Workgroup: Internet Engineering Task Force
Internet-Draft:
draft-huque-dane-client-cert-04
Updates: 6698, 7671 (if approved)
Published: 30 October 2020
Intended Status: Standards Track
Expires: 3 May 2021
Authors: S. Huque V. Dukhovni A. Wilson
Salesforce Two Sigma Valimail

TLS Client Authentication via DANE TLSA records

Abstract

The DANE TLSA protocol [RFC6698] [RFC7671] describes how to publish Transport Layer Security (TLS) server certificates or public keys in the DNS. This document updates RFC 6698 and RFC 7671. It describes how to additionally use the TLSA record to publish client certificates or public keys, and also the rules and considerations for using them with 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 https://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 3 May 2021.

Table of Contents

1. Introduction and Motivation

The Transport Layer Security (TLS) protocol [RFC5246] [RFC8446] optionally supports the authentication of clients using X.509 certificates [RFC5280] or raw public keys [RFC7250]. TLS applications that perform DANE authentication of servers using TLSA records may also desire to authenticate clients using the same mechanism, especially if the client identity is in the form of or can be represented by a DNS domain name. Some design patterns from the Internet of Things (IoT) plan to make use of this form of authentication, where large networks of physical objects identified by DNS names may authenticate themselves using TLS to centralized device management and control platforms.

In this document, the term TLS is used generically to describe both the TLS and DTLS (Datagram Transport Layer Security) [RFC6347] protocols.

2. Associating Client Identities in TLSA Records

Different applications may have quite different conventions for naming clients via domain names. This document thus does not proscribe a single format, but mentions a few that may have wide applicability.

2.1. Format 1: Service specific client identity

In this format, the owner name of the client TLSA record has the following structure:

```
_service.[client-domain-name]
```

The first label identifies the application service name. The remaining labels are composed of the client domain name.

Encoding the application service name into the owner name allows the same client domain name to have different authentication credentials for different application services. There is no need to encode the transport label - the same name form is usable with both TLS and DTLS.

The _service label could be a custom string for an application, but more commonly is expected to be a service name registered in the IANA Service Name Registry [SRVREG].

The RDATA or data field portion of the TLSA record is formed exactly as specified in RFC 6698 and RFC 7671, and carries the same meaning.

2.2. Format 2: DevId: IOT Device Identity

The DevID form of the TLSA record has the following structure:

```
[devicename]._device.[org-domain-name]
```

It is loosely based on the proposed PKI Certificate Identifier Format for Devices [CERTDEVID], but is simpler in form. It makes no distinction between manufacturer issued and locally issued certificates, and does away with the "serial" and "type" labels. The "_device" label that precedes the organization domain name allows all the device identities to be delegated to a subzone or to another party.

3. Authentication Model

The authentication model assumed in this document is the following:

The client is assigned an identity corresponding to a DNS domain name. This domain name doesn't necessarily have any relation to its network layer addresses. Clients often have dynamic or unpredictable addresses, and may move around the network, so tying their identity to network addresses is not feasible or wise in the general case.

The client generates (or has generated for it) a private and public key pair. Where client certificates are being used, the client also has a certificate binding the name to its public key. The certificate or public key has a corresponding TLSA record published in the DNS, which allows it to be authenticated directly via the DNS (using the DANE-TA or DANE-EE certificate usage modes) or via a PKIX public CA system constraint if the client's certificate was issued by a public CA (using the PKIX-TA or PKIX-EE DANE usage modes).

4. Client Identifiers in X.509 certificates

If the TLS DANE Client Identity extension (see Section 5) is not being used, the client certificate MUST have have the client's DNS name specified in the Subject Alternative Name extension's dNSName type.

If the TLS DANE Client Identity extension is in use, then with DANE-EE(3), the subject name need not be present in the certificate.

5. Signaling the Client's DANE Identity in TLS

The client SHOULD explicitly signal that it has a DANE identity. The most important reason is that the server may want an explicit indication from the client that it has a DANE record, so as to avoid unnecessary DNS queries in-band with the TLS handshake.

The DANE Client Identity TLS extension [TLSCLIENTID] is used for this purpose. This extension can also be used to convey the actual DANE client identity (i.e. domain name) that the TLS server should attempt to authenticate. This is required when using TLS raw public key authentication, since there is no client certificate from which to extract the client's DNS identity. It is also helpful when the client certificate contains multiple identities, and only a specific one has a DANE record.

An additional case where such client signaling is helpful, is one where DANE client authentication is optional, and there is a population of buggy client software that does not react gracefully to receipt of a Certificate Request message from the TLS server. This extension allows TLS servers to deal with this situation by selectively sending a Certificate Request message only to clients that have sent this extension.

6. Example TLSA records for clients

The following examples are provided in the textual presentation format of the TLSA record.

6.1. Format 1: Service Specific Client Identity

An example TLSA record for the client "device1.example.com." and the application "smtp-client". This record specifies the SHA-256 hash of the subject public key component of the end-entity certificate corresponding to the client. The certificate usage for this record is 3 (DANE-EE) and thus is validated in accordance with section 5.1 of RFC 7671.

```
_smtp-client.device1.example.com. IN TLSA (
   3 1 1 d2abde240d7cd3ee6b4b28c54df034b9
         7983a1d16e8a410e4561cb106618e971 )
```

6.2. Format 2: DevID

An example TLSA record for the device named "sensor7" managed by the organization "example.com" This record specifies the SHA-512 hash of the subject public key component of an EE certificate corresponding to the client.

```
sensor7._device.example.com. IN TLSA (
   3 1 2 0f8b48ff5fd94117f21b6550aaee89c8
         d8adbc3f433c8e587a85a14e54667b25
         f4dcd8c4ae6162121ea9166984831b57
         b408534451fd1b9702f8de0532ecd03c )
```

7. Changes to Client and Server behavior

A TLS Client conforming to this specification MUST have a signed DNS TLSA record published corresponding to its DNS name and X.509 certificate or public key. The client presents this certificate or public key in the TLS handshake with the server. The client should not offer ciphersuites that are incompatible with its certificate or public key. If the client's certificate has a DANE record with a certificate usage other than DANE-EE, then the presented client certificate MUST have have the client's DNS name specified in the Subject Alternative Name extension's dNSName type.

Additionally, when using raw public key authentication, the client MUST send the TLS DANE Client Identity extension [TLSCLIENTID] in its Client Hello message. When using X.509 certificate authentication, it SHOULD send this extension.

A TLS Server implementing this specification performs the following steps:

*Request a client certificate in the TLS handshake (the "Client Certificate Request" message). This could be done

unconditionally, or only when it receives the TLS DANE Client Identity extension from the client.

* If the client has sent a non-empty DANE Client Identity extension, then extract the client's domain name from the extension. Otherwise, extract the client identity from the Subject Alternative Name extension's dNSName type.

* Construct the DNS query name for the corresponding TLSA record. If the TLS DANE client identity extension was present, then this name should be used. Otherwise, identities from the client certificate are used.

* Look up the TLSA record set in the DNS. The response MUST be cryptographically validated using DNSSEC. The server could perform the DNSSEC validation itself. It could also be configured to trust responses obtained via a validating resolver to which it has a secure connection.

* Extract the RDATA of the TLSA records and match them to the presented client certificate according to the rules specified in the DANE TLS protocol [RFC6698] [RFC7671]. If successfully matched, the client is authenticated and the TLS session proceeds. If unsuccessful, the server MUST treat the client as unauthenticated (e.g. it could terminate the session, or proceed with the session giving the client access to resources as a generic unauthenticated user).

* If there are multiple records in the TLSA record set, then the client is authenticated as long as at least one of the TLSA records matches, subject to RFC7671 digest agility, which SHOULD be implemented.

If the DANE Client Identity extension is not present, and the presented client certificate has multiple distinct reference identifier types (e.g. a dNSName, and an rfc822Name) then TLS servers configured to perform DANE authentication according to this specification should only examine and authenticate the dNSName.

If the presented client certificate has multiple dNSName identities, then the client MUST use the TLS DANE client identity extension to unambiguously indicate its intended name to the server.

Specific applications may be designed to require additional validation steps. For example, a server might want to verify the client's IP address is associated with the certificate in some manner, e.g. by confirming that a secure reverse DNS lookup of that address ties it back to the same domain name, or by requiring an iPAddress component to be included in the certificate. Such details are outside the scope of this document, and should be outlined in

other documents specific to the applications that require this behavior.

Servers may have their own whitelisting and authorization rules for which certificates they accept. For example a TLS server may be configured to only allow TLS sessions from clients with certificate identities within a specific domain or set of domains.

8. Raw Public Keys

When using raw public keys in TLS [RFC7250], this specification requires the use of the TLS DANE Client Identity extension. The associated DANE TLSA records employ only certificate usage 3 (DANE-EE) and a selector value of 1 (SPKI), as described in [RFC7671].

9. Acknowledgements

TBD.

10. IANA Considerations

This document includes no request to IANA.

11. Security Considerations

This document updates RFC 6698 by defining the use of the TLSA record for client TLS certificates. There are no security considerations for this document beyond those described in RFC 6698 and RFC 7671 and in the specifications for TLS and DTLS [RFC8446], [RFC5246], [RFC6347].

12. References

12.1. Normative References

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[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, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <https://www.rfc-editor.org/info/rfc6698>.

[RFC7250] Wouters, P., Ed., Tschofenig, H., Ed., Gilmore, J., Weiler, S., and T. Kivinen, "Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", RFC 7250, DOI 10.17487/RFC7250, June 2014, <https://www.rfc-editor.org/info/rfc7250>.

[RFC7671] Dukhovni, V. and W. Hardaker, "The DNS-Based Authentication of Named Entities (DANE) Protocol: Updates and Operational Guidance", RFC 7671, DOI 10.17487/RFC7671, October 2015, <https://www.rfc-editor.org/info/rfc7671>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[TLSCLIENTID] Huque, S. and V. Dukhovni, "TLS Extension for DANE Client Identity", <https://tools.ietf.org/html/draft-huque-tls-dane-clientid>.

12.2. Informative References

[CERTDEVID] Friel, O. and R. Barnes, "PKI Certificate Identifier Format for Devices", <https://tools.ietf.org/id/draft-friel-pki-for-devices-00.html>.

[SRVREG] IANA, "Service Name and Transport Protocol Port Number Registry", <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt>.

Authors' Addresses

Shumon Huque
Salesforce

Email: shuque@gmail.com

Viktor Dukhovni
Two Sigma

Email: ietf-dane@dukhovni.org

Ash Wilson

Valimail

Email: ash.wilson@valimail.com