DANE   V. Dukhovni
Internet-Draft   Unaffiliated
Intended status: Experimental   W. Hardaker
Expires: April 11, 2014   Parsons
                                                        October 08, 2013

SMTP security via opportunistic DANE TLS
draft-ietf-dane-smtp-with-dane-00

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

This Internet-Draft will expire on April 11, 2014.

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 provide certificates which match a trustable identifier. 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 Transport 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. As a result, secure verification of MX host certificates is not possible without DNSSEC, as an active attacker can forge DNS replies with fake MX records, and can direct traffic to a server of their choice. 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 the key management is operationally challenging at large scale unless 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, nor 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 only 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 names (or content digests) to expect in the presented MX host certificates of those domains. Such statically configured SMTP secure channels are also used rarely, and only 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, although this is not yet widely deployed). 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 reply on the existing public CA PKI. This is not simply a result of complacency on the part SMTP server administrators and MTA developers. Nor is it just a result of the relative maturity of the SMTP infrastructure when 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 section 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 unencrypted and unauthenticated). 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. A downgrade resistant mechanism for a server to advertise TLS support based on DANE TLSA records is specified below. 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.

Typically, the next-hop destination defaults to 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. In the language of [RFC5321] Section 5.1, we'll refer to this next-hop destination host or domain as "the initial name".

2.1.1. MX resolution

If the initial name is a next-hop domain subject to MX resolution, a DNSSEC validated "MX" lookup is performed, to obtain the list of associated MX hosts. If no MX records are found, or if the initial name is a next-hop host not subject to MX resolution, it is resolved to one or more network addresses, by performing DNSSEC validated "A" and/or "AAAA" lookups.

Following [RFC5321] Section 5.1, if the "A", "AAAA" or "MX" lookup of the initial name yields a CNAME, we replace it with the resulting name as if it were the initial name and try the same lookup again with the new name. MTAs typically support limited recursion in CNAME expansion so this replacement is performed recursively. If initially, or at any stage of recursion, the response is "bogus", MX resolution fails with a temporary error. Mail delivery SHOULD either be deferred or attempted via any alternative delivery channel configured by the MTA administrator (which may also employ opportunistic DANE TLS).

With a next-hop destination domain subject to MX resolution which has MX records, if at least one lookup in the (possibly empty) chain of CNAMEs leading to the MX RRset is "insecure", opportunistic DANE TLS is not applicable, and mail delivery may proceed with pre-DANE opportunistic TLS (subject to its various MITM attacks).

With a next-hop destination host not subject to MX resolution or a domain with no MX records, if at least one lookup in the (possibly empty) chain of CNAMEs leading to the A or AAAA RRset is "insecure", the TLSA base domain is the initial next-hop name, and opportunistic DANE TLS is applicable only when a "secure" TLSA RRset is found at that base domain.

Otherwise, if at each and every stage of CNAME expansion the DNS response is "secure", and either the initial name is a next-hop host name not subject to MX resolution or no MX records are found, the resulting final name becomes the next-hop destination and is the base domain for TLSA record lookup. In summary, the TLSA base domain is the fully CNAME expanded name that is "secure" or else is the initial name.

Finally, if at each and every stage the DNS response is "secure", and and one or more MX records are found, the MX records MUST be sorted by preference. A better (numerically lower) MX preference for a host

that does not support TLS MUST NOT be preempted by a worse (numerically higher) MX preference for a host that does support TLS. In other words, avoiding delivery loops trumps any preference for channel security. In each delivery attempt via a candidate MX host, the MX host MUST be treated as though it were the initial next-hop destination host (which is, of course, not subject to further MX resolution). The associated TLSA base domain is equal to the CNAME resolved MX exchange name if CNAME expansion of MX exchange names is supported and all CNAMEs encountered are "secure". Otherwise, the unexpanded name of the MX exchange is the TLSA base domain.

CNAMEs are not legal in the exchange field of MX records, thus MTAs MAY skip over MX records in which the MX exchange is a CNAME. There is some additional risk, in this case, 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 whether opportunistic DANE TLS is applicable and if so the associated TLSA RRset base domain.

2.1.2. TLSA record lookup

When all the DNSSEC lookups, "CNAME", "MX", "A" or "AAAA", used to obtain a given TLSA base domain (one for each candidate MX host if multiple DNSSEC validated MX hosts were found) are "secure", and the SMTP client is configured for opportunistic DANE TLS, it SHOULD locate the TLSA RRset corresponding to this base domain. 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>" 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. DNAME processing with DNSSEC can be done using standard DNAME resolution techniques and will not be discussed in detail here. The SMTP client MUST check the security status of the response.

If the response is "bogus", delivery via the host in question SHOULD NOT proceed, otherwise the SMTP client is vulnerable to man in the middle STARTTLS downgrade attacks. If the response is "insecure", opportunistic DANE TLS is not applicable for the host in question, and the SMTP client SHOULD proceed with ordinary opportunistic TLS.

If the response is "secure" and the record is a CNAME or DNAME, the SMTP client restarts the TLSA query at the target domain, following CNAMEs as appropriate (such CNAME expansion does not change the SMTP client's notion of the TLSA base domain).

If, after possible CNAME indirection, the response is "secure" and at least one 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.

When no TLSA records are found at a CNAME-expanded initial name (insecure response or no records), the unexpanded initial name MUST be tried instead. This supports clients of hosting providers where the provider zone is not DNSSEC validated, but the client has shared appropriate key material with the hosting provider to enable TLS via SNI.

When usable TLSA records are available, a client SHOULD NOT deliver mail via a server that fails to match at least one TLSA record. This is not a "must" because clients may incrementally deploy opportunistic DANE TLS only for selected peer domains. At times, clients may need to disable opportunistic DANE TLS for peers that fail to interoperate due to misconfiguration or software defects on either end. For opportunistic DANE TLS to be robust (resistant to failures), servers MUST live up to the promises stated by the existence of the TLSA record, but it is not always possible to compel clients to use a security policy chosen by the server. Given a robust security protocol, clients will hopefully, over time, willingly choose to adopt it.

SMTP clients employing opportunistic DANE TLS and TLSA record publishers for SMTP servers need to follow the guidance outlined in [I-D.dukhovni-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

As noted in the introduction, the existing public CA PKI is not viable for the Internet email backbone. TLSA records for MX hosts or submission servers that are to be found via SRV records SHOULD NOT include certificate usage "0" or "1", as in both cases SMTP clients cannot be expected to perform [RFC5280] PKIX validation or [RFC6125]

identity verification. Clients MAY treat such TLSA records as unusable.

SMTP clients may also to the extent possible map these usages to the corresponding non-PKIX certificate usages (0 to 2 and 1 to 3). Servers publishing these certificate usages hoping to be protected by both the public CA PKI and by DNSSEC will typically be protected by neither.

TLSA Publishers should follow the TLSA publication size guidance found in [I-D.dukhovni-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.

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 the public key of the server's default certificate). It uses the SHA-256 digest which all clients are obligated to support, and works across certificate renewals with the same key.

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 regular 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 trust chain, clients can safely enable support for these RRs. If sufficiently many server administrators are negligent in deploying these RRs, SMTP clients will be hesitant to support them, 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 never forget to include trust-anchor certificates in server trust chains. SMTP client implementations SHOULD support these TLSA RRs, unless server operators fail publish certificate chains that include the required TA certificate.

2.2.1.3. Certificate usage 1

SMTP servers SHOULD NOT publish TLSA RRs with certificate usage "1". Clients MAY treat such TLSA records as unusable. Alternatively, SMTP clients that implement this specification MAY ignore the PKIX validation requirement when they encounter certificate usage "1", and authenticate the server per the content of the TLSA record alone. That is, SMTP clients may treat certificate usage "1" as certificate usage "3".

2.2.1.4. Certificate usage 0

SMTP servers SHOULD NOT publish TLSA RRs with certificate usage "0". Clients MAY treat such TLSA records as unusable. Alternatively,

since PKIX validation is not possible with opportunistic DANE TLS, SMTP clients MAY treat certificate usage "0" RRs as though they were certificate usage "2" RRs. But, with certificate usage "0" the usability of the TLSA record depends more strongly on its matching type.

If the matching type is "0" (the server should also avoid this matching type and should publish usage "3" or "2" public key or certificate digests), the TLSA record contains the full certificate or full public key of the trusted certificate authority. In this case the client has all the information it needs to match the server trust-chain to the TLSA record. The client SHOULD ignore the PKIX validation requirement, and verify the server's trust chain via its DANE TLSA records only (name checks still apply as with usage "2").

If the matching type is not "0", the TLSA record contains only a digest of the trust certificate authority certificate or public key. The server operator publishing usage 0 TLSA records may expect that clients already have the issuing authority certificate on hand, and may omit it from the server's certificate chain. As a result, the client may not be able to match the server trust chain against the TLSA record if it, in fact, does not have a copy of the certificate authority certificate or public key.

SMTP clients that implement this specification SHOULD treat TLSA records with certificate usage "0" and a digest matching type as unusable, but MAY be explicitly configured to support them when it is believed that clients posses a sufficiently complete set of trusted public CA certificates. This is most plausible with an MUA which only needs enough CA certificates to authenticate its preferred submission service.

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. The SMTP client MUST accept the TLSA base domain as a valid DNS name in the server certificate. Clients should also accept securely looked up TLSA base domain obtained indirectly via an MX lookup, or a CNAME resolved expansion.

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 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 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. Since DANE-aware clients are obligated to send SNI information, which requires at least TLS 1.0, SMTP servers for which DANE TLSA records are published MUST support TLS 1.0 or later with any client authorized to use the service.

Each SMTP server MUST present a certificate trust 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 indication 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 client may even offer to use anonymous TLS ciphersuites and servers SHOULD support these, no security is gained by forcing the use of a certificate the client will ignore. Indeed support for anonymous ciphersuites in the server makes audit trails more useful if the chosen ciphersuite is logged, as this will in many cases

record which clients did not care to authenticate the server. (The Postfix SMTP server supports 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.

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 opportunistically secure the submission service association. 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.

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 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 also authorized to obtain certificates 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.

- 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.dukhovni-dane-ops]
Dukhovni, V., "DANE TLSA implementation and operational guidance", draft-dukhovni-dane-ops-00 (work in progress), May 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