI extension containing the TLSA base
domain. This precludes the use of the backward compatible SSL 2.0
compatible SSL HELLO by the SMTP client. The minimum SSL/TLS client
HELLO version for SMTP clients performing DANE authentication is SSL
3.0, but a client that offers SSL 3.0 MUST also offer at least TLS
1.0 and MUST include the SNI extension. Servers that don't make use
of SNI MAY negotiate SSL 3.0 if offered by the client.

Each SMTP server MUST present a certificate chain (see [RFC5246]
Section 7.4.2) that matches at least one of the TLSA records. The
server MAY rely on SNI to determine which certificate chain to
present to the client. Clients that don't send SNI information may
not see the expected certificate chain.

If the server's TLSA records match the server's default certificate
chain, the server need not support SNI. In either case, the server
need not include the SNI extension in its TLS HELLO as simply
returning a matching certificate chain is sufficient. Servers MUST
NOT enforce the use of SNI by clients, as the client may be using
unauthenticated opportunistic TLS and may not expect any particular
certificate from the server. If the client sends no SNI extension,
or sends an SNI extension for an unsupported domain, the server MUST
simply send its default certificate chain. The reason for not
enforcing strict matching of the requested SNI hostname is that DANE
TLS clients are typically willing to accept multiple server names,
but can only send one name in the SNI extension. The server's

default certificate may match a different name acceptable to the
client, e.g., the original next-hop domain.

5.2. Anonymous TLS cipher suites

Since many SMTP servers either do not support or do not enable any
anonymous TLS cipher suites, SMTP client TLS HELLO messages SHOULD
offer to negotiate a typical set of non-anonymous cipher suites
required for interoperability with such servers. An SMTP client
employing pre-DANE opportunistic TLS MAY in addition include one or
more anonymous TLS cipher suites in its TLS HELLO. SMTP servers,
that need to interoperate with opportunistic TLS clients SHOULD be
prepared to interoperate with such clients by either always selecting
a mutually supported non-anonymous cipher suite or by correctly
handling client connections that negotiate anonymous cipher suites.

Note that while SMTP server operators are under no obligation to
enable anonymous cipher suites, no security is gained by sending
certificates to clients that will ignore them. Indeed support for
anonymous cipher suites in the server makes audit trails more
informative. Log entries that record connections that employed an
anonymous cipher suite record the fact that the clients did not care
to authenticate the server.

6. Operational Considerations

6.1. Client Operational Considerations

An operational error on the sending or receiving side that cannot be
corrected in a timely manner may, at times, lead to consistent
failure to deliver time-sensitive email. The sending MTA
administrator may have to choose between letting email queue until
the error is resolved and disabling opportunistic or mandatory DANE
TLS for one or more destinations. The choice to disable DANE TLS
security should not be made lightly. Every reasonable effort should
be made to determine that problems with mail delivery are the result
of an operational error, and not an attack. A fallback strategy may
be to configure explicit out-of-band TLS security settings if
supported by the sending MTA.

SMTP clients may deploy opportunistic DANE TLS incrementally by
enabling it only for selected sites, or may occasionally need to
disable opportunistic DANE TLS for peers that fail to interoperate
due to misconfiguration or software defects on either end. Some
implementations MAY support DANE TLS in an "audit only" mode in which
failure to achieve the requisite security level is logged as a
warning and delivery proceeds at a reduced security level. Unless
local policy specifies "audit only" or that opportunistic DANE TLS is

not to be used for a particular destination, an SMTP client MUST NOT
deliver mail via a server whose certificate chain fails to match at
least one TLSA record when usable TLSA records are found for that
server.

6.2. Publisher Operational Considerations

SMTP servers that publish certificate usage DANE-TA(2) associations
MUST include the TA certificate in their TLS server certificate
chain, even when that TA certificate is a self-signed root
certificate.

TLSA Publishers must follow the digest agility guidelines in
Section 2.3.4 and must make sure that all objects published in digest
form for a particular usage and selector are published with the same
set of digest algorithms.

TLSA Publishers should follow the TLSA publication size guidance
found in [I-D.ietf-dane-ops] about "DANE DNS Record Size Guidelines".

7. Security Considerations

This protocol leverages DANE TLSA records to implement MITM resistant
opportunistic channel security for SMTP. For destination domains
that sign their MX records and publish signed TLSA records for their
MX hostnames, this protocol allows sending MTAs to securely discover
both the availability of TLS and how to authenticate the destination.

This protocol does not aim to secure all SMTP traffic, as that is not
practical until DNSSEC and DANE adoption are universal. The
incremental deployment provided by following this specification is a
best possible path for securing SMTP. This protocol coexists and
interoperates with the existing insecure Internet email backbone.

The protocol does not preclude existing non-opportunistic SMTP TLS
security arrangements, which can continue to be used as before via
manual configuration with negotiated out-of-band key and TLS
configuration exchanges.

Opportunistic SMTP TLS depends critically on DNSSEC for downgrade
resistance and secure resolution of the destination name. If DNSSEC
is compromised, it is not possible to fall back on the public CA PKI
to prevent MITM attacks. A successful breach of DNSSEC enables the
attacker to publish TLSA usage 3 certificate associations, and
thereby bypass any security benefit the legitimate domain owner might
hope to gain by publishing usage 0 or 1 TLSA RRs. Given the lack of
public CA PKI support in existing MTA deployments, avoiding
certificate usages 0 and 1 simplifies implementation and deployment
with no adverse security consequences.

Implementations must strictly follow the portions of this
specification that indicate when it is appropriate to initiate a non-
authenticated connection or cleartext connection to a SMTP server.
Specifically, in order to prevent downgrade attacks on this protocol,
implementation must not initiate a connection when this specification
indicates a particular SMTP server must be considered unreachable.

8. IANA considerations

This specification requires no support from IANA.

9. Acknowledgements

The authors would like to extend great thanks to Tony Finch, who
started the original version of a DANE SMTP document. His work is
greatly appreciated and has been incorporated into this document.
The authors would like to additionally thank Phil Pennock for his
comments and advice on this document.

Acknowledgments from Viktor: Thanks to Paul Hoffman who motivated me
to begin work on this memo and provided feedback on early drafts.
Thanks to Patrick Koetter, Perry Metzger and Nico Williams for
valuable review comments. Thanks also to Wietse Venema who created
Postfix, and whose advice and feedback were essential to the
development of the Postfix DANE implementation.

10. References

10.1. Normative References

[I-D.ietf-dane-ops]
            Dukhovni, V. and W. Hardaker, "DANE TLSA implementation
            and operational guidance", draft-ietf-dane-ops-00 (work in
            progress), October 2013.

[RFC1035]   Mockapetris, P., "Domain names - implementation and
            specification", STD 13, RFC 1035, November 1987.

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

   [RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over
              Transport Layer Security", RFC 3207, February 2002.

   [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S.
              Rose, "DNS Security Introduction and Requirements", RFC
              4033, March 2005.

   [RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S.
              Rose, "Resource Records for the DNS Security Extensions",
              RFC 4034, March 2005.

   [RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S.
              Rose, "Protocol Modifications for the DNS Security
              Extensions", RFC 4035, March 2005.

   [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security
              (TLS) Protocol Version 1.2", RFC 5246, August 2008.

   [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, May 2008.

   [RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321,
              October 2008.

   [RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions:
              Extension Definitions", RFC 6066, January 2011.

   [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, March 2011.

   [RFC6186] Daboo, C., "Use of SRV Records for Locating Email
              Submission/Access Services", RFC 6186, March 2011.

   [RFC6672] Rose, S. and W. Wijngaards, "DNAME Redirection in the
              DNS", RFC 6672, June 2012.

   [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication
              of Named Entities (DANE) Transport Layer Security (TLS)
              Protocol: TLSA", RFC 6698, August 2012.

10.2. Informative References

   [I-D.ietf-dane-registry-acronyms]
              Gudmundsson, O., "Adding acronyms to simplify DANE
              conversations", draft-ietf-dane-registry-acronyms-01 (work
              in progress), October 2013.

   [I-D.ietf-dane-srv]
              Finch, T., "Using DNS-Based Authentication of Named
              Entities (DANE) TLSA records with SRV and MX records.",
              draft-ietf-dane-srv-02 (work in progress), February 2013.

   [RFC5598] Crocker, D., "Internet Mail Architecture", RFC 5598, July
              2009.

   [RFC6409] Gellens, R. and J. Klensin, "Message Submission for Mail",
              STD 72, RFC 6409, November 2011.

Authors' Addresses

   Viktor Dukhovni
   Two Sigma

   Email: ietf-dane@dukhovni.org

   Wes Hardaker
   Parsons
   P.O. Box 382
   Davis, CA 95617
   US

   Email: ietf@hardakers.net