DANE
Internet-Draft
Intended status: Standards Track
Expires: May 28, 2014

V. Dukhovni
Unaffiliated
W.H. Hardaker
Parsons

November 24, 2013

SMTP security via opportunistic DANE TLS
draft-ietf-dane-smtp-with-dane-03

Abstract

This memo describes a protocol for opportunistic TLS security based on the DANE TLSA DNS record. The protocol is downgrade resistant when the SMTP client supports DANE TLSA and the server domain publishes TLSA records for its MX hosts. This enables an incremental transition of the Internet email backbone (MTA to MTA SMTP traffic) to TLS encrypted and authenticated delivery.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Lacking verified DNS and "Server Name Indication" (SNI), there has historically been no scalable way for SMTP server operators to deploy certificates with a client-trusted subject name. It's only with the deployment of DNSSEC and DANE that authenticated TLS for SMTP to MX becomes possible between parties that have not already established an identity convention out-of-band.

1.1. Background

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

As described in the introduction of [RFC6698], TLS authentication via the existing public Certificate Authority (CA) Public Key Infrastructure (PKI) suffers from an over-abundance of trusted certificate authorities capable of issuing certificates for any domain of their choice. DNS-Based Authentication of Named Entities (DANE) leverages the DNSSEC infrastructure to publish trusted keys and certificates for use with TLS via a new TLSA record type. With DANE, the public CA PKI can be augmented or replaced by DNSSEC validated TLSA records.

The Transport Layer Security (TLS [RFC5246]) protocol enables secure TCP communication. In the context of this memo, channel security is assumed to be provided by TLS. Used without authentication, TLS protects only against eavesdropping. With authentication, TLS also protects against man-in-the-middle (MITM) attacks.

1.2. SMTP Channel Security

The Simple Mail Transfer Protocol (SMTP) ([RFC5321]) is multi-hop store & forward, while TLS security is hop-by-hop. The number of hops from the sender's Mail User Agent to the recipient mailbox is rarely less than 2 and is often higher. Some hops may be TLS protected, some may not. The same SMTP TCP endpoint can serve both TLS and non-TLS clients, with TLS negotiated via the SMTP STARTTLS command ([RFC3207]). DNS MX records abstract the next-hop transport end-point. SMTP addresses are not transport addresses and are security agnostic. Unlike HTTP, there is no URI scheme for email addresses to designate whether the SMTP server should be contacted with or without security.

A Mail Transport Agent (MTA) may need to forward a message to a particular email recipient <user@example.com>. To deliver the message, the MTA needs to retrieve the MX hosts of example.com from DNS, and then deliver the message to one of them. Absent DNSSEC, the MX lookup is vulnerable to man-in-the-middle and cache poisoning attacks. An active attacker can forge DNS replies with fake MX records, and can direct traffic to a server of his choice. Therefore, secure verification of MX host certificates is not possible without DNSSEC. A man in the middle can also suppress the MX host's STARTTLS EHLO response, convincing the client that communication over TLS is unavailable.

One might try to harden STARTTLS with SMTP against DNS attacks by requiring each MX host to posess an X.509 certificate for the recipient domain that is obtained from the message envelope and is not subject to DNS reply forgery. Unfortunately, this is impractical, as email for many domains is handled by third parties, which are not in a position to obtain certificates for all the

domains they serve. Deployment of SNI (see [RFC6066] Section 3.1) 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 authentication identity, and neither can the MX hostname without DNSSEC, large scale deployment of authenticated TLS for SMTP requires secure DNS. At this time, DNSSEC is not yet widely deployed and MTA to MTA traffic between Internet connected organizations rarely uses TLS at all, or simply uses TLS opportunistically without authentication and protects against only passive eavesdropping attacks.

The exceptions are cases in which the sending MTA is statically configured to use TLS for mail sent to specific selected peer domains and is configured with appropriate subject names (or content digests) to expect in the presented MX host certificates of those domains. Such statically configured SMTP secure channels are used rarely, generally between domains that make bilateral arrangements with their business partners. Internet email, on the other hand, requires contacting many new domains for which security configurations can not be established in advance.

Note, the above does not apply to mail submission [RFC6409], where a mail user agent is pre-configured to send all email to a fixed Mail Submission Agent (MSA). Submission servers usually offer TLS and the Mail User Agent (MUA) can be statically configured to require TLS with its chosen MSA. The situation changes when submission servers are configured dynamically via SRV records (see [RFC6186] Section 6). Applications to submission via SRV records will be discussed later in this memo.

With little opportunity to use TLS authentication, MX hosts that support STARTTLS often use self-signed or private CA issued X.509 certificates. Sending systems are rarely configured with a comprehensive list of trusted CAs and do not check CRLs or implement OCSP. In essence, they don't and can't rely on the existing public CA PKI. This is not a result of complacency on the part SMTP server administrators and MTA developers. Nor is it just a consequence of the relative maturity of the SMTP infrastructure at the time that TLS was introduced. Rather, 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.

This does not mean, however, that the Internet email backbone cannot benefit from TLS. The fact that transport security is not explicitly

specified in either the recipient address or the MX record means that new protocols can furnish out-of-band information to SMTP, making it possible to simultaneously discover both which peer domains support secure delivery via TLS and how to verify the authenticity of the associated MX hosts. The first such mechanism that can work an Internet scale is DANE TLSA, but use of DANE TLSA with MTA to MTA SMTP must be cognizant of the lack of any realistic role for the existing public CA PKI.

1.3. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Hardening Opportunistic TLS

This memo describes opportunistic SMTP over TLS security, where traffic from DANE TLSA aware SMTP clients to domains that implement DANE TLSA records in accordance with this memo is secure. Traffic to other domains continues to be sent in the same manner as before (either manually configured for security or unauthenticated and often unencrypted). It is hoped that, over time, more domains will implement DNSSEC and publish DANE TLSA records for their MX hosts. This will enable an incremental transition of the email backbone to authenticated TLS delivery.

Since email addresses and MX hostnames (or submission SRV records) neither signal nor deny support for TLS by the receiving domain, it is possible to use DANE TLSA records to securely signal TLS support and simultaneously to provide the means by which SMTP clients can successfully authenticate legitimate SMTP servers.

2.1. TLS discovery

As noted previously (Section 1.2), opportunistic TLS with SMTP servers that advertise TLS support via STARTTLS is subject to a man in the middle downgrade attack. Some SMTP servers erroneously advertise STARTTLS in default configurations that are not, in fact, TLS capable, and clients need to be prepared to retry plaintext delivery after STARTTLS fails. This memo specifies a downgrade resistant mechanism that allows a server to advertise TLS support based on DANE TLSA records. DNSSEC validated TLSA records are unlikely to be accidentally published for servers that do not in fact support TLS, and thus clients can safely interpret their presence as a commitment by the server operator to implement STARTTLS.

SMTP is a store & forward protocol. An MTA that is not the final destination for a message recipient forwards the message one hop closer to the recipient's mailbox. To do so, it must determine the appropriate next-hop destination, and locate one or more associated SMTP servers. When DNSSEC validated TLSA records are available for a given next-hop SMTP server, the TLS connection to that server will be downgrade resistant. If the records in question are "usable" ([RFC6698], Section 4.1) to authenticate the server, the connection will also be authenticated and thus immune to eavesdropping or tampering (unless DNSSEC itself is compromised).

Typically, the next-hop destination will be the domain part of the recipient address, which is then subject to MX resolution. The next-hop destination may also be configured by the MTA administrator to be a next-hop destination host (explicitly exempt from MX resolution), or a next-hop destination domain (subject to MX resolution) which takes the place of the domain part of the recipient address.

The protocol in this memo is "opportunistic"; it should be used whenever possible but communication should continue when it is not available. Absent "secure" (DNSSEC validated) TLSA records, mail delivery should fall back to pre-DANE opportunistic TLS. The SMTP client MAY be configured to require DANE verified delivery for some or all destinations, in which case mail delivery will be deferred when "secure" TLSA records are absent.

Below we explain how to determine for a given next-hop destination the associated SMTP servers, the TLSA base domain and TLSA records.

2.1.1. Non-MX destinations

As mentioned above, the next-hop destination domain may in some cases be exempt from MX lookups. In addition, MX lookups for the next-hop domain may yield no results. In either case, the destination server for such a domain is determined by looking up the corresponding A or AAAA records.

When "bogus" records are encountered either during CNAME expansion, or when retrieving the associated TLSA RRset, the SMTP client MUST proceed as if the next-hop domain were unreachable. Delivery should either be deferred or may be attempted via any fallback next-hop configured by the SMTP client administrator. Fallback next-hop destinations may also employ opportunistic DANE TLS. Proceeding with the original next-hop despite "bogus" DNS responses would destroy protection against downgrade attacks.

Following [RFC5321] Section 5.1, if the A 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, although MTAs typically support only limited recursion in CNAME expansion. We consider the following cases:

Non-CNAME: The next-hop destination domain is not a CNAME alias. The lookup key for the DNSSEC validated TLSA records is obtained by prepending service labels ("_<port-number>._tcp") to the initial next-hop destination domain. If associated "secure" TLSA records are found (see Section 2.1.3) the TLSA base domain is the next-hop domain. If no secure TLSA records are found, opportunistic DANE TLS is not applicable and mail delivery proceeds with pre-DANE opportunistic TLS.

Insecure CNAME: The next-hop destination domain is a CNAME alias, but at least one of the CNAME RRs leading to the ultimate target of this alias (during recursive CNAME expansion) is "insecure". We treat this case just like the non-CNAME case above.

Secure CNAME, no TLSA: The next-hop destination domain is a CNAME alias, and all the CNAME RRs leading to the ultimate target of this alias (during recursive CNAME expansion) are "secure" (DNSSEC validated), but no "secure" TLSA RRs are found after prefixing the service labels to the CNAME-expanded next-hop domain. This case is also treated just like the non-CNAME case.

Secure CNAME, TLSA: The next-hop destination domain is a CNAME alias, all the CNAME RRs leading to the ultimate target of this alias (during recursive CNAME expansion) are "secure", and in addition "secure" TLSA RRs are found after prefixing the service labels to the CNAME-expanded next-hop domain. In this case the CNAME-expanded next-hop domain is taken as the TLSA base domain. The original next-hop domain is (see Section 2.2.2) used only as an alternative name in certificate peername verification if applicable.

In summary, if it is possible to securely obtain the full, CNAME-expanded, DNSSEC-validated address records for the non-MX next-hop domain then that name is the preferred TLSA base domain. If that is not possible, then the original next-hop domain is used as the TLSA base domain. When no "secure" TLSA records are found at either the CNAME expanded or original next-hop domain, then opportunistic DANE TLS does not apply for mail delivery to the non-MX destination in question.

2.1.2. MX resolution

In this section we consider next-hop domains that are subject to MX resolution and have MX records. When DANE TLS is applicable, the TLSA base domain will be associated with the MX host selected for message delivery. Therefore, the MX host names must be determined securely by performing a DNSSEC validated MX lookup to obtain the list of associated MX hosts. If the MX RRset is "insecure", DANE TLSA does not apply and mail delivery proceeds with pre-DANE opportunistic TLS (subject to its various MITM attacks and unecrypted transmission when STARTTLS is not supported by the destination).

When "bogus" DNSSEC records are encountered during CNAME expansion of the next-hop domain or when processing the actual MX RRset, delivery MUST either be deferred, or MAY be attempted via any fallback next-hop (which may also employ opportunistic DANE TLS) configured by the SMTP client administrator. Proceeding with the original next-hop despite "bogus" DNS responses would destroy protection against downgrade attacks. When "bogus" DNSSEC records are encountered with CNAME expansion or TLSA RRset lookup for a particular MX host, delivery MUST proceed as if MX host in question were unreachable.

MX records MUST be sorted by preference; an MX host with a better preference and no TLSA records MUST NOT be preempted by a host with a worse MX preference but with TLSA records. In other words, avoiding delivery loops by following MX preferences must take place even if it means insecure delivery.

In accordance with Section 5.1 of [RFC5321], if the MX lookup of the initial name yields a CNAME, we replace the initial name with the resulting name and perform 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 (or the limit on the number of CNAMEs to examine is reached). If at any stage of CNAME expansion the DNS result is "bogus", MX resolution fails with a temporary error. In that case, mail delivery MUST either be deferred, or attempted via any alternative delivery channel configured by the MTA administrator. We consider the following cases:

Non-CNAME: The next-hop destination domain is not a CNAME alias, that is, it resolves directly to a set of DNSSEC validated ("secure") MX hosts. With each MX host, if MX host CNAME expansion is supported by the MTA, and the full CNAME expansion of the MX host name is "secure", then the CNAME expanded MX host name is the TLSA base domain provided secure TLSA records are found there after prefixing service labels ("_<port-number>._tcp"). Otherwise, the initial MX host name is the TLSA base domain provided secure TLSA records are found there after prefixing service labels. With the MX hostname (or its CNAME expansion) as

the TLSA base domain, the original next-hop domain SHOULD be used only in certificate name checks. If no "secure" TLSA RRs are found, and no "bogus" records encountered, DANE TLSA is not applicable with the MX host in question and delivery proceeds as with pre-DANE opportunistic TLS.

CNAME: The next-hop destination domain is a CNAME alias, and resolves via a chain of "secure" CNAME records to a final domain with "secure" MX records. The TLSA base domain for each MX host in this case is the same as in the "Non-CNAME" case above, but now both the initial domain and its CNAME-expansion are candidate names in certificate name checks. If the CNAME chain contains "insecure" elements, DANE TLSA does not apply to the next-hop domain, and delivery proceeds via pre-DANE opportunistic TLS.

Note: CNAMEs are not legal in the exchange field of MX records, thus MTAs are not obligated to perform MX exchange CNAME expansion. If an MTA does not perform CNAME expansion, there is potential risk, that the MTA may fail to notice that it is one of the MX hosts for the destination and that it must skip MX records with equal or worse (numerically higher precedence). If an MTA does allow CNAMEs to be used in MX records, it SHOULD process them recursively as described above to determine the most appropriate TLSA RRset base domain.

2.1.3. TLSA record lookup

Each TLSA base domain obtained above (for a non-MX destination, or for a particular MX host of an MX destination), when prefixed with appropriate service labels leads to associated "secure" TLSA RRs (possibly via a chain of "secure" CNAME RRs). If, for example, the base domain is "mail.example.com", the TLSA RRset is obtained via a DNSSEC query of the form:

```
_25._tcp.mail.example.com. IN TLSA ?
```

Typically, the destination TCP port is 25, but this may be different with custom routes specified by the MTA administrator or when an MUA connects to a submission server on port 587. The SMTP client MUST use the appropriate "_<port-number>" prefix in place of "_25" when the port number is not equal to 25. The query response may be a CNAME (or a DNAME + CNAME combination), or the TLSA RRset. If the record is a CNAME or DNAME, the SMTP client restarts the TLSA query at the target domain, following CNAMEs as appropriate.

CNAMEs encountered during TLSA record lookups can be used to share a single TLSA RRset specifying a common certificate authority or a common leaf certificate for multiple TLS services. Such CNAME

expansion does not change the SMTP client's notion of the TLSA base domain, thus when _25._tcp.mail.example.com is a CNAME the base domain remains mail.example.com and is still used in peer certificate name checks. For example:

```
example.com.                IN MX 0 mail.example.com.
example.com.                IN MX 0 mail2.example.com.
_25._tcp.mail.example.com.  IN CNAME 2.1.1._dane.example.com.
_25._tcp.mail2.example.com. IN CNAME 2.1.1._dane.example.com.
2.1.1._dane.example.com.    IN  TLSA 2 1 1 e3b0c44298fc1c14
                                           9afbf4c8996fb924
                       27ae41e4649b934c
                       a495991b7852b855
```

Here, mail.example.com and mail2.example.com have certificates issued under a common trust-anchor, but each MX host's TLSA base domain remains its hostname and MUST match the subject name (or subject alternative name) in its certificate.

If, after possible CNAME indirection, at least one "secure" TLSA record is found (even if not usable because it is unsupported by the implementation or administratively disabled) the next-hop host has committed to TLS support. The SMTP client SHOULD NOT deliver mail via such a next-hop host unless a TLS session is negotiated via STARTTLS. This avoids man in the middle STARTTLS downgrade attacks.

As noted previously (Section 2.1.1, Section 2.1.2), when no TLSA records are found at a CNAME-expanded name (due to an insecure response or a lack of TLSA records verified by DNSSEC's proof-of-non-existence), the unexpanded name MUST be tried instead. This supports clients of hosting providers where the provider's zone is not DNSSEC validated, but the client has shared appropriate key material with the hosting provider to enable TLS via SNI.

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. Unless local policy specifies that opportunistic DANE TLS is not to be used for a particular destination, client MUST NOT deliver mail via a server whose certificate chain fails to match at least one TLSA record when usable TLSA records are available.

SMTP clients employing opportunistic DANE TLS and TLSA record publishers for SMTP servers need to follow the guidance outlined in [I-D.ietf-dane-ops]'s "Certificate Name Check Conventions", "Service Provider and TLSA Publisher Synchronization" and "TLSA Base Domain and CNAMEs" sections.

2.2. DANE authentication

2.2.1. TLSA certificate usages

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

2.2.1.1. Certificate usage 3

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. TLSA records published for SMTP servers SHOULD be "3 1 1" records to support opportunistic SMTP over TLS with DANE. This record specifies the SHA-256 digest of the server's public key. Since all DANE implementations are required to support SHA-256, this record works for all clients and need not change across certificate renewals with the same key.

Authentication via certificate usage "3" TLSA records involves simply checking that the server's leaf certificate matches the TLSA record. Other than extracting the relevant certificate elements for comparison, no other use is made of the certificate content. Authentication via certificate usage "3" TLSA records involves no certificate authority signature checks. It also involves no server name checks, and thus does not impose any new requirements on the names contained in the server certificate (SNI is not required when the TLSA record matches server's default certificate).

Two TLSA records will need to be published before updating a server's public key, 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 time out the previous TLSA RRset, which contains only the old key, the server may be reconfigured to use the new private key and associated public key certificate. The amount of time a server should wait before using a new key that is referenced by new TLSA records should be at least twice the TTL of the previously published TLSA records. Once the server is using a new key, the obsolete TLSA RR can be removed from DNS, leaving only the RR that matches the new key.

2.2.1.2. Certificate usage 2

Some domains may prefer to reduce the operational complexity of publishing unique TLSA RRs for each TLS service. If the domain employs a common issuing certificate authority to create certificates for multiple TLS services, it may be simpler to publish the issuing authority's public key as a trust-anchor for the certificate chains of all relevant services. The TLSA RRs for each service issued by the same TA may then be CNAMEs to a common TLSA RRset that matches the TA. In this case, the certificate chain presented in the TLS handshake of each service SHOULD include the TA certificate, as SMTP clients cannot generally be expected to have domain-issued trust-anchor certificates in their trusted certificate store. TLSA Publishers should publish either "2 1 1" or "2 0 1" TLSA parameters, which specify the SHA-256 digest of the trust-anchor public key or certificate respectively. As with leaf certificate rollover discussed in Section 2.2.1.1, two such TLSA RRs need to be published to facilitate TA certificate rollover.

The usability of "2 1 1" or "2 0 1" TLSA RRs with SMTP is not assured. If server operators employing these RRs universally ensure that the corresponding TA certificate is included in the SMTP server's TLS handshake certificate chain, clients can safely enable support for these RRs. If sufficiently many server administrators negligently omit the TA certificate from the server's TLS certificate chain, SMTP clients will be hesitant to support usage "2" TLSA RRs, since mail delivery will not work to many destination domains if they do. Server operators are encouraged to implement these RRs, if they are operationally a better fit for their organization, provided they do so with care. It is critical to not forget to include trust-anchor certificates in server trust chains. SMTP client implementations SHOULD support these TLSA RRs, unless, despite the above warning, a non-trivial fraction of server operators fail to publish certificate chains that include the required TA certificate.

2.2.1.3. Certificate usages 0 and 1

SMTP servers SHOULD NOT publish TLSA RRs with certificate usage "0" or "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) perform [RFC6125] server identity verification.

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

2.2.2. Certificate matching

When at least one usable "secure" TLSA record is found, the SMTP client SHOULD use TLSA records to authenticate the next-hop host, mail SHOULD not be delivered via this next-hop host if authentication fails, otherwise the SMTP client is vulnerable to TLS man in the middle attacks.

To match a server via a TLSA record with certificate usage "2", the client MUST perform name checks to ensure that it has reached the correct server. In all cases the SMTP client MUST accept the TLSA base domain as a valid DNS name in the server certificate.

MX: If the TLSA base domain was obtained indirectly via an MX lookup (it is the name of an MX exchange that may be securely CNAME expanded), then the initial query name used in the MX lookup SHOULD be accepted in the peer certificate. The CNAME-expanded initial query name SHOULD also be accepted if different from the initial query name.

Non-MX: If no MX records were found and the TLSA base domain is the CNAME-expanded initial query name, then the initial query name SHOULD also be accepted.

Accepting certificates with the next-hop domain in addition to the next-hop MX host allows a domain with multiple MX hosts to field a single certificate bearing the email domain name across all the MX hosts, this is also compatible with pre-DANE SMTP clients that are configured to look for the email domain name in server certificates.

The SMTP client MUST NOT perform certificate usage name checks with certificate usage "3", since with usage "3" the server is authenticated directly by matching the TLSA RRset to its certificate or public key without resort to any issuing authority. The certificate content is ignored except in so far as it is used to match the certificate or public key (ASN.1 object or its digest) with the TLSA RRset.

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 SSLv2-compatible SSL HELLO by the SMTP client. The minimum SSL/TLS version for SMTP clients performing DANE authentication is SSLv3.

Each SMTP server MUST present a certificate chain (see [RFC2246] 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 RRset includes records with a matching type indicating a digest record (i.e., a value other than "0"), the SHA-256 digest of any object SHOULD be provided along with any other digest published, since clients may support only SHA-256. Unless SHA-256 proves vulnerable to a "second preimage" attack, it should be the only digest algorithm used in TLSA records.

If the server's TLSA records match the server's default certificate chain, the server need not support SNI. The server need not include the extension in its TLS HELLO, simply returning a matching certificate chain is sufficient. Servers MUST NOT enforce the use of SNI by clients, if the client sends no SNI extension, or sends an SNI extension for an unsupported domain the server MUST simply use its default certificate chain. The client may be using unauthenticated opportunistic TLS and may not expect any particular certificate from the server.

The SMTP client MAY include anonymous TLS ciphersuites in its SSL HELO. MX hosts that receive email from the Internet MUST interoperate with opportunistic TLS SMTP clients. If they advertise support for STARTTLS in their SMTP EHLO response, they MUST NOT fail to complete the TLS handshake merely because the SMTP client offered some ciphersuites that do not provide for server authentication.

While server operators are under no obligation to implement or enable anonymous ciphers, no security is gained by sending certificates clients are willing to ignore. Indeed support for anonymous ciphersuites in the server makes audit trails more useful when the chosen ciphersuite is logged, as this will in many cases record which clients did not care to authenticate the server. For example, the Postfix SMTP server enables anonymous TLS ciphersuites by default, and the Postfix SMTP client offers these at its highest preference when server authentication is not applicable.

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, therefore authentication via this protocol is expected to be less prone to connection failure caused by incompatible configuration of the client and server.

2.2.3. Digest algorithm agility

While [RFC6698] specifies multiple digest algorithms it does not explicitly specify a protocol by which the publisher can agree on the strongest shared algorithm, and thereby avoind exposure to any deprecated weaker algorithms that are published out of interoperability concerns, but should if possible be ignored. We specify such a protocol below.

Suppose that a DANE TLS client authenticating TLS server considers digest algorithm X stronger than digest algorithm Y. Suppose further that that a server's TLSA RRset contains some records with X as the digest algorithm. 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 digest Y (with some usage and selector) it also appears with digest X (with the same usage and selector). In that case our client can safely ignore TLSA records with the weaker digest Y, because it suffices to check the records with the stronger algorithm X.

We take the simplest appraoch and mandate that all published TLSA RRsets conform to the above assumptions. Then clients can unconditionally ignore all but the (equal) strongest digest records with a given usage and selector.

Records with a matching type of "0", that publish the verbatim object value play no role in digest algorithm agility. They neither preempt the processing of records that employ digests, nor are they ignored in the presence of any digest records.

Therefore, server operators MUST ensure that for any given usage and selector, ALL objects with certificate association data with that usage and selector that are published with a digest matching type are published with the SAME SET of digests (non-zero matching types). In other words, for each usage and selector, the records with non-zero matching types will be a cross-product of a set of underlying objects and a fixed set of digests that apply uniformly to all the objects.

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 only process any usable records with a matching type of "0" and any usable records whose digest is the strongest among usable records with the same 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 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 currently deployed certificates causing connectivity issues until new keys or certificates are fielded. Similarly, publishing new records with fewer digests could cause problems once the new keys are deployed.

Therefore, server operators SHOULD follow the following rule. When adding or removing objects from the TLSA RRset (e.g. during key rotation), DO NOT change the set of digests used, change just the list of objects. When changing the set of digests used, change only the digests, and generate a new RRset in which all the existing objects are re-published with the new set of digests.

The client-side of this "digest algorithm agility" protocol is enabled by default in the first DANE for SMTP implementation. For key rotation to work non-disruptively server operators MUST ensure that their TLSA records conform with the above specification.

3. Opportunistic TLS for Submission

Prior to [RFC6409], the SMTP submission protocol was a poster-child for PKIX TLS. The MUA typically connects to one or more submission servers explicitly configured by the user. There is no indirection via insecure MX records, and unlike web browsers, there is no need to authenticate a large set of TLS servers. Once TLS is enabled for the desired submission server or servers, provided the server certificate is correctly maintained, the MUA is able to reliably use TLS to authenticate the submission server.

[RFC6186] aims to simplify the configuration of the MUA submission service by dynamically deriving the submission service from the user's email address. This is done via SRV records, but at the cost of introducing the same TLS security problems faced by MTA to MTA SMTP. Prompting the user when the SRV record domain is different from the email domain is not a robust solution.

The protocol defined in this memo can also be used to secure submission service discovery. If the email domain is DNSSEC signed, the SRV records are "secure" and the SRV host publishes secure TLSA records for submission, then the MUA can safely auto-configure to authenticate the submission server via DANE. When DANE TLSA records are not available, the client SHOULD fall back to legacy behavior (this may involve prompting the user to accept the resulting server and perhaps "pin" its certificate).

Specifically, MUAs that dynamically determine the submission server via SRV records SHOULD support DNSSEC and DANE TLSA records. They

SHOULD use TLSA records to authenticate the server. The processing of usage 2 and 3 TLSA associations by an MUA is the same as by an MTA with SRV records replaced by corresponding MX records.

Just as with MX service on port 25, SMTP submission servers SHOULD NOT publish usage 0 or 1 TLSA associations, and MUAs that support DANE TLSA are not expected to trust a full list of public CAs. Server certificate subjectAltNames should include at least the server name. When the server administrator is able to obtain a certificate for the email domain, the server certificate should also include the email domain name. MUAs that are not able to support DNSSEC may then be able to authenticate the server domain. If it is practical to field additional certificates for hosted domains, SNI may be used by the server to select the appropriate domain's certificate.

4. Mandatory TLS Security

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

Specifically, with domains that publish DANE TLSA records for their MX hosts a sending MTA can be configured to use the receiving domains's DANE TLSA records to authenticate the corresponding MX hosts, thereby obviating the complex manual provisioning process. In anticipation of, or in response to, a failure to obtain the expected TLSA records, the sending system's administrator may choose from a selection of fallback options, if supported by the sending MTA:

- Defer mail if no usable TLSA records are found. This is useful when the destination is known to publish TLSA records, and lack of TLSA records is most likely a transient misconfiguration.

- Authenticate the peer via a manually configured certificate digest. This may be obtained, for example, after a problem is detected and confirmed to be valid by some out-of-band mechanism.

- Authenticate the peer via the existing public CA PKI, if the peer server has usable CA issued certificates. In many cases the sending MTA will need custom certificate name matching rules to match the destination's gateways. And the sending server must explicitly configure policy for the destination to always require TLS to prevent MITM attacks.

- o Send via unauthenticated mandatory TLS. This is useful if the requirement is merely to always encrypt transmissions to protect against only eavesdropping, and the possibility of MITM attacks is less of a concern than timely email delivery.

It should be noted that barring administrator intervention, email SHOULD be deferred when DNSSEC lookups fail, (as distinct from "secure" non-existence of TLSA records, or secure evidence that the domain is no longer signed). In addition to configuring fallback strategies when TLSA records are unexpectedly absent, administrators may, in hopefully rare cases, need to disable DNSSEC lookups for a destination to work around a DNSSEC outage.

5. 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 Tony Finch who finally prodded me into participating in DANE working group discussions. Thanks to Paul Hoffman who motivated me to produce this memo and provided feedback on early drafts. Thanks also to Wietse Venema who created Postfix, and patiently guided the Postfix DANE implementation to production quality.

6. 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 hosts, this protocol allows sending MTAs (and perhaps dynamically configured MUAs) 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 and 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, deprecating certificate usages 0 and 1 in this specifications improves interoperability without degrading security.

7. 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.

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

[RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

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

[RFC3546] Blake-Wilson, S., Nystrom, M., Hopwood, D., Mikkelsen, J., and T. Wright, "Transport Layer Security (TLS) Extensions", RFC 3546, June 2003.

[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.

[RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[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.

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

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

Authors' Addresses

Viktor Dukhovni
Unaffiliated

Email: ietf-dane@dukhovni.org

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

Email: ietf@hardakers.net