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

DANE TLSA implementation and operational guidance
draft-ietf-dane-ops-00

Abstract

This memo provides operational 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 to use for server authentication. Finally, guidance is given to protocol designers who wish to make use of TLSA records to secure protocols using a TLS and TLSA combination.

Table of Contents

1. Introduction

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

In the context of this memo, channel security is assumed to be provided by TLS or DTLS. The Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) protocols provide secured TCP and UDP communication over the Internet Protocol. By convention, "TLS" will be used through this document and, unless otherwise specified, the text applies equally as well to the DTLS protocol. Used without authentication, TLS provides protection only against eavesdropping. With authentication, TLS also provides protection

against man-in-the-middle (MITM) attacks. Since the publication of the TLS 1.0 specification in [RFC2246], two updates to the protocol have been published: TLS 1.1 [RFC4346] and TLS 1.2 [RFC5246]. The DTLS protocol was later documented in [RFC6347].

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

When a TLS client goes to the trouble of authenticating a certificate presented by a TLS server, it should not continue to use the server in case of authentication failure or else authentication serves no purpose. Consequently, if a client cannot reliably authenticate correctly configured, legitimate servers via a particular combination of TLSA parameters, then the client should treat that combination of parameters as unusable. Otherwise, the client risks routinely dropping connections to legitimate servers. Servers publishing TLSA records MUST be configured to allow correctly configured clients to successfully authenticate the server's TLS certificate.

If a TLSA record is found as unusable because of a parameter combination, it is protocol specific as to whether the connection should be established anyway without security, with only TLS encryption and not authentication, or to refuse to connect entirely.

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 [RFC2119].

This memo is being discussed on the dane@ietf.org mailing list.

The following terms are used throughout this document:

Service Provider: A company or organization that offers to host a service on behalf of a Client Domain. The original domain name associated with the service is typically still within the control of the client, however, and the service provider is frequently referred to by a redirection resource record. Example redirection records include MX, SRV, and CNAME. Many times the Service Provider provides services for many customers and must carefully

manage TLS credentials offered to their clients to ensure name matching is handled easily by clients.

Client Domain: Clients that make use of a Service Provider to outsource their services will be referred to as "Client Domains".

TLSA Publisher: The entity responsible for publishing a TLSA record within a DNS zone. This zone will be considered DNSSEC signed, unless otherwise specified. If the Client Domain is not outsourcing their DNS service, the TLSA Publisher will be the client themselves. Otherwise the TLSA Publisher may be the outsourced DNS service instead.

public key: The term "public key" will be an informal short-hand for the subjectPublicKeyInfo from a PKIX certificate.

SNI: The "Server Name Indication", or SNI, describes the process by which a TLS client requests to connect to a particular service name for a TLS server ([RFC3546]). Without this TLS extension, a TLS server has no choice but to offer a PKIX certificate with a default server name. Service Providers that are expected to host services for many clients need to present the correct certificate for the correct client, and the SNI extension provides a hint to the server which certificate should be transmitted to the client.

2. DANE TLSA record overview

[RFC6698] specifies a protocol for publishing TLS server certificate associations via DNSSEC. The DANE TLSA specification defines multiple TLSA RR types via combinations of the following 3 parameters:

- The TLSA Certificate Usage field. Section 2.1.1 of [RFC6698] specifies 4 values ranging from 0 to 3.

- The selector field. Section 2.1.2 of [RFC6698] specifies 2 values ranging from 0 to 1.

- The matching type field. Section 2.1.3 of [RFC6698] specifies 3 values ranging from 0 to 2.

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

- When the low bit is set (TLSA Certificate Usages 1 and 3) the TLSA record matches an EE (server) certificate.

- When the low bit is not set (TLSA Certificate Usages 0 and 2) the TLSA record matches a trust-anchor (a certificate authority) that issued a certificate somewhere in the certificate chain that authenticates the final end-entity certificate.

- When the high bit is set (TLSA Certificate Usages 2 and 3) the server certificate chain is domain-issued and may be verified without reference to the 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 (TLSA Certificate Usages 0 and 1) the TLSA record publishes a server policy stating that its certificate chain must pass PKIX validation [RFC5280], with the DANE TLSA record 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 or just its subjectPublicKeyInfo (i.e. an ASN.1 DER encoding of the certificate's algorithm id, any parameters and the public key data). A selector field of "0" specifies the whole certificate. A selector field of "1" specifies just the public key.

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 "0" means exact match, the DER encoding of the certificate or public key is given in the TLSA RR. A "1" value means a SHA-256 digest and "2" means a SHA-512 digest. Of these, only SHA-256 is mandatory to implement. Clients SHOULD implement SHA-512, but servers SHOULD NOT exclusively publish SHA-512 digests. Unless a "second preimage" attack is found against SHA-256, servers should only publish SHA-256 digests.

2.1. Example TLSA record

In the example TLSA record below:

```
_25._tcp.mail.example.com. IN TLSA 3 0 1 (
                              E8B54E0B4BAA815B06D3462D65FBC7C0
                              CF556ECCF9F5303EBFBB77D022F834C0 )
```

The TLSA Certificate Usage is "3", the selector is "0" and the matching type is "1". The rest of the record is the certificate association data field, which is in this case the SHA-256 digest of the server certificate.

3. General DANE Guidelines

These guidelines provide guidance for using or designing protocols for DANE, regardless of what type the TLSA record will actually contain.

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 is the size of the resulting TLSA record based on the parameters chosen.

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 yet widely deployed or permitted through many firewalls. TCP must be expected to be deployed on all the DNS servers and DNS clients for it to be a truly viable large-record solution.

3.2.2. Packet Size Considerations for TLSA Parameters

Server operators SHOULD NOT publish "TLSA * 0 0" records, as even a single certificate is generally too large to be reliably delivered via DNS without TCP being widely available. Furthermore, two full certificates may need to be published in the TLSA RRset for certificate rollover.

While "TLSA * 1 0" records, which publish full public keys without the full X.509 wrapping, are generally more compact, these too should be used with caution. Servers SHOULD publish digests within TLSA records instead. The complete certificate should, instead, be transmitted to the client in band during the TLS handshake.

3.3. Certificate Name Check Conventions

Certificates presented by a TLS server will contain either a Common Name (CN) or subjectAltName (or both), according to [RFC5280]. The server's hostname should be published within these fields, ideally within the subjectAltName. This section discusses what must be done to match an expected name against the name found within a certificate, if required.

The TLSA Publisher for TLSA records for a given service MUST ensure that at least one of these TLSA records will match the server's certificate chain. If SNI is not employed for a TLS connection, the TLSA record must match the server's default certificate. If the SNI extension is sent by the client with a host_name (see [RFC3546] Section 3.1) equal to the base domain of the TLSA RRset, at least one TLSA record must match the certificate presented by the server for that host_name.

When, for example, the TLSA RRset is published at

    _25._tcp.mail.example.com

the TLSA base domain is mail.example.com. At least one of the TLSA records in the _25._tcp.mail.example.com RRset MUST match the server certificate chain, provided the client TLS hanshake included the SNI extension with a host_name of "mail.example.com".

Note: Except with TLSA Certificate Usage "3", where name checks are not applicable (see Section 4.1), DANE aware clients SHOULD use the base domain of the TLSA RRset to verify that the client has reached the correct server by checking that the TLSA base domain is matched by one of the subjectAltName ([RFC5280]) in the server certificate. The commonName from the certificate subject DN MAY be used only when no subjectAltNames of type 'dns' are present. Additional acceptable names may be specified by protocol specific DANE RFCs. For example, with SMTP both the destination domain name and the MX host name are acceptable in the server certificate.

Since the server's ability to respond with the right certificate chain requires the TLS client to provide the correct SNI information, DANE PKI aware clients SHOULD send the SNI extension with a host_name value of the base domain of the TLSA RRset (otherwise they risk failure to authenticate the server).

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 don't fall out of sync with the server certificate configuration.

Ideally, the TLSA Publisher and the Service Provider should be the same entity. If a TLSA record must be published in the client's base domain, CNAME records can be easily used to point at the real TLSA record in the Service Provider's zone assuming certificate usage 3 TLSA records are published by the Service Provider (see Section 3.5). 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.

For example, with SMTP, the customer's MX records can be pointed at the Service Provider's MX hosts. When the customer's DNS zone is signed, the MX records can be securely used as the base names for TLSA records managed by the Service Provider.

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. Also Unlike MX or SRV records, CNAME records may chain (though clients will generally impose implementation dependent maximum nesting depths).

When CNAMEs are employed, the best place to seek DANE TLSA records is in the Service Provider's domain, as discussed in Section 3.4. Therefore, DANE PKI 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) and use the resulting target host as the base domain for TLSA lookups. Standards defining how to use DANE anchored TLS for each application protocol are expected to specify where to locate TLSA RRs when the destination is referred to by a CNAME.

If CNAMEs were not followed, Client Domains would need to publish TLSA records that match the Service Provider's certificate chain or always use an entity that was both the Service Provider and the TLSA publisher. Having the TLSA base domain be different than the Service Provider's domain imposes a difficult key management burden on the Client Domain and the Service Provider.

It is possible to publish CNAMEs in the Client Domain pointing to the Service Provider's TLSA RRset if the TLSA certificate usage field is set to 3. Otherwise, a client that used the alias name (from the hosted domain rather than the Service Provider's domain) as the base

domain to obtain the TLSA RRset would look for the hosted domain in the server certificate when performing name checks, and would generally fail to authenticate the server except in the rare cases when the server's certificate does include the Client Domain. SNI SHOULD be used to help perform the right certificate selection by the server, although this imposes a management burden on the TLS server that could be avoided by ensuring the TLSA base domain is within the Service Provider's control in the first place.

Example CNAME record for a TLSA domain:

```
 ; TLSA RRs aliased to Service Provider, but the base domain is
 ; the hosted domain.  Likely to fail name check unless Service
 ; Provider usage is "3".
 ;
 _25._tcp.mail.example.com. IN CNAME _25._tcp.mail.example.net.
     _25._tcp.mail.example.net. IN TLSA 3 1 1 ...
```

Note: when the TLSA RRset query domain (base domain plus port and protocol prefixes) resolves to a DNSSEC validated CNAME that points to a DNSSEC signed zone with the actual TLSA records, as the above example indicates, it has no effect on the value of the base domain, which remains the original domain to which the client prefixed the port and protocol. In the example above, the base domain is "mail.example.com" and not "mail.example.net".

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. In this case, if the MX or SRV host is a CNAME alias the client MAY "chase" the CNAME and SHOULD use the target hostname as the base domain for TLSA records as well as the host_name in SNI, provided the CNAME RR is found to be "secure" at each step in the CNAME expansion.

3.6. TLSA Base Name Priorities

There are multiple steps within a chaining DNS lookup process that TLSA base names can be pulled from. This section will discuss what the preferred selection points are. TBD.

1. Final Domain Name

2. Redirect Name

3. Initial Name

3.7. Interaction with Certificate Transparency

[RFC6962] Certificate Transparency or CT for short, defines an 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 a 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 DANE TLSA RRs constrain the end-entity certificate to a fixed public key, there is no role for CT, and clients SHOULD NOT apply CT checks.

When a server is authenticated via a DANE TLSA RR with TLSA Certificate Usage "1" or "3" (that is an end-entity certificate association), the domain owner has unambiguously specified the certificate associated with the given service. Even if a rogue CA were able to issue an unauthorized end-entity certificate that binds a public key to a name in that domain, barring "second preimage" attacks on the hashing algorithms in use, any such certificate would not match the TLSA record and would be rejected. Therefore, when a TLS client authenticates the TLS server via a TLSA certificate association with usage "1" or "3", CT checks SHOULD NOT be performed. Publication of the server certificate or public key (digest) in a DNSSEC signed zone by the domain owner assures the 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 "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 root). Since TLSA Certificate Usage "2" is generally intended to support non-PKIX trust anchors, clients SHOULD NOT perform CT checks with usage "2" using unknown root CAs. A server operator that wants CT checks SHOULD publish TLSA RRs with usage "0", or can obviate them with usage "1" or "3".

CT checks remain applicable with TLSA Certificate Usage "0" when the client supports both DANE and CT and the trusted PKIX root issuer is a known public root.

3.8. Design Considerations for Protocols Using DANE

3.8.1. Design Considerations for non-PKIX Protocols

For some application protocols, the existing public CA PKI may not be viable. For these (non-PKIX) protocols, servers SHOULD NOT suggest publishing TLSA records with TLSA Certificate Usage "0" or "1", as 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 "0" or "1" as unusable. After verifying that the only available TLSA Certificate Usage types are "0" or "1", protocol definitions MAY instruct clients to either refuse to initiate a connection or to connect via unauthenticated mandatory TLS if no alternative authentication mechanisms are available.

If non-PKIX protocols do allow for publication of TLSA records with TLSA Certificate Usage "0" or "1", clients SHOULD make use of the TLSA verification to the fullest extent possible.

3.8.1.1. TLSA Certificate Usage 1

With non-PKIX protocols, clients using TLSA Certificate Usage "1" records MAY ignore the PKIX validation requirement, and authenticate the server per the content of the TLSA record alone. Since servers will hopefully rely on SNI to select the correct certificate for presentation, the client SHOULD use the SNI extension to signal the base domain of the TLSA RRset.

3.8.1.2. TLSA Certificate Usage 0

With TLSA Certificate Usage "0" in non-PKIX protocols, the usability of the TLSA records depends on its matching type.

If the matching type is "0", the TLSA record contains the full certificate or full public key of the trusted certificate authority. In this case the client has all the information it needs to match the server trust-chain to the TLSA record. The client MAY ignore the PKIX validation requirement and authenticate the server via its DANE TLSA records alone (sending SNI with the base domain as usual). The client SHOULD use the base domain of the TLSA record(s) in certificate name checks.

If the matching type is not "0", the TLSA record contains only a digest of the trust certificate authority certificate or public key. The full certificate may not be included in the server's certificate chain and the client may not be able to match the server trust chain against the TLSA record when a non-PKIX protocol is being used, as the client won't have a default CA trust list. See Section 3.9.1 for a more complete discussion of this case. The client cannot reliably authenticate the server in this case and SHOULD treat the TLSA record as unusable.

If the client is configured with a set of trusted CAs that are believed to be sufficiently complete to authenticate all the servers it expects to communicate with, then it MAY elect to honor

certificate usage "0" TLSA records that publish digests of the trusted CA certificate or public key.

3.9. TLSA Records and Trust Anchor Digests

With TLSA records that match the EE certificate, the TLS client has no difficulty matching the TLS record against the server certificate, as this certificate is always present in the TLS server certificate chain. The TLS client can, if necessary, extract the public key from the server certificate, and can compute the appropriate digest.

With DANE TLSA records that 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. 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. We will consider each trust-anchor TLSA Certificate Usage in turn.

3.9.1. Trust Anchor Digests With TLSA Certificate Usage 0

In this case, from the server's perspective, the omission of the root CA seems reasonable, since in addition to authentication via DANE TLSA records, the client is expected to perform [RFC5280] PKIX validation of the server's trust chain and thus to already have a copy of the omitted root certificate.

From the client's perspective the situation is more nuanced. Despite the server's indicated preference for PKIX validation, the client may not possess (or may not fully trust) a complete set of public root CAs. This is especially likely in protocols where the existing public CA PKI is not applicable, as described in Section 3.8.1. If it is likely that a client lacks a sufficiently complete list of trusted CAs, and that a non-negligible number of DNS servers publish TLSA Certificate Usage 0 TLSA records with digests of omitted root

CAs, then such a client SHOULD treat such TLSA records as "unusable". Simply ignoring PKIX validation is not an option, since the client will also be unable to match the TLSA record without position of the root certificate. The client MAY choose fall back to unauthenticated TLS, if PKIX is also not an option (see [I-D.ietf-dane-srv]) or refuse to initiate a connection.

3.9.2. Trust Anchor Digests With TLSA Certificate Usage 2

With TLSA Certificate Usage "2", there is no expectation that the client is pre-configured with the trust anchor certificate. With TLSA Certificate Usage "2" clients are expecting to rely on the TLSA records alone. But, with a matching type other than "0" the TLSA records contain 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, clients will be unable to authenticate the server.

TLSA Publishers that publish TLSA Certificate Usage "2" with a non-zero matching type MUST ensure that the corresponding server is configured to include the associated 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.

Since servers are expected to always provide usage "2" trust anchor certificates (either via DNS or else via the TLS hanshake), clients SHOULD fully support this TLSA Certificate Usage. Clients MAY choose to treat it as unusable if experience proves that servers don't consistently live up to their obligations.

3.10. Trust anchor public keys

TLSA records with TLSA Certificate Usage "0" or "2", selector "1" and a matching type of "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 "0", when the client is able to perform PKIX validation, the client can construct a complete PKIX trust chain as it will have access to the trust anchor name. So in that case, the client can verify that the server certificate chain is issued by a trust anchor that matches the TLSA record.

With TLSA Certificate Usage "2", the client may not have the missing 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" 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 chain when the TLSA Certificate Usage is "2".

If none of the server's certificate chain elements match a public key specified in full (selector = 0, match type = 0) in a TLSA record, clients SHOULD attempt to check whether the topmost certificate in the chain is signed by the provided public key, and if so consider the server trust chain valid, with authentication complete if name checks are also successful.

4. Type Specific DANE Guidelines

4.1. Type 3 Guidelines

4.2. Type 2 Guidelines

4.3. Type 1 Guidelines

4.4. Type 0 Guidelines

TLSA Certificate Usage "0" allows a domain to publish constraints on the set of certificate authorities trusted to issue certificates for its TLS servers. It is expected that clients will only accept trust chains which contain a match for one of the published TLSA records. This is simple for TLSA Certificate Usage "1" where the PKIX trust chain always contains the leaf server certificate. The situation for TLSA Certificate Usage "0" is more subtle.

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 set of trusted certificates 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 chain may omit any trusted roots 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 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.

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 public root CAs 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 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 of 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 every day. For more domains with less critical data, a reasonable signature lifetime is a couple of weeks to a month, and the zone should be resigned every week. 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. 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.

7. Security Considerations

Application protocols that cannot make use of the existing public CA PKI (so called non-PKIX protocols), may choose to not implement certain PKIX-dependent TLSA record types defined in [RFC6698], or may choose to make a best-effort use of such records. In neither case is security compromised, since by assumption PKIX verification is simply not an option for these protocols. When the TLS server is authenticated based on the TLSA records alone, the client is as well authenticated as possible, treating the TLSA records as unusable would lead to weaker security.

Therefore, when TLSA records are used with protocols where PKIX does not apply, the recommended trade-off is for servers to not publish PKIX-dependent TLSA records, and for clients to use them as best they can, but otherwise treat them unusable. Of course when PKIX validation is an option clients SHOULD perform PKIX validation per [RFC6698].

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.

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

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

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

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

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

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

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

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[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.

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

8.2. Informative References

[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.

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