T. Finch
University of Cambridge
February 18, 2013

Using DNS-Based Authentication of Named Entities (DANE) TLSA records with SRV and MX records.
draft-ietf-dane-srv-00

Abstract

The DANE specification [RFC6698] describes how to use TLSA resource records in the DNS to associate a server's host name with its TLS certificate. The association is secured with DNSSEC. Some application protocols can use SRV records [RFC2782] to indirectly name the server hosts for a service domain. (SMTP uses MX records for the same purpose.) This specification gives generic instructions for how these application protocols locate and use TLSA records. Separate documents give the details that are specific to particular application protocols.

Table of Contents

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. 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, but instead use a service domain. The domain's servers are located indirectly via SRV records [RFC2782] (or MX records in the case of SMTP [RFC5321]). When they do not use host names [RFC6698] does not direcly apply to these protocols.

This document describes how to use DANE TLSA records with SRV and MX 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 SRV or MX query.

- o The TLSA records are located alongside the SRV target host names.

- o Clients always use TLS when connecting to servers with TLSA records.

- o The server's certificate is expected to match the server host name, rather than the service domain.

Separate documents give the details that are specific to particular application protocols. For examples, see [I-D.ietf-dane-smtp] and [I-D.ietf-dane-mua].

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

2. Using TLSA records with SRV and MX

2.1. MX records

For the purpose of this specification (to avoid cluttering the description with special cases) each MX record ([RFC5321] section 5) is treated as being equivalent to a SRV record [RFC2782] with corresponding fields copied from the MX record and the remaining fields having fixed values as follows:

Service smtp

Proto tcp

Name MX owner name

TTL MX TTL

Class MX Class

Priority MX Priority

Weight 0

Port 25

Target MX Target

For example this MX record is treated as if it were the following SRV record:

```
example.com.            86400 IN MX  10      mx.example.net.

_smtp._tcp.example.com. 86400 IN SRV 10 0 25 mx.example.net.
```

2.2. SRV query

When the client makes a SRV query, a successful result can be a (possible chain of CNAME / DNAME aliases referring to a) list of one or more SRV records.

For this specification to take effect, all of these DNS RRsets MUST be "secure" according to DNSSSEC validation ([RFC4033] section 5). In the case of a (chain of) aliases, the whole chain MUST be secure as well as the ultimate target. (This corresponds to the AD bit being set in the response(s) - see [RFC4035] section 3.2.3.)

If they are not all secure, this protocol has not been fully deployed. The client SHOULD fall back to its non-DNSSEC non-DANE behaviour. (This corresponds to the AD bit being unset.)

If any of the responses is "bogus" according to DNSSEC validation the client MUST abort. (This usually corresponds to a "server failure" response.)

The client now has an authentic list of 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.

2.3. TLSA queries

This sub-section applies to each server host name individually.

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

For example this SRV record leads to the following TLSA query:

```
_imap._tcp.example.com. 86400 IN SRV 10 0 143 imap.example.net.

_143._tcp.imap.example.net. IN TLSA ?
```

- o A secure answer containing one or more TLSA records, in which case the client SHALL proceed as descrbed below.
- o A bogus answer or other failure, which the client MUST treat as a temporary error.
- o If there is no TLSA record or its DNSSEC validation state is insecure or indeterminate, this protocol has not been fully deployed. The client SHOULD deliver to this server insecurely (which might be over unauthenticated TLS, as described in the introduction).

3. Guidelines for application protocols

Separate documents describe how to apply this specification to particular application protocols. If you are writing such as document the following points ought to be covered:

- o How should the client react to a "bogus" DNSSEC status?

4. Security considerations

4.1. Mixed security status

We do not specify that clients check that all of a service domain's 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 fall-back server, for example.

The SRV and MX 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.

4.2. 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, and publishing a TLSA record under that name. If these are not correct then connections from TLSA-aware clients might fail.

4.3. 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 checking might appear to be unnecessary, since DNSSEC provides a secure binding between the server name and the TLSA record, which in turn authenticates the certificate. However this latter step can be indirect, via a chain of certificates. A usage=0 TLSA record only authenticates the CA that issued the certificate, and third parties can obtain certificates from the same CA.

So this specification says that clients check that the server's certificate matches the server host name, to ensure that the certificate was issued by the CA to the server that the client is connecting to. The client always performs this check regardless of the TLSA usage, to simplify implementation and so that this specification is less likely to need updating when new TLSA usages are added.

4.4. Deliberate omissions

We do not specify that clients check the DNSSEC state of the server address records. This is not necessary since the certificate checks ensure that the client has connected to the correct server. (The address records will normally have the same security state as the TLSA records, but they can differ if there are CNAME or DNAME indirections.)

5. Internationalization Considerations

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

6. IANA Considerations

No IANA action is required.

7. Acknowledgements

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

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.

[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.

[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.

[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.

[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

[I-D.ietf-dane-smtp]
Finch, T., "DNS-Based Authentication of Named Entities (DANE) for secure SMTP", draft-ietf-dane-smtp (work in progress), March 2013.

[I-D.ietf-dane-mua]
Finch, T., "DNS-Based Authentication of Named Entities (DANE) for POP, IMAP, and message submission", draft-ietf-dane-mua (work in progress), March 2013.

Appendix A. Example

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

```
; mail domain
example.com.                MX      1 mx.example.net.
example.com.                RRSIG   MX ...

; SMTP server host name
mx.example.net.             A      192.0.2.1
mx.example.net.             AAAA   2001:db8:212:8::e:1

; TLSA resource record
_25._tcp.mx.example.net.  TLSA   ...
_25._tcp.mx.example.net.  RRSIG  TLSA ...
```

Mail for addresses at example.com is delivered by SMTP to mx.example.net. Connections to mx.example.net port 25 that use STARTTLS will get a server certificate that authenticates the name mx.example.net.

Appendix B. Rationale

The long-term goal of this specification is to settle on TLS

certificates that verify the server host name rather than the service domain, since this is more convenient for servers hosting multiple domains and scales up more easily to larger numbers of service domains.

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

- The certificate is part of the server configuration, so it makes sense to associate it with the server name rather than the service domain.

- 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.

- 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.

- The same TLSA records work with this specification, and with direct connections to the host name in the style of [RFC6698].

- 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.

- It is common for SMTP servers to act in multiple roles, 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.

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 puttling the TLSA records under the server host name.

The disadvantage is that clients which do not do 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 Server Name Indication support is necessary for backwards compatibility.

Author's Address

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/