

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 12, 2014

P. Wouters
Red Hat
April 10, 2014

Best Common Practise for using OPENPGPKEY records
draft-ietf-dane-openpgpkey-usage-00

Abstract

The OPENPGPKEY DNS Resource Record can be used to match an email address to an OpenPGP key. This document specifies a Best Common Practise ("BCP") for email clients, MUA's and MTA's for using the OPENPGPKEY DNS Resource Record.

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 12, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	2
2. The OPENPGPKEY record presence	2
3. OpenPGP public key considerations	3
3.1. Public Key UIDs and email addresses	3
3.2. Public Key UIDs and IDNA	3
3.3. Public Key UIDs and synthesized DNS records	3
3.4. OpenPGP Key size and DNS	4
4. Security Considerations	4
4.1. Email address information leak	4
4.2. OpenPGP security and DNSSEC	5
4.3. MTA behaviour	5
4.4. MUA behaviour	6
4.5. Email client behaviour	6
5. References	7
5.1. Normative References	7
5.2. Informative References	7
Author's Address	8

1. Introduction

This document describes a Best Current Practise ("BCP") for using OPENPGPKEY DNS Resource Records xref target="OPENPGPKEY"/ in email clients, MUA's and MTA.

1.1. 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 RFC 2119 [RFC2119].

This document also makes use of standard DNSSEC and DANE terminology. See DNSSEC [RFC4033], [RFC4034], [RFC4035], and DANE [RFC6698] for these terms.

2. The OPENPGPKEY record presence

A user who publishes an OPENPGPKEY record in DNS explicitly prefers receiving encrypted email over receiving unencrypted email.

A user who publishes an OPENPGPKEY record in DNS still expects senders to perform their due diligence by additional verification of their public key via other out-of-band methods before sending any confidential or sensitive information

In other words, the OPENPGPKEY record in DNS, without any additional verification, should be used only as an alternative to sending plaintext email. It SHOULD NOT be used to change one's opinion on whether it is safe or appropriate to sent the content via email in the first place.

3. OpenPGP public key considerations

Once an OPENPGPKEY resource record has been found and the OpenPGP public keyring has been decoded, the right public key must be located inside the keyring. For a public key in the keyring to be usable, the public key has to have a key uid as specified in [RFC4648] that matches the email address for which the OPENPGPKEY RR lookup was performed.

3.1. Public Key UIDs and email addresses

An OpenPGP public key can be associated with multiple email addresses by specifying multiple key uids. The OpenPGP public key obtained from a OPENPGPKEY RR can be used as long as the target recipient's email address appears as one of the OpenPGP public key uids. The name part (left of the @) should appear in the native format, not its SHA2-224 hash that was used to lookup the OPENPGPKEY RR.

3.2. Public Key UIDs and IDNA

Internationalized domains that use non-ascii characters (U-label) are encoded in DNS using IDNA [RFC5891] - also referred to as punycode or A-label. When matching OpenPGP public key uids, both the email address specified using U-label and A-label should be considered as valid public key uids.

3.3. Public Key UIDs and synthesized DNS records

CNAME's (see [RFC2181]) and DNAME's (see [RFC6672]) can be followed to obtain an OPENPGPKEY RR, as long as the original recipient's email address appears as one of the OpenPGP public key uids. For example, if the OPENPGPKEY RR query for hugh@example.com (8d57[...]b7._openpgpkey.example.com) yields a CNAME to 8d57[...]b7._openpgpkey.example.net, and an OPENPGPKEY RR for 8d57[...]b7._openpgpkey.example.net exists, then this OpenPGP public key can be used, provided one of the key uids contains "hugh@example.com". This public key cannot be used if it would only contain the key uid "hugh@example.net".

If one of the OpenPGP key uids contains only a single wildcard as the LHS of the email address, such as "*@example.com", the OpenPGP public key may be used for any email address within that domain. Wildcards

at other locations (eg hugh@*.com) or regular expressions in key uids are not allowed, and any OPENPGPKEY RR containing these should be ignored.

3.4. OpenPGP Key size and DNS

Although the reliability of the transport of large DNS Resource Records has improved in the last years, it is still recommended to keep the DNS records as small as possible without sacrificing the security properties of the public key. The algorithm type and key size of OpenPGP keys should not be modified to accomodate this section.

OpenPGP supports various attributes that do not contribute to the security of a key, such as an embedded image file. It is recommended that these properties are not exported to OpenPGP public keyrings that are used to create OPENPGPKEY Resource Records. Some OpenPGP software, for example GnuPG, have support for a "minimal key export" that is well suited to use as OPENPGPKEY RDATA.

4. Security Considerations

The main goal of the OPENPGPKEY resource record is to stop passive attacks against plaintext emails. While it can also thwart some active attacks (such as people uploading rogue keys to key servers in the hopes that others will encrypt to these rogue keys), this resource record is not a replacement for verifying OpenPGP public keys via the web of trust signatures, or manually via a fingerprint verification.

Various components could be responsible for encrypting an email message to a target recipient. It could be done by the sender's email client or software plugin, the sender's Mail User Agent (MUA) or the sender's Mail Transfer Agent (MTA). Each of these have their own characteristics. An email client can direct the human to make a decision before continuing. The MUA can either accept or refuse a message. The MTA must deliver the message as-is, or encrypt the message before delivering. Each of these programs should ensure that the security of an email message is never downgraded, and that an unencrypted received message will be encrypted whenever possible.

Organisations that require to be able to read everyone's encrypted email should publish the escrow key as the OPENPGPKEY record. Upon receipt, such mail servers can optionally re-encrypt the message to the individual's OpenPGP key.

4.1. Email address information leak

DNS zones that are signed with DNSSEC using NSEC for denial of existence are susceptible to zone-walking, a mechanism that allow someone to enumerate all the names in the zone. Someone who wanted to collect email addresses from a zone that uses OPENPGPKEY might use such a mechanism. DNSSEC-signed zones using NSEC3 for denial of existence are significantly less susceptible to zone-walking. Someone could still attempt a dictionary attack on the zone to find OPENPGPKEY records, just as they can use dictionary attacks on an SMTP server or grab the entire contents of existing PGP key servers to see which addresses are valid.

4.2. OpenPGP security and DNSSEC

DNSSEC key sizes are chosen based on the fact that these keys can be rolled with next to no requirement for security in the future. If one doubts the strength or security of the DNSSEC key for whatever reason, one simply rolls to a new DNSSEC key with a stronger algorithm or larger key size.

This effectively means that anyone who can obtain a DNSSEC private key of a domain name via coercion, theft or brute force calculations, can replace any OPENPGPKEY record in that zone and all of the delegated child zones, irrespective of the key length strength of the OpenPGP keypair.

Therefor, DNSSEC is not an alternative for the "web of trust" or for manual fingerprint verification by humans. It is a solution aimed to ease obtaining someone's public key, and without manual verification should be treated as "better than plaintext" only. While this thwarts all passive attacks that simply capture and log all plaintext email content, it is not a security measure against active attacks.

4.3. MTA behaviour

An MTA could be operating in a stand-alone mode, without access to the sender's OpenPGP public keyring, or in a way where it can access the user's OpenPGP public keyring. Regardless, the MTA MUST NOT modify the user's OpenPGP keyring.

An MTA sending an email MUST NOT add the public key obtained from an OPENPGPKEY resource record to a permanent public keyring for future use beyond the TTL.

If the obtained public key is revoked, the MTA MUST NOT use the key for encryption, even if that would result in sending the message in plaintext.

If a message is already encrypted, the MTA SHOULD NOT re-encrypt the message, even if different encryption schemes or different encryption keys were used.

If an OPENPGPKEY resource record is received without DNSSEC protection, it MAY still be used for encryption.

If the DNS request for an OPENPGPKEY record returned an "indeterminate" or "bogus" answer, the MTA MUST NOT send the message and queue the plaintext message for delivery at a later time. If the problem persists, the email should be returned via the regular bounce methods.

If multiple non-revoked OPENPGPKEY resource records are found, the MTA SHOULD pick the most secure RR based on its local policy. [or should it encrypt to both?]

4.4. MUA behaviour

If the public key for a recipient obtained from the locally stored sender's public keyring differs from the recipient's OPENPGPKEY RR, the MUA MUST NOT accept the message for delivery.

If the public key for a recipient obtained from the locally stored sender's public keyring contains contradicting properties for the same key obtained from an OPENPGPKEY RR, the MUA SHOULD NOT accept the message for delivery.

If multiple non-revoked OPENPGPKEY resource records are found, the MUA SHOULD pick the most secure OpenPGP public key based on its local policy.

4.5. Email client behaviour

Email clients should adhere to the above listed MUA behaviour. Additionally, an email client MAY interact with the user to resolve any conflicts between locally stored keyrings and OPENPGPKEY RRdata.

An email client that is encrypting a message SHOULD clearly indicate to the user the difference between encrypting to a locally stored and humanly verified public key and encrypting to an unverified (by the human sender) public key obtained via an OPENPGPKEY resource record.

5. References

5.1. Normative References

- [OPENPGPKEY] Wouters, P., "DANE for OpenPGP public keys", draft-ietf-wouters-dane-openpgp (work in progress), April 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [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.
- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.
- [RFC4880] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, November 2007.
- [RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, August 2010.

5.2. Informative References

- [RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, July 1997.
- [RFC2822] Resnick, P., "Internet Message Format", RFC 2822, April 2001.
- [RFC4255] Schlyter, J. and W. Griffin, "Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints", RFC 4255, January 2006.
- [RFC6530] Klensin, J. and Y. Ko, "Overview and Framework for Internationalized Email", RFC 6530, February 2012.

[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.

Author's Address

Paul Wouters
Red Hat

Email: pwouters@redhat.com

DANE
Internet-Draft
Intended status: Standards Track
Expires: January 04, 2015

V. Dukhovni
Unaffiliated
W. Hardaker
Parsons
July 03, 2014

Updates to and Operational Guidance for the DANE Protocol
draft-ietf-dane-ops-05

Abstract

This memo clarifies and updates the DANE TLSA protocol based on implementation experience since the publication of the original DANE specification in [RFC6698]. It also contains guidance for DANE implementers and operators.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 04, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	3
2. DANE TLSA Record Overview	4
2.1. Example TLSA record	6
3. DANE TLS Requirements	6
4. Certificate-Usage-Specific DANE Updates and Guidelines	6
4.1. Certificate Usage DANE-EE(3)	6
4.2. Certificate Usage DANE-TA(2)	8
4.3. Certificate Usage PKIX-EE(1)	11
4.4. Certificate Usage PKIX-TA(0)	11
5. Service Provider and TLSA Publisher Synchronization	12
6. TLSA Base Domain and CNAMEs	15
7. TLSA Publisher Requirements	15
7.1. Rolling a Key Without Changing TLSA Parameters	16
7.2. Switching to DANE-CA from DANE-EE	17
7.3. Switching to New TLSA Parameters	17
7.4. TLSA Publisher Requirements Summary	18
8. Digest Algorithm Agility	18
9. General DANE Guidelines	19
9.1. DANE DNS Record Size Guidelines	19
9.2. Certificate Name Check Conventions	20
9.3. Design Considerations for Protocols Using DANE	21
10. Interaction with Certificate Transparency	22
11. Note on DNSSEC Security	23
12. Summary of Updates to RFC6698	24
13. Security Considerations	25
14. IANA Considerations	25
15. Acknowledgements	25
16. References	25
16.1. Normative References	25
16.2. Informative References	26
Authors' Addresses	27

1. Introduction

[RFC6698] specifies a new DNS resource record "TLSA" that associates a public certificate or public key of a trusted leaf or issuing authority with the corresponding TLS transport endpoint. These DANE TLSA records, when validated by DNSSEC, can be used to augment or replace the trust model of the existing public Certification Authority (CA) Public Key Infrastructure (PKI).

[RFC6698] defines three TLSA record fields with respectively 4, 2 and 3 currently specified values. These yield 24 distinct combinations of TLSA record types. This many options have been lead to implementation and operational complexity. This memo will recommend

best-practice choices to help simplify implementation and deployment given these plethora of choices.

Implementation complexity also arises from the fact that the TLS transport endpoint is often specified indirectly via Service Records (SRV), Mail Exchange (MX) records, CNAME records or other mechanisms that map an abstract service domain to a concrete server domain. With service indirection there are multiple potential places for clients to find the relevant TLSA records. Service indirection is often used to implement "virtual hosting", where a single Service Provider transport endpoint simultaneously supports multiple hosted domain names. With services that employ TLS, such hosting arrangements may require the Service Provider to deploy multiple pairs of private keys and certificates with TLS clients signaling the desired domain via the Server Name Indication (SNI) extension ([RFC6066], section 3). This memo provides operational guidelines intended to maximize interoperability between DANE TLS clients and servers.

In the context of this memo, channel security is assumed to be provided by TLS or DTLS. The Transport Layer Security (TLS) [RFC5246] and Datagram Transport Layer Security (DTLS) [RFC6347] protocols provide secured TCP and UDP communication over IP. By convention, "TLS" will be used throughout this document and, unless otherwise specified, the text applies equally well to the DTLS protocol. Used without authentication, TLS provides protection only against eavesdropping through its use of encryption. With authentication, TLS also provides integrity protection and authentication, which protect the transport against man-in-the-middle (MITM) attacks.

Other related documents that build on [RFC6698] are [I-D.ietf-dane-srv] and [I-D.ietf-dane-smtp-with-dane]. In Section 12 we summarize the updates this document makes to [RFC6698].

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 are used throughout this document:

Service Provider: A company or organization that offers to host a service on behalf of a Customer Domain. The original domain name associated with the service often remains under the control of the customer. Connecting applications may be directed to the Service

Provider via a redirection resource record. Example redirection records include MX, SRV, and CNAME. The Service Provider frequently provides services for many customers and must carefully manage any TLS credentials offered to connecting applications to ensure name matching is handled easily by the applications.

Customer Domain: As described above, a client may be interacting with a service that is hosted by a third party. We will refer to the domain name used to locate the service prior to any redirection, as the "Customer Domain".

TLSA Publisher: The entity responsible for publishing a TLSA record within a DNS zone. This zone will be assumed DNSSEC-signed and validatable to a trust anchor, unless otherwise specified. If the Customer Domain is not outsourcing their DNS service, the TLSA Publisher will be the customer themselves. Otherwise, the TLSA Publisher is sometimes the operator of the outsourced DNS service.

public key: The term "public key" is short-hand for the subjectPublicKeyInfo component of a PKIX [RFC5280] certificate.

SNI: The "Server Name Indication" (SNI) TLS protocol extension allows a TLS client to request a connection to a particular service name of a TLS server ([RFC6066], section 3). Without this TLS extension, a TLS server has no choice but to offer a PKIX certificate with a default list of server names, making it difficult to host multiple Customer Domains at the same IP-addressed based TLS service endpoint (i.e., "secure virtual hosting").

TLSA parameters: In [RFC6698] the TLSA record is defined to consist of four fields. The first three of these are numeric parameters that specify the meaning of the data in fourth and final field. To avoid language contortions when we need to distinguish between the first three fields that together define a TLSA record "type" and the fourth that provides a data value of that type, we will call the first three fields "TLSA parameters", or sometimes just "parameters" when obvious from context.

2. DANE TLSA Record Overview

DANE TLSA [RFC6698] specifies a protocol for publishing TLS server certificate associations via DNSSEC [RFC4033] [RFC4034] [RFC4035]. The DANE TLSA specification defines multiple TLSA RR types via combinations of numeric values of the first three fields of the TLSA record (i.e. the "TLSA parameters"). The numeric values of these parameters were later given symbolic names in [I-D.ietf-dane-registry-acronyms]. These parameters are:

The 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 PKIX, or public PKI issued, and DANE, or domain-issued trust anchors:

- o When the low bit is set (PKIX-EE(1) and DANE-EE(3)) the TLSA record matches an EE certificate (also commonly referred to as a leaf or server certificate.)
- o When the low bit is not set (PKIX-TA(0) and DANE-TA(2)) the TLSA record matches a trust anchor (a Certification Authority) that issued one of the certificates in the server certificate chain.
- o When the high bit is set (DANE-TA(2) and DANE-EE(3)), the server certificate chain is domain-issued and may be verified without reference to any pre-existing public certification authority PKI. Trust is entirely placed on the content of the TLSA records obtained via DNSSEC.
- o When the high bit is not set (PKIX-TA(0) and PKIX-EE(1)), the TLSA record publishes a server policy stating that its certificate chain must pass PKIX validation [RFC5280] and the DANE TLSA record is used to signal an additional requirement that the PKIX validated server certificate chain also contains the referenced CA or EE certificate.

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. Of the two digest algorithms, for now only SHA2-256(1) is mandatory to implement. Clients SHOULD implement SHA2-512(2), but servers SHOULD NOT exclusively publish SHA2-512(2) digests. The digest algorithm agility protocol defined in Section 8 SHOULD be used by clients to decide how to process TLSA RRsets that employ multiple digest algorithms. Server operators MUST publish TLSA RRsets that are compatible with digest algorithm agility.

2.1. Example TLSA record

In the example TLSA record below:

```
_25._tcp.mail.example.com. IN TLSA PKIX-TA Cert SHA2-256 (  
    E8B54E0B4BAA815B06D3462D65FBC7C0  
    CF556ECCF9F5303EBFBB77D022F834C0 )
```

The TLSA Certificate Usage is DANE-TA(2), the selector is Cert(0) and the matching type is SHA2-256(1). The last field is the Certificate Association Data Field, which in this case contains the SHA2-256 digest of the server certificate.

3. DANE TLS Requirements

[RFC6698] does not discuss what versions of TLS are required when using DANE records. This document specifies that TLS clients that support DANE/TLSA MUST support at least TLS 1.0 and SHOULD support TLS 1.2. TLS clients and servers using DANE SHOULD support the "Server Name Indication" (SNI) extension of TLS.

4. Certificate-Usage-Specific DANE Updates and Guidelines

The four Certificate Usage values from the TLSA record, DANE-EE(3), DANE-TA(2), PKIX-EE(1) and PKIX-TA(0), are discussed below.

4.1. Certificate Usage DANE-EE(3)

In this section the meaning of DANE-EE(3) is updated from [RFC6698] to specify that peer identity matching and that validity interval compliance is based solely on the TLSA RRset properties. We also extend [RFC6698] to cover the use of DANE authentication of raw public keys [I-D.ietf-tls-oob-pubkey] via TLSA records with Certificate Usage DANE-EE(3) and selector SPKI(1).

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, with DANE-EE(3), 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 signatures.

With DANE-EE(3) servers that know all the connecting clients are making use of DANE, they need not employ SNI (i.e., they may ignore the client's SNI message) even when the server is known under multiple domain names that would otherwise require separate certificates. It is instead sufficient for the TLSA RRsets for all the domain names in question to match the server's primary certificate. For application protocols where the server name is obtained indirectly via SRV, MX or similar records, it is simplest to publish a single hostname as the target server name for all the hosted domains.

In organizations where it is practical to make coordinated changes in DNS TLSA records before server key rotation, it is generally best to publish end-entity DANE-EE(3) certificate associations in preference to other choices of certificate usage. DANE-EE(3) TLSA records support multiple server names without SNI, don't suddenly stop working when leaf or intermediate certificates expire, and don't fail when a server operator neglects to include all the required issuer certificates in the server certificate chain.

TLSA records published for DANE servers SHOULD, as a best practice, 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. This TLSA record type easily supports hosting arrangements with a single certificate matching all hosted domain. It is also the easiest to implement correctly in the client.

Another advantage of "DANE-EE(3) SPKI(1)" (with any suitable matching type) TLSA records is that they are compatible with the raw public key TLS extension specified in [I-D.ietf-tls-oob-pubkey]. DANE clients that support this extension can use the TLSA record to authenticate servers that negotiate the use of raw public keys in place of X.509 certificate chains. Provided the server adheres to the requirements of Section 7, the fact that raw public keys are not compatible with any other TLSA record types will not get in the way of successful authentication. Clients that employ DANE to authenticate the peer server SHOULD NOT negotiate the use of raw public keys unless the server's TLSA RRset includes compatible TLSA records.

While it is, in principle, also possible to authenticate raw public keys via "DANE-EE(3) Cert(0) Full(0)" records by extracting the public key from the certificate in DNS, this is in conflict with the indicated selector and requires extra logic on clients that not all implementations are expected to provide. Servers SHOULD NOT rely on "DANE-EE(3) Cert(0) Full(0)" TLSA records to publish authentication data for raw public keys.

4.2. Certificate Usage DANE-TA(2)

This section updates [RFC6698] by specifying a new operational requirement for servers publishing TLSA records with a usage of DANE-TA(2): such servers MUST include the trust-anchor certificate in their TLS server certificate message.

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:

```
www1.example.com.      IN A 192.0.2.1
www2.example.com.      IN A 192.0.2.2
_443._tcp.www1.example.com. IN CNAME tlsa201._dane.example.com.
_443._tcp.www2.example.com. IN CNAME tlsa201._dane.example.com.
tlsa201._dane.example.com. IN TLSA 2 0 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 SHOULD employ SNI to select the appropriate certificate to present to the client.

4.2.1. Recommended record combinations

TLSA records with selector Full(0) are NOT RECOMMENDED. 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 DNS interoperability issues with large TLSA records containing full certificates or keys (see Section 9.1.1).

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 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, enable a subsidiary CA to issue a chain that violates the trust anchor's path length or name constraints.

4.2.2. Trust anchor digests and server certificate chain

With DANE-TA(2) (TLSA records should match the digest of a TA certificate or public key), a complication arises when the TA certificate is omitted from the server's certificate chain, perhaps on the basis of Section 7.4.2 of [RFC5246]:

The sender's certificate MUST come first in the list. Each following certificate MUST directly certify the one preceding it. Because certificate validation requires that root keys be distributed independently, the self-signed certificate that specifies the root certification authority MAY be omitted from the chain, under the assumption that the remote end must already possess it in order to validate it in any case.

With TLSA Certificate Usage DANE-TA(2), there is no expectation that the client is pre-configured with the trust anchor certificate. In fact, client implementations are free to ignore all locally configured trust anchors when processing usage DANE-TA(2) TLSA records and may rely exclusively on the certificates provided in the server's certificate chain. But, with a digest in the TLSA

record, the TLSA record contains neither the full trust anchor certificate nor the full public key. If the TLS server's certificate chain does not contain the trust anchor certificate, DANE clients will be unable to authenticate the server.

TLSA Publishers that publish TLSA Certificate Usage DANE-TA(2) associations with a selector of SPKI(1) or using a digest-based matching type (not Full(0)) MUST ensure that the corresponding server is configured to also include the trust anchor certificate in its TLS handshake certificate chain, even if that certificate is a self-signed root CA and would have been optional in the context of the existing public CA PKI.

4.2.3. Trust anchor public keys

TLSA records with TLSA Certificate Usage DANE-TA(2), selector SPKI(1) and a matching type of Full(0) will publish the full public key of a trust anchor via DNS. In section 6.1.1 of [RFC5280] the definition of a trust anchor consists of the following four parts:

1. the trusted issuer name,
2. the trusted public key algorithm,
3. the trusted public key, and
4. optionally, the trusted public key parameters associated with the public key.

Items 2-4 are precisely the contents of the `subjectPublicKeyInfo` published in the TLSA record. The issuer name is not included in the `subjectPublicKeyInfo`.

With TLSA Certificate Usage DANE-TA(2), the client may not have the associated trust anchor certificate, and cannot generally verify whether a particular certificate chain is "issued by" the trust anchor described in the TLSA record.

When the server certificate chain includes a CA certificate whose public key matches the TLSA record, the client can match that CA as the intended issuer. Otherwise, the client can only check that the topmost certificate in the server's chain is "signed by" the trust anchor's public key in the TLSA record. Such a check may be difficult to implement, and cannot be expected to be supported by all clients.

Thus, servers should not rely on "DANE-TA(2) SPKI(1) Full(0)" TLSA records to be sufficient to authenticate chains issued by the

associated public key in the absense of a corresponding certificate in the server's TLS certificate message. Servers SHOULD include the TA certificate in their certificate chain.

If none of the server's certificate chain elements match a public key specified in a TLSA record, and at least one "DANE-TA(2) SPKI(1) Full(0)" TLSA record is available, clients are encouraged to check whether the topmost certificate in the chain is signed by the provided public key and has not expired, and in that case consider the server authenticated, provided the rest of the chain passes validation including leaf certificate name checks.

4.3. Certificate Usage PKIX-EE(1)

This Certificate Usage is similar to DANE-EE(3), but in addition PKIX verification is required. Therefore, name checks, certificate expiration, etc., apply as they would without DANE. When, for a given application protocol, DANE clients support both DANE-EE(3) and PKIX-EE(1) usages, it should be noted that an attacker who can compromise DNSSEC can replace these with usage DANE-EE(3) or DANE-TA(2) TLSA records of their choosing, and thus bypass any PKIX verification requirements.

Therefore, except when applications support only the PKIX Certificate Usages (0 and 1), this Certificate Usage offers only illusory incremental security over usage DANE-EE(3). It provides lower operational reliability than DANE-EE(3) since some clients may not be configured with the required root CA, the server's chain may be incomplete or name checks may fail. PKIX-EE(1) also requires more complex coordination between the Customer Domain and the Service Provider in hosting arrangements. This certificate usage is NOT RECOMMENDED.

4.4. Certificate Usage PKIX-TA(0)

This section updates [RFC6698] by specifying new client implementation requirements. Clients that trust intermediate certificates MUST be prepared to construct longer PKIX chains than would be required for PKIX alone.

TLSA Certificate Usage PKIX-TA(0) allows a domain to publish constraints on the set of PKIX certification authorities trusted to issue certificates for its TLS servers. This TLSA record matches PKIX-verified trust chains which contain an issuer certificate (root or intermediate) that matches its association data field (typically a certificate or digest).

As with PKIX-EE(1) case, an attacker who can compromise DNSSEC can replace these with usage DANE-EE(3) or DANE-TA(2) TLSA records of his choosing and thus bypass the PKIX verification requirements. Therefore, except when applications support only the PKIX Certificate Usages (0 and 1), this Certificate Usage offers only illusory incremental security over usage DANE-TA(2). It provides lower operational reliability than DANE-TA(2) since some clients may not be configured with the required root CA. PKIX-TA(0) also requires more complex coordination between the Customer Domain and the Service Provider in hosting arrangements. This certificate usage is NOT RECOMMENDED.

TLSA Publishers who publish TLSA records for a particular public root CA, will expect that clients will then only accept chains anchored at that root. It is possible, however, that the client's trusted certificate store includes some intermediate CAs, either with or without the corresponding root CA. When a client constructs a trust chain leading from a trusted intermediate CA to the server leaf certificate, such a "truncated" chain might not contain the trusted root published in the server's TLSA record.

If the omitted root is also trusted, the client may erroneously reject the server chain if it fails to determine that the shorter chain it constructed extends to a longer trusted chain that matches the TLSA record. Thus, when matching a usage PKIX-TA(0) TLSA record, a client SHOULD NOT always stop extending the chain when the first locally trusted certificate is found. If no TLSA records have matched any of the elements of the chain, and the trusted certificate found is not self-issued, the client MUST attempt to build a longer chain in the hope that a certificate closer to the root may in fact match the server's TLSA record.

5. Service Provider and TLSA Publisher Synchronization

Complications arise when the TLSA Publisher is not the same entity as the Service Provider. In this situation, the TLSA Publisher and the Service Provider must cooperate to ensure that TLSA records published by the TLSA Publisher don't fall out of sync with the server certificate used by the Service Provider.

Whenever possible, the TLSA Publisher and the Service Provider should be the same entity. Otherwise, changes in the service certificate chain must be carefully coordinated between the parties involved. Such coordination is difficult and service outages will result when coordination fails.

Having the master TLSA record in the Service Provider's zone avoids the complexity of bilateral coordination of server certificate

configuration and TLSA record management. Even when the TLSA RRset must be published in the Customer Domain's DNS zone (perhaps the client application does not "chase" CNAMEs to the TLSA base domain), it is possible to employ CNAME records to delegate the content of the TLSA RRset to a domain operated by the Service Provider. Certificate name checks generally constrain the applicability of TLSA CNAMEs across organizational boundaries to Certificate Usages DANE-EE(3) and DANE-TA(2):

Certificate Usage DANE-EE(3): In this case the Service Provider can publish a single TLSA RRset that matches the server certificate or public key digest. The same RRset works for all Customer Domains because name checks do not apply with DANE-EE(3) TLSA records (see Section 4.1). A Customer Domain can create a CNAME record pointing to the TLSA RRset published by the Service Provider.

Certificate Usage DANE-TA(2): When the Service Provider operates a private certification authority, the Service Provider is free to issue a certificate bearing any customer's domain name. Without DANE, such a certificate would not pass trust verification, but with DANE, the customer's TLSA RRset that is aliased to the provider's TLSA RRset can delegate authority to the provider's CA for the corresponding service. The Service Provider can generate appropriate certificates for each customer and use the SNI information provided by clients to select the right certificate chain to present to each client.

Below are example DNS records (assumed "secure" and shown without the associated DNSSEC information, such as record signatures) that illustrate both of the above models in the case of an HTTPS service whose clients all support DANE TLS. These examples work even with clients that don't "chase" CNAMEs when constructing the TLSA base domain (see Section 6 below).

```
; The hosted web service is redirected via a CNAME alias.
; The associated TLSA RRset is also redirected via a CNAME alias.
;
; A single certificate at the provider works for all Customer
; Domains due to the use of the DANE-EE(3) Certificate Usage.
;
www1.example.com.      IN CNAME w1.example.net.
_443._tcp.www1.example.com. IN CNAME _443._tcp.w1.example.net.
_443._tcp.w1.example.net. IN TLSA DANE-EE SPKI SHA2-256 (
                        8A9A70596E869BED72C69D97A8895DFA
                        D86F300A343FCEFF19E89C27C896BC9 )

;
; A CA at the provider can also issue certificates for each Customer
```

```

; Domain, and use the DANE-TA(2) Certificate Usage type to
; indicate a trust anchor.
;
www2.example.com.          IN CNAME w2.example.net.
_443._tcp.www2.example.com. IN CNAME _443._tcp.w2.example.net.
_443._tcp.w2.example.net.   IN TLSA DANE-TA Cert SHA2-256 (
                             C164B2C3F36D068D42A6138E446152F5
                             68615F28C69BD96A73E354CAC88ED00C )

```

With protocols that support explicit transport redirection via DNS MX records, SRV records, or other similar records, the TLSA base domain is based on the redirected transport end-point, rather than the origin domain. With SMTP, for example, when an email service is hosted by a Service Provider, the Customer Domain's MX hostnames will point at the Service Provider's SMTP hosts. When the Customer Domain's DNS zone is signed, the MX hostnames can be securely used as the base domains for TLSA records that are published and managed by the Service Provider. For example (without the required DNSSEC information, such as record signatures):

```

; Hosted SMTP service
;
example.com.          IN MX 0 mx1.example.net.
example.com.          IN MX 0 mx2.example.net.
_25._tcp.mx1.example.net. IN TLSA DANE-EE SPKI SHA2-256 (
                             8A9A70596E869BED72C69D97A8895DFA
                             D86F300A343FECEFF19E89C27C896BC9 )
_25._tcp.mx2.example.net. IN TLSA DANE-EE SPKI SHA2-256 (
                             C164B2C3F36D068D42A6138E446152F5
                             68615F28C69BD96A73E354CAC88ED00C )

```

If redirection to the Service Provider's domain (via MX or SRV records or any similar mechanism) is not possible, and aliasing of the TLSA record is not an option, then more complex coordination between the Customer Domain and Service Provider will be required. Either the Customer Domain periodically provides private keys and a corresponding certificate chain to the Provider (after making appropriate changes in its TLSA records), or the Service Provider periodically generates the keys and certificates and must wait for matching TLSA records to be published by its Customer Domains before deploying newly generated keys and certificate chains. In Section 6 below, we describe an approach that employs CNAME "chasing" to avoid the difficulties of coordinating key management across organization boundaries.

For further information about combining DANE and SRV, please see [I-D.ietf-dane-srv].

6. TLSA Base Domain and CNAMEs

When the application protocol does not support service location indirection via MX, SRV or similar DNS records, the service may be redirected via a CNAME. A CNAME is a more blunt instrument for this purpose, since unlike an MX or SRV record, it remaps the entire origin domain to the target domain for all protocols.

The complexity of coordinating key management is largely eliminated when DANE TLSA records are found in the Service Provider's domain, as discussed in Section 5. Therefore, DANE TLS clients connecting to a server whose domain name is a CNAME alias SHOULD follow the CNAME hop-by-hop to its ultimate target host (noting at each step whether the CNAME is DNSSEC-validated). If at each stage of CNAME expansion the DNSSEC validation status is "secure", the final target name SHOULD be the preferred base domain for TLSA lookups.

Implementations failing to find a TLSA record using a base name of the final target of a CNAME expansion SHOULD issue a TLSA query using the original destination name. That is, the preferred TLSA base domain should be derived from the fully expanded name, and failing that should be the initial domain name.

When the TLSA base domain is the result of "secure" CNAME expansion, the resulting domain name MUST be used as the HostName in SNI, and MUST be the primary reference identifier for peer certificate matching with certificate usages other than DANE-EE(3).

Protocol-specific TLSA specifications may provide additional guidance or restrictions when following CNAME expansions.

Though CNAMEs are illegal on the right hand side of most indirection records, such as MX and SRV records, they are supported by some implementations. For example, if the MX or SRV host is a CNAME alias, some implementations may "chase" the CNAME. If they do, they SHOULD use the target hostname as the preferred TLSA base domain as described above (and if the TLSA records are found there, use the CNAME expanded domain also in SNI and certificate name checks).

7. TLSA Publisher Requirements

This section updates [RFC6698] by specifying a requirement on the TLSA Publisher to ensure that each combination of Certificate Usage, selector and matching type that is present in the server's TLSA RRset MUST include at least one record that matches the server's present (rather than future or past) certificate chain. We describe a TLSA record update algorithm that ensures this requirement is met.

While a server is to be considered authenticated when its certificate chain is matched by any of the published TLSA records, not all clients support all combinations of TLSA record parameters. Some clients may not support some digest algorithms, others may either not support, or may exclusively support, the PKIX Certificate Usages. Some clients may prefer to negotiate [I-D.ietf-tls-oob-pubkey] raw public keys, which are only compatible with TLSA records whose Certificate Usage is DANE-EE(3) with selector SPKI(1).

A consequence of the above uncertainty as to which TLSA parameters are supported by any given client is that servers need to ensure that each and every parameter combination that appears in the TLSA RRset is, on its own, sufficient to match the server's current certificate chain. In particular, when deploying new keys or new parameter combinations some care is required to not generate parameter combinations that only match past or future certificate chains (or raw public keys). The rest of this section explains how to update the TLSA RRset in a manner that ensures the above requirement is met.

7.1. Rolling a Key Without Changing TLSA Parameters

The simplest case is key rollover while retaining the same set of published parameter combinations. In this case, TLSA records matching the existing server certificate chain (or raw public keys) are first augmented with corresponding records matching the future keys, at least two TTLs or longer before the the new chain is deployed. This allows the obsolete RRset to age out of client caches before the new chain is used in TLS handshakes. Once sufficient time has elapsed and all clients performing DNS lookups are retrieving the updated TLSA records, the server administrator may deploy the new certificate chain, verify that it works, and then remove any obsolete records matching the no longer active chain:

```
; The initial TLSA RRset
;
_443._tcp.www.example.org. IN TLSA 3 1 1 01d09d19c2139a46...

; The transitional TLSA RRset published at least 2*TTL seconds
; before the actual key change
;
_443._tcp.www.example.org. IN TLSA 3 1 1 01d09d19c2139a46...
_443._tcp.www.example.org. IN TLSA 3 1 1 7aa7a5359173d05b...

; The final TLSA RRset after the key change
;
_443._tcp.www.example.org. IN TLSA 3 1 1 7aa7a5359173d05b...
```


The next case to consider is adding or switching to a new combination of TLSA parameters. In this case publish the new parameter combinations for the server's existing certificate chain first, and only then deploy new keys if desired:

```
; Initial TLSA RRset
;
_443._tcp.www.example.org. IN TLSA 1 1 1 01d09d19c2139a46...

; New TLSA RRset, same key re-published as DANE-EE(3)
;
_443._tcp.www.example.org. IN TLSA 3 1 1 01d09d19c2139a46...
```

7.2. Switching to DANE-CA from DANE-EE

A more complex involves switching to a trust-anchor or PKIX usage from a chain that is either self-signed, or issued by a private CA and thus not compatible with PKIX. Here the process is to first add TLSA records matching the future chain that is issued by the desired future CA (private or PKIX), but initially with the same parameters as the legacy chain. Then, after deploying the new keys, switch to the new TLSA parameter combination.

```
; The initial TLSA RRset
;
_443._tcp.www.example.org. IN TLSA 3 1 1 01d09d19c2139a46...

; A transitional TLSA RRset, published at least 2*TTL before the
; actual key change. The new keys are issued by a DANE-TA(2) CA,
; but for now specified via a DANE-EE(3) association.
;
_443._tcp.www.example.org. IN TLSA 3 1 1 01d09d19c2139a46...
_443._tcp.www.example.org. IN TLSA 3 1 1 7aa7a5359173d05b...

; The final TLSA RRset after the key change. Now that the old
; self-signed EE keys are not an impediment, specify the issuing
; TA of the new keys.
;
_443._tcp.www.example.org. IN TLSA 2 0 1 c57bce38455d9e3d...
```

7.3. Switching to New TLSA Parameters

When employing a new digest algorithm in the TLSA RRset, for compatibility with digest agility specified in Section 8 below, administrators should publish the new digest algorithm with each combinations of Certificate Usage and selector for each associated key or chain used with any other digest algorithm. When removing an algorithm, remove it entirely. Each digest algorithm employed should match the same set of chains (or raw public keys).

```
; The initial TLSA RRset with EE SHA2-256 associations for two keys.
;
_443._tcp.www.example.org. IN TLSA 3 1 1 01d09d19c2139a46...
_443._tcp.www.example.org. IN TLSA 3 1 1 7aa7a5359173d05b...

; The new TLSA RRset also with SHA2-512 associations for each key
;
_443._tcp.www.example.org. IN TLSA 3 1 1 01d09d19c2139a46...
_443._tcp.www.example.org. IN TLSA 3 1 2 d9947c35089310bc...
_443._tcp.www.example.org. IN TLSA 3 1 1 7aa7a5359173d05b...
_443._tcp.www.example.org. IN TLSA 3 1 2 89a7486a4b6ae714...
```

7.4. TLSA Publisher Requirements Summary

In summary, server operators updating TLSA records should make one change at a time. Either pre-publish new keys with existing TLSA parameters, remove records matching stale keys, or add new TLSA parameters for all current keys. Ensure that at all times, each combination of parameter values matches the same set of underlying objects (trust anchors, leaf certificates or raw public keys). Another way of saying the same thing is that no combination of Certificate Usage, selector and matching type in a server's TLSA RRset should ever match only some combination of future or past keys. Such combinations of parameters should be removed before corresponding keys are retired, or added only after new keys become active.

8. Digest Algorithm Agility

While [RFC6698] specifies multiple digest algorithms, it does not specify a protocol by which the TLS 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 "BetterAlg" stronger than digest algorithm "WorseAlg". Suppose further that a server's TLSA RRset contains some

records with "BetterAlg" as the digest algorithm. Suppose also that the server adheres to the requirements of Section 7 and ensures that each combination of TLSA parameters contains at least one record that matches the server's current certificate chain (or raw public keys). Under the above assumptions the client can safely ignore TLSA records with the weaker algorithm "WorseAlg", because it suffices to only check the records with the stronger algorithm "BetterAlg".

To make digest algorithm agility possible, all published TLSA RRsets for use with DANE TLS MUST conform to the requirements of Section 7. With servers publishing compliant TLSA RRsets, TLS clients can, for each combination of usage and selector, ignore all digest records except those that employ their notion of the strongest digest algorithm. (The server should only publish algorithms it deems acceptable at all.) The ordering of digest algorithms by strength is not specified in advance; it is entirely up to the TLS client. TLS client implementations SHOULD make the digest algorithm preference ordering a configurable option.

Note, TLSA records with a matching type of Full(0) that publish an entire 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.

TLS clients SHOULD use digest algorithm agility when processing the DANE TLSA records of an TLS 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 considered by the client to be the strongest among usable records with the given usage and selector.

9. General DANE Guidelines

These guidelines provide guidance for using or designing protocols for DANE.

9.1. DANE DNS Record Size Guidelines

Selecting a combination of TLSA parameters to use requires careful thought. One important consideration to take into account is the size of the resulting TLSA record after its parameters are selected.

9.1.1. UDP and TCP Considerations

Deployments SHOULD avoid TLSA record sizes that cause UDP fragmentation.

Although DNS over TCP would provide the ability to more easily transfer larger DNS records between clients and servers, it is not universally deployed and is still prohibited by some firewalls. Clients that request DNS records via UDP typically only use TCP upon receipt of a truncated response in the DNS response message sent over UDP. Setting the TC bit alone will be insufficient if the response containing the TC bit is itself fragmented.

9.1.2. Packet Size Considerations for TLSA Parameters

Server operators SHOULD NOT publish TLSA records using both a TLSA Selector of Cert(0) and a TLSA Matching Type of Full(0), as even a single certificate is generally too large to be reliably delivered via DNS over UDP. Furthermore, two TLSA records containing full certificates will need to be published simultaneously during a certificate rollover, as discussed in Section 7.1.

While TLSA records using a TLSA Selector of SPKI(1) and a TLSA Matching Type of Full(0) (which publish the bare public keys without the overhead of a containing X.509 certificate) are generally more compact, these too should be used with caution as they are still larger than necessary. Rather, servers SHOULD publish digest-based TLSA Matching Types in their TLSA records. The complete corresponding certificate should, instead, be transmitted to the client in-band during the TLS handshake, which can be easily verified using the digest value.

In summary, the use of a TLSA Matching Type of Full(0) is NOT RECOMMENDED and the use of a digest-based matching type, such as SHA2-256(1) SHOULD be used.

9.2. Certificate Name Check Conventions

Certificates presented by a TLS server will generally contain a subjectAltName (SAN) extension or a Common Name (CN) element within the subject distinguished name (DN). The TLS server's DNS domain name is normally published within these elements, ideally within the subjectAltName extension. (The use of the CN field for this purpose is deprecated.)

When a server hosts multiple domains at the same transport endpoint, the server's ability to respond with the right certificate chain is predicated on correct SNI information from the client. DANE clients MUST send the SNI extension with a HostName value of the base domain of the TLSA RRset.

Except with TLSA Certificate Usage DANE-EE(3), where name checks are not applicable (see Section 4.1), DANE clients MUST verify that the client has reached the correct server by checking that the server name is listed in the server certificate's SAN or CN. The server name used for this comparison SHOULD be the base domain of the TLSA RRset. Additional acceptable names may be specified by protocol-specific DANE standards. For example, with SMTP both the destination domain name and the MX host name are acceptable names to be found in the server certificate (see [I-D.ietf-dane-smtp-with-dane]).

It is the responsibility of the service operator, in coordination with the TLSA Publisher, to ensure that at least one of the TLSA records published for the service will match the server's certificate chain (either the default chain or the certificate that was selected based on the SNI information provided by the client).

Given the DNSSEC validated DNS records below:

```
example.com.           IN MX 0 mail.example.com.
mail.example.com.      IN A 192.0.2.1
_25._tcp.mail.example.com. IN TLSA DANE-TA Cert SHA2-256 (
                        E8B54E0B4BAA815B06D3462D65FBC7C0
                        CF556ECCF9F5303EBFBB77D022F834C0 )
```

The TLSA base domain is "mail.example.com" and is required to be the HostName in the client's SNI extension. The server certificate chain is required to be signed by a trust anchor with the above certificate SHA2-256 digest. Finally, one of the DNS names in the server certificate is required to be either "mail.example.com" or "example.com" (this additional name is a concession to compatibility with prior practice, see [I-D.ietf-dane-smtp-with-dane] for details).

The semantics of wildcards in server certificates are left to individual application protocol specifications.

9.3. Design Considerations for Protocols Using DANE

When a TLS client goes to the trouble of authenticating a certificate chain presented by a TLS server, it will typically not continue to use that server in the event of authentication failure, or else authentication serves no purpose. Some clients may, at times, operate in an "audit" mode, where authentication failure is reported to the user or in logs as a potential problem, but the connection proceeds despite the failure. Nevertheless servers publishing TLSA records MUST be configured to allow correctly configured clients to successfully authenticate their TLS certificate chains.

A service with DNSSEC-validated TLSA records implicitly promises TLS support. When all the TLSA records for a service are found "unusable", due to unsupported parameter combinations or malformed associated data, DANE clients cannot authenticate the service certificate chain. When authenticated TLS is dictated by the application, the client SHOULD NOT connect to the associated server. If, on the other hand, the use of TLS is "opportunistic", then the client SHOULD generally use the server via an unauthenticated TLS connection, but if TLS encryption cannot be established, the client MUST NOT use the server. Standards for DANE specific to the particular application protocol may modify the above requirements, as appropriate, to specify whether the connection should be established anyway without relying on TLS security, with only encryption but not authentication, or whether to refuse to connect entirely. Application protocols need to specify when to prioritize security over the ability to connect under adverse conditions.

9.3.1. Design Considerations for non-PKIX Protocols

For some application protocols (such as SMTP to MX with opportunistic TLS), the existing public CA PKI is not a viable alternative to DANE. For these (non-PKIX) protocols, new DANE standards SHOULD NOT suggest publishing TLSA records with TLSA Certificate Usage PKIX-TA(0) or PKIX-EE(1), as TLS clients cannot be expected to perform [RFC5280] PKIX validation or [RFC6125] identity verification.

Protocols designed for non-PKIX use SHOULD choose to treat any TLSA records with TLSA Certificate Usage PKIX-TA(0) or PKIX-EE(1) as unusable. After verifying that the only available TLSA Certificate Usage types are PKIX-TA(0) or PKIX-EE(1), protocol specifications MAY instruct clients to either refuse to initiate a connection or to connect via unauthenticated TLS if no alternative authentication mechanisms are available.

10. Interaction with Certificate Transparency

Certificate Transparency (CT) [RFC6962] defines an experimental approach to mitigate the risk of rogue or compromised public CAs issuing unauthorized certificates. This section clarifies the interaction of CT and DANE. CT is an experimental protocol and auditing system that applies only to public CAs, and only when they are free to issue unauthorized certificates for a domain. If the CA is not a public CA, or a DANE-EE(3) TLSA RR directly specifies the end entity certificate, there is no role for CT, and clients need not apply CT checks.

When a server is authenticated via a DANE TLSA RR with TLSA Certificate Usage DANE-EE(3), the domain owner has directly specified

the certificate associated with the given service without reference to any PKIX certification authority. Therefore, when a TLS client authenticates the TLS server via a TLSA certificate association with usage DANE-EE(3), CT checks SHOULD NOT be performed. Publication of the server certificate or public key (digest) in a TLSA record in a DNSSEC signed zone by the domain owner assures the TLS client that the certificate is not an unauthorized certificate issued by a rogue CA without the domain owner's consent.

When a server is authenticated via a DANE TLSA RR with TLSA usage DANE-TA(2) and the server certificate does not chain to a known public root CA, CT cannot apply (CT logs only accept chains that start with a known, public root). Since TLSA Certificate Usage DANE-TA(2) is generally intended to support non-PKIX trust anchors, TLS clients SHOULD NOT perform CT checks with usage DANE-TA(2) using unknown root CAs.

A server operator who wants clients to perform CT checks should publish TLSA RRs with usage PKIX-TA(0) or PKIX-EE(1).

11. Note on DNSSEC Security

Clearly the security of the DANE TLSA PKI rests on the security of the underlying DNSSEC infrastructure. While this memo is not a guide to DNSSEC security, a few comments may be helpful to TLSA implementers.

With the existing public CA PKI, name constraints are rarely used, and a public root CA can issue certificates for any domain of its choice. With DNSSEC, under the Registry/Registrar/Registrant model, the situation is different: only the registrar of record can update a domain's DS record in the registry parent zone (in some cases, however, the registry is the sole registrar). With many gTLDs, for which multiple registrars compete to provide domains in a single registry, it is important to make sure that rogue registrars cannot easily initiate an unauthorized domain transfer, and thus take over DNSSEC for the domain. DNS Operators SHOULD use a registrar lock of their domains to offer some protection against this possibility.

When the registrar is also the DNS operator for the domain, one needs to consider whether the registrar will allow orderly migration of the domain to another registrar or DNS operator in a way that will maintain DNSSEC integrity. TLSA Publishers SHOULD ensure their registrar publishes a suitable domain transfer policy.

DNSSEC signed RRsets cannot be securely revoked before they expire. Operators should plan accordingly and not generate signatures with excessively long duration periods. For domains publishing high-value

keys, a signature lifetime of a few days is reasonable, and the zone should be resigned daily. For domains with less critical data, a reasonable signature lifetime is a couple of weeks to a month, and the zone should be resigned weekly. Monitoring of the signature lifetime is important. If the zone is not resigned in a timely manner, one risks a major outage and the entire domain will become bogus.

12. Summary of Updates to RFC6698

Authors note: is this section needed? Or is it sufficiently clear above that we don't need to restate things here?

- o In Section 3 we update [RFC6698] to specify a requirement for clients to support at least TLS 1.0, and to support SNI.
- o In Section 4.1 we update [RFC6698] to specify peer identity matching and certificate validity interval based solely on the basis of the TLSA RRset. We also specify DANE authentication of raw public keys [I-D.ietf-tls-oob-pubkey] via TLSA records with Certificate Usage DANE-EE(3) and selector SPKI(1).
- o In Section 4.2 we update [RFC6698] to require that servers publishing digest TLSA records with a usage of DANE-TA(2) MUST include the trust-anchor certificate in their TLS server certificate message. This extends to the case of "2 1 0" TLSA records which publish a full public key.
- o In Section 4.3 and Section 4.4, we explain that PKIX-EE(1) and PKIX-TA(0) are generally NOT RECOMMENDED. With usage PKIX-TA(0) we note that clients may need to process extended trust chains beyond the first trusted issuer, when that issuer is not self-signed.
- o In Section 6, we recommend that DANE application protocols specify that when possible securely CNAME expanded names be used to derive the TLSA base domain.
- o In Section 7, we specify a strategy for managing TLSA records that interoperates with DANE clients regardless of what subset of the possible TLSA record types (combinations of TLSA parameters) is supported by the client.
- o In Section 8, we propose a digest algorithm agility protocol. [Note: This section does not yet represent the rough consensus of the DANE working group and requires further discussion. Perhaps this belongs in a separate document.]

- o In Section 9.1 we recommend against the use of Full(0) TLSA records, as digest records are generally much more compact.

13. Security Considerations

Application protocols that cannot make use of the existing public CA PKI (so called non-PKIX protocols), may choose not to implement certain PKIX-dependent TLSA record types defined in [RFC6698]. If such records are published despite not being supported by the application protocol, they are treated as "unusable". When TLS is opportunistic, the client may proceed to use the server with mandatory unauthenticated TLS. This is stronger than opportunistic TLS without DANE, since in that case the client may also proceed with a plaintext connection. When TLS is not opportunistic, the client MUST NOT connect to the server.

Therefore, when TLSA records are used with protocols where PKIX does not apply, the recommended policy is for servers to not publish PKIX-dependent TLSA records, and for opportunistic TLS clients to use them to enforce the use of (albeit unauthenticated) TLS, but otherwise treat them as unusable. Of course, when PKIX validation is supported by the application protocol, clients SHOULD perform PKIX validation per [RFC6698].

14. IANA Considerations

This specification requires no support from IANA.

15. Acknowledgements

The authors would like to 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 Samuel Dukhovni for editorial assistance.

16. References

16.1. Normative References

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

- [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.
- [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.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 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.

16.2. Informative References

- [I-D.ietf-dane-registry-acronyms]
Gudmundsson, O., "Adding acronyms to simplify DANE conversations", draft-ietf-dane-registry-acronyms-03 (work in progress), January 2014.
- [I-D.ietf-dane-smtp-with-dane]
Dukhovni, V. and W. Hardaker, "SMTP security via opportunistic DANE TLS", draft-ietf-dane-smtp-with-dane-06 (work in progress), February 2014.

[I-D.ietf-dane-srv]

Finch, T., Miller, M., and P. Saint-Andre, "Using DNS-Based Authentication of Named Entities (DANE) TLSA records with SRV and MX records.", draft-ietf-dane-srv-05 (work in progress), February 2014.

[I-D.ietf-tls-oob-pubkey]

Wouters, P., Tschofenig, H., Gilmore, J., Weiler, S., and T. Kivinen, "Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", draft-ietf-tls-oob-pubkey-11 (work in progress), January 2014.

[RFC6962] Laurie, B., Langley, A., and E. Kasper, "Certificate Transparency", RFC 6962, June 2013.

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

INTERNET-DRAFT
DANE Working Group
Intended status: Proposed Standard
Expires: December 31, 2014
Updates: 6698 (if approved)

J. Gilmore
Electronic Frontier Foundation
July 3, 2014

Authenticating Raw Public Keys with DANE TLSA
draft-ietf-dane-rawkeys-00

Abstract

This document standardizes how the Domain Name System can authenticate Raw Public Keys. Transport Level Security now has the option to use Raw Public Keys, but they require some form of external authentication. The document updates RFC 6698 to allow the Domain Name System to standardize the authentication of more types of keying material.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

1 Background and Introduction

The Internet uses many kinds of encryption, and many kinds of keying material. These keys are authenticated in an attempt to prove that the public keys used in communication are the correct keys needed to interact with a particular server or client across the Internet.

The Domain Name System (DNS) [RFC1034, RFC1035] provides a globally distributed database for brief information about names used in the Internet. The DNS Security Extensions (DNSSEC) [RFC4033, RFC4034, RFC4035] provide authentication for this database, proving whether the information in the DNS was truly published by the owner of the associated domain name.

Transport Level Security (TLS) [RFC5246] and Datagram TLS (DTLS) [RFC6347] define a protocol that protects an Internet datastream or a series of datagrams from eavesdropping and modification. They initially used certificates in PKIX [RFC 5280] formats to store their keying material, and authenticated them via a series of trust anchors embedded in client applications.

Domain name system Authentication of Named Entities (DANE) provides a way to store application level public keys in the DNS and authenticate them using DNSSEC. The DANE TLS Authentication (TLSA) resource record [RFC6698] initially provided authentication for the PKIX certificates used in TLS and DTLS.

1.1 Summary of Changes

This document extends TLSA records to be able to authenticate more kinds of keying material than PKIX certificates. Protocols can then use their keying material with DANE by standardizing new forms of TLSA records.

As a first example of such a new form, this document extends DANE to provide authentication for Raw Public Keys. Raw Public Keys are used in place of PKIX certificates in an extension to TLS and DTLS [RFC7250]. Client applications using Raw Public Keys with TLS or DTLS can use DNSSEC to prove whether those public keys were truly published by the owner of the domain name whose server they are

accessing.

1.2 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 RFC 2119 [RFC2119].

2 Extending TLSA records to support non-PKIX keying material

This document relaxes the restriction that TLSA records can only authenticate PKIX certificates (RFC 6698, section 1.3). The DANE protocol and TLSA records can now apply to encryption keying material in general. This protocol and record type continue to apply to PKIX [RFC5280] certificates, but new standards are free to define non-PKIX keying material formats.

Wherever the term "certificate" is used in RFC 6698 to refer to fields in the TLSA record, this document extends it to refer more generally to "keying material". Thus the "certificate usage" field can be thought of as a "keying material usage" field, the "certificate association data" can now be used as "keying material association data", etc.

In addition, this document relaxes the requirement that certificate usage value 3 can only be used for PKIX format certificates (RFC 6698, section 2.1.1). Certificate usage 3 can now be used with any standardized keying material format. Certificate usages 0, 1 and 2 remain restricted to apply only to PKIX-formatted certificates in DER encoding [X.690].

3 Supporting Raw Public Keys in TLSA records

This document extends the DANE TLSA record definition to allow TLSA records to describe raw public keys as well as PKIX certificates. This extension does not define any new field values; it merely defines how existing fields are processed when being used with raw public keys, such as those provided by TLS and DTLS servers.

There are two different ways to use raw public keys in TLSA records. One is to store the public key itself, so that it can be accessed directly from the DNS. The second is to store a hash of the public key, so that a public key obtained in some other way can be authenticated via the DNS. These cases are distinguished by the matching type field in the TLSA record.

When a raw public key is to be stored in a TLSA record, the record MUST specify a certificate usage / keying material usage of 3

(domain-provided), a selector of 1 (SubjectPublicKeyInfo), and a matching type of 0. The SubjectPublicKeyInfo structure that holds the public key is placed in the certificate association / keying material association data field.

This SubjectPublicKeyInfo structure MUST be encoded in DER encoding [X.660] of Abstract Syntax Notation One (ASN.1) [X.208]. It is identical to the SubjectPublicKeyInfo structure that is described in RFC 3279 [RFC3279], which is used as a component of PKIX certificates. It is identical to the SubjectPublicKeyInfo structure that is used in TLSA records that use a selector value of 1 and a matching type of 0 to match PKIX certificates. It contains an algorithm identifier, any optional parameters needed with that algorithm identifier, and the public key itself.

When a raw public key (that was obtained in some other way, such as in a TLS or DTLS transaction) is to be merely matched by a TLSA record, matching type 0 MAY be used, or matching types other than 0 MAY also be used, by placing the hash value of the SubjectPublicKeyInfo structure into the certificate association / keying material association data.

This document extends the meaning of the certificate usage / keying material usage value of 3 (from RFC 6699 section 2.1.1) by defining how the TLSA record is used by a client communicating with a TLS or DTLS server that uses raw public keys. This extension adds to, rather than replacing, the definition of certificate usage 3 with TLS or DTLS servers that use PKIX certificates.

3 -- Keying material usage 3 is also used to specify a raw public key that MUST match the raw public key presented by the server in TLS or DTLS. When the server provides a raw public key, there is no PKIX certificate and no PKIX validation is done. The server's raw public key MUST match the raw public key provided in the TLSA record. This keying material usage is sometimes referred to as "domain-issued" because it allows a domain administrator to directly certify a domain's public keys.

4 Security Considerations

The encoding used in the TLSA resource record for Raw Public Keys is identical to the encoding used to match the public key of a PKIX certificate. This allows a single TLSA record to match both a PKIX certificate used in traditional TLS or DTLS, and to also match a Raw Public Key provided in extended TLS or DTLS. This offers TLS or DTLS servers an easy way to interoperate with both traditional and extended clients. They can use the same public and private key when communicating with either extended or traditional clients.

Since TLSA records use a protocol type and port number as a prefix on the domain name, services that use Raw Public Keys on various ports accessed through the same domain name are free to use different keying material. Using diverse keying material for different services can improve the robustness of the services after a key compromise. For example, email service on port 25 can continue with full security, even after the private key protecting HTTPS service on port 443 has been compromised. This is a tighter binding between public keys and services than that provided by PKIX certificates, which do not distinguish port numbers. When PKIX certificates are authenticated with TLSA usages 0, 1, or 2, a PKIX certificate that was originally used with HTTPS could be used for a man-in-the-middle attack on email service as well, after its corresponding private key has been compromised. This cross-port attack does not work when the domain name uses TLSA usage 3 to authenticate different Raw Public Keys (or PKIX certificates) for the different services on different ports.

In the TLS and DTLS protocol, certificate types are often negotiated before the relevant TLSA records are available to the client. Server operators who anticipate using TLSA records to authenticate the server should always ensure that if their server offers support for Raw Public Keys, then their server's domain name(s) SHOULD contain TLSA records that match the public key that the server offers. Failure to publish such TLSA records would otherwise lead to an authentication failure in clients that opt to use Raw Public Keys, even if TLSA records exist that authenticate PKIX certificates with usages 0, 1, or 2. This is not an issue when Raw Public Keys are used with out-of-band non-DANE authentication.

When using Raw Public Keys and TLSA records, the security of the domain name system records directly affects the security of the communications protected by TLS or DTLS. If the domain's DNS records are compromised, or the DNS records that delegate name service to this domain are compromised, communications can be blocked, redirected, intercepted, or modified. The DANE TLSA Security Considerations section [RFC6698] provides further details.

5 IANA Considerations

In the IANA "TLSA Certificate Usages" registry created by Section 7.2 of RFC 6698, the value "3" ("Domain-issued certificate") should have its short description changed to "Domain-issued keying material", and should have this document added as a reference document.

6 References

6.1 Normative References

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.
- [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.
- [RFC3279] Polk, T., Housley, R., Bassham, L., "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3279, April 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.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.
- [RFC6698] Hoffman, P., Schlyter, J., "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.
- [RFC7250] Wouters, P., Tschofenig, H., Gilmore, J., Weiler, S., Kivinen, T., "Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", RFC 7250, May 2014
- [X.208] CCITT Recommendation X.208: Specification of Abstract Syntax Notation One (ASN.1), 1988.
- [X.690] "Recommendation ITU-T X.690 (2002) | ISO/IEC 8825-1:2002, Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules

(DER)", July 2002.

6.2 Informative References

- [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.

Authors' Addresses

John Gilmore
Electronic Frontier Foundation
815 Eddy Street
San Francisco, CA 94117
United States

EMail: gnu@ietf.toad.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 18, 2014

P. Hoffman
VPN Consortium
J. Schlyter
Kirei AB
February 14, 2014

Using Secure DNS to Associate Certificates with Domain Names For S/MIME
draft-ietf-dane-smime-06

Abstract

This document describes how to use secure DNS to associate an S/MIME user's certificate with the intended domain name, similar to the way that DANE (RFC 6698) does for 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 August 18, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	3
2. The SMIMEA Resource Record	3
3. Domain Names for S/MIME Certificate Associations	4
4. Mandatory-to-Implement Features	5
5. IANA Considerations	5
5.1. SMIMEA RRtype	5
6. Security Considerations	5
7. Acknowledgements	5
8. References	6
8.1. Normative References	6
8.2. Informative References	6
Authors' Addresses	7

1. Introduction

S/MIME [RFC5751] messages often contain a certificate (some messages contain more than one certificate). These certificates assist in authenticating the sender of the message and can be used for encrypting messages that will be sent in reply. In order for the S/MIME receiver to authenticate that a message is from the sender who is identified in the message, the receiver's mail user agent (MUA) must validate that this certificate is associated with the purported sender. Currently, the MUA must trust a trust anchor upon which the sender's certificate is rooted, and must successfully validate the certificate. There are other requirements on the MUA, such as associating the identity in the certificate with that of the message, that are out of scope for this document.

Some people want to authenticate the association of the sender's certificate with the sender without trusting a configured trust anchor. Given that the DNS administrator for a domain name is authorized to give identifying information about the zone, it makes sense to allow that administrator to also make an authoritative binding between email messages purporting to come from the domain name and a certificate that might be used by someone authorized to send mail from those servers. The easiest way to do this is to use the DNS.

This document describes a mechanism for associating a user's certificate with the domain that is similar to that described in DANE itself [RFC6698]. Most of the operational and security considerations for using the mechanism in this document are described in RFC 6698, and are not described here at all. Only the major differences between this mechanism and those used in RFC 6698 are described here. Thus, the reader must be familiar with RFC 6698 before reading this document.

NOTE FOR FUTURE DRAFTS OF THIS DOCUMENT: The DANE WG needs to have a serious discussion about what the DANE set of specifications covering TLS for HTTP, TLS for SMTP, S/MIME, OpenPGP, and so on are meant for. They could be used for acquisition of key association material, for discovering services that use the keying material, for having assurance that a service that uses the keying material should be available, or some combination of these.

1.1. 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 RFC 2119 [RFC2119].

This document also makes use of standard PKIX, DNSSEC, and S/MIME terminology. See PKIX [RFC5280], DNSSEC [RFC4033], [RFC4034], [RFC4035], and SMIME [RFC5751] for these terms.

2. The SMIMEA Resource Record

The SMIMEA DNS resource record (RR) is used to associate an end entity certificate or public key with the associated email address, thus forming a "SMIMEA certificate association". The semantics of how the SMIMEA RR is interpreted are given later in this document. Note that the information returned in the SMIMEA record might be for the end entity certificate, or it might be for the trust anchor or an intermediate certificate.

The type value for the SMIMEA RRtype is defined in Section 5.1. The SMIMEA resource record is class independent. The SMIMEA resource record has no special TTL requirements.

The SMIMEA wire format and presentation format are the same as for the TLSA record as described in section 2.1 of RFC 6698. The certificate usage field, the selector field, and the matching type field have the same format; the semantics are also the same except where RFC 6698 talks about TLS at the target protocol for the certificate information.

3. Domain Names for S/MIME Certificate Associations

Domain names are prepared for requests in the following manner.

1. The user name (the "left-hand side" of the email address, called the "local-part" in the mail message format definition [RFC2822] and the "local part" in the specification for internationalized email [RFC6530]), is hashed using the SHA2-224 [RFC5754] algorithm (with the hash being represented in its hexadecimal representation, to become the left-most label in the prepared domain name. This does not include the "@" character that separates the left and right sides of the email address. The string that is used for the local part is a Unicode string encoded in UTF-8.
2. The string "_smimecert" becomes the second left-most label in the prepared domain name.
3. The domain name (the "right-hand side" of the email address, called the "domain" in RFC 2822) is appended to the result of step 2 to complete the prepared domain name.

For example, to request a SMIMEA resource record for a user whose address is "chris@example.com", calculate the SHA-224 of "chris", which is 0x3f51f4663b2b798560c5b9e16d6069a28727f62518c3a1b33f7f5214. The request is thus:

```
3f51f4663b2b798560c5b9e16d6069a28727f62518c3a1b33f7f5214._smimecert.example.com
```

The corresponding resource record in the example.com zone might look like:

```
3f51f4663b2b798560c5b9e16d6069a28727f62518c3a1b33f7f5214._smimecert.example.com.  
IN SMIMEA (  
0 0 1 d2abde240d7cd3ee6b4b28c54df034b9  
7983ald16e8a410e4561cb106618e971 )
```

Design note: Hashing the user name with SHA-224 and using the hexadecimal encoding of that hash allows local parts that have characters that would prevent their use in domain names in typical applications. Even though the DNS protocol itself can use any octet value in a label, most applications that use DNS names are limited to a much smaller set of allowed characters. For example, a period (".") is a valid character in a local part, but would wreak havoc in a domain name unless the application using the name somehow quoted it. Similarly, RFC 6530 allows non-ASCII characters in local parts, and encoding a local part with non-ASCII characters as the hex of the SHA-224 renders the name usable in applications that use the DNS.

Wildcards can be more useful for SMIMEA than they are for TLSA. If a site publishes a trust anchor certificate for all users on the site (certificate usage 0 or 2), it could make sense to use a wildcard resource record such as `*._smimecert.example.com`.

4. Mandatory-to-Implement Features

S/MIME MUAs conforming to this specification MUST be able to correctly interpret SMIMEA records with certificate usages 0, 1, 2, and 3. S/MIME MUAs conforming to this specification MUST be able to compare a certificate association with a certificate offered by another S/MIME MUA using selector types 0 and 1, and matching type 0 (no hash used) and matching type 1 (SHA-256), and SHOULD be able to make such comparisons with matching type 2 (SHA-512).

5. IANA Considerations

5.1. SMIMEA RRtype

This document uses a new DNS RRtype, SMIMEA, whose value will be allocated by IANA from the Resource Record (RR) TYPEs subregistry of the Domain Name System (DNS) Parameters registry.

TODO: there needs to be new registries for certificate usages, selectors, and matching types, pre-populated with the values from TLSA.

6. Security Considerations

DNS zones that are signed with DNSSEC using NSEC for denial of existence are susceptible to zone-walking, a mechanism that allow someone to enumerate all the names in the zone. Someone who wanted to collect email addresses from a zone that uses SMIMEA might use such a mechanism. DNSSEC-signed zones using NSEC3 for denial of existence are significantly less susceptible to zone-walking. Someone could still attempt a dictionary attack on the zone to find SMIMEA records, just as they can use dictionary attacks on an SMTP server to see which addresses are valid.

Client treatment of any information included in the trust anchor is a matter of local policy. This specification does not mandate that such information be inspected or validated by the domain name administrator.

7. Acknowledgements

Brian Dickson, Miek Gieben, and Martin Pels contributed technical ideas and support to this document.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [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.
- [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.
- [RFC5751] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, January 2010.
- [RFC5754] Turner, S., "Using SHA2 Algorithms with Cryptographic Message Syntax", RFC 5754, January 2010.
- [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.

8.2. Informative References

- [RFC2822] Resnick, P., "Internet Message Format", RFC 2822, April 2001.
- [RFC6530] Klensin, J. and Y. Ko, "Overview and Framework for Internationalized Email", RFC 6530, February 2012.

Authors' Addresses

Paul Hoffman
VPN Consortium

Email: paul.hoffman@vpnc.org

Jakob Schlyter
Kirei AB

Email: jakob@kirei.se

DANE
Internet-Draft
Intended status: Standards Track
Expires: November 26, 2014

V. Dukhovni
Two Sigma
W. Hardaker
Parsons
May 25, 2014

SMTP security via opportunistic DANE TLS
draft-ietf-dane-smtp-with-dane-10

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 November 26, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Terminology	3
1.2. Background	5
1.3. SMTP channel security	6
1.3.1. STARTTLS downgrade attack	6
1.3.2. Insecure server name without DNSSEC	7
1.3.3. Sender policy does not scale	7
1.3.4. Too many certification authorities	8
2. Identifying applicable TLSA records	8
2.1. DNS considerations	8
2.1.1. DNS errors, bogus and indeterminate responses	8
2.1.2. DNS error handling	11
2.1.3. Stub resolver considerations	11
2.2. TLS discovery	12
2.2.1. MX resolution	13
2.2.2. Non-MX destinations	15
2.2.3. TLSA record lookup	17
3. DANE authentication	19
3.1. TLSA certificate usages	19
3.1.1. Certificate usage DANE-EE(3)	20
3.1.2. Certificate usage DANE-TA(2)	21
3.1.3. Certificate usages PKIX-TA(0) and PKIX-EE(1)	22
3.2. Certificate matching	23
3.2.1. DANE-EE(3) name checks	23
3.2.2. DANE-TA(2) name checks	23
3.2.3. Reference identifier matching	24
4. Server key management	25
5. Digest algorithm agility	26
6. Mandatory TLS Security	27
7. Note on DANE for Message User Agents	28
8. Interoperability considerations	29
8.1. SNI support	29
8.2. Anonymous TLS cipher suites	29
9. Operational Considerations	30
9.1. Client Operational Considerations	30
9.2. Publisher Operational Considerations	30
10. Security Considerations	31
11. IANA considerations	31
12. Acknowledgements	31
13. References	32
13.1. Normative References	32
13.2. Informative References	33
Authors' Addresses	33

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 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. It 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 legacy 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.

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 and Section 3 which describe respectively how to locate and use DANE TLSA records with SMTP. In Section 6, we discuss application of DANE TLS to destinations for which channel integrity and confidentiality are mandatory. In Section 7 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 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 signaled 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. Identifying applicable TLSA records

2.1. DNS considerations

2.1.1. 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 cannot 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.

2.1.2. DNS error handling

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 securely identified as the delivery destination, a set of DNS lookups (Section 2.2) MUST be performed to locate 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.

2.1.3. Stub resolver considerations

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 a stub resolver returns a response containing a CNAME alias that does not also contain the corresponding query results for the target

of the alias, the SMTP client will need to repeat 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 CNAME expansion an error is detected, the lookup of the original requested records MUST be considered to have failed.

Whether a chain of CNAME records was returned in a single stub resolver response or via explicit recursion by the SMTP client, if at any stage of recursive expansion an "insecure" CNAME record is encountered, then it and all subsequent results (in particular, the final result) MUST be considered "insecure" regardless of whether any earlier CNAME records leading to the "insecure" record were "secure".

Note, a security-aware non-validating stub resolver may return to the SMTP client an "insecure" reply received from a validating recursive resolver that contains a CNAME record along with additional answers recursively obtained starting at the target of the CNAME. In this all that one can say is that some record in the set of records returned is "insecure", but it is possible that the initial CNAME record and a subset of the subsequent records are "secure".

If the SMTP client needs to determine the security status of the DNS zone containing the initial CNAME record, it may need to issue an a separate query of type "CNAME" that returns only the initial CNAME record. In particular in Section 2.2.2 when insecure A or AAAA records are found for an SMTP server via a CNAME alias, it may be necessary to perform an additional CNAME query to determine whether the DNS zone in which the alias is published is signed.

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 2.1.1 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.

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. DANE TLS can authenticate message

delivery to the intended next-hop domain only when the MX records are obtained securely via a DNSSEC validated 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 2.1.1), DANE TLS need not apply, and delivery MAY proceed via pre-DANE opportunistic TLS. That said, the protocol in this memo is an "opportunistic security" protocol, meaning that it strives to communicate with each peer as securely as possible, while maintaining broad interoperability. Therefore, the SMTP client MAY proceed to use DANE TLS (as described in Section 2.2.2 below) even with MX hosts obtained via an "insecure" MX RRset. For example, when a hosting provider has a signed DNS zone and publishes TLSA records for its SMTP servers, hosted domains that are not signed may still benefit from the provider's TLSA records. Deliveries via the provider's SMTP servers will not be subject to active attacks when sending SMTP clients elect to make use of the provider's TLSA records.

When the MX records are not (DNSSEC) signed, an active attacker can redirect SMTP clients to MX hosts of his choice. Such redirection is tamper-evident when SMTP servers found via "insecure" MX records are recorded as the next-hop relay in the MTA delivery logs in their original (rather than CNAME expanded) form. Sending MTAs SHOULD log unexpanded MX hostnames when these result from insecure MX lookups. Any successful authentication via an insecurely determined MX host MUST NOT be misrepresented in the mail logs as secure delivery to the intended next-hop domain. When DANE TLS is mandatory (Section 6) for a given destination, delivery MUST be delayed when the MX RRset is not "secure".

Otherwise, assuming no DNS errors (Section 2.1.1), 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 2.1.1, 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" (see Section 2.1.3), 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 2.1.1). 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 2.1.1). 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 MAY proceed via pre-DANE opportunistic TLS. SMTP clients MAY elect to use "insecure" TLSA records to avoid STARTTLS downgrades or even to skip SMTP servers that fail authentication, but MUST NOT misrepresent authentication success as either a secure connection to the SMTP server or as a secure delivery to the intended next-hop domain.

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 MUST 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 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".

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.

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.

3.1.1.1. 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.

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.

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

As noted in the introduction, SMTP clients cannot, without relying on DNSSEC for secure MX records and DANE for STARTTLS support signaling, perform server identity verification or prevent STARTTLS downgrade attacks. The use of PKIX 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 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. Lacking a complete set of public CAs, clients would not be able to verify the certificates of SMTP servers whose issuing root CAs are not trusted by the client.

Opportunistic DANE TLS needs to interoperate without bilateral coordination of security settings between client and server systems. Therefore, parameter choices that are fragile in the absence of bilateral coordination are unsupported. Nothing is lost since the PKIX certificate usages cannot aid SMTP TLS security, they can only impede SMTP TLS interoperability.

SMTP client treatment of TLSA RRs with certificate usages PKIX-TA(0) or PKIX-EE(1) is undefined. SMTP clients should generally treat such TLSA records as unusable.

3.2. Certificate matching

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

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 3.1.1 above.

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 a "secure" 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 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. When the client MTA is employing DANE TLS security despite "insecure" MX redirection the MX hostname is the only reference identifier.

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".

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.

4. Server key management

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.

As described in Section 3.1.2, when server certificates are validated via a DANE-TA(2) trust anchor, and CNAME records are employed to store the TA association data at a single location, the responsibility of updating the TLSA RRset shifts to the operator of the trust anchor. Before a new trust anchor is used to sign any new server certificates, its certificate (digest) is added to the relevant TLSA RRset. After enough time elapses for the original TLSA RRset to age out of DNS caches, the new trust anchor can start issuing new server certificates. Once all certificates issued under the previous trust anchor have expired, its associated RRs can be removed from the TLSA RRset.

In the DANE-TA(2) key management model server operators do not generally need to update DNS TLSA records after initially creating a CNAME record that references the centrally operated DANE-TA(2) RRset. If a particular server's key is compromised, its TLSA CNAME SHOULD be replaced with a DANE-EE(3) association until the certificate for the compromised key expires, at which point it can return to using CNAME record. If the central trust anchor is compromised, all servers need to be issued new keys by a new TA, and a shared DANE-TA(2) TLSA RRset needs to be published containing just the new TA. SMTP servers cannot expect broad SMTP client CRL or OCSP support.

5. 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 "BetterAlg" stronger than digest algorithm "WorseAlg". Suppose further that a server's TLSA RRset contains some records with "BetterAlg" 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 "WorseAlg" with some usage and selector it also appears with algorithm "BetterAlg" with the same usage and selector. In that case our client can safely ignore TLSA records with the weaker algorithm "WorseAlg", because it suffices to check the records with the stronger algorithm "BetterAlg".

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.

6. 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.

7. 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.

8. Interoperability considerations

8.1. 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 some fallback certificate chain of its choice. 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 fallback certificate may match a different name acceptable to the client, e.g., the original next-hop domain.

8.2. 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.

9. Operational Considerations

9.1. 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.

9.2. 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 5 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".

10. 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.

11. IANA considerations

This specification requires no support from IANA.

12. 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.

13. References

13.1. 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.

13.2. 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

DNS-Based Authentication of Named Entities (DANE)
Internet-Draft

Intended status: Standards Track

Expires: December 12, 2014

T. Finch
University of Cambridge

M. Miller

Cisco Systems, Inc.

P. Saint-Andre

&yet

June 10, 2014

Using DNS-Based Authentication of Named Entities (DANE) TLSA Records
with SRV Records
draft-ietf-dane-srv-06

Abstract

The DANE specification (RFC 6698) describes how to use TLSA resource records in the DNS to associate a server's host name with its TLS certificate, where the association is secured with DNSSEC. However, application protocols that use SRV records (RFC 2782) to indirectly name the target server host names for a service domain cannot apply the rules from RFC 6698. Therefore this document provides guidelines that enable such protocols to locate and use TLSA records.

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 December 12, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. DNS Checks	3
3.1. SRV Query	4
3.2. Address Queries	4
3.3. TLSA Queries	5
3.4. Impact on TLS Usage	5
4. TLS Checks	5
4.1. SRV Records Only	5
4.2. TLSA Records	6
5. Guidance for Application Protocols	7
6. Guidance for Server Operators	7
7. Internationalization Considerations	8
8. IANA Considerations	8
9. Security Considerations	8
9.1. Mixed Security Status	8
9.2. A Service Domain Trusts its Servers	8
9.3. Certificate Subject Name Matching	9
10. Acknowledgements	9
11. References	9
11.1. Normative References	9
11.2. Informative References	10
Appendix A. Examples	11
A.1. IMAP	11
A.2. XMPP	11
Appendix B. Rationale	12
Authors' Addresses	13

1. Introduction

The base DANE specification [RFC6698] describes how to use TLSA resource records in the DNS to associate a server's host name with its TLS certificate, where the association is secured using DNSSEC. That document "only relates to securely associating certificates for TLS and DTLS with host names" (see the last paragraph of section 1.2 of [RFC6698]).

Some application protocols do not use host names directly; instead, they use a service domain, and the relevant target server host names

are located indirectly via SRV records [RFC2782]. Because of this intermediate resolution step, the normal DANE rules specified in [RFC6698] cannot be applied to protocols that use SRV records. (Rules for SMTP [RFC5321], which uses MX records instead of SRV records, are described in [I-D.ietf-dane-smtp-with-dane].)

This document describes how to use DANE TLSA records with SRV records. To summarize:

- o We rely on DNSSEC to secure the association between the service domain and the target server host names (i.e., the host names that are discovered by the SRV query).
- o The TLSA records are located using the port, protocol, and target server host name fields (not the service domain).
- o Clients always use TLS when connecting to servers with TLSA records.
- o Assuming that the association is secure, the server's certificate is expected to authenticate the target server host name, rather than the service domain.

Note: The "CertID" specification [RFC6125] does not use the terms "service domain" and "target server host name", but refers to the same entities with the terms "source domain" and "derived domain".

2. Terminology

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

This draft uses the definitions for "secure", "insecure", "bogus", and "indeterminate" from [RFC4033]. This draft uses the acronyms from [RFC7218] for the values of TLSA fields where appropriate.

3. DNS Checks

To expedite connection to the intended service, where possible the queries described in the following sections SHOULD be performed in parallel (this is similar to the "happy eyeballs" approach for IPv4 and IPv6 connections described in [RFC6555]).

3.1. SRV Query

When the client makes an SRV query, a successful result will typically be a list of one or more SRV records (or possibly a chain of CNAME / DNAME aliases leading to such a list).

For this specification to apply, the entire DNS RRset that is returned MUST be "secure" according to DNSSEC validation ([RFC4033] section 5). In the case of aliases, the whole chain of CNAME and DNAME RRsets MUST be secure as well. This corresponds to the AD bit being set in the response(s); see [RFC4035] section 3.2.3.

If the the entire RRset is not secure, this protocol has not been correctly deployed. The client SHOULD fall back to its non-DNSSEC, non-DANE behavior (this corresponds to the AD bit being unset).

If a particular response is "bogus" or "indeterminate" according to DNSSEC validation, the client MUST ignore that target server host name.

In the successful case, the client now has an authentic list of target server host names with weight and priority values. It performs server ordering and selection using the weight and priority values without regard to the presence or absence of DNSSEC or TLSA records. It also takes note of the DNSSEC validation status of the SRV response for use when checking certificate names (see Section 4). The client can now proceed to making address queries on the target server host names as described in the next section.

3.2. Address Queries

For each SRV target server host name, the client makes A / AAAA queries, performs DNSSEC validation on the address (A, AAAA) response, and continues as follows based on the results:

- o If the response is "secure" and usable, the client MUST perform a TLSA query for that target server host name as described in the next section.
- o If the response is "insecure", the client MUST NOT perform a TLSA query for that target server host name; the TLSA query will most likely fail.
- o If the response is "bogus" or "indeterminate", the client MUST NOT connect to this target server; instead it uses the next most appropriate SRV target.

3.3. TLSA Queries

The client SHALL construct the TLSA query name as described in [RFC6698] section 3, based on fields from the SRV record: the port from the SRV RDATA, the protocol from the SRV query name, and the TLSA base domain set to the SRV target server host name.

For example, the following SRV record for IMAP (see [RFC6186]) leads to the TLSA query shown below:

```
_imap._tcp.example.com. 86400 IN SRV 10 0 9143 imap.example.net.  
_9143._tcp.imap.example.net. IN TLSA ?
```

3.4. Impact on TLS Usage

The client SHALL determine if the TLSA record(s) returned in the previous step are usable according to section 4.1 of [RFC6698]. This affects the use TLS as follows:

- o If the TLSA response is "secure" and usable, then the client MUST use TLS when connecting to the target server. The TLSA records are used when validating the server's certificate as described under Section 4.
- o If the TLSA response is "insecure", then the client SHALL proceed as if the target server had no TLSA records. It MAY connect to the target server with or without TLS, subject to the policies of the application protocol or client implementation.
- o If the TLSA response is "bogus" or "indeterminate", then the client MUST NOT connect to the target server (the client can still use other SRV targets).

4. TLS Checks

When connecting to a server, the client MUST use TLS if the responses to the SRV and TLSA queries were "secure" as described above. The rules described in the next two sections apply.

4.1. SRV Records Only

If the client received zero usable TLSA certificate associations, it SHALL validate the server's TLS certificate using the normal PKIX rules [RFC5280] or protocol-specific rules (e.g., following [RFC6125]) without further input from the TLSA records.

In this case, the client uses the information in the server certificate and the DNSSEC validation status of the SRV query in its authentication checks. It SHOULD use the Server Name Indication extension (TLS SNI) [RFC6066] or its functional equivalent in the relevant application protocol (e.g., in XMPP [RFC6120] this is the 'to' address of the initial stream header). The preferred name SHALL be chosen as follows, and the client SHALL verify the identity asserted by the server's certificate according to section 6 of [RFC6125], using a list of reference identifiers constructed as follows (note again that in RFC 6125 the terms "source domain" and "derived domain" refer to the same things as "service domain" and "target server host name" in this document). The examples below assume a service domain of "im.example.com" and a target server host name of "xmpp23.hosting.example.net".

SRV is insecure: The reference identifiers SHALL include the service domain and MUST NOT include the SRV target server host name (e.g., include "im.example.com" but not "xmpp23.hosting.example.net"). The service domain is the preferred name for TLS SNI or its equivalent.

SRV is secure: The reference identifiers SHALL include both the service domain and the SRV target server host name (e.g., include both "im.example.com" and "xmpp23.hosting.example.net"). The target server host name is the preferred name for TLS SNI or its equivalent.

In the latter case, the client will accept either identity to ensure compatibility with servers that support this specification as well as servers that do not support this specification.

4.2. TLSA Records

If the client received one or more usable TLSA certificate associations, it SHALL process them as described in section 2.1 of [RFC6698].

If the TLS server's certificate -- or the public key of the server's certificate -- matches a usable TLSA record with Certificate Usage "DANE-EE", the client MUST consider the server to be authenticated. Because the information in such a TLSA record supersedes the non-key information in the certificate, all other [RFC5280] and [RFC6125] authentication checks (e.g., reference identifier, key usage, expiration, issuance) MUST be ignored or omitted.

5. Guidance for Application Protocols

This document describes how to use DANE with application protocols in which target servers are discovered via SRV records. Although this document attempts to provide generic guidance applying to all such protocols, additional documents for particular application protocols could cover related topics, such as:

- o Fallback logic in the event that a client is unable to connect securely to a target server by following the procedures defined in this document.
- o How clients ought to behave if they do not support SRV lookups, or if clients that support SRV lookups encounter service domains that do not offer SRV records.
- o Whether the application protocol has a functional equivalent for TLS SNI that is preferred within that protocol.

For example, [I-D.ietf-xmpp-dna] covers such topics for the Extensible Messaging and Presence Protocol (XMPP).

6. Guidance for Server Operators

To conform to this specification, the published SRV records and subsequent address (A, AAAA) records MUST be secured with DNSSEC. There SHOULD also be at least one TLSA record published that authenticates the server's certificate.

When using TLSA records with Certificate Usage "DANE-EE", it is not necessary for the deployed certificate to contain an identifier for either the source domain or target server host name. However, servers that rely solely on validation using Certificate Usage "DANE-EE" TLSA records might prevent clients that do not support this specification from successfully connecting with TLS.

For TLSA records with Certificate Usage types other than "DANE-EE", the certificate(s) MUST contain an identifier that matches:

- o the service domain name (the "source domain" in [RFC6125] terms, which is the SRV query domain); and/or
- o the target server host name (the "derived domain" in [RFC6125] terms, which is the SRV target).

Servers that support multiple service domains (i.e., so-called "multi-tenanted environments") can implement the Transport Layer Security Server Name Indication (TLS SNI) [RFC6066] or its functional

equivalent to determine which certificate to offer. Clients that do not support this specification will indicate a preference for the service domain name, while clients that support this specification will indicate the target server host name. However, the server determines what certificate to present in the TLS handshake; e.g., the presented certificate might only authenticate the target server host name.

7. Internationalization Considerations

If any of the DNS queries are for an internationalized domain name, then they need to use the A-label form [RFC5890].

8. IANA Considerations

No IANA action is required.

9. Security Considerations

9.1. Mixed Security Status

We do not specify that clients checking all of a service domain's target server host names are consistent in whether they have or do not have TLSA records. This is so that partial or incremental deployment does not break the service. Different levels of deployment are likely if a service domain has a third-party fallback server, for example.

The SRV sorting rules are unchanged; in particular they have not been altered in order to prioritize secure servers over insecure servers. If a site wants to be secure it needs to deploy this protocol completely; a partial deployment is not secure and we make no special effort to support it.

9.2. A Service Domain Trusts its Servers

By signing their zone with DNSSEC, service domain operators implicitly instruct their clients to check their server TLSA records. This implies another point in the trust relationship between service domain holders and their server operators. Most of the setup requirements for this protocol fall on the server operator: installing a TLS certificate with the correct name (where necessary), and publishing a TLSA record for that certificate. If these are not correct then connections from TLSA-aware clients might fail.

9.3. Certificate Subject Name Matching

Section 4 of the TLSA specification [RFC6698] leaves the details of checking names in certificates to higher level application protocols, though it suggests the use of [RFC6125].

Name checks are not necessary if the matching TLSA record is of Certificate Usage "DANE-EE". Because such a record identifies the specific certificate (or public key of the certificate), additional checks are superfluous and potentially conflicting.

Otherwise, while DNSSEC provides a secure binding between the server name and the TLSA record, and the TLSA record provides a binding to a certificate, this latter step can be indirect via a chain of certificates. For example, a Certificate Usage "PKIX-TA" TLSA record only authenticates the CA that issued the certificate, and third parties can obtain certificates from the same CA. Therefore, clients need to check whether the server's certificate matches one of the expected reference identifiers to ensure that the certificate was issued by the CA to the server the client expects.

10. Acknowledgements

Thanks to Mark Andrews for arguing that authenticating the target server host name is the right thing, and that we ought to rely on DNSSEC to secure the SRV lookup. Thanks to James Cloos, Viktor Dukhovni, Ned Freed, Olafur Gudmundsson, Paul Hoffman, Phil Pennock, Hector Santos, Jonas Schneider, and Alessandro Vesely for helpful suggestions.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, 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.

- [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.
- [RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.
- [RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, January 2011.
- [RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 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.
- [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.
- [RFC7218] Gudmundsson, O., "Adding Acronyms to Simplify Conversations about DNS-Based Authentication of Named Entities (DANE)", RFC 7218, April 2014.

11.2. Informative References

- [I-D.ietf-dane-smtp-with-dane]
Dukhovni, V. and W. Hardaker, "SMTP security via opportunistic DANE TLS", draft-ietf-dane-smtp-with-dane-05 (work in progress), February 2014.
- [I-D.ietf-xmpp-dna]
Saint-Andre, P. and M. Miller, "Domain Name Associations (DNA) in the Extensible Messaging and Presence Protocol (XMPP)", draft-ietf-xmpp-dna-05 (work in progress), February 2014.

[RFC6555] Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", RFC 6555, April 2012.

Appendix A. Examples

In the following, most of the DNS resource data is elided for simplicity.

A.1. IMAP

```
; mail domain
_imap._tcp.example.com. SRV 10 0 9143 imap.example.net.
example.com.           RRSIG  SRV ...

; target server host name
imap.example.net.      A      192.0.2.1
imap.example.net.      RRSIG  A ...

imap.example.net.      AAAA    2001:db8:212:8::e:1
imap.example.net.      RRSIG  ...

; TLSA resource record
_9143._tcp.imap.example.net. TLSA  ...
_9143._tcp.imap.example.net. RRSIG  TLSA ...
```

Mail messages submitted for addresses at example.com are sent via IMAP to imap.example.net. Connections to imap.example.net port 9143 that use STARTTLS will get a server certificate that authenticates the name imap.example.net.

A.2. XMPP

```
; XMPP domain
_xmpp-client.example.com. SRV      1 0 5222 im.example.net.
_xmpp-client.example.com. RRSIG    SRV ...

; target server host name
im.example.net.      A      192.0.2.3
im.example.net.      RRSIG  A ...

im.example.net.      AAAA    2001:db8:212:8::e:4
im.example.net.      RRSIG  AAAA ...

; TLSA resource record
_5222._tcp.im.example.net. TLSA  ...
_5222._tcp.im.example.net. RRSIG  TLSA ...
```

XMPP sessions for addresses at example.com are established at im.example.net. Connections to im.example.net port 5222 that use STARTTLS will get a server certificate that authenticates the name im.example.net.

Appendix B. Rationale

The long-term goal of this specification is to settle on TLS certificates that verify the target server host name rather than the service domain, since this is more convenient for servers hosting multiple domains (so-called "multi-tenanted environments") and scales up more easily to larger numbers of service domains.

There are a number of other reasons for doing it this way:

- o The certificate is part of the server configuration, so it makes sense to associate it with the server host name rather than the service domain.
- o In the absence of TLS SNI, if the certificate identifies the host name then it does not need to list all the possible service domains.
- o When the server certificate is replaced it is much easier if there is one part of the DNS that needs updating to match, instead of an unbounded number of hosted service domains.
- o The same TLSA records work with this specification, and with direct connections to the host name in the style of [RFC6698].
- o Some application protocols, such as SMTP, allow a client to perform transactions with multiple service domains in the same connection. It is not in general feasible for the client to specify the service domain using TLS SNI when the connection is established, and the server might not be able to present a certificate that authenticates all possible service domains. See [I-D.ietf-dane-smtp-with-dane] for details.
- o It is common for SMTP servers to act in multiple roles, for example as outgoing relays or as incoming MX servers, depending on the client identity. It is simpler if the server can present the same certificate regardless of the role in which it is to act. Sometimes the server does not know its role until the client has authenticated, which usually occurs after TLS has been established. See [I-D.ietf-dane-smtp-with-dane] for details.

This specification does not provide an option to put TLSA records under the service domain because that would add complexity without

providing any benefit, and security protocols are best kept simple. As described above, there are real-world cases where authenticating the service domain cannot be made to work, so there would be complicated criteria for when service domain TLSA records might be used and when they cannot. This is all avoided by putting the TLSA records under the target server host name.

The disadvantage is that clients which do not complete DNSSEC validation must, according to [RFC6125] rules, check the server certificate against the service domain, since they have no other way to authenticate the server. This means that SNI support or its functional equivalent is necessary for backward compatibility.

Authors' Addresses

Tony Finch
University of Cambridge Computing Service
New Museums Site
Pembroke Street
Cambridge CB2 3QH
ENGLAND

Phone: +44 797 040 1426
Email: dot@dotat.at
URI: <http://dotat.at/>

Matthew Miller
Cisco Systems, Inc.
1899 Wynkoop Street, Suite 600
Denver, CO 80202
USA

Email: mamille2@cisco.com

Peter Saint-Andre
&yet
P.O. Box 787
Parker, CO 80134
USA

Email: peter@andyet.com