Internet Engineering Task Force
Internet-Draft
Updates: 6698, 7671 (if approved)
Intended status: Standards Track
Expires: 17 March 2022
13 September 2021

A. Wilson
Valimail
S. Huque
Salesforce

PKI-Authenticated Certificate Discovery Using DANE TLSA records
draft-wilson-dane-pkix-cd-02

Abstract

The DNS-Based Authentication of Named Entities (DANE) TLSA specification [RFC6698] and The DNS-Based Authentication of Named Entities (DANE) Protocol: Updates and Operational Guidance [RFC7671] describe how to publish Transport Layer Security (TLS) server certificates or public keys in the DNS. This document updates [RFC6698] and [RFC7671]. It describes how to use the TLSA record to enable entity and CA certificate discovery for object security and trust chain discovery use cases, and how to use PKIX validation for TLSA records queried without the benefit of DNSSEC.

Status of This Memo

Copyright Notice

Table of Contents

1. Background and Motivation

1.1. Background

A digital identity consists of at least two essential elements: a name, and a method for proving ownership of the name. Digital identities are often represented using X.509 certificates [RFC5280], which bind a name to a public key under a certification authority's signature. This allows all entities which trust the certification authority to trust that the public key associated with the name in the certificate is authentic and unaltered. The public key may be used for authentication, in the establishment of a secure session between two entities using a protocol like TLS or DTLS. The public key in the certificate may also be used to provide object security mechanisms like cryptographic signature verification or payload encryption. The certificate discovery process for object security usually relies on either in-band transmission of the certificate by the sender (IEEE 802.11p DSRC), or out-of-band via a proprietary API presented by the certification authority.

1.2. Motivation

Internet of Things (IoT) applications increasingly need to authenticate messages sent through decoupled applications. Without some sort of message security mechanism in place, trust in sender identity is assumed based on the trustworthiness of all systems and networks involved in handling the message between the sending entity and the recipient. This document proposes the use of the DANE TLSA record for certificate discovery, and provides an optional method for certificate authentication, in the event DNSSEC is not available.

2. 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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Authentication Model

3.1. Object Signing And Encryption

An entity which generates a cryptographic signature for a message has an associated name in DNS, where a TLSA record may be found containing the entity's certificate. The entity's certificate contains the public key which can be used to authenticate the signature.

Likewise, an entity which can receive messages encrypted using a public key (either directly, or by way of a key-wrapping technique) has a DNS-based identity where a public key suitable for message encryption can be found.

3.2. Trust Anchors

A PKIX-CD identity has a discoverable certificate and trust anchor. The trust anchor MAY be used by a TLS server to populate a local certificate store for the purpose of enabling traditional PKI-based mutual TLS. This approach comes with constraints and caveats, as described in Security (Section 7), below. The use of PKIX-CD for enabling mutual TLS is a wrapper for PKI-based mutual authentication, and does not change the way that TLS operates. This approach provides a transitional state from PKIX-based DANE to DNSSEC-based DANE for mutual TLS authentication. Trust anchor discovery MAY happen on-the-fly within the TLS handshake only when using DNSSEC-based DANE.

3.3. Trust Model For DNS-Based Identities

DNSSEC SHOULD be used to ensure the integrity of the certificate presented via the TLSA record. For a variety of reasons, DNSSEC is not ubiquitous across the entire DNS namespace. For zones which are not protected by DNSSEC, the CA certificate which can be used to verify the certificates presented by the TLSA records MUST be retrieved from a known location derived from the identity's full DNS name.

3.4. DNS Naming Convention Or Labeling Format

This document defines a specific label or DNS naming format for the TLSA DNS records which carry entity certificates, and this format is compatible with the [TLSCLIENTID]. This is necessary to provide a well-known location for securely retrieving a CA certificate which can be used to verify certificates retrieved from DNS without the benefit of DNSSEC.

For a device identity with DNS name a1b2c3._device.subdomain.organization.example, we can decompose the DNS name into a few important parts:

* a1b2c3: device identifier

* _device: identity grouping label

* subdomain: organizational label

* organization.example: organizational domain

* _device.subdomain.organization.example: identity domain

The device identifier MAY consist of multiple labels, but for simplicity's sake we will represent it here as only one label.

The identity grouping label is the rightmost label prefixed by an underscore, to the left of the organizational domain. This does not need to be immediately adjacent to the organizational domain; labels MAY exist between the identity grouping label and the organizational domain. In the above example, this is _device. Another example might be abc123._messagesender.subdomain.example.net, where the identity grouping label would be _messagesender.

The organizational label(s) is any label existing between the identity grouping label and the organizational domain.

The organizational domain is the domain that was registered with a domain name registrar.

The identity domain is all labels from the top-level domain label through the identity grouping label.

3.5. Trust Anchor Discovery

In order to authenticate device certificates presented in TLSA records, without DNSSEC, we must safely obtain a CA certificate which can be used to verify the entity certificate.

Using the components of the device name, together with the authorityKeyIdentifier (AKI) found in the entity certificate (described in [RFC5280]), we can compose the URL where the signing certificate can be found.

The process whereby the signing certificate URL can be constructed for a device named a1b2c3._device.environment.example.net is as follows:

1. Perform a DNS query for a1b2c3._device.environment.organization.example rrtype==TLSA.

2. Extract the AKI from the certificate. We will use AA-BB-CC as the placeholder in this example.

3. Identify the organizational domain: organization.example

4. Identify any organizational labels: environment

5. Identify the identity grouping label, and remove the underscore prefix: device

6. Using the following pattern, create the hostname: ${IDENTITY_GROUPING_LABEL}.${ORGANIZATIONAL_LABELS}.${ORGANIZATIONAL_DOMAIN}: device.environment.organization.example

7. Adding HTTPS and the AKI, build the authority URI: https://device.environment.organization.example/.well-known/ca/AA-BB-CC.pem

The file found at the URL MUST contain exactly one PEM-encoded CA certificate. The HTTPS server presenting the file MUST use TLS with a certificate that can be validated to the TLS client's client root certificate store, if DANE is not used for the server's identity. The certificate in the PEM file does not need to chain to the TLS client's root certificate store.

A PEM-encoded certificate being presented by a server possessing a DANE or Web PKI-validated identity is sufficient to indicate the trustworthiness of the CA certificate for validating certificates presented for associated identities. In this example, the certificate found at https://device.environment.organization.example/.well-known/ca/AA-BB-CC.pem MAY be cached and used to validate PKIX-CD certificates with identities matching *._device.environment.example.net, if the certificates contain AKI AA-BB-CC.

The chain of trust from signing certificate to root certificate may be discovered using the authorityInfoAccess ([RFC5280] section 4.2.2.1) extension within the certificate, or by substituting the authorityKeyIdentifier in the signing certificate for the AKI in the authority URI, and querying the updated authority URI. This process may be repeated to retrieve the entire trust chain, the root certificate of which is useful for enabling certificate-based authentication for TLS clients.

Benefits of this approach:

* This method uses an HTTPS server as a content-addressable storage mechanism for public keys in CA certificates. The HTTPS server MAY host any number of CA certificates, and the HTTPS server does not need to provide any directory listing service. This makes the discovery of CA certificates possible by already knowing an identity's AKI, and the entire CA bundle for the zone is not required to be distributable as a bundle.

* Device identities exist in a zone apart from other identities, which may have granular management access controls applied.

* The DNS record used for locating the CA certificates does not exist in the same zone as the records it is used to verify. The zone where the authority record is located may have granular rules applied which create compartmentalized failure zones. If the device identity zone is compromised, altered certificates will not validate unless the authority server can also be impersonated.

4. Update to DANE

The method of certificate discovery via TLSA records, without DNSSEC, SHALL be referred to as PKIX-CD, and shall use the DANE Certificate Usage value of 4. The presence of the certificate usage value 4 indicates that the party requesting the certificate is responsible for validating it using the CA certificate located at the authority URL, which can be composed using the process described above.

5. PKIX Constraints

The certificates presented with PKIX-CD MUST contain the subjectAlternativeName OID, with at least one dNSName which matches the DNS name used to retrieve the certificate. A certificate which is delivered without the benefit of DNSSEC, even if the CA's signature is valid, MUST NOT be trusted without alignment between the DNS name used to retrieve the certificate and one DNSName attribute within the certificate. The SubjectAlternativeName field MAY have other entries in addition to the one which aligns with the DNS name where the certificate can be discovered.

6. IANA Considerations

This draft updates the TLSA Certificate Usages registry maintained at https://www.iana.org/assignments/dane-parameters/dane-parameters.xhtml#certificate-usages to add the following value:

Value	Acronym	Short Description	Reference
4	PKIXCD	PKI-Auth Cert Discovery	[PKIXCD]

Table 1

7. Security Considerations

This document updates RFC 6698 by defining the use of the TLSA record for discovering client TLS certificates and associated CA certificates. However, the security model presented here is quite different than the security model presented in RFC 6698. RFC 6698 relies on DNSSEC as the public key infrastructure (PKI) used for validating certificates (or certificate metadata) presented via TLSA resource records in DNS. This draft proposes the use of Web PKI as the root of trust, in the event the zone presenting TLSA records for this use case is not protected by DNSSEC. Web PKI's root of trust is in the browser CA bundle, and the substitution of Web PKI for the DNSSEC PKI trades the security implications of DNSSEC for Web PKI.

Because PKIX-CD can provide an avenue for certificate chain discovery, care must be taken to ensure that identities are authenticated via the correct chain. There may be a desire to locally cache all discovered CA certificates into a general certificate store. This must not happen without certain controls in place. Without the identity consumer (authenticator) ensuring that the signing certificate for a particular zone is only used to verify certificates ONLY from that particular zone, cross-domain

impersonation (is this a cousin of the confused deputy problem?) may be possible. An example of this is an application where organization.example and supplier.example both present identities via PKIX-CD. If the CA certificates from organization.example and supplier.example are loaded into a local certificate store and used for authenticating certificates from both zones, then devices under supplier.example may be deemed authentic even if signed by organization.example.

8. References

8.1. Normative References

[PKIXCD] Wilson, A. and S. Huque, "DANE PKIX Certificate Discovery", <https://tools.ietf.org/html/draft-wilson-dane-pkix-cd>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[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>.

[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>.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

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

Authors' Addresses

Ash Wilson
Valimail

Email: ash.d.wilson@gmail.com

Shumon Huque
Salesforce

Email: shuque@gmail.com