DANE
Internet-Draft
Intended status: Best Current Practice
Expires: August 18, 2014

V. Dukhovni
Unaffiliated
W. Hardaker
Parsons

February 14, 2014

DANE TLSA implementation and operational guidance
draft-ietf-dane-ops-03

Abstract

This memo provides guidance to server operators to help ensure that clients will be able to authenticate a server's certificate chain via published TLSA records. Guidance is also provided to clients for selecting reliable TLSA record parameters and using them for server authentication. Finally, guidance is given to protocol designers who wish to make use of TLSA records when securing protocols using a combination of the Transport Layer Security (TLS) protocol and TLSA records.

Status of This Memo

This Internet-Draft will expire on August 18, 2014.

Copyright Notice

Table of Contents

1. Introduction

Section 2 of [RFC6698] specifies a new "TLSA" DNS resource record which associates a TLS transport endpoint with a corresponding trusted leaf or issuing authority certificates or public keys. DNSSEC-validated DANE TLSA records can be used to augment or replace the trust model of the existing public Certificate Authority (CA) Public Key Infrastructure (PKI).

[RFC6698] defines 24 combinations of TLSA record parameters. Additional complexity arises when the TLS transport endpoint is obtained indirectly via a Service Record (SRV), Mail Exchange (MX) record, 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 employ multiple pairs of private keys and certificates with TLS clients signalling the desired domain via an 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 the 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 protects the transport against man-in-the-middle (MITM) attacks.

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 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: Customers that make use of a Service Provider to outsource their service(s) will be referred to as "Customer Domains".

TLSA Publisher: The entity responsible for publishing a TLSA record within a DNS zone. This zone will be considered 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 may be the operator of the outsourced DNS service.

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

SNI: "Server Name Indication", or SNI, describes the TLS protocol extension by which a TLS client requests to connect 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 TLS service endpoint (i.e., "secure virtual hosting").

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 3 numeric parameters. The numeric values of these parameters were later given symbolic names in [I-D.ietf-dane-registry-acronyms]. These 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 PKIX, or public PKI issued, and TA, or domain-issued trust anchors:

- 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.)

- When the low bit is not set (PKIX-TA(0) and DANE-TA(2)) the TLSA record matches a trust anchor (a Certificate Authority) that issued one of the certificates in the server certificate chain.

- 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 certificate authority PKI. Trust is entirely placed on the content of the TLSA records obtained via DNSSEC.

- 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 constrain the server certificate chain to contain 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. A digest algorithm agility protocol is proposed in section 2.3.3 of [I-D.ietf-dane-smtp-with-dane] that 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. 300 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 rest of the record is the certificate association data field, which is in this case the SHA2-256 digest of the server certificate.

3. General DANE Guidelines

These guidelines provide guidance for using or designing protocols for DANE, regardless of what sort of TLSA record will be used.

3.1. TLS Requirements

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" extension of TLS.

3.2. 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.

3.2.1. UDP and TCP Considerations

Deployments SHOULD avoid TLSA record sizes that cause UDP fragmentation.

Although DNS over TCP would provide the ability to transfer larger DNS records between clients and servers, it is not universally deployed and is still blocked by some firewalls. Clients that request DNS records via UDP typically only use TCP upon receipt of a truncated response in TCP.

3.2.2. 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 may need to be published simultaneously during a certificate rollover.

While TLSA records using a TLSA Selector of SPKI(1) and a TLSA Matching Type of Full(0) (which publishes the bare public key 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.

In summary, the use of a TLSA Matching Type of Full(0) is NOT RECOMMENDED and the use of SHA2-256(1) and SHA2-512(2) is strongly preferred.

3.3. Certificate Name Check Conventions

Certificates presented by a TLS server will generally contain a subjectAltName (SAN) extension or a Common Name (CN) element in the subject distinguished name (DN). The server's DNS domain name should be published within these elements, ideally within the subjectAltName extension as use of the CN field for this purpose is deprecated. Name checks SHOULD NOT consider the subject CN when SAN values of type 'dns' are present.

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. 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 from the client). With certificate usage values other than DANE-EE(3), the EE certificate SHOULD include the TLSA base domain as one of its names. If other acceptable names are specified by a protocol-specific DANE standard, one of those MAY be used in place of the TLSA base domain.

Given the DNSSEC validated DNS records below:

```
example.com.               300 IN MX 0 mail.example.com.
_25._tcp.mail.example.com. 300 IN TLSA DANE-TA Cert SHA2-256  (
                               E8B54E0B4BAA815B06D3462D65FBC7C0
                               CF556ECCF9F5303EBFBB77D022F834C0 )
```

The TLSA base domain is "mail.example.com" and this MUST be the HostName in the client's SNI extension. The server certificate chain MUST be signed by a trust anchor with the above certificate SHA2-256 digest. One of the DNS names in the server certificate MUST be either "mail.example.com" or "example.com".

3.4. 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, it is possible to employ CNAME records (see Section 3.5) 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 certificate 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 SNI to select the right certificate chain to present to each client.

Below are example DNS records that illustrate both of of the above cases in the case of an HTTPS service whose clients all support DANE TLS.

```
; Hosted web service redirected via a CNAME alias.
; Associated TLSA RRset redirected via a CNAME alias.
;
; Single certificate at provider works for all Customer Domains
;
www1.example.com.             300 IN CNAME w3.example.net.
_443._tcp.www3.example.com.   300 IN CNAME _443._tcp.w3.example.net.
_443._tcp.w3.example.net.     300 IN TLSA  DANE-EE SPKI SHA2-256 (
                              8A9A70596E869BED72C69D97A8895DFA
                              D86F300A343FECEFF19E89C27C896BC9 )
;
; CA at provider can issue certificates for each Customer Domain.
;
www2.example.com.             300 IN CNAME w2.example.net.
_443._tcp.www2.example.com.   300 IN CNAME _443._tcp.w2.example.net.
_443._tcp.w2.example.net.     300 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 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:

```
; Hosted SMTP service
;
example.com.               300 IN MX 0 mx1.example.net.
example.com.               300 IN MX 0 mx2.example.net.
_25._tcp.mx1.example.net. 300 IN TLSA DANE-EE SPKI SHA2-256 (
                              8A9A70596E869BED72C69D97A8895DFA
                              D86F300A343FECEFF19E89C27C896BC9 )
_25._tcp.mx2.example.net. 300 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 is 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.

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

3.5. TLSA Base Domain and CNAMEs

When the 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 origin domain to the target domain for all protocols.

The complexity of coordinating key rollover is largely eliminated when DANE TLSA records are found in the Service Provider's domain, as discussed in Section 3.4. 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.

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. They SHOULD use the target hostname as the preferred TLSA base domain as well as the HostName in SNI, provided the CNAME RR is found to be "secure" at each step in the CNAME expansion.

3.6. 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 certificate 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).

3.7. 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 should not continue to use that server in the event of authentication failure, or else authentication serves no purpose. 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 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. Protocols must choose whether to prioritize security or robustness.

3.7.1. 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.

3.8. TLSA Records and Trust Anchor Digests

With TLSA records that match the EE certificate (i.e., DANE-EE(3) or PKIX-EE(1)), the TLS client has no difficulty matching TLSA records against the server certificate, as this certificate is always present in the TLS server certificate chain.

With DANE TLSA records that match the digest of a TA certificate or public key (i.e., DANE-TA(2) or PKIX-TA(0)), a complication arises when the TA certificate is omitted from the server's certificate chain. This can happen when the trust anchor is a root certificate authority, as stated in 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 certificate 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.

This means that TLSA records that match a TA certificate or public key digest are not entirely sufficient to validate the peer certificate chain. If no matching certificate is found in the server's certificate chain, the chain may be signed by an omitted root CA whose digest matches the TLSA record. With Certificate Usage PKIX-TA(0), this is not a problem, since the client is expected to be pre-configured with the issuing TA certificate.

With TLSA Certificate Usage DANE-TA(2), however, there is no expectation that the client is pre-configured with the trust anchor certificate. Rather, with TLSA Certificate Usage DANE-TA(2) clients must be able to rely on the TLSA records alone. 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) with a digest (not Full(0)) matching type 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.

3.9. Trust anchor public keys

TLSA records with TLSA Certificate Usage DANE-TA(2), selector SPKI(1) and a matching type of Full(0) 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, but the issuer name is not included in the public key.

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. If 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 public key in the TLSA record.

Since trust chain validation via bare public keys rather than trusted CA certificates may be difficult to implement using existing TLS libraries, servers SHOULD include the trust anchor certificate in their certificate chains when the TLSA Certificate Usage is DANE-TA(2).

If none of the server's certificate chain elements match a public key specified in full in a TLSA record, clients SHOULD check whether the topmost certificate in the chain is signed by the provided public key and has not expired, and if that is the case, and the rest of the chain passes validation, consider the server authenticated if name checks are also successful.

4. Certificate Usage Specific DANE Guidelines

4.1. Certificate Usage DANE-EE(3) Guidelines

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

Two TLSA records will need to be published before updating a server's public key, one matching the currently deployed key and the other matching the new key scheduled to replace it. Once sufficient time has elapsed for all the previous TLSA RRsets, which contains only the old key, to expire from DNS caches, the server may be reconfigured to use the new private key and associated certificate chain. Once the server is using the new key, the TLSA RR that matches the retired key can be removed from DNS, leaving only the TLSA RR that matches the new key.

TLSA records for servers SHOULD, when possible, be DANE-EE(3), SPKI(1), SHA2-256(1) records. Such records specify the SHA2-256 digest of the public key of the server certificate. Since all DANE implementations are required to support SHA2-256, this record works for all clients and need not change across certificate renewals with

the same key. With no name checks required, this TLSA record type supports hosting arrangements with a single certificate matching all client domains! It is also the easiest to implement correctly in the client.

4.2. Certificate Usage DANE-TA(2) Guidelines

Some domains may prefer to reduce the operational complexity of maintaining a distinct TLSA RRset for each TLS service. If the domain employs a common issuing certificate authority to create certificates for multiple TLS services, it may be simpler to publish the issuing authority as a trust anchor (TA) for the certificate chains of all relevant services. The TLSA RRs for each service issued by the same TA may then be CNAMEs to a common TLSA RRset that matches the TA. This certificate usage also allows Service Providers to independently generate appropriate certificates for each Customer Domain (see Section 3.4).

As explained in Section 3.8, servers that employ Certificate Usage DANE-TA(2) TLSA records MUST include the TA certificate as part of the certificate chain presented in the TLS handshake even when it is a self-signed root certificate. TLSA Publishers should publish either "DANE-TA(2) SPKI(1) SHA2-256(1)" or "DANE-TA(2) Cert(0) SHA2-256(1)" TLSA parameters. As with leaf certificate rollover discussed in Section 4.1, two such TLSA RRs need to be published to facilitate TA certificate rollover.

4.3. Certificate Usage PKIX-EE(1) Guidelines

From a TLSA record perspective 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. 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 his choosing, and thus bypass the PKIX verification requirements.

Therefore, in most cases this certificate usage offers only illusory incremental security over usage DANE-EE(3). It provides lower operational reliability than usage 3 since when 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) Guidelines

TLSA Certificate Usage PKIX-TA(0) allows a domain to publish constraints on the set of certificate authorities trusted to issue certificates for its TLS servers. Clients MUST only accept PKIX-verified trust chains which contain a match for one of the published TLSA records.

TLSA Publishers MAY publish TLSA records for a particular public root CA, expecting 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 a trusted root published in the server's TLSA records.

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 records. This means that, 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, it MUST attempt to build a longer chain if the trusted certificate found is not self-issued, in the hope that a certificate closer to the root may in fact match the server's TLSA records.

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, in most cases this certificate usage offers only illusory incremental security over usage DANE-TA(2). It provides lower reliability than usage 2, though, since some clients may not be configured with the required root CA, and additionally requires more complex coordination between the Customer Domain and the Service Provider in hosting arrangements. This certificate usage is NOT RECOMMENDED.

5. 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 implementors.

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, 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. 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 with the entire domain becoming invalid.

6. 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].

7. IANA considerations

This specification requires no support from IANA.

8. 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.

9. References

9.1. 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.

9.2. 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.

[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