

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 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 <http://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 August 18, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	3
2. DANE TLSA record overview	4
2.1. Example TLSA record	5
3. General DANE Guidelines	6
3.1. TLS Requirements	6
3.2. DANE DNS Record Size Guidelines	6
3.3. Certificate Name Check Conventions	7
3.4. Service Provider and TLSA Publisher Synchronization	8
3.5. TLSA Base Domain and CNAMEs	10
3.6. Interaction with Certificate Transparency	11
3.7. Design Considerations for Protocols Using DANE	11
3.8. TLSA Records and Trust Anchor Digests	12
3.9. Trust anchor public keys	13
4. Certificate Usage Specific DANE Guidelines	14
4.1. Certificate Usage DANE-EE(3) Guidelines	14
4.2. Certificate Usage DANE-TA(2) Guidelines	15
4.3. Certificate Usage PKIX-EE(1) Guidelines	15
4.4. Certificate Usage PKIX-TA(0) Guidelines	15
5. Note on DNSSEC security	16
6. Security Considerations	17
7. IANA considerations	17
8. Acknowledgements	18
9. References	18
9.1. Normative References	18
9.2. Informative References	19
Authors' Addresses	19

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 be 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:

- o 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.)
- o 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.
- o 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.
- o 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 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

DANE
Internet-Draft
Intended status: Standards Track
Expires: August 18, 2014

V. Dukhovni
Unaffiliated
W. Hardaker
Parsons
February 14, 2014

SMTP security via opportunistic DANE TLS
draft-ietf-dane-smtp-with-dane-07

Abstract

This memo describes a downgrade-resistant protocol for SMTP transport security between Mail Transfer Agents (MTAs) based on the DNS-Based Authentication of Named Entities (DANE) TLSA DNS record. Adoption of this protocol enables an incremental transition of the Internet email backbone to one using encrypted and authenticated Transport Layer Security (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 <http://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 August 18, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	3
1.2. Background	4
1.3. SMTP channel security	5
1.3.1. STARTTLS downgrade attack	5
1.3.2. Insecure server name without DNSSEC	6
1.3.3. Sender policy does not scale	7
1.3.4. Too many certificate authorities	7
2. Hardening (pre-DANE) Opportunistic TLS	8
2.1. DNS errors, bogus and indeterminate responses	8
2.2. TLS discovery	11
2.2.1. MX resolution	13
2.2.2. Non-MX destinations	14
2.2.3. TLSA record lookup	16
2.3. DANE authentication	17
2.3.1. TLSA certificate usages	18
2.3.2. Certificate matching	20
2.3.3. Digest algorithm agility	23
3. Mandatory TLS Security	25
4. Operational Considerations	25
4.1. Client Operational Considerations	25
4.2. Publisher Operational Considerations	25
5. Security Considerations	26
6. IANA considerations	26
7. Acknowledgements	27
8. References	27
8.1. Normative References	27
8.2. Informative References	28
Authors' Addresses	28

1. Introduction

This memo specifies a new connection security model for Message Transfer Agents (MTAs). This model is motivated by key features of inter-domain SMTP delivery, in particular the fact that the destination server is selected indirectly via DNS Mail Exchange (MX) records and that with MTA to MTA SMTP the use of TLS is generally opportunistic.

We note that the SMTP protocol is also used between Message User Agents (MUAs) and Message Submission Agents (MSAs) [RFC6409]. In [RFC6186] a protocol is specified that enables an MUA to dynamically locate the MSA based on the user's email address. SMTP connection

security requirements for MUAs implementing [RFC6186] are largely analogous to connection security requirements for MTAs, and this specification could be applied largely verbatim with DNS MX records replaced by corresponding DNS Service (SRV) records [I-D.ietf-dane-srv].

However, until MUAs begin to adopt the dynamic configuration mechanisms of [RFC6186] they are adequately served by more traditional static TLS security policies. This document will not discuss the MUA use case further, leaving specification of DANE TLS for MUAs to future documents that focus specifically on SMTP security between MUAs and MSAs. The rest of this memo will focus on securing MTA to MTA SMTP connections.

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 or concepts are used through the document:

secure, bogus, insecure, indeterminate: DNSSEC validation results, as defined in Section 4.3 of [RFC4035].

Validating Security-Aware Stub Resolver and Non-Validating Security-Aware Stub Resolver:
Capabilities of the stub resolver in use as defined in [RFC4033]; note that this specification requires the use of a Security-Aware Stub Resolver; Security-Oblivious stub-resolvers MUST NOT be used.

opportunistic DANE TLS: Best-effort use of TLS, resistant to downgrade attacks for destinations with DNSSEC-validated TLSA records. When opportunistic DANE TLS is determined to be unavailable, clients should fall back to opportunistic TLS below. Opportunistic DANE TLS requires support for DNSSEC, DANE and STARTTLS on the client side and STARTTLS plus a DNSSEC published TLSA record on the server side.

(pre-DANE) opportunistic TLS: Best-effort use of TLS that is generally vulnerable to DNS forgery and STARTTLS downgrade attacks. When a TLS-encrypted communication channel is not available, message transmission takes place in the clear. MX record indirection generally precludes authentication even when TLS is available.

MX hostname: The RRDATA of an MX record consists of a 16 bit preference followed by a Mail Exchange domain name (see [RFC1035], Section 3.3.9). We will use the term "MX hostname" to refer to the latter, that is, the DNS domain name found after the preference value in an MX record. Thus an "MX hostname" is specifically a reference to a DNS domain name, rather than any host that bears that name.

SMTP server: An SMTP server whose name appears in an MX record for a particular domain. Used to refer specifically to the host and SMTP service itself, not its DNS name.

delayed delivery: Email delivery is a multi-hop store & forward process. When an MTA is unable forward a message that may become deliverable later, the message is queued and delivery is retried periodically. Some MTAs may be configured with a fallback next-hop destination that handles messages that the MTA would otherwise queue and retry. In these cases, messages that would otherwise have to be delayed, may be sent to the fallback next-hop destination instead. The fallback destination may itself be subject to opportunistic or mandatory DANE TLS as though it were the original message destination.

original next hop destination: The logical destination for mail delivery. By default this is the domain portion of the recipient address, but MTAs may be configured to forward mail for some or all recipients via designated relays. The original next hop destination is, respectively, either the recipient domain or the associated configured relay.

MTA: Message Transfer Agent ([RFC5598], Section 4.3.2).

MSA: Message Submission Agent ([RFC5598], Section 4.3.1).

MUA: Message User Agent ([RFC5598], Section 4.2.1).

RR: A DNS Resource Record

RRset: A set of DNS Resource Records for a particular class, domain and record type.

1.2. Background

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

As described in the introduction of [RFC6698], TLS authentication via the existing public Certificate Authority (CA) PKI suffers from an over-abundance of trusted certificate authorities capable of issuing certificates for any domain of their choice. DANE leverages the DNSSEC infrastructure to publish trusted public keys and certificates for use with the Transport Layer Security (TLS) [RFC5246] protocol via a new "TLSA" DNS record type. With DNSSEC each domain can only vouch for the keys of its directly delegated sub-domains.

The TLS protocol enables secure TCP communication. In the context of this memo, channel security is assumed to be provided by TLS. Used without authentication, TLS provides only privacy protection against eavesdropping attacks. With authentication, TLS also provides data integrity protection to guard against man-in-the-middle (MITM) attacks.

1.3. SMTP channel security

With HTTPS, Transport Layer Security (TLS) employs X.509 certificates [RFC5280] issued by one of the many Certificate Authorities (CAs) bundled with popular web browsers to allow users to authenticate their "secure" websites. Before we specify a new DANE TLS security model for SMTP, we will explain why a new security model is needed. In the process, we will explain why the familiar HTTPS security model is inadequate to protect inter-domain SMTP traffic.

The subsections below outline four key problems with applying traditional PKI to SMTP that are addressed by this specification. Since SMTP channel security policy is not explicitly specified in either the recipient address or the MX record, a new signaling mechanism is required to indicate when channel security is possible and should be used. The publication of TLSA records allows server operators to securely signal to SMTP clients that TLS is available and should be used. DANE TLSA makes it possible to simultaneously discover which destination domains support secure delivery via TLS and how to verify the authenticity of the associated SMTP services, providing a path forward to ubiquitous SMTP channel security.

1.3.1. STARTTLS downgrade attack

The Simple Mail Transfer Protocol (SMTP) [RFC5321] is a single-hop protocol in a multi-hop store & forward email delivery process. SMTP envelope recipient addresses are not transport addresses and are security-agnostic. Unlike the Hypertext Transfer Protocol (HTTP) and its corresponding secured version, HTTPS, there is no URI scheme for email addresses to designate whether communication with the SMTP server should be conducted via a cleartext or a TLS-encrypted channel. Indeed, no such URI scheme could work well with SMTP since

TLS encryption of SMTP protects email traffic on a hop-by-hop basis while email addresses could only express end-to-end policy.

With no mechanism available to signal transport security policy, SMTP relays employ a best-effort "opportunistic" security model for TLS. A single SMTP server TCP listening endpoint can serve both TLS and non-TLS clients; the use of TLS is negotiated via the SMTP STARTTLS command ([RFC3207]). The server signals TLS support to the client over a cleartext SMTP connection, and, if the client also supports TLS, it may negotiate a TLS encrypted channel to use for email transmission. The server's indication of TLS support can be easily suppressed by a man in the middle attacker. Thus pre-DANE SMTP TLS security can be subverted by simply downgrading a connection to cleartext. No TLS security feature, such as the use of PKIX, can prevent this. The attacker can simply bypass TLS.

1.3.2. Insecure server name without DNSSEC

With SMTP, DNS Mail Exchange (MX) records abstract the next-hop transport endpoint and allow administrators to specify a set of target servers to which SMTP traffic should be directed for a given domain.

A PKIX TLS client is vulnerable to man in the middle (MITM) attacks unless it verifies that the server's certificate binds its public key to its name. However, with SMTP server names are obtained indirectly via MX records. Without DNSSEC, the MX lookup is vulnerable to MITM and DNS cache poisoning attacks. Active attackers can forge DNS replies with fake MX records and can redirect email to servers with names of their choice. Therefore, secure verification of SMTP TLS certificates is not possible without DNSSEC.

One might try to harden the use of TLS with SMTP against DNS attacks by requiring each SMTP server to possess a trusted certificate for the envelope recipient domain rather than the MX hostname. Unfortunately, this is impractical, as email for many domains is handled by third parties that are not in a position to obtain certificates for all the domains they serve. Deployment of the Server Name Indication (SNI) extension to TLS (see [RFC6066] Section 3) is no panacea, since SNI key management is operationally challenging except when the email service provider is also the domain's registrar and its certificate issuer; this is rarely the case for email.

Since the recipient domain name cannot be used as the SMTP server authentication identity, and neither can the MX hostname without DNSSEC, large-scale deployment of authenticated TLS for SMTP requires that the DNS be secure.

Since SMTP security depends critically on DNSSEC, it is important to point out that consequently SMTP with DANE is the most conservative possible trust model. It trusts only what must be trusted and no more. Adding any other trusted actors to the mix can only reduce SMTP security. A sender may choose to harden DNSSEC for selected high value receiving domains, by configuring explicit trust anchors for those domains instead of relying on the chain of trust from the root domain. In such a case there is not an "additional" trusted authority, rather the root trust anchor is replaced with a more specific trust anchor for each of the domains in question. Detailed discussion of DNSSEC security practices is out of scope for this document.

1.3.3. Sender policy does not scale

Sending systems are in some cases explicitly configured to use TLS for mail sent to specifically selected peer domains. This requires MTAs to be configured with appropriate subject names or certificate content digests to expect in the presented host certificates. Because of the heavy administrative burden, such statically configured SMTP secure channels are used rarely (generally only between domains that make bilateral arrangements with their business partners). Internet email, on the other hand, requires regularly contacting new domains for which security configurations cannot be established in advance.

The abstraction of the SMTP transport endpoint via DNS MX records, often across organization boundaries, limits the use of public CA PKI with SMTP to a small set of sender-configured peer domains. With little opportunity to use TLS authentication, sending MTAs are rarely configured with a comprehensive list of trusted CAs. SMTP services that support STARTTLS often use X.509 certificates that are self-signed or issued by a private CA.

1.3.4. Too many certificate authorities

Even if it were generally possible to determine a secure server name, the SMTP client would still need to verify that the server's certificate chain is issued by a trusted certificate authority (a trust anchor). MTAs are not interactive applications where a human operator can make a decision (wisely or otherwise) to selectively disable TLS security policy when certificate chain verification fails. With no user to "click OK", the MTAs list of public CA trust anchors would need to be comprehensive in order to avoid bouncing mail addressed to sites that employ unknown certificate authorities.

On the other hand, each trusted CA can issue certificates for any domain. If even one of the configured CAs is compromised or operated

by an adversary, it can subvert TLS security for all destinations. Any set of CAs is simultaneously both overly inclusive and not inclusive enough.

2. Hardening (pre-DANE) Opportunistic TLS

Neither email addresses nor MX hostnames (or submission SRV records) signal a requirement for either secure or cleartext transport. Therefore, SMTP transport security is of necessity generally opportunistic (barring manually configured exceptions).

This specification uses the presence of DANE TLSA records to securely signal TLS support and to publish the means by which SMTP clients can successfully authenticate legitimate SMTP servers. This becomes "opportunistic DANE TLS" and is resistant to downgrade and MITM attacks, and enables an incremental transition of the email backbone to authenticated TLS delivery, with increased global protection as adoption increases.

With opportunistic DANE TLS, traffic from SMTP clients to domains that publish "usable" DANE TLSA records in accordance with this memo is authenticated and encrypted. Traffic from non-compliant clients or to domains that do not publish TLSA records will continue to be sent in the same manner as before, via manually configured security, (pre-DANE) opportunistic TLS or just cleartext SMTP.

2.1. DNS errors, bogus and indeterminate responses

An SMTP client that implements opportunistic DANE TLS per this specification depends critically on the integrity of DNSSEC lookups, as discussed in Section 1.3. This section lists the DNS resolver requirements needed to avoid downgrade attacks when using opportunistic DANE TLS.

A DNS lookup may signal an error or return a definitive answer. A security-aware resolver must be used for this specification. Security-aware resolvers will indicate the security status of a DNS RRset with one of four possible values defined in Section 4.3 of [RFC4035]: "secure", "insecure", "bogus" and "indeterminate". In [RFC4035] the meaning of the "indeterminate" security status is:

An RRset for which the resolver is not able to determine whether the RRset should be signed, as the resolver is not able to obtain the necessary DNSSEC RRs. This can occur when the security-aware resolver is not able to contact security-aware name servers for the relevant zones.

Note, the "indeterminate" security status has a conflicting definition in section 5 of [RFC4033].

There is no trust anchor that would indicate that a specific portion of the tree is secure.

SMTP clients following this specification SHOULD NOT distinguish between "insecure" and "indeterminate" in the [RFC4033] sense. Both "insecure" and RFC4033 "indeterminate" are handled identically: in either case unvalidated data for the query domain is all that is and can be available, and authentication using the data is impossible. In what follows, when we say "insecure", we include also DNS results for domains that lie in a portion of the DNS tree for which there is no applicable trust anchor. With the DNS root zone signed, we expect that validating resolvers used by Internet-facing MTAs will be configured with trust anchor data for the root zone. Therefore, RFC4033-style "indeterminate" domains should be rare in practice. From here on, when we say "indeterminate", it is exclusively in the sense of [RFC4035].

As noted in section 4.3 of [RFC4035], a security-aware DNS resolver MUST be able to determine whether a given non-error DNS response is "secure", "insecure", "bogus" or "indeterminate". It is expected that most security-aware stub resolvers will not signal an "indeterminate" security status in the RFC4035-sense to the application, and will signal a "bogus" or error result instead. If a resolver does signal an RFC4035 "indeterminate" security status, this MUST be treated by the SMTP client as though a "bogus" or error result had been returned.

An MTA making use of a non-validating security-aware stub resolver MAY use the stub resolver's ability, if available, to signal DNSSEC validation status based on information the stub resolver has learned from an upstream validating recursive resolver. In accordance with section 4.9.3 of [RFC4035]:

... a security-aware stub resolver MUST NOT place any reliance on signature validation allegedly performed on its behalf, except when the security-aware stub resolver obtained the data in question from a trusted security-aware recursive name server via a secure channel.

To avoid much repetition in the text below, we will pause to explain the handling of "bogus" or "indeterminate" DNSSEC query responses. These are not necessarily the result of a malicious actor; they can, for example, occur when network packets are corrupted or lost in transit. Therefore, "bogus" or "indeterminate" replies are equated in this memo with lookup failure.

There is an important non-failure condition we need to highlight in addition to the obvious case of the DNS client obtaining a non-empty "secure" or "insecure" RRset of the requested type. Namely, it is not an error when either "secure" or "insecure" non-existence is determined for the requested data. When a DNSSEC response with a validation status that is either "secure" or "insecure" reports either no records of the requested type or non-existence of the query domain, the response is not a DNS error condition. The DNS client has not been left without an answer; it has learned that records of the requested type do not exist.

Security-aware stub resolvers will, of course, also signal DNS lookup errors in other cases, for example when processing a "ServFail" RCODE, which will not have an associated DNSSEC status. All lookup errors are treated the same way by this specification, regardless of whether they are from a "bogus" or "indeterminate" DNSSEC status or from a more generic DNS error: the information that was requested can not be obtained by the security-aware resolver at this time. A lookup error is thus a failure to obtain the relevant RRset if it exists, or to determine that no such RRset exists when it does not.

In contrast to a "bogus" or an "indeterminate" response, an "insecure" DNSSEC response is not an error, rather it indicates that the target DNS zone is either securely opted out of DNSSEC validation or is not connected with the DNSSEC trust anchors being used. Insecure results will leave the SMTP client with degraded channel security, but do not stand in the way of message delivery. See section Section 2.2 for further details.

When a stub resolver receives a response containing a CNAME alias, it will generally restart the query at the target of the alias, and should do so recursively up to some configured or implementation-dependent recursion limit. If at any stage of recursive CNAME expansion a query fails, the stub resolver's lookup of the original requested records will result in a failure status being returned. If at any stage of recursive expansion the response is "insecure", then it and all subsequent results (in particular, the final result) MUST be considered "insecure" regardless of whether the other responses received were deemed "secure". If at any stage of recursive expansion the validation status is "bogus" or "indeterminate" or associated with another DNS lookup error, the resolution of the requested records MUST be considered to have failed.

When a DNS lookup failure (error or "bogus" or "indeterminate" as defined above) prevents an SMTP client from determining which SMTP server or servers it should connect to, message delivery MUST be delayed. This naturally includes, for example, the case when a "bogus" or "indeterminate" response is encountered during MX

resolution. When multiple MX hostnames are obtained from a successful MX lookup, but a later DNS lookup failure prevents network address resolution for a given MX hostname, delivery may proceed via any remaining MX hosts.

When a particular SMTP server is selected as the delivery destination, a set of DNS lookups must be performed to discover any related TLSA records. If any DNS queries used to locate TLSA records fail (be it due to "bogus" or "indeterminate" records, timeouts, malformed replies, ServFails, etc.), then the SMTP client MUST treat that server as unreachable and MUST NOT deliver the message via that server. If no servers are reachable, delivery is delayed.

In what follows, we will only describe what happens when all relevant DNS queries succeed. If any DNS failure occurs, the SMTP client MUST behave as described in this section, by skipping the problem SMTP server, or the problem destination. Queries for candidate TLSA records are explicitly part of "all relevant DNS queries" and SMTP clients MUST NOT continue to connect to an SMTP server or destination whose TLSA record lookup fails.

2.2. TLS discovery

As noted previously (in Section 1.3.1), opportunistic TLS with SMTP servers that advertise TLS support via STARTTLS is subject to an MITM downgrade attack. Also some SMTP servers that are not, in fact, TLS capable erroneously advertise STARTTLS by default and clients need to be prepared to retry cleartext delivery after STARTTLS fails. In contrast, DNSSEC validated TLSA records MUST NOT be published for servers that do not support TLS. Clients can safely interpret their presence as a commitment by the server operator to implement TLS and STARTTLS.

This memo defines four actions to be taken after the search for a TLSA record returns secure usable results, secure unusable results, insecure or no results or an error signal. The term "usable" in this context is in the sense of Section 4.1 of [RFC6698]. Specifically, if the DNS lookup for a TLSA record returns:

- A secure TLSA RRset with at least one usable record: A connection to the MTA MUST be made using authenticated and encrypted TLS, using the techniques discussed in the rest of this document. Failure to establish an authenticated TLS connection MUST result in falling back to the next SMTP server or delayed delivery.
- A Secure non-empty TLSA RRset where all the records are unusable: A connection to the MTA MUST be made via TLS, but authentication is not required. Failure to establish an encrypted TLS connection

MUST result in falling back to the next SMTP server or delayed delivery.

An insecure TLSA RRset or DNSSEC validated proof-of-non-existent TLSA records:

A connection to the MTA SHOULD be made using (pre-DANE) opportunistic TLS, this includes using cleartext delivery when the remote SMTP server does not appear to support TLS. The MTA may optionally retry in cleartext when a TLS handshake fails.

Any lookup error: Lookup errors, including "bogus" and "indeterminate", as explained in Section 2.1 MUST result in falling back to the next SMTP server or delayed delivery.

An SMTP client MAY be configured to require DANE verified delivery for some destinations. We will call such a configuration "mandatory DANE TLS". With mandatory DANE TLS, delivery proceeds only when "secure" TLSA records are used to establish an encrypted and authenticated TLS channel with the SMTP server.

An operational error on the sending or receiving side that cannot be corrected in a timely manner may, at times, lead to consistent failure to deliver time-sensitive email. The sending MTA administrator may have to choose between letting email queue until the error is resolved and disabling opportunistic or mandatory DANE TLS for one or more destinations. The choice to disable DANE TLS security should not be made lightly. Every reasonable effort should be made to determine that problems with mail delivery are the result of an operational error, and not an attack. A fallback strategy may be to configure explicit out-of-band TLS security settings if supported by the sending MTA.

A note about DNAME aliases: a query for a domain name whose ancestor domain is a DNAME alias returns the DNAME RR for the ancestor domain, along with a CNAME that maps the query domain to the corresponding sub-domain of the target domain of the DNAME alias [RFC6672]. Therefore, whenever we speak of CNAME aliases, we implicitly allow for the possibility that the alias in question is the result of an ancestor domain DNAME record. Consequently, no explicit support for DNAME records is needed in SMTP software, it is sufficient to process the resulting CNAME aliases. DNAME records only require special processing in the validating stub-resolver library that checks the integrity of the combined DNAME + CNAME reply. When DNSSEC validation is handled by a local caching resolver, rather than the MTA itself, even that part of the DNAME support logic is outside the MTA.

When the original next-hop destination is an address literal, rather than a DNS domain, DANE TLS does not apply. Delivery proceeds using any relevant security policy configured by the MTA administrator. Similarly, when an MX RRset incorrectly lists an network address in lieu of an MX hostname, if the MTA chooses to connect to the network address DANE TLSA does not apply for such a connection.

In the subsections that follow we explain how to locate the SMTP servers and the associated TLSA records for a given next-hop destination domain. We also explain which name or names are to be used in identity checks of the SMTP server certificate.

2.2.1. MX resolution

In this section we consider next-hop domains that are subject to MX resolution and have MX records. The TLSA records and the associated base domain are derived separately for each MX hostname that is used to attempt message delivery. Clearly, if DANE TLS security is to apply to message delivery via any of the SMTP servers, the MX records must be obtained securely via a DNSSEC validated MX lookup.

MX records MUST be sorted by preference; an MX hostname with a worse (numerically higher) MX preference that has TLSA records MUST NOT preempt an MX hostname with a better (numerically lower) preference that has no TLSA records. In other words, prevention of delivery loops by obeying MX preferences MUST take precedence over channel security considerations. Even with two equal preference MX records, an MTA is not obligated to choose the MX hostname that offers more security. Domains that want secure inbound mail delivery need to ensure that all their SMTP servers and MX records are configured accordingly.

In the language of [RFC5321] Section 5.1, the original next-hop domain is the "initial name". If the MX lookup of the initial name results in a CNAME alias, the MTA replaces the initial name with the resulting name and performs a new lookup with the new name. MTAs typically support recursion in CNAME expansion, so this replacement is performed repeatedly until the ultimate non-CNAME domain is found.

If the MX RRset (or any CNAME leading to it) is "insecure" (see Section 2.1), DANE TLS does not apply, and delivery proceeds via pre-DANE opportunistic TLS. Otherwise (assuming no DNS errors or "bogus" / "indeterminate" responses), the MX RRset is "secure", and the SMTP client MUST treat each MX hostname as a separate non-MX destination for opportunistic DANE TLS as described in Section 2.2.2. When, for a given MX hostname, no TLSA records are found, or only "insecure" TLSA records are found, DANE TLSA is not applicable with the SMTP server in question and delivery proceeds to that host as with pre-

DANE opportunistic TLS. To avoid downgrade attacks, any errors during TLSA lookups **MUST**, as explained in Section 2.1, cause the SMTP server in question to be treated as unreachable.

2.2.2. Non-MX destinations

This section describes the algorithm used to locate the TLSA records and associated TLSA base domain for an input domain not subject to MX resolution. Such domains include:

- o Each MX hostname used in a message delivery attempt for an original next-hop destination domain subject to MX resolution. Note, MTAs are not obligated to support CNAME expansion of MX hostnames.
- o Any administrator configured relay hostname, not subject to MX resolution. This frequently involves configuration set by the MTA administrator to handle some or all mail.
- o A next-hop destination domain subject to MX resolution that has no MX records. In this case the domain's name is implicitly also the hostname of its sole SMTP server.

Note that DNS queries with type TLSA are mishandled by load balancing nameservers that serve the MX hostnames of some large email providers. The DNS zones served by these nameservers are not signed and contain no TLSA records, but queries for TLSA records fail, rather than returning the non-existence of the requested TLSA records.

To avoid problems delivering mail to domains whose SMTP servers are served by the problem nameservers the SMTP client **MUST** perform any A and/or AAAA queries for the destination before attempting to locate the associated TLSA records. This lookup is needed in any case to determine whether the destination domain is reachable and the DNSSEC validation status of each stage of the chain of CNAME queries required to reach the final result.

If no address records are found, the destination is unreachable. If address records are found, but the DNSSEC validation status of the first query response is "insecure" (there may be additional queries if the initial response is a CNAME alias), the SMTP client SHOULD NOT proceed to search for any associated TLSA records. With the problem domains, TLSA queries will lead to DNS lookup errors and cause messages to be consistently delayed and ultimately returned to the sender. We don't expect to find any "secure" TLSA records associated with a TLSA base domain that lies in an unsigned DNS zone. Therefore, skipping TLSA lookups in this case will also reduce latency with no detrimental impact on security.

If the A and/or AAAA lookup of the "initial name" yields a CNAME, we replace it with the resulting name as if it were the initial name and perform a lookup again using the new name. This replacement is performed recursively.

We consider the following cases for handling a DNS response for an A or AAAA DNS lookup:

Not found: When the DNS queries for A and/or AAAA records yield neither a list of addresses nor a CNAME (or CNAME expansion is not supported) the destination is unreachable.

Non-CNAME: The answer is not a CNAME alias. If the address RRset is "secure", TLSA lookups are performed as described in Section 2.2.3 with the initial name as the candidate TLSA base domain. If no "secure" TLSA records are found, DANE TLS is not applicable and mail delivery proceeds with pre-DANE opportunistic TLS (which, being best-effort, degrades to cleartext delivery when STARTTLS is not available or the TLS handshake fails).

Insecure CNAME: The input domain is a CNAME alias, but the ultimate network address RRset is "insecure" (see Section 2.1). If the initial CNAME response is also "insecure", DANE TLS does not apply. Otherwise, this case is treated just like the non-CNAME case above, where a search is performed for a TLSA record with the original input domain as the candidate TLSA base domain.

Secure CNAME: The input domain is a CNAME alias, and the ultimate network address RRset is "secure" (see Section 2.1). Two candidate TLSA base domains are tried: the fully CNAME-expanded initial name and, failing that, then the initial name itself.

In summary, if it is possible to securely obtain the full, CNAME-expanded, DNSSEC-validated address records for the input domain, then that name is the preferred TLSA base domain. Otherwise, the unexpanded input-MX domain is the candidate TLSA base domain. When

no "secure" TLSA records are found at either the CNAME-expanded or unexpanded domain, then DANE TLS does not apply for mail delivery via the input domain in question. And, as always, errors, bogus or indeterminate results for any query in the process MUST result in delaying or abandoning delivery.

2.2.3. TLSA record lookup

Each candidate TLSA base domain (the original or fully CNAME-expanded name of a non-MX destination or a particular MX hostname of an MX destination) is in turn prefixed with service labels of the form "_<port>._tcp". The resulting domain name is used to issue a DNSSEC query with the query type set to TLSA ([RFC6698] Section 7.1).

For SMTP, the destination TCP port is typically 25, but this may be different with custom routes specified by the MTA administrator in which case the SMTP client MUST use the appropriate number in the "_<port>" prefix in place of "_25". If, for example, the candidate base domain is "mail.example.com", and the SMTP connection is to port 25, the TLSA RRset is obtained via a DNSSEC query of the form:

```
_25._tcp.mail.example.com. IN TLSA ?
```

The query response may be a CNAME, or the actual TLSA RRset. If the response is a CNAME, the SMTP client (through the use of its security-aware stub resolver) restarts the TLSA query at the target domain, following CNAMEs as appropriate and keeping track of whether the entire chain is "secure". If any "insecure" records are encountered, or the TLSA records don't exist, the next candidate TLSA base is tried instead.

If the ultimate response is a "secure" TLSA RRset, then the candidate TLSA base domain will be the actual TLSA base domain and the TLSA RRset will constitute the TLSA records for the destination. If none of the candidate TLSA base domains yield "secure" TLSA records then delivery should proceed via pre-DANE opportunistic TLS.

TLSA record publishers may leverage CNAMEs to reference a single authoritative TLSA RRset specifying a common certificate authority or a common end entity certificate to be used with multiple TLS services. Such CNAME expansion does not change the SMTP client's notion of the TLSA base domain; thus, when _25._tcp.mail.example.com is a CNAME, the base domain remains mail.example.com and is still the name used in peer certificate name checks.

Note, shared end entity certificate associations expose the publishing domain to substitution attacks, where an MITM attacker can reroute traffic to a different server that shares the same end entity

certificate. Such shared end entity records should be avoided unless the servers in question are interchangeable.

For example, given the DNSSEC validated records below:

```
example.com.           IN MX 0 mail.example.com.
example.com.           IN MX 0 mail2.example.com.
_25._tcp.mail.example.com. IN CNAME tlsa211._dane.example.com.
_25._tcp.mail2.example.com. IN CNAME tlsa211._dane.example.com.
tlsa211._dane.example.com. IN  TLSA 2 1 1 e3b0c44298fc1c14....
```

The SMTP servers mail.example.com and mail2.example.com will be expected to have certificates issued under a common trust anchor, but each MX hostname's TLSA base domain remains unchanged despite the above CNAME records. Each SMTP server's certificate subject name (or one of the subject alternative names) is expected to match either the corresponding MX hostname or else "example.com".

If, during TLSA resolution (including possible CNAME indirection), at least one "secure" TLSA record is found (even if not usable because it is unsupported by the implementation or support is administratively disabled), then the corresponding host has signaled its commitment to implement TLS. The SMTP client SHOULD NOT deliver mail via the corresponding host unless a TLS session is negotiated via STARTTLS. This is required to avoid MITM STARTTLS downgrade attacks.

As noted previously (in Section 2.2.2), when no "secure" TLSA records are found at the fully CNAME-expanded name, the original unexpanded name MUST be tried instead. This supports customers of hosting providers where the provider's zone cannot be validated with DNSSEC, but the customer has shared appropriate key material with the hosting provider to enable TLS via SNI. Intermediate names that arise during CNAME expansion that are neither the original, nor the final name, are never candidate TLSA base domains, even if "secure".

2.3. DANE authentication

This section describes which TLSA records are applicable to SMTP opportunistic DANE TLS and how to apply such records to authenticate the SMTP server. With opportunistic DANE TLS, both the TLS support implied by the presence of DANE TLSA records and the verification parameters necessary to authenticate the TLS peer are obtained together, therefore authentication via this protocol is expected to be less prone to connection failure caused by incompatible configuration of the client and server.

2.3.1. TLSA certificate usages

The DANE TLSA specification [RFC6698] 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]. The rest of the TLSA record is the "certificate association data field", which specifies the full or digest value of a certificate or public key. The 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 public PKI issued and domain-issued certificates.

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.

The certificate usage element of a TLSA record plays a critical role in determining how the corresponding certificate association data field is used to authenticate server's certificate chain. The next two subsections explain the process for certificate usages DANE-EE(3) and DANE-TA(2). The third subsection briefly explains why certificate usages PKIX-TA(0) and PKIX-EE(1) are not applicable with opportunistic DANE TLS.

2.3.1.1. Certificate usage DANE-EE(3)

Since opportunistic DANE TLS will be used by non-interactive MTAs, with no user to "press OK" when authentication fails, reliability of peer authentication is paramount.

Authentication via certificate usage DANE-EE(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 DANE-EE(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 (SNI is not required when the TLSA record matches the server's default certificate).

Two TLSA records MUST 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 DNS caches to expire the previous TLSA RRset and related signature RRsets, the server may be reconfigured to use the new private key and associated public key certificate. Once the server is using the new key, the TLSA RR that matches the retired key can be removed from DNS, leaving only the RR that matches the new key.

TLSA records published for SMTP servers SHOULD, in most cases, be "DANE-EE(3) SPKI(1) SHA2-256(1)" records. 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.

2.3.1.2. Certificate usage DANE-TA(2)

Some domains may prefer to avoid the operational complexity of publishing unique TLSA RRs 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 query domain (TLSA base domain with port

and protocol prefix labels) for each service issued by the same TA may then be set to a CNAME alias that points to a common TLSA RRset that matches the TA.

SMTP servers that rely on certificate usage DANE-TA(2) TLSA records for TLS authentication MUST include the TA certificate as part of the certificate chain presented in the TLS handshake server certificate message even when it is a self-signed root certificate. At this time, many SMTP servers are not configured with a comprehensive list of trust anchors, nor are they expected to at any point in the future. Some MTAs will ignore all locally trusted certificates when processing usage DANE-TA(2) TLSA records. Thus even when the TA happens to be a public Certificate Authority known to the SMTP client, authentication is likely to fail unless the TA is included in the TLS server certificate message.

TLSA Publishers should publish either "DANE-TA(2) SPKI(1) Full(0)" or "DANE-TA(2) Cert(0) SHA2-256(1)" TLSA parameters. As with leaf certificate rollover discussed in Section 2.3.1.1, two such TLSA RRs need to be published to facilitate TA certificate rollover.

2.3.1.3. Certificate usages PKIX-TA(0) and PKIX-EE(1)

SMTP servers SHOULD NOT publish TLSA RRs with certificate usage "PKIX-TA(0)" or "PKIX-EE(1)". SMTP clients cannot be expected to be configured with a suitably complete set of trusted public CAs. Even with a full set of public CAs, SMTP clients cannot (without relying on DNSSEC for secure MX records and DANE for STARTTLS support signalling) perform [RFC6125] server identity verification or prevent STARTTLS downgrade attacks. The use of trusted public CAs offers no added security since an attacker capable of compromising DNSSEC is free to replace any PKIX-TA(0) or PKIX-EE(1) TLSA records with records bearing any convenient non-PKIX certificate usage.

SMTP client treatment of TLSA RRs with certificate usages "PKIX-TA(0)" or "PKIX-EE(1)" is undefined. For example, clients MAY (will likely) treat such TLSA records as unusable.

2.3.2. Certificate matching

When at least one usable "secure" TLSA record is found, the SMTP client SHOULD use TLSA records to authenticate the SMTP server. Messages SHOULD NOT be delivered via the SMTP server if authentication fails, otherwise the SMTP client is vulnerable to MITM attacks.

To match a server via a TLSA record with certificate usage DANE-TA(2), the client MUST perform name checks to ensure that it has

reached the correct server. In all cases the SMTP client MUST accept the TLSA base domain as a valid DNS name in the server certificate.

TLSA records for MX hostnames: If the TLSA base domain was obtained indirectly via an MX lookup (including any CNAME-expanded name of an MX hostname), then the original next-hop domain used in the MX lookup MUST be accepted in the peer certificate. The CNAME-expanded original next-hop domain MUST also be accepted if different from the initial query name.

TLSA records for Non-MX hostnames: If MX records were not used (e.g., if none exist) and the TLSA base domain is the CNAME-expanded original next-hop domain, then the original next-hop domain MUST also be accepted.

Accepting certificates with the original next-hop domain in addition to the MX hostname allows a domain with multiple MX hostnames to field a single certificate bearing a single domain name (i.e., the email domain) across all the SMTP servers. This also aids interoperability with pre-DANE SMTP clients that are configured to look for the email domain name in server certificates. For example, with "secure" DNS records as below:

```
exchange.example.org.      IN CNAME mail.example.org.
mail.example.org.          IN CNAME example.com.
example.com.               IN MX      10 mx10.example.com.
example.com.               IN MX      15 mx15.example.com.
example.com.               IN MX      20 mx20.example.com.
;
mx10.example.com.          IN A        192.0.2.10
_25._tcp.mx10.example.com. IN TLSA    2 0 1 ...
;
mx15.example.com.          IN CNAME mxbackup.example.com.
mxbackup.example.com.      IN A        192.0.2.15
; _25._tcp.mxbackup.example.com. IN TLSA ? (NXDOMAIN)
_25._tcp.mx15.example.com. IN TLSA    2 0 1 ...
;
mx20.example.com.          IN CNAME mxbackup.example.net.
mxbackup.example.net.      IN A        198.51.100.20
_25._tcp.mxbackup.example.net. IN TLSA    2 0 1 ...
```

Certificate name checks for delivery of mail to exchange.example.org via any of the associated SMTP servers MUST accept at least the names "exchange.example.org" and "example.com", which are respectively the original and fully expanded next-hop domain. When the SMTP server is mx10.example.com, name checks MUST accept the TLSA base domain "mx10.example.com". If, despite the fact that MX hostnames are required to not be aliases, the MTA supports delivery via

"mx15.example.com" or "mx20.example.com" then name checks MUST accept the respective TLSA base domains "mx15.example.com" and "mxbackup.example.net".

The SMTP client MUST NOT perform certificate usage name checks with certificate usage DANE-EE(3), since with usage DANE-EE(3) the server is authenticated directly by matching the TLSA RRset to its certificate or public key without resorting to any issuing authority. The certificate content is ignored except to match the certificate or public key (ASN.1 DER encoding or its digest) with the TLSA RRset.

To ensure that the server sends the right certificate chain, the SMTP client MUST send the TLS SNI extension containing the TLSA base domain. This precludes the use of the backward compatible SSL 2.0 compatible SSL HELLO by the SMTP client. The minimum SSL/TLS client HELLO version for SMTP clients performing DANE authentication is SSL 3.0, but a client that offers SSL 3.0 MUST also offer at least TLS 1.0 and MUST include the SNI extension. Servers that don't make use of SNI MAY negotiate SSL 3.0 if offered by the client.

Each SMTP server MUST present a certificate chain (see [RFC5246] Section 7.4.2) that matches at least one of the TLSA records. The server MAY rely on SNI to determine which certificate chain to present to the client. Clients that don't send SNI information may not see the expected certificate chain.

If the server's TLSA RRset includes records with a matching type indicating a digest record (i.e., a value other than Full(0)), a TLSA record with a SHA2-256(1) matching type SHOULD be provided along with any other digest published, since some SMTP clients may support only SHA2-256(1).

If the server's TLSA records match the server's default certificate chain, the server need not support SNI. In either case, the server need not include the SNI extension in its TLS HELLO as simply returning a matching certificate chain is sufficient. Servers MUST NOT enforce the use of SNI by clients, as the client may be using unauthenticated opportunistic TLS and may not expect any particular certificate from the server. If the client sends no SNI extension, or sends an SNI extension for an unsupported domain, the server MUST simply send its default certificate chain. The reason for not enforcing strict matching of the requested SNI hostname is that DANE TLS clients are typically willing to accept multiple server names, but can only send one name in the SNI extension. The server's default certificate may match a different name acceptable to the client, e.g., the original next-hop domain.

An SMTP client employing pre-DANE opportunistic TLS MAY include some anonymous TLS cipher suites in its TLS HELLO in addition to at least one non-anonymous cipher suite (since servers often do support any of the anonymous ones). Therefore, an SMTP server MUST either select some suitable non-anonymous cipher suite offered by the client, or if it selects an anonymous cipher suite, it MUST NOT fail to complete the handshake merely because an anonymous cipher suite was chosen.

Note that while SMTP server operators are under no obligation to enable anonymous cipher suites, no security is gained by sending certificates to clients that will ignore them. Indeed support for anonymous cipher suites in the server makes audit trails more informative. Log entries that record connections that employed an anonymous cipher suite record the fact that the clients did not care to authenticate the server.

2.3.3. Digest algorithm agility

While [RFC6698] specifies multiple digest algorithms, it does not specify a protocol by which the SMTP client and TLSA record publisher can agree on the strongest shared algorithm. Such a protocol would allow the client and server to avoid exposure to any deprecated weaker algorithms that are published for compatibility with less capable clients, but should be ignored when possible. We specify such a protocol below.

Suppose that a DANE TLS client authenticating a TLS server considers digest algorithm BETTER stronger than digest algorithm WORSE. Suppose further that a server's TLSA RRset contains some records with BETTER as the digest algorithm. Finally, suppose that for every raw public key or certificate object that is included in the server's TLSA RRset in digest form, whenever that object appears with algorithm WORSE with some usage and selector it also appears with algorithm BETTER with the same usage and selector. In that case our client can safely ignore TLSA records with the weaker algorithm WORSE, because it suffices to check the records with the stronger algorithm BETTER.

Server operators MUST ensure that for any given usage and selector, each object (certificate or public key), for which a digest association exists in the TLSA RRset, is published with the SAME SET of digest algorithms as all other objects that published with that usage and selector. In other words, for each usage and selector, the records with non-zero matching types will correspond to on a cross-product of a set of underlying objects and a fixed set of digest algorithms that apply uniformly to all the objects.

To achieve digest algorithm agility, all published TLSA RRsets for use with opportunistic DANE TLS for SMTP MUST conform to the above requirements. Then, for each combination of usage and selector, SMTP clients can simply ignore all digest records except those that employ the strongest digest algorithm. The ordering of digest algorithms by strength is not specified in advance, it is entirely up to the SMTP client. SMTP client implementations SHOULD make the digest algorithm preference order configurable. Only the future will tell which algorithms might be weakened by new attacks and when.

Note, TLSA records with a matching type of Full(0), that publish the full value of a certificate or public key object, play no role in digest algorithm agility. They neither trump the processing of records that employ digests, nor are they ignored in the presence of any records with a digest (i.e. non-zero) matching type.

SMTP clients SHOULD use digest algorithm agility when processing the DANE TLSA records of an SMTP server. Algorithm agility is to be applied after first discarding any unusable or malformed records (unsupported digest algorithm, or incorrect digest length). Thus, for each usage and selector, the client SHOULD process only any usable records with a matching type of Full(0) and the usable records whose digest algorithm is believed to be the strongest among usable records with the given usage and selector.

The main impact of this requirement is on key rotation, when the TLSA RRset is pre-populated with digests of new certificates or public keys, before these replace or augment their predecessors. Were the newly introduced RRs to include previously unused digest algorithms, clients that employ this protocol could potentially ignore all the digests corresponding to the current keys or certificates, causing connectivity issues until the new keys or certificates are deployed. Similarly, publishing new records with fewer digests could cause problems for clients using cached TLSA RRsets that list both the old and new objects once the new keys are deployed.

To avoid problems, server operators SHOULD apply the following strategy:

- o When changing the set of objects published via the TLSA RRset (e.g. during key rotation), DO NOT change the set of digest algorithms used; change just the list of objects.
- o When changing the set of digest algorithms, change only the set of algorithms, and generate a new RRset in which all the current objects are re-published with the new set of digest algorithms.

After either of these two changes are made, the new TLSA RRset should be left in place long enough that the older TLSA RRset can be flushed from caches before making another change.

3. Mandatory TLS Security

An MTA implementing this protocol may require a stronger security assurance when sending email to selected destinations. The sending organization may need to send sensitive email and/or may have regulatory obligations to protect its content. This protocol is not in conflict with such a requirement, and in fact can often simplify authenticated delivery to such destinations.

Specifically, with domains that publish DANE TLSA records for their MX hostnames, a sending MTA can be configured to use the receiving domains's DANE TLSA records to authenticate the corresponding SMTP server. Authentication via DANE TLSA records is easier to manage, as changes in the receiver's expected certificate properties are made on the receiver end and don't require manually communicated configuration changes. With mandatory DANE TLS, when no usable TLSA records are found, message delivery is delayed. Thus, mail is only sent when an authenticated TLS channel is established to the remote SMTP server.

Administrators of mail servers that employ mandatory DANE TLS, need to carefully monitor their mail logs and queues. If a partner domain unwittingly misconfigures their TLSA records, disables DNSSEC, or misconfigures SMTP server certificate chains, mail will be delayed.

4. Operational Considerations

4.1. Client Operational Considerations

SMTP clients may deploy opportunistic DANE TLS incrementally by enabling it only for selected sites, or may occasionally need to disable opportunistic DANE TLS for peers that fail to interoperate due to misconfiguration or software defects on either end. Unless local policy specifies that opportunistic DANE TLS is not to be used for a particular destination, an SMTP client MUST NOT deliver mail via a server whose certificate chain fails to match at least one TLSA record when usable TLSA records are found for that server.

4.2. Publisher Operational Considerations

SMTP servers that publish certificate usage DANE-TA(2) associations MUST include the TA certificate in their TLS server certificate chain, even when that TA certificate is a self-signed root certificate.

TLSA Publishers must follow the digest agility guidelines in Section 2.3.3 and must make sure that all objects published in digest form for a particular usage and selector are published with the same set of digest algorithms.

TLSA Publishers should follow the TLSA publication size guidance found in [I-D.ietf-dane-ops] about "DANE DNS Record Size Guidelines".

5. Security Considerations

This protocol leverages DANE TLSA records to implement MITM resistant opportunistic channel security for SMTP. For destination domains that sign their MX records and publish signed TLSA records for their MX hostnames, this protocol allows sending MTAs to securely discover both the availability of TLS and how to authenticate the destination.

This protocol does not aim to secure all SMTP traffic, as that is not practical until DNSSEC and DANE adoption are universal. The incremental deployment provided by following this specification is a best possible path for securing SMTP. This protocol coexists and interoperates with the existing insecure Internet email backbone.

The protocol does not preclude existing non-opportunistic SMTP TLS security arrangements, which can continue to be used as before via manual configuration with negotiated out-of-band key and TLS configuration exchanges.

Opportunistic SMTP TLS depends critically on DNSSEC for downgrade resistance and secure resolution of the destination name. If DNSSEC is compromised, it is not possible to fall back on the public CA PKI to prevent MITM attacks. A successful breach of DNSSEC enables the attacker to publish TLSA usage 3 certificate associations, and thereby bypass any security benefit the legitimate domain owner might hope to gain by publishing usage 0 or 1 TLSA RRs. Given the lack of public CA PKI support in existing MTA deployments, avoiding certificate usages 0 and 1 simplifies implementation and deployment with no adverse security consequences.

Implementations must strictly follow the portions of this specification that indicate when it is appropriate to initiate a non-authenticated connection or cleartext connection to a SMTP server. Specifically, in order to prevent downgrade attacks on this protocol, implementation must not initiate a connection when this specification indicates a particular SMTP server must be considered unreachable.

6. IANA considerations

This specification requires no support from IANA.

7. Acknowledgements

The authors would like to extend great thanks to Tony Finch, who started the original version of a DANE SMTP document. His work is greatly appreciated and has been incorporated into this document. The authors would like to additionally thank Phil Pennock for his comments and advice on this document.

Acknowledgments from Viktor: Thanks to Paul Hoffman who motivated me to begin work on this memo and provided feedback on early drafts. Thanks to Patrick Koetter, Perry Metzger and Nico Williams for valuable review comments. Thanks also to Wietse Venema who created Postfix, and whose advice and feedback were essential to the development of the Postfix DANE implementation.

8. References

8.1. Normative References

- [I-D.ietf-dane-ops] Dukhovni, V. and W. Hardaker, "DANE TLSA implementation and operational guidance", draft-ietf-dane-ops-02 (work in progress), January 2014.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, February 2002.
- [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.
- [RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, October 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.
- [RFC6186] Daboo, C., "Use of SRV Records for Locating Email Submission/Access Services", RFC 6186, March 2011.
- [RFC6672] Rose, S. and W. Wijngaards, "DNAME Redirection in the DNS", RFC 6672, June 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.

8.2. 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-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.
- [RFC5598] Crocker, D., "Internet Mail Architecture", RFC 5598, July 2009.
- [RFC6409] Gellens, R. and J. Klensin, "Message Submission for Mail", STD 72, RFC 6409, November 2011.

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

DNS-Based Authentication of Named Entities (DANE)
Internet-Draft

Intended status: Standards Track

Expires: August 17, 2014

T. Finch
University of Cambridge

M. Miller

Cisco Systems, Inc.

P. Saint-Andre

&yet

February 13, 2014

Using DNS-Based Authentication of Named Entities (DANE) TLSA records
with SRV and MX records.
draft-ietf-dane-srv-05

Abstract

The DANE specification (RFC 6698) 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 use SRV records (RFC 2782) 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 when technologies such as SRV records are used. Separate documents give the details that are specific to particular application protocols.

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 <http://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 August 17, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. Relation between SRV and MX records	3
4. DNS Checks for TLSA and SRV Records	4
4.1. SRV Query	4
4.2. TLSA Queries	5
5. TLS Checks for TLSA and SRV Records	6
6. Guidance for Application Protocols	7
7. Guidance for Server Operators	7
8. Internationalization Considerations	8
9. IANA Considerations	8
10. Security Considerations	8
10.1. Mixed Security Status	8
10.2. A Service Domain Trusts its Servers	8
10.3. Certificate Subject Name Matching	9
11. Acknowledgements	9
12. References	9
12.1. Normative References	9
12.2. Informative References	10
Appendix A. Mail Example	11
Appendix B. XMPP Example	11
Appendix C. Rationale	12
Authors' Addresses	13

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; instead, they use a service domain and the relevant host names are located indirectly via SRV records [RFC2782], or MX records in the case of

SMTP [RFC5321] (Note: in the "CertID" specification [RFC6125], the source domain and host name are referred to as the "source domain" and the "derived domain"). Because of this intermediate resolution step, the normal DANE rules specified in [RFC6698] do not directly apply to protocols that use SRV or MX records.

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 host names that are discovered by the SRV or MX query).
- o The TLSA records are located using the port, protocol, and target host name fields (not the service domain).
- o Clients always use TLS when connecting to servers with TLSA records.
- o Assuming that the association is secure, the server's certificate is expected to authenticate the target server host name, rather than the service domain.

Separate documents give the details that are specific to particular application protocols, such as SMTP [I-D.ietf-dane-smtp-with-dane] and XMPP [I-D.ietf-xmpp-dna].

2. Terminology

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

This draft uses the definitions for "secure", "insecure", "bogus", and "indeterminate" from [RFC4035]. This draft uses the acronyms from [I-D.ietf-dane-registry-acronyms] for the values of TLSA fields where appropriate.

3. Relation between SRV and MX records

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

Table 1: SRV Fields and MX Equivalents

DNS SRV Field	Equivalent MX Value
Service	smtp
Proto	tcp
Name	MX owner name (mail domain)
TTL	MX TTL
Class	MX Class
Priority	MX Priority
Weight	0
Port	25
Target	MX Target

Thus we can treat the following MX record as if it were the SRV record shown below:

```
example.com.          86400 IN MX  10      mx.example.net.
_smtptcp.example.com. 86400 IN SRV 10 0 25 mx.example.net.
```

Other details that are specific to SMTP are described in [I-D.ietf-dane-smtp-with-dane].

4. DNS Checks for TLSA and SRV Records

4.1. SRV Query

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

For this specification to apply, all of these DNS RRsets MUST be "secure" according to DNSSEC validation ([RFC4033] section 5). In the case of aliases, the whole chain of CNAME and DNAME RRsets MUST be secure as well. 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 correctly deployed. The client SHOULD fall back to its non-DNSSEC non-DANE behavior (this corresponds to the AD bit being unset).

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

In the successful case, 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. It takes note of the DNSSEC validation status of the SRV response for use when checking certificate names (see Section 5).

4.2. TLSA Queries

If the SRV response was insecure, the client MUST NOT perform any TLSA queries. If the SRV response is "secure" according to DNSSEC validation, the client performs a TLSA query for each SRV target as described in this section.

For each SRV target host name, the client performs DNSSEC validation on the address (A, AAAA) response and continues based on the results:

- o if the response is "insecure", the client MUST NOT perform a TLSA query for that target; the TLSA query will most likely fail.
- o If the response is "bogus" or "indeterminate", the client MUST NOT connect to this host name; instead it uses the next most appropriate SRV target.

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

For example, the following SRV record leads to the TLSA query shown below:

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

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

The client SHALL determine if the TLSA record(s) are usable according to section 4.1 of [RFC6698]. This affects SRV handling as follows:

If the TLSA response is "secure", the client MUST use TLS when connecting to the server. The TLSA records are used when validating the server's certificate as described under Section 5.

If the TLSA response is "insecure", the client SHALL proceed as if this server has no TLSA records. It MAY connect to the server with or without TLS.

If the TLSA response is "bogus" or "indeterminate", then the client MUST NOT connect to this server (the client can still use other SRV targets).

5. TLS Checks for TLSA and SRV Records

When connecting to a server, the client MUST use TLS if the responses to the SRV and TLSA queries were "secure" as described above. If the client received zero usable TLSA certificate associations, it SHALL validate the server's TLS certificate using the normal PKIX rules [RFC5280] or protocol-specific rules (e.g., following [RFC6125]) without further input from the TLSA records. If the client received one or more usable TLSA certificate associations, it SHALL process them as described in [RFC6698] section 2.1.

If the TLS server's certificate -- or the public key of the server's certificate -- matches a usable TLSA record with Certificate Usage "DANE-EE", the client MUST consider the server to be authenticated. Because the information in such a TLSA record supersedes the non-key information in the certificate, all other [RFC5280] and [RFC6125] authentication checks (e.g., reference identifier, key usage, expiration, issuance, etc.) MUST be ignored or omitted.

Otherwise, the client uses the information in the server certificate and DNSSEC validation status of the SRV query in its authentication checks. It SHOULD use the Server Name Indication extension (TLS SNI) [RFC6066] or its functional equivalent in the relevant application protocol (e.g., in XMPP [RFC6120] this is the 'to' address of the initial stream header). The preferred name SHALL be chosen as follows, and the client SHALL verify the identity asserted by the server's certificate according to [RFC6125] section 6, using a list of reference identifiers constructed as follows (note again that in RFC 6125 the terms "source domain" and "derived domain" refer to the same things as "service domain" and "target host name" in this document).

SRV is insecure: The reference identifiers SHALL include the service domain and MUST NOT include the SRV target host name. The service domain is the preferred name for TLS SNI or its equivalent.

SRV is secure: The reference identifiers SHALL include both the service domain and the SRV target host name. The target host name is the preferred name for TLS SNI or its equivalent.

In the latter case, the client will accept either identity so that it is compatible with servers that do and do not support this specification.

6. Guidance for Application Protocols

Separate documents describe how to apply this specification to particular application protocols. Such documents ought to cover the following points:

- o Fallback logic in the event of bogus replies and the like.
- o The use of TLS SNI or its functional equivalent.
- o Appropriate mappings for non-SRV technologies such as MX.
- o Compatibility with clients that do not support SRV lookups.

7. Guidance for Server Operators

To conform to this specification, the published SRV records and subsequent address (A, AAAA) records MUST be secured with DNSSEC. There SHOULD also be at least one TLSA record published that authenticates the server's certificate.

When using TLSA records with Certificate Usage "DANE-EE", the deployed certificate does not need to contain any of the possible reference identifiers discussed below. Indeed, none of the certificate's information is necessary for such certificates. However, servers that rely solely on validation using Certificate Usage "DANE-EE" TLSA records might prevent clients that do not support this specification from successfully connecting with TLS.

For TLSA records with Certificate Usage types other than "DANE-EE", the certificate(s) MUST contain a reference identifier that matches:

- o the service domain name (the "source domain" in [RFC6125] terms, which is the SRV query domain); and/or

- o the server host name (the "derived domain" in [RFC6125] terms, which is the SRV target).

Servers that support multiple service domains (i.e., multi-tenant) can implement Server Name Indicator (TLS SNI) [RFC6066] or its functional equivalent to determine which certificate to offer. Clients that do not support this specification will indicate a preference for the service domain name, while clients that support this specification will indicate the server host name. However, the server determines what certificate to present in the TLS handshake; e.g., the presented certificate might only authenticate the server host name.

8. Internationalization Considerations

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

9. IANA Considerations

No IANA action is required.

10. Security Considerations

10.1. Mixed Security Status

We do not specify that clients checking 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 fallback 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.

10.2. 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 (where necessary), and publishing a TLSA record for that certificate. If these are not correct then connections from TLSA-aware clients might fail.

10.3. 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 checks are not necessary if the matching TLSA record is of Certificate Usage "DANE-EE". Because such a record identifies the specific certificate (or public key of the certificate), additional checks are superfluous and potentially conflicting.

Otherwise, while DNSSEC provides a secure binding between the server name and the TLSA record, and the TLSA record provides a binding to a certificate, this latter step can be indirect via a chain of certificates. For example, a Certificate Usage "PKIX-TA" TLSA record only authenticates the CA that issued the certificate, and third parties can obtain certificates from the same CA. Therefore, clients need to check whether the server's certificate matches one of the expected reference identifiers to ensure the certificate was issued by the CA to the server the client expects.

11. 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, Viktor Dukhovni, Ned Freed, Olafur Gudmundsson, Paul Hoffman, Phil Pennock, Hector Santos, Jonas Schneider, and Alessandro Vesely for helpful suggestions.

12. References

12.1. Normative 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.
- [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.
- [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.
- [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.
- [RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, January 2011.
- [RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 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.
- [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.

12.2. Informative References

- [I-D.ietf-dane-smtp-with-dane]
Dukhovni, V. and W. Hardaker, "SMTP security via opportunistic DANE TLS", draft-ietf-dane-smtp-with-dane-05 (work in progress), February 2014.
- [I-D.ietf-xmpp-dna]
Saint-Andre, P. and M. Miller, "Domain Name Associations (DNA) in the Extensible Messaging and Presence Protocol (XMPP)", draft-ietf-xmpp-dna-05 (work in progress), February 2014.

Appendix A. Mail 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.   RRSIG   A ...

mx.example.net.   AAAA    2001:db8:212:8::e:1
mx.example.net.   RRSIG   ...

; 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. XMPP Example

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

```
; XMPP domain
_xmpp-client.example.com. SRV      1 0 5222 im.example.net.
_xmpp-client.example.com. RRSIG    SRV ...

; XMPP server host name
im.example.net.   A       192.0.2.3
im.example.net.   RRSIG   A ...

im.example.net.   AAAA    2001:db8:212:8::e:4
im.example.net.   RRSIG   AAAA ...

; TLSA resource record
_5222._tcp.im.example.net.  TLSA    ...
_5222._tcp.im.example.net.  RRSIG   TLSA ...
```

XMPP sessions for addresses at example.com are established at im.example.net. Connections to im.example.net port 5222 that use STARTTLS will get a server certificate that authenticates the name im.example.net.

Appendix C. 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 (so-called "multi-tenanted environments") and scales up more easily to larger numbers of service domains.

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

- o The certificate is part of the server configuration, so it makes sense to associate it with the server host name rather than the service domain.
- o 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.
- o 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.
- o The same TLSA records work with this specification, and with direct connections to the host name in the style of [RFC6698].
- o 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.

- o It is common for SMTP servers to act in multiple roles, for example 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 putting 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 SNI support or its functional equivalent is necessary for backward compatibility.

Authors' Addresses

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/>

Matthew Miller
Cisco Systems, Inc.
1899 Wynkoop Street, Suite 600
Denver, CO 80202
USA

Email: mamille2@cisco.com

Peter Saint-Andre
&yet

Email: ietf@stpeter.im

DANE
Internet-Draft
Intended status: Informational
Expires: August 18, 2014

O. Gudmundsson
Shinkuro Inc.
February 14, 2014

Harmonizing how applications specify DANE-like usage
draft-ogud-dane-vocabulary-02

Abstract

There is no standard terminology as how to talk about use of DNS in various application contexts, this document goal is to facilitate creation of such a vocabulary/taxonomy.

This document started out as proposal for specific word usage for specifications of adding DANE like technology by different protocols/services. DANE is a method for specifying in DNS records acceptable keys/certificates for application servers.

The terms defined in this document should be applicable to all uses of service specification that uses DNS records.

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 <http://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 August 18, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements notation	3
2. Proposed Terms	3
2.1. DNS Navigation Records	3
2.2. DNS Integrity	4
2.3. Service Specification Records (SSR)	4
2.4. Service Address Records (SAR)	5
2.5. Application Authentication Records (AAR)	6
2.6. Offered Name: Name used when indirection records are used	6
3. Example specification	7
4. IANA considerations	7
5. Security considerations	7
6. Internationalization Considerations	8
7. Acknowledgements	8
8. References	8
8.1. Normative References	8
8.2. Informative References	8
Appendix A. Document history	9
Author's Address	9

1. Introduction

DNS [RFC1034] is being used by many protocols to express where services are located on the internet, today there is no good way to express exactly what people have in mind when specifying a new service/protocol exactly and in concise manner how the service is looked up in the DNS.

DANE [RFC6698] is a powerful new way to provide/amend how authentication/authorization/confidentiality of a connection to a server can be protected by leveraging DNSSEC [RFC4033] [RFC4034] [RFC4035] for the establishment of TLS connection [RFC5246] [RFC6347] which in many cases uses PKIX [RFC5280]. All of these technologies are complicated. People familiar with one or two are not necessarily familiar with all the parts that needed to apply DANE like mechanism to other protocols.

The goal of this document is three fold:

- o To provide common vocabulary for usage of DNS records in service specification.
- o To provide an overview of the non protocol specific parts needed to specify an DANE like addition.
- o To provide a common framework for such specifications making it easy to review/compare the specifications. An important goal is to allow the new specifications to avoid repeating explanations and/or definitions.

Number of RFC's in the past have tried to use consistent terminology when specifying how to access services both in the context of security TLS with X.509 [RFC6125] and without security [RFC2782]. The terminology in this document is not identical but concepts are similar. The hope is that once the standard terminology is specified, as simple documents can provide a mapping if one is needed.

This version of the document aims to hide complexity and focus on generalities. This is done to make it easier for the reader to decide if the terms here are of use and if it is worthwhile for the DANE WG to adopt this document. Descriptions of complexities can be added in later versions if the WG decides that is needed.

When notation "foo/bar" is used below that is because the editor is not sure if both apply or which one is more appropriate, please advise.

1.1. Requirements notation

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].

2. Proposed Terms

The terms below are being proposed to avoid confusion when reading protocol specifications related to DNS and DANE, for various application protocols.

At this point all the terms below are proposals and better terms are welcome.

2.1. DNS Navigation Records

DNS Navigation refers to any records used to traverse the DNS tree to find the records requested. This includes

NS records: that provide a referral to DNS servers for more specific part of the name being looked up. Example: name server for "example." will hand out a referral to server for "bar.example." when asked about "foo.bar.example."

CNAME records: records that change the location of an record, this for all practical purposes a pointer that only applies to that specific name.

DNAME records: specify a rewrite rule for a name to a new name. Example: "bar.example." DNAME "foo.example." means that "www.bar.example." is to be looked up as "www.foo.example.". DNAME applies to names that are longer than the name it, i.e. "bar.example." is not rewritten but "www.bar.example." is.

DANE specification explicitly requires all of these records to be validated by DNSSEC.

See section Section 2.2

While traversing the DNS tree other records like A and AAAA are used but these records do not change the "navigation", these records do not explicitly need to be protected as the data retrieved from the addresses is expected to be protected.

2.2. DNS Integrity

DNSSEC defines a records and procedures to provide integrity and authentication to data stored in DNS [RFC4034]. The records used to provide the keying information and chain of trust are DNSKEY, DS records. NSEC/NSEC3 provide information about existence/non-existence of the requested information. RRSIG provides a digital signature for a RRset.

DNSSEC provides both Integrity and Authenticity i.e. it says the records came from the right source and have not been changed.

Any DNS record that is DNS Integrity protected, will pass DNSSEC validation for all DNS Navigation records leading to the name and the record itself also passes DNSSEC validation.

In the case of CNAME and DNAME that go "sideways" i.e. to a different branch of the DNS tree, both branches MUST be validated.

2.3. Service Specification Records (SSR)

Protocols have different ways to express servers.

- o Web servers are frequently specified by name i.e. the "www" prefix, thus its service specification record is: "address record stored at www.<domain>".
- o Email servers have a special RR type (MX): SRR= "MX record at <domain>"),
- o Jabber uses SRV records: SSR="SRV record at _xmpp-server.tcp.<domain>",
- o ENUM uses NAPTR records etc.
- o In addition there are also protocols that use a combination like S-NAPTR a schema where NAPTR records are used to specify where to look for SRV records. For all practical purposes NAPTR + SRV should combined be treated as the Service Specification.

For a DANE like specification it has to be clear as what the service specification records are and these records require DNS Integrity.

NOTE: when a client supplies a string to the server as a indicator of what service the the client wants, the string supplied MAY depend on redirection in DNS navigation as well as results of NAPTR records, etc. See section Section 2.6.

NOTE: when NAPTR records as are used they should be treated same way as DNS Navigation records even though strictly speaking it is the application that evaluates the NAPTR record.

NOTE: When there is a CNAME at the name service is expected to be specified at, that can be either a DNS Navigation record or a Service Specification Record. Protocol specification should provide guidance on interpretation.

2.4. Service Address Records (SAR)

These are the address records for the servers that offer the service.

In some cases the Service Specification records reside at the same name or are the same as the Service Address records. Example: original TLS/DANE[RFC6698], thus both SSR and SAR records are covered by the same DNS integrity rule.

2.5. Application Authentication Records (AAR)

This term refers to the records that provide information about what are acceptable keys or certificates for the servers to offer.

Application Authentication Records MUST be protected by DNS Integrity and each protocol specification MUST explicitly state where/how to look up the Authentication records.

In some cases all the servers for a service will have the same authentication information, in other cases it is going to be on a server by server case. In the first case it is "natural" to store the Authentication records "at" the Service Specification records. In the second case it more natural to store them "at" the Address Records. In this context "at" means the authentication records are stored at name that is an extension of the location example: "_443._tcp.www.example.com" for [RFC6698]. It is possible that neither of these locations is the right one and in that case the specification MUST explicitly express rules as how to find the Authentication Records.

Note: above that there is no a requirement that the Application Address records be covered by DNS Integrity. This is because when the Application Authentication records reside "at" the address records, DNS Integrity is inherited. On the other hand when when Application Authentication Records are stored "at" the Service Specification Record, DNS Integrity for the address records is optional, as any connection to a bogus/wrong server should fail the Authentication tests performed at connection time.

Note: When a Address record search has a CNAME at or DNAME above, the name queried, where should the Authentication Records reside ? With CNAME or with final address record ?

2.6. Offered Name: Name used when indirection records are used

In many protocols one of the first items presented by the application is a <name> that is "related to"/"derived from" the original query name. When DNAME is used the name queried for might be required to be rewritten into a new name.

To disambiguate these cases following prefix terms are defined. Similar rules apply NAPTR + SRV combinations. It is important for many applications to be able to express what name is presented by the application to the server at connection time.

Query: The name the application issued the query for to discover SSR /service.

Final: The name after all the indirection records have been applied.

SRV The name on the SRV record used.

NAPTR The name on the first NAPTR record used, prefix with Final if that is the one wanted.

Intermediate A particular location in the indirection chain. The specification needs to handle this case if it ever occurs.

NOTE: not sure this is needed???ogud???

3. Example specification

This section is an short example for a protocol that is like SSH [RFC4253] we will call this protocol HISS. This is not an actual full specification, just here to give an idea of how to go about extending DANE-like to a random protocol using the terminology from this document.

Location of HISS protocol DNS records:

Service Specification Records:

HISS uses address records as the service specification record. This record MUST have "DNS Integrity" as explained in RFC-to-be-this-document. CNAME/DNAME are treated as a DNS Navigation record.

Service Address Records:

see: Service Specification Records.

Application Authentication Records:

The protocol uses the DNS HISSFP that is stored at the same name as the service is specified. The HISSFP record, if present, takes precedence over keys stored in client cache.

Offered Name

Not used.

The HISS protocol and HISSFP DNS RR do not exist

4. IANA considerations

None

[RFC Editor: Please remove this section before publication]

5. Security considerations

This documents goal is to improve specifications of adding security via DANE technology to protocols, thus the overwriting goal is to decrease confusion and increase clarity, with the end goal of improving security. This document does not specify a protocol. XX
Needs more work XX

6. Internationalization Considerations

When selecting terms to use in standards documents it is important to select words that do not confuse international readers. This document goes out of its way in selecting English terms that are dissimilar to avoid confusions.

7. Acknowledgements

Number of people have commented that this is interesting work. Peter Saint-Andre tried to apply the terms to one of his documents and provided many good 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.
- [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

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.
- [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.
- [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.
- [RFC4253] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, January 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.

Appendix A. Document history

[RFC Editor: Please remove this section before publication]

02 Textual improvements, applied comments from Peter Saint-Andre.

01 Added definition of offered names, expanded DNAME/CNAME text added NAPTR and SRV.

00 Initial version

Author's Address

Olafur Gudmundsson
Shinkuro Inc.
4922 Fairmont Av, Suite 250
Bethesda, MD 20814
USA

Email: ogud@ogud.com

DANE
Internet-Draft
Intended status: Standards Track
Expires: August 17, 2014

E. Osterweil
G. Wiley
VeriSign, Inc.
D. Mitchell
Twitter
A. Newton
American Registry for Internet
Numbers
February 13, 2014

Opportunistic Encryption with DANE Semantics and IPsec: IPSECA
draft-osterweil-dane-ipsec-00

Abstract

The query/response transactions of the Domain Name System (DNS) can disclose valuable meta-data about the online activities of DNS' users. The DNS Security Extensions (DNSSEC) provide object-level security, but do not attempt to secure the DNS transaction itself. For example, DNSSEC does not protect against information leakage, and only protects DNS data until the last validating recursive resolver. Stub resolvers are vulnerable to adversaries in the network between themselves and their validating resolver ("the last mile"). This document details a new DANE-like DNS Resource Record (RR) type called IPSECA, and explains how to use it to bootstrap DNS transactions through informing entries in IPsec Security Policy Databases (SPDs) and to subsequently verifying Security Associations (SAs) for OE IPsec tunnels.

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 <http://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 August 17, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. What IPSECA Adds to DNSSEC Transactions	4
1.2. IP-Centric IPsec Tunnel Discovery Using IPSECKEY	4
1.3. Service-Centric IPsec Tunnel Discovery Using IPSECA and DANE	5
2. The IPSECA Resource Record	6
2.1. IPSECA RDATA Wire Format	7
2.1.1. The Usage Field	7
2.1.2. The Selector Field	7
2.1.3. The Matching Field	8
2.1.4. The Gateway Field	8
2.1.5. The Certificate Association Data Field	9
2.2. IPSECA RR Presentation Format	9
2.3. Domain Names used for IPSEC Records	9
2.4. IPSECA RR Examples	10
2.4.1. OE to a DNS Name Server Example	10
3. Operational Considerations	11
4. IANA Considerations	12
5. Security Considerations	12
5.1. Interactions	12
5.2. Last Mile Security Analysis	13
6. Acknowledgements	14
7. References	14
7.1. Normative References	14
7.2. Informative References	15
Appendix A. Name Server OE Configuration Example	16
Appendix B. Recursive Resolver OE Configuration Example	16
Authors' Addresses	17

1. Introduction

The query/response transactions of the Domain Name System (DNS) [RFC1035] can disclose valuable meta-data about the online activities of DNS' users. The DNS Security Extensions' (DNSSEC's) [RFC4033], [RFC4034], [RFC4035] core services (integrity, source authenticity, and secure denial of existence) are designed to secure data in DNS transactions by providing object-level security, but do not attempt to secure the DNS transaction itself. For example, DNSSEC does not attempt to protect the confidentiality of DNS transactions, does not protect data outside of the RRsets (including the DNS header, OPT record, etc.), and its DNS-specific protections expose opportunities for adversaries to identify DNS traffic, eavesdrop on DNS messages, and target DNS and its meta-data for attacks. As a result, a clever adversary may target just DNS traffic, discover the nature of a user's online browsing (from fully qualified domain names), interfere with the delivery of specific messages (though the DNS objects are not forgeable), or even attack "the last mile," between a resolver and a remote validating recursive resolver.

For example, the information leakage exposed by observing DNSSEC transactions, could enable an adversary to not only learn what Second Level Domains (SLDs) a user is querying (such as their bank, a funding agency, a security contractor, etc.), but could also inspect the fully qualified domain name(s) to learn the specific hosts visited, or in the case of certain DNS-based chat programs, information about ongoing conversations.

In addition, DNSSEC's design only protects DNS data until the last validating recursive resolver. If a client issues DNS queries from a stub resolver to a remote DNSSEC-aware resolver, then the network between these two ("the last mile") can be leveraged by an adversary to spoof responses, drop traffic, etc.

Clearly, these limitations do not invalidate the benefits of DNSSEC. DNSSEC still protects the actual DNS objects, protects against cache poisoning attacks, and more. Rather, these limitations simply illustrate that there is more at stake than just valid DNS data.

This document details the motivation for, the synergy from, and a protocol to advertise and verify security credentials that can be used to verify Opportunistic Encryption (OE) IPsec [RFC4301], [RFC6071] tunnels for DNS transactions. Securing DNS transactions in this way is both necessary and sufficient for providing confidentiality of many types of DNS-transaction meta data, which can betray user privacy. This document details a new DANE-like [RFC6698] DNS Resource Record (RR) type called IPSECA, and explains how to use it to bootstrap entries in IPsec Security Policy Databases (SPDs) and

to subsequently verify Security Associations (SAs) for OE IPsec tunnels.

1.1. What IPSECA Adds to DNSSEC Transactions

DNSSEC's focus on object level security leaves the types of protections offered by IPsec unaddressed. Specifically, the way (or ways) to associate certificate(s) used by IPsec with a DNSSEC-aware name server need to be codified. This can be especially complicated if different IPsec certificates need to be discovered for different services that are running on the same IP address. This can become complicated if certificates are learned solely by the IP addresses of networked-services. This gap is inherently overcome during certificate discovery in DANE protocols by the concept of "Service Address Records," [I-D.draft-ogud-dane-vocabulary]. These Security Associations are defined by, and discovered by, domain names rather than just IP addresses. [RFC6698] standardizes a way for security associations of certificates to be made with service domains for TLS, rather than just IP addresses. As one of the underlying facilities of DANE's approach to certificate verification, this adds a necessary enhancement to IPsec certificate learning over approaches that are based solely on IP addresses in DNS (such as described in [RFC4025] and [RFC4322]).

The advantages of using DANE for IPsec OE also include other simplifications that the DANE protocol inherently offers all of its protocols. Such as, the automatic deauthorization of certificates that happens when they are removed from a DNS zone, which may (under many circumstances) obviate the need for extensive use of revocation mechanisms (OCSP [RFC6960] or CRL [RFC5280]). Details of these relative trade offs is described in more detail in [DANE_SATIN12].

It is also noteworthy that DANE offers flexibility that is not available in IP-centric certificate discovery and IP-centric OE [RFC4322], while still being backwards compatible with them. That is, while users can use IPSECA records to map OE IPsec tunnels to service names, they can also use IPSECA records in their reverse DNS zone in a similar fashion to the IPSECKEY [RFC4025] record used in [RFC4322]. However, while this document illustrates an example usage of DANE with IPsec OE, any specification for how the IPSECA resource record MUST get used with OE is beyond the scope of this document.

1.2. IP-Centric IPsec Tunnel Discovery Using IPSECKEY

In contrast to a DANE-centric discovery, [RFC4025] specifies a DNS resource record called IPSECKEY. The IPsec certificate learning described therein prescribes that relying parties learn the intended usage of IPsec certificates after they locate them in DNS and

retrieve them. The types of information that relying parties learn from IPSECKEY responses include: precedence, gateway type, algorithm, gateway, and possibly the public key. After learning the key and creating the Security Association, the relying party can use techniques like [RFC4322] to initialize an OE IPsec tunnel.

The inherent key learning and verification technique in [RFC4322] is based on learning tunnels from IP addresses only (IP-centric). Because of this technique's focus on IP-centric learning, operational entities running services on a specific IP address may not have access to annotate the reverse DNS zone for their services (especially if they are shared environments). So, this type of OE may often be a non-starter. One example would be when zones are hosted and/or served by cloud service providers. In this case, customers are almost certainly not allowed to annotate the reverse DNS zone for their providers.

1.3. Service-Centric IPsec Tunnel Discovery Using IPSECA and DANE

The suggested usage of this document is to aid in discovering where OE IPsec tunnels exist, and to act as an out of band verification substrate that can validate the certificates received during IPsec key exchange. For example, if a DNS caching recursive resolver is configured to attempt OE IPsec tunnels to DNS name servers (using a specific key exchange protocol, like [RFC2409], [RFC5996], etc.), then when it receives a referral it SHOULD query name servers for corresponding IPSECA resource records. (we discuss the format of the resource record and domain names below in Section 2). When an IPSECA record is discovered by a resolver, that resolver SHOULD follow its configurations and setup an SPD entry, in order to signal its IPsec layer to attempt to attempt to establish an SA. Note, this document does not specify a new, or any modifications to any existing, IPsec key exchange protocols. Rather, after adding an SPD and after a successful tunnel establishment, the credentials used for the Security Association with the name server SHOULD be cross-checked with the IPSECA resource record(s).

When using IPSECA resource records to verify OE tunnels, clients MUST perform full DNSSEC validation of the DNSSEC chain of trust that leads to IPSECA RRs. As specified in [RFC6698]:

"A [IPSECA] RRSet whose DNSSEC validation state is secure MUST be used as a certificate association for [IPsec] unless a local policy would prohibit the use of the specific certificate association in the secure TLSA RRSet.

If the DNSSEC validation state on the response to the request for the [IPSECA] RRSet is bogus, this MUST cause IPsec not to be

started or, if the IPsec negotiation is already in progress, MUST cause the connection to be aborted.

A [IPSECA] RRSset whose DNSSEC validation state is indeterminate or insecure cannot be used for [IPsec] and MUST be considered unusable."

This is to ensure that the SPD entries and SA(s) used for tunnels are fully verified. This verification MAY include local trust anchor processing, such that local DNSKEY resource records can be used to verify corresponding RRSIGs. Trust anchors (which may be distributed during dynamic host configuration) may be useful for bootstrapping. For example, consider the case where private address space [RFC1918] is used for internal recursive resolvers. Here, the locally provisioned DNS names for the private address space (in the reverse tree) that are secured using DNSSEC MAY use local trust anchors. That is, if an [RFC1918] address is used internally, the corresponding domain name MUST also resolve and be verifiable through DNS and DNSSEC, but a local trust anchor MAY be used to verify covered RRSIGs. This shifts the onus of securing DNS transactions to the initial configuration step. The intuition behind this reasons that if the first (configuration) step was already where the local resolver was configured, then the security of the DNS transactions already hinged on learning the valid resolver this way. So, this step is already used to convey trusted configurations (bootstrapping). Adversaries attempting to subvert an end host have only the narrow attack window that is associated with learning configurations. In contrast, an insecure DNS resolver offers an attack window every time it issues or responds to a query. We discuss this further in Section 5.2.

2. The IPSECA Resource Record

The IPSECA resource record is modeled heavily off of the IPSECKEY RR [RFC4025], but it differs in significant ways. The format of IPSECA is harmonized with the architectural direction set by other DANE work [RFC6698], [I-D.draft-ogud-dane-vocabulary].

2.1. IPSECA RDATA Wire Format

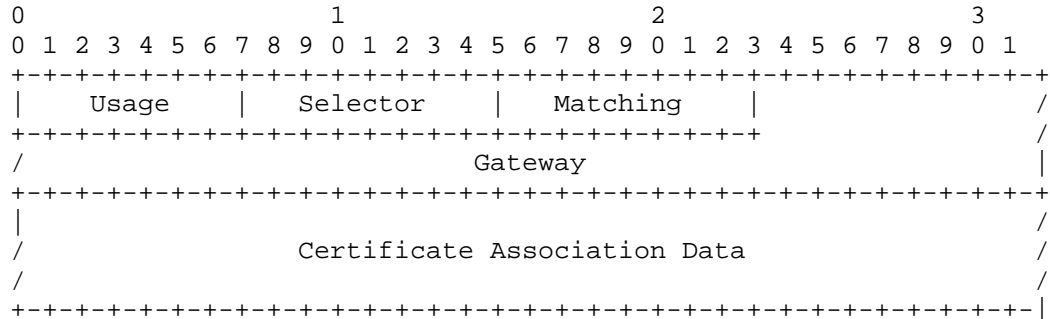


Figure 1

2.1.1. The Usage Field

The meaning, semantics, and interpretation of the Usage field of the IPSECA resource record follow the specification described in Section 2.1 of [I.D.draft-ietf-dane-registry-acronyms]:

Value	Acronym	Short Description	Reference
0	PKIX-TA	CA constraint	[RFC6698]
1	PKIX-EE	Service certificate constraint	[RFC6698]
2	DANE-TA	Trust anchor assertion	[RFC6698]
3	DANE-EE	Domain-issued certificate	[RFC6698]
4-254		Unassigned	
255	PrivCert	Reserved for Private Use	[RFC6698]

Table 1: TLSA Certificate Usages

2.1.2. The Selector Field

The meaning, semantics, and interpretation of the Selector field of the IPSECA resource record follow the specification described in Section 2.2 of [I.D.draft-ietf-dane-registry-acronyms]:

Value	Acronym	Short Description	Reference
0	Cert	Full certificate	[RFC6698]
1	SPKI	SubjectPublicKeyInfo	[RFC6698]
2	DANE-TA	Trust anchor assertion	[RFC6698]
3-254		Unassigned	
255	PrivSel	Reserved for Private Use	[RFC6698]

Table 2: TLSA Selectors

2.1.3. The Matching Field

The meaning, semantics, and interpretation of the Matching field of the IPSECA resource record follow the specification described in Section 2.3 of [I.D.draft-ietf-dane-registry-acronyms]:

Value	Acronym	Short Description	Reference
0	Full	No hash used	[RFC6698]
1	SHA2-256	256 bit hash by SHA2	[RFC6698]
2	SHA2-512	512 bit hash by SHA2	[RFC6698]
3-254		Unassigned	
255	PrivMatch	Reserved for Private Use	[RFC6698]

Table 3: TLSA Matching Types

2.1.4. The Gateway Field

The Gateway field follows the similar logic to that specified in [RFC4025], Section 2.5:

"The gateway field indicates a gateway to which an IPsec tunnel may be created in order to reach the entity named by this resource record.

There are three formats:

A 32-bit IPv4 address is present in the gateway field. The data portion is an IPv4 address as described in section 3.4.1 of [RFC1035]. This is a 32-bit number in network byte order.

A 128-bit IPv6 address is present in the gateway field. The data portion is an IPv6 address as described in section 2.2 of [RFC3596]. This is a 128-bit number in network byte order.

The gateway field is a normal wire-encoded domain name, as described in section 3.3 of [RFC1035]. Compression MUST NOT be used."

It is at the gateway specified in this field that any key exchange should be conducted.

2.1.5. The Certificate Association Data Field

The meaning, semantics, and interpretation of the Certificate Association Data field of the IPSECA resource record follow the specification of the same field in the TLSA resource record, described in Section 2.1.4 of [RFC6698]:

"This field specifies the 'certificate association data' to be matched. These bytes are either raw data (that is, the full certificate or its SubjectPublicKeyInfo, depending on the selector) for matching type 0, or the hash of the raw data for matching types 1 and 2. The data refers to the certificate in the association, not to the TLS ASN.1 Certificate object."

2.2. IPSECA RR Presentation Format

</STUBBED OUT SECTION>

2.3. Domain Names used for IPSEC Records

The IPSECA resource record SHOULD be mapped to a domain name that is intuitive when discovering OE IPsec tunnels for specific services. The expected procedure for constructing the domain names for IPSECA records that enable OE for DNS (port 53) are:

1. The left-most label begins with an underscore character (_), followed by the decimal representation of the port number that corresponds to the service that should be conducted over IPsec. For example, the DNS transactions discussed in this document would result in "_53".
2. Next, the fully qualified domain name [RFC1035] of the service is appended to the right side. In the case of a DNS name server, that is its domain name. In the case of a service that is located using an IP address, the service address records MUST be its full reverse octet name (including the appropriate suffix, such as .in-addr.arpa. for IPv4 addresses and .ip6.arpa for IPv6

addresses).

Any custom configured tunnels and port mappings may result local policies that use their own domain name format. Such custom OE tunnels are non-standard, and may not be discoverable by other relying parties.

2.4. IPSECA RR Examples

Because the IPSECA record is intended to be associated with a Service Address Records, it (implicitly) can also be associated with an IP address (through the reverse DNS). A few illustrative mappings are presented here as examples. These domain name / resource record mappings are not necessarily intended to update the processing of protocols like IKEv1 [RFC2409], IKEv2 [RFC5996], etc. or other OE protocols [RFC4322]. Rather, these mappings are intended to serve as examples of IPsec tunnels, and their proper configuration. They MAY be used in verifying Security Associations, but a protocol to do this is beyond the scope of this document.

2.4.1. OE to a DNS Name Server Example

Suppose a DNS zone example.com is served by the name servers ns1.example.com and ns2.example.com. If the zone operators want to advertise their willingness to offer OE to their name servers using IKEv2 [RFC5996], then the following domain names MUST be placed under the example.com zone (the contents of the resource records, below, are exemplary only and MAY have whatever values a zone operator chooses):

```
_53.ns1.example.com. IN IPSECA (  
    0 1 1 ns1.example.com  
    edeff39034cd2ee83446633a9fba  
    d815a579134ecd7636e51af92ec7  
    207fd490 ) ; Verify IPsec for DNS txns  
  
_53.ns2.example.com. IN IPSECA (  
    0 1 1 tunnel2.example.com  
    edeff39034cd2ee83446633a9fba  
    d815a579134ecd7636e51af92ec7  
    207fd490 ) ; Verify IPsec for DNS txns
```

This example illustrates how a zone MAY indicate where an SPD entry and SA establishment endpoints exist for its name servers (note, they are not required to be the name servers themselves). Here, each name server is mapped to a tunnel endpoint (ns1.example.com acts as its own endpoint, and ns2.example.com points to another gateway), and

these two name servers are mapped to service ports for DNS (port 53). The IPSECA records above indicate that they verify the CA who must have issued the IPsec certificate used and they represent a SHA256 hash of that certificate's SPKI.

Alternately, suppose an enterprise wants to configure OE for DNS transactions between its desktop clients and its recursive resolver. In this case, if the enterprise has configured their desktop clients (perhaps through DHCP) to forward their DNS queries to a caching recursive resolver at the IP address 192.168.1.2, then the following IPSECA mapping should be placed in an internally managed DNS reverse zone:

```
_53.2.1.168.192.in-addr.arpa. IN IPSECA (
  3 0 2 192.168.1.2
    8f6ea3c50b5c488bef74c7c4a17a
    24e8b0f4777d13c211a29223b69a
    ea7a89184ac4d272a2e3d9760966
    fb3f220b39f7fdfb325998289e50
    311ce0748f13cled ) ; Verify data in IKEv2 SA
```

This example illustrates how a caching recursive resolver MAY indicate where it will accept IPsec tunnel establishment and what the certificate used for a SA should be. Here the DNS service port and the IPSECA records describe the nature of the authentic certificate that SHOULD be used in an SA with this endpoint. In this example, the IPSECA records both specify that a DANE-EE cert should be expected in an SA with this resolver, and the SHA-512 hash of that full certificate should match the encoded value in the IPSECA resource record.

Of note here is that since SAs MAY be identified by domain names (which map to IP addresses), some IP addresses may host services that offer IPsec, and some that do not. The IPSECA record allows hosts to advertise these nuanced configurations in the same way that these services are discovered (through the DNS itself).

3. Operational Considerations

Scaling IPsec connections to the full capacity that large recursive resolvers or large authoritative name servers operate at could be cause for concern. The additional overhead required to establish and maintain SAs could exceed the provisioning capacity of deployed systems. However, there are several relevant observations:

1. If a resolver enables OE, but no (or relatively few) name servers provision IPSECA records, then no IPsec tunnels will be

established, and the load will remain static (or marginally increase).

2. If an authoritative name server provisions IPSECA record, it will only result in additional load if querying resolvers are configured to attempt OE.
3. Using white-listing techniques (such as those used during pilot deployments of AAAA records) would allow authoritative name servers to only return IPSECA responses to clients that have been white-listed. This would allow name servers to control the amount of IPsec overhead they incur. For the same reason, resolvers can be configured to only query for IPSECA records from white-listed name servers.

4. IANA Considerations

This document uses a new DNS resource record type, called IPSECA. This resource record will need to have a new value assigned to it. Current implementations are advised to use a type number TYPE65347.

This document uses the same semantics and values as the TLSA resource record [RFC6698] for its Usage, Selector, and Matching fields. Any future use or modification of an IANA registry for that resource record will have similar effects on this resource record.

5. Security Considerations

This document details some of the benefits of using IPsec OE for DNS transactions. Such a utility does not reduce the benefits of other security protections. For example, the object-level security assurances that are offered by DNSSEC are cooperative with the session-level security of IPsec. Additional discussions are available in [IPSEC_APPEAL]. Moreover, the protections described herein also offer cooperative benefits with higher layer protocol protections, like TLS [RFC5246]. Any combination of these types of protections offer both defense-in-depth (securing transactions at multiple levels) and offer security practitioners a larger mosaic of security tools from which to construct and maintain their security postures.

5.1. Interactions

This document requires that all fully qualified domain names [RFC1035] must be secured by DNSSEC. This includes domains in the reverse tree of DNS (which represent IP addresses).

The use of IPSECA resource records does not constitute a source of information leakage. Rather, it provides a mechanism to help bolster confidentiality, by obfuscating DNS transactions.

Expressing tunnel endpoints through DNS may allow adversaries a vehicle to learn where OE is being offered by name servers. However, OE tunnels to these name servers will only be attempted if querying resolvers are configured to attempt IPsec. As a result, adversaries may be able to learn of potential tunnel endpoints, but if they aim to disrupt active IPsec traffic, they must still observe which resolvers are trying to initiate IPsec communications. Therefore, adversaries would have no greater opportunity to disrupt IPsec traffic than they already do. They would still begin by (for example) observing VPN tunnel setup on wireless LANs (such as at public WiFi hot-spots).

5.2. Last Mile Security Analysis

For the last mile, we define one type of attack as the case where an adversary intercepts messages that can be undetectably spoofed. For example, if a zone (like example.com) has deployed DNSSEC, then if an adversary responds to a DNS query for www.exmaple.com, a validating DNS resolver should be able to detect the forgery. However, if an adversary responds to a query that is sent for a non-DNSSEC zone, a resolver cannot distinguish the spoofed response from an authentic response. In addition to this, many bootstrapping protocols (such as DHCP [RFC2131]) represent the first opportunity for an adversary to disrupt DNS transactions (by subverting the bootstrapping of the resolver itself on stub-resolvers). Under this model, a DNS stub-resolver's security posture is enhanced by keeping an adversary's attack window to the smallest value possible.

Therefore, the attack window offered by DNS clients in a given time span T is comprised of the set of transactions that bootstrap configurations $W_{\text{cfg}}(T)$, plus any DNS transactions that are not verifiable. Of note, however, is that the DNSSEC transactions between stub-resolvers and recursive resolvers are not protected by DNSSEC's cryptography. The only indication of protections is a header bit (the AD bit), which is spoofable. As a result, the attack window includes all DNS transactions $W_{\text{rDNS}}(T)$.

From this, the attack window can be expressible as:

$$W(T) = W_{\text{cfg}}(T) + W_{\text{rDNS}}(T)$$

Of note is that under most circumstances, resolvers issue many more queries than configuration requests. So,

$W_cfg(T) = 1$, and $W_rDNS(T) \gg W_cfg(T)$.

However, consider the attack window when using OE: $\{W(T)\}$. If the initial configuration includes a DNSKEY trust anchor that can be used to verify DNSSEC data that corresponds to a resolver's corresponding reverse zone (i.e., the IPSECA RR under in-addr.arpa or ip6.arpa), then $\{W_cfg(T)\} = 1$ and $\{W_rDNS(T)\} = 0$. Therefore, since $W_rDNS(T) \gg W_cfg(T)$ and $\{W_rDNS(T)\} = 0$, then by the transitive property,

$W(T) \gg \{W(T)\}$.

6. Acknowledgements

The editors would like to express their thanks for the early support and insights given by Danny McPherson.

7. References

7.1. Normative References

- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.
- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- [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.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.
- [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.

7.2. Informative References

- [DANE_SATIN12]
Osterweil, E., Kaliski, B., Larson, M., and D. McPherson,
"Reducing the X.509 Attack Surface with DNSSEC's DANE",
Proceedings of Securing and Trusting Internet Names, SATIN
'12, March 2012.
- [I-D.draft-ogud-dane-vocabulary]
Gudmundsson, O., "Harmonizing how applications specify
DANE-like usage", October 2013.
- [I-D.draft-ietf-dane-registry-acronyms]
Gudmundsson, O., "Adding acronyms to simplify DANE
conversations", January 2014.
- [IPSEC_APPEAL]
Osterweil, E. and D. McPherson, "IPsec's Appeal:
Protecting DNS Under the Covers", Verisign Labs Technical
Report #1130006 Revision 1, January 2013.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol",
RFC 2131, March 1997.
- [RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange
(IKE)", RFC 2409, November 1998.
- [RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi,
"DNS Extensions to Support IP Version 6", RFC 3596,
October 2003.
- [RFC4025] Richardson, M., "A Method for Storing IPsec Keying
Material in DNS", RFC 4025, March 2005.
- [RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol",
RFC 4306, December 2005.
- [RFC4322] Richardson, M. and D. Redelmeier, "Opportunistic
Encryption using the Internet Key Exchange (IKE)",
RFC 4322, December 2005.
- [RFC4945] Korver, B., "The Internet IP Security PKI Profile of
IKEv1/ISAKMP, IKEv2, and PKIX", RFC 4945, August 2007.
- [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.

- [RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.
- [RFC6071] Frankel, S. and S. Krishnan, "IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap", RFC 6071, February 2011.
- [RFC6960] Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 6960, June 2013.

Appendix A. Name Server OE Configuration Example

<STUBBED OUT SECTION>

NAME SERVER SIDE

- o Config SPD to accept connections from any on port 53 only
- o Zones add IPSECA RRs for each NS domain name and configure DNSSEC:
<examples>

RESOLVER SIDE

- o resolver processing logic to intercept referrals and look for IPSECA RR(s).
- o When an IPSECA RR is found, create SPD for that IP and port 53.

</STUBBED OUT SECTION>

Appendix B. Recursive Resolver OE Configuration Example

<STUBBED OUT SECTION>

RESOLVER SIDE

- o If public resolver, create SPD entry that only allows IPsec from port 53. If internal resolver, limit to addresses serviced.

REVERSE DNS ZONE

- o Add IPSECA RR(s) and configure DNSSEC

STUB SIDE

- o Configure reverse zone DNSKEY (if 1918) as a local TA (such as over DHCP). Then do onetime DNSSEC validation for fetching IPSECA RR.
- o Tools include dnskey-grab and/or NLnet Labs' xxxxx.

</STUBBED OUT SECTION>

Authors' Addresses

Eric Osterweil
VeriSign, Inc.
Reston, VA
USA

Email: eosterweil@verisign.com

Glen Wiley
VeriSign, Inc.
Reston, VA
USA

Email: gwiley@verisign.com

Dave Mitchell
Twitter
355 Market Street, Suite 900
San Francisco, CA
USA

Email: dave@twitter.com

Andrew Newton
American Registry for Internet Numbers
3635 Concorde Parkway
Chantilly, VA
USA

Email: andy@arin.net

Network Working Group
Internet-Draft
Intended status: Informational
Expires: June 27, 2014

V. Smyslov
ELVIS-PLUS
December 24, 2013

The NULL Authentication Method in IKEv2 Protocol
draft-smyslov-ipsecme-ikev2-null-auth-00

Abstract

This document defines the NULL Authentication Method for IKEv2 Protocol. This method provides a way to omit peer authentication in IKEv2 and to explicitly indicate it in the protocol run. This method may be used to preserve anonymity or in situations, where no trust relationship exists between the parties.

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 <http://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 June 27, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Conventions Used in This Document	3
2. Using NULL Authentication Method	4
2.1. Authentication Payload	4
2.2. Identity Payload	4
3. Security Considerations	5
4. IANA Considerations	6
5. Normative References	7
Author's Address	8

1. Introduction

The Internet Key Exchange Protocol version 2 (IKEv2), specified in [RFC5996], provides a way for two parties to perform authenticated key exchange. Mutual authentication is mandatory in the IKEv2, so that each party must be authenticated by the other, but authentication methods, used by the peers, need not be the same.

In some situations mutual authentication is undesirable or impossible. For example:

- o User wants to get anonymous access to some resource. In this situation he/she should be able to authenticate server, but to leave out his/her own authentication to prevent anonymity. In this case one-way authentication is desirable.
- o Two peers without any trust relationship want to get some level of security in their communications. Without trust relationship they cannot prevent active Man-in-the-Middle attacks, but it is still possible to prevent passive eavesdropping with opportunistic encryption. In this case they have to perform unauthenticated key exchange.

To meet this needs the document introduces NULL Authentication Method, which is effectively a "dummy" method, that provides no authentication. This allows peer to explicitly indicate to the other side that he is unwilling or unable to certify his identity.

1.1. Conventions Used in This Document

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].

2. Using NULL Authentication Method

NULL Authentication Method affects how Authentication and Identity Payloads are formed in IKE_AUTH Exchange.

2.1. Authentication Payload

Even when peer uses NULL Authentication, the AUTH Payload must still be present in IKE_AUTH Exchange and must be properly formed, as it cryptographically links IKE_SA_INIT Messages with the other Messages sent over IKE SA.

With NULL Authentication Method the content of AUTH Payload MUST be computed using the syntax for pre-shared secret authentication, described in Section 2.15 of [RFC5996]. The values SK_pi and SK_pr MUST be used as shared secrets for AUTH Payloads generated by Initiator and Responder respectively. Note, that this is exactly how content of the two last AUTH Payloads is calculated in case of using non-key generating EAP Method (see Section 2.16 of [RFC5996] for details). The field Auth Method MUST be set to <to be assigned by IANA>.

2.2. Identity Payload

NULL Authentication Method provides no authentication of the party using it. For that reason Identity Payload content cannot be verified by the other party and SHOULD be ignored. It MAY be used for the purpose of audit, but it MUST NOT be used for any authorization decisions. As peer identity is meaningless in this case, Identification Data MAY be omitted from ID Payload, in which case ID Type MAY be set to any value. Implementations supporting NULL Authentication Method MUST NOT fail if they receive such "empty" ID Payload.

3. Security Considerations

IKEv2 protocol provides mutual authentication of the peers. If one peer uses NULL Authentication Method, then this peer cannot be authenticated by the other side, and it makes authentication in IKEv2 to become one-way. If both peers use NULL Authentication method, key exchange becomes unauthenticated, that makes it subject to the Man-in-the-Middle attack.

The identity of the peer using NULL Authenticated Method cannot be verified by the other side and, therefore, MUST NOT be used neither for authorization purposes, nor for policy decisions. All peers who use NULL Authenticated Method should be considered by the other party as "guests" and get the least possible privileges.

4. IANA Considerations

This document defines new value in the "IKEv2 Authentication Method" registry:

<TBA>	NULL Authentication Method
-------	----------------------------

5. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

Author's Address

Valery Smyslov
ELVIS-PLUS
PO Box 81
Moscow (Zelenograd) 124460
RU

Phone: +7 495 276 0211
Email: svan@elvis.ru

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 17, 2014

P. Wouters
Red Hat
February 13, 2014

Using DANE to Associate OpenPGP public keys with email addresses
draft-wouters-dane-openpgp-02

Abstract

OpenPGP is a message format for email (and file) encryption, that lacks a standardized lookup mechanism to obtain OpenPGP public keys. This document specifies a standardized method for securely publishing and locating OpenPGP public keys in DNS using a new OPENPGPKEY DNS Resource Record.

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 <http://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 August 17, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	3
2. The OPENPGPKEY Resource Record	3
2.1. The OPENPGPKEY RDATA component	3
2.2. The OPENPGPKEY RDATA wire format	3
2.3. The OPENPGPKEY RDATA presentation format	3
3. Location of the OpenPGPKEY record	4
4. OpenPGP Key size and DNS	4
5. Security Considerations	5
5.1. Email address information leak	5
5.2. Forward security of OpenPGP versus DNSSEC	5
6. IANA Considerations	6
6.1. OPENPGPKEY RRtype	6
7. Acknowledgements	6
8. References	6
8.1. Normative References	6
8.2. Informative References	7
Appendix A. Generating OPENPGPKEY records	7
Author's Address	8

1. Introduction

To encrypt a message to a target recipient using OpenPGP [RFC4880], possession of the recipient's OpenPGP public key is required. To obtain that public key, two problems need to be solved by the sender's email client, MUA or MTA. Where does one find the recipient's public key and how does one trust that the found key actually belongs to the intended recipient.

Obtaining a public key is not a straightforward process as there are no trusted standardized locations for publishing OpenPGP public keys indexed by email address. Instead, OpenPGP clients rely on "well-known key servers" that are accessed using the web based HKP protocol or manually by users using a variety of differently formatted front-end web pages.

Currently deployed key servers have no method of validating any uploaded OpenPGP public key. The key servers simply store and publish. Anyone can add public keys with any identities and anyone can add signatures to any other public key using forged malicious identities. Furthermore, once uploaded, public keys cannot be deleted. People who did not pre-sign a key revocation can never remove their public key from these key servers once they lost their private key.

The lack of association of email address and public key lookup is also preventing email clients, MTAs and MUAs from encrypting a received message to the target recipient forcing the software to send the message unencrypted. Currently deployed MTA's only support encrypting the transport of the email, not the email contents itself.

This document describes a mechanism to associate a user's OpenPGP public key with their email address, using a new DNS RRtype.

The proposed new DNS Resource Record type is secured using DNSSEC. This trust model is not meant to replace the "web of trust" model. However, it can be used to encrypt a message that would otherwise have to be sent out unencrypted, where it could be monitored by a third party in transit or located in plaintext on a storage or email server.

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 RFC 2119 [RFC2119].

This document also makes use of standard DNSSEC and DANE terminology. See DNSSEC [RFC4033], [RFC4034], [RFC4035], and DANE [RFC6698] for these terms.

2. The OPENPGPKEY Resource Record

The OPENPGPKEY DNS resource record (RR) is used to associate an end entity OpenPGP public key with an email address, thus forming a "OpenPGP public key association".

The type value allocated for the OPENPGPKEY RR type is [TBD]. The OPENPGPKEY RR is class independent. The OPENPGPKEY RR has no special TTL requirements.

2.1. The OPENPGPKEY RDATA component

The RDATA (or RHS) of an OPENPGPKEY Resource Record contains a single value consisting of a [RFC4880] formatted OpenPGP public keyring.

2.2. The OPENPGPKEY RDATA wire format

The RDATA Wire Format is the binary OpenPGP public keyring as specified in [RFC4880] without any ascii armor or base64 encoding.

2.3. The OPENPGPKEY RDATA presentation format

The RDATA Presentation Format, as visible in textual zone files, consists of the [RFC4880] formatted OpenPGP public keyring encoded in Base64 [RFC4648]

3. Location of the OpenPGPKEY record

Email addresses are mapped into DNS using the following method:

1. The user name (the "left-hand side" of the email address, called the "local-part" in the mail message format definition [RFC2822] and the "local part" in the specification for internationalized email [RFC6530]), is hashed using the SHA2-224 [RFC5754] algorithm to become the left-most label in the prepared domain name. This does not include the at symbol ("@") that separates the left and right sides of the email address.
2. The DNS does not allow the use of all characters that are supported in "local-part" of email addresses as defined in [RFC2822] and [RFC6530]. The SHA2-224 hashing of the user name ensures that none of these characters would need to be placed directly in the DNS.
3. The string "_openpgpkey" becomes the second left-most label in the prepared domain name.
4. The domain name (the "right-hand side" of the email address, called the "domain" in RFC 2822) is appended to the result of step 2 to complete the prepared domain name.

For example, to request an OPENPGPKEY resource record for a user whose email address is "hugh@example.com", an OPENPGPKEY query would be placed for the following QNAME: "8d5730bd8d76d417bf974c03f59eedb7af98cb5c3dc73ea8ebbd54b7._openpgpkey.example.com" The corresponding RR in the example.com zone might look like:

```
8d5730bd8d76d417bf974c03f59eedb7af98cb5c3dc73ea8ebbd54b7._openpgpkey.example.  
com. IN OPENPGPKEY <encoded public key>
```

4. OpenPGP Key size and DNS

Although the reliability of the transport of large DNS Resource Records has improved in the last years, it is still recommended to keep the DNS records as small as possible without sacrificing the security properties of the public key. The algorithm type and key size of OpenPGP keys should not be modified to accomodate this section.

OpenPGP supports various attributes that do not contribute to the security of a key, such as an embedded image file. It is recommended that these properties are not exported to OpenPGP public keyrings that are used to create OPENPGPKEY Resource Records. Some OpenPGP software, for example GnuPG, have support for a "minimal key export" that is well suited to use as OPENPGPKEY RDATA. See Appendix A

5. Security Considerations

OPENPGPKEY usage considerations are published in [OPENPGPKEY-USAGE]

5.1. Email address information leak

Email addresses are not secret. Using them causes its publication. The hashing of the user name in this document is not a security feature. Publishing OPENPGPKEY records however, will create a list of hashes of valid email addresses, which could simplify obtaining a list of valid email addresses for a particular domain. It is desirable to not ease the harvesting of email addresses where possible.

The domain name part of the email address is not used as part of the hash so that hashes can be used in multiple zones deployed using DNAME [RFC6672]. This does makes it slightly easier and cheaper to brute-force the SHA2-224 hashes into common and short user names, as single rainbow tables can be re-used accross domains. This can be somewhat countered by using NSEC3.

DNS zones that are signed with DNSSEC using NSEC for denial of existence are susceptible to zone-walking, a mechanism that allows someone to enumerate all the OPENPGPKEY hashes in a zone. This can be used in combination with previously hashed common or short user names (in rainbow tables) to deduce valid email addresses. DNSSEC-signed zones using NSEC3 for denial of existence instead of NSEC are significantly harder to brute-force after performing a zone-walk.

5.2. Forward security of OpenPGP versus DNSSEC

DNSSEC key sizes are chosen based on the fact that these keys can be rolled with next to no requirement for security in the future. If one doubts the strength or security of the DNSSEC key for whatever reason, one simply rolls to a new DNSSEC key with a stronger algorithm or larger key size. On the other hand, OpenPGP key sizes are chosen based on how many years (or decades) their encryption should remain unbreakable by adversaries that own large scale computational resources.

This effectively means that anyone who can obtain a DNSSEC private key of a domain name via coercion, theft or brute force calculations, can replace any OPENPGPKEY record in that zone and all of the delegated child zones, irrespective of the key size of the OpenPGP keypair. Any future messages encrypted with the malicious OpenPGP key could then be read.

Therefor, an OpenPGP key obtained via an OPENPGPKEY record can only be trusted as much as the DNS domain can be trusted, and are no substitute for in-person key verification of the "Web of Trust". See [OPENPGPKEY-USAGE] for more in-depth information on safe usage of OPENPGPKEY based OpenPGP keys.

6. IANA Considerations

6.1. OPENPGPKEY RRtype

This document uses a new DNS RR type, OPENPGPKEY, whose value [TBD] has been allocated by IANA from the Resource Record (RR) TYPES subregistry of the Domain Name System (DNS) Parameters registry.

7. Acknowledgements

This document is based on RFC-4255 and draft-ietf-dane-smime whose authors are Paul Hoffman, Jacob Schlyter and W. Griffin. Olafur Gudmundsson provided feedback and suggested various improvements. Willem Toorop contributed the gpg and hexdump command options.

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.
- [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.

- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.
- [RFC4880] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, November 2007.
- [RFC5754] Turner, S., "Using SHA2 Algorithms with Cryptographic Message Syntax", RFC 5754, January 2010.

8.2. Informative References

- [OPENPGPKEY-USAGE] Wouters, P., "Usage considerations with the DNS OPENPGPKEY record", draft-wouters-openpgpkey-usage (work in progress), January 2014.
- [RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, July 1997.
- [RFC2822] Resnick, P., "Internet Message Format", RFC 2822, April 2001.
- [RFC3597] Gustafsson, A., "Handling of Unknown DNS Resource Record (RR) Types", RFC 3597, September 2003.
- [RFC6530] Klensin, J. and Y. Ko, "Overview and Framework for Internationalized Email", RFC 6530, February 2012.
- [RFC6672] Rose, S. and W. Wijngaards, "DNAME Redirection in the DNS", RFC 6672, June 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.

Appendix A. Generating OPENPGPKEY records

The commonly available GnuPG software can be used to generate the RRdata portion of an OPENPGPKEY record:

```
gpg --export --export-options export-minimal \  
hugh@example.com | base64
```

The `--armor` or `-a` option of the `gpg` command should NOT be used, as it adds additional markers around the armored key.

When DNS software reading or signing the zone file does not yet support the OPENPGPKEY RRtype, the Generic Record Syntax of [RFC3597] can be used to generate the RDATA. One needs to calculate the number of octets and the actual data in hexadecimal:

```
gpg --export --export-options export-minimal \  
hugh@example.com | wc -c
```

```
gpg --export --export-options export-minimal \  
hugh@example.com | hexdump -e \  
    '\t' /1 "%.2x" -e '/32 "\n"'
```

These values can then be used to generate a generic record:

```
<SHA2-224(hugh)>._openpgpkey.example.com. IN TYPE65280 \# <numOctets> <keydat  
a in hex>
```

The openpgpkey command in the hash-slinger software can be used to generate complete OPENPGPKEY records

```
~> openpgpkey --output rfc hugh@example.com  
8d5730bd8d[...]bbd54b7._openpgpkey.example.com. IN OPENPGPKEY mQCNazIG[...]  
  
~> openpgpkey --output generic hugh@example.com  
8d5730bd8d[...]bbd54b7._openpgpkey.example.com. IN TYPE65280 \# 2313 99008d03  
[...]
```

Author's Address

Paul Wouters
Red Hat

Email: pwouters@redhat.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 18, 2014

P. Wouters
Red Hat
February 14, 2014

Best Common Practise for using OPENPGPKEY records
draft-wouters-dane-openpgpkey-usage-00

Abstract

The OPENPGPKEY DNS Resource Record can be used to match an email address to an OpenPGP key. This document specifies a Best Common Practise ("BCP") for email clients, MUA's and MTA's for using the OPENPGPKEY DNS Resource Record.

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 <http://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 August 18, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	2
2. The OPENPGPKEY record presence	2
3. OpenPGP public key considerations	3
3.1. Public Key UIDs and email addresses	3
3.2. Public Key UIDs and IDNA	3
3.3. Public Key UIDs and synthesized DNS records	3
3.4. OpenPGP Key size and DNS	4
4. Security Considerations	4
4.1. Email address information leak	4
4.2. OpenPGP security and DNSSEC	5
4.3. MTA behaviour	5
4.4. MUA behaviour	6
4.5. Email client behaviour	6
5. References	7
5.1. Normative References	7
5.2. Informative References	7
Author's Address	8

1. Introduction

This document describes a Best Current Practise ("BCP") for using OPENPGPKEY DNS Resource Records xref target="OPENPGPKEY"/ in email clients, MUA's and MTA.

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 RFC 2119 [RFC2119].

This document also makes use of standard DNSSEC and DANE terminology. See DNSSEC [RFC4033], [RFC4034], [RFC4035], and DANE [RFC6698] for these terms.

2. The OPENPGPKEY record presence

A user who publishes an OPENPGPKEY record in DNS explicitly prefers receiving encrypted email over receiving unencrypted email.

A user who publishes an OPENPGPKEY record in DNS still expects senders to perform their due diligence by additional verification of their public key via other out-of-band methods before sending any confidential or sensitive information

In other words, the OPENPGPKEY record in DNS, without any additional verification, should be used only as an alternative to sending plaintext email. It SHOULD NOT be used to change one's opinion on whether it is safe or appropriate to send the content via email in the first place.

3. OpenPGP public key considerations

Once an OPENPGPKEY resource record has been found and the OpenPGP public keyring has been decoded, the right public key must be located inside the keyring. For a public key in the keyring to be usable, the public key has to have a key uid as specified in [RFC4648] that matches the email address for which the OPENPGPKEY RR lookup was performed.

3.1. Public Key UIDs and email addresses

An OpenPGP public key can be associated with multiple email addresses by specifying multiple key uids. The OpenPGP public key obtained from a OPENPGPKEY RR can be used as long as the target recipient's email address appears as one of the OpenPGP public key uids. The name part (left of the @) should appear in the native format, not its SHA2-224 hash that was used to lookup the OPENPGPKEY RR.

3.2. Public Key UIDs and IDNA

Internationalized domains that use non-ascii characters (U-label) are encoded in DNS using IDNA [RFC5891] - also referred to as punycode or A-label. When matching OpenPGP public key uids, both the email address specified using U-label and A-label should be considered as valid public key uids.

3.3. Public Key UIDs and synthesized DNS records

CNAME's (see [RFC2181]) and DNAME's (see [RFC6672]) can be followed to obtain an OPENPGPKEY RR, as long as the original recipient's email address appears as one of the OpenPGP public key uids. For example, if the OPENPGPKEY RR query for hugh@example.com (8d57[...]b7._openpgpkey.example.com) yields a CNAME to 8d57[...]b7._openpgpkey.example.net, and an OPENPGPKEY RR for 8d57[...]b7._openpgpkey.example.net exists, then this OpenPGP public key can be used, provided one of the key uids contains "hugh@example.com". This public key cannot be used if it would only contain the key uid "hugh@example.net".

If one of the OpenPGP key uids contains only a single wildcard as the LHS of the email address, such as "*@example.com", the OpenPGP public key may be used for any email address within that domain. Wildcards

at other locations (eg hugh@*.com) or regular expressions in key uids are not allowed, and any OPENPGPKEY RR containing these should be ignored.

3.4. OpenPGP Key size and DNS

Although the reliability of the transport of large DNS Resource Records has improved in the last years, it is still recommended to keep the DNS records as small as possible without sacrificing the security properties of the public key. The algorithm type and key size of OpenPGP keys should not be modified to accommodate this section.

OpenPGP supports various attributes that do not contribute to the security of a key, such as an embedded image file. It is recommended that these properties are not exported to OpenPGP public keyrings that are used to create OPENPGPKEY Resource Records. Some OpenPGP software, for example GnuPG, have support for a "minimal key export" that is well suited to use as OPENPGPKEY RDATA.

4. Security Considerations

The main goal of the OPENPGPKEY resource record is to stop passive attacks against plaintext emails. While it can also thwart some active attacks (such as people uploading rogue keys to key servers in the hopes that others will encrypt to these rogue keys), this resource record is not a replacement for verifying OpenPGP public keys via the web of trust signatures, or manually via a fingerprint verification.

Various components could be responsible for encrypting an email message to a target recipient. It could be done by the sender's email client or software plugin, the sender's Mail User Agent (MUA) or the sender's Mail Transfer Agent (MTA). Each of these have their own characteristics. An email client can direct the human to make a decision before continuing. The MUA can either accept or refuse a message. The MTA must deliver the message as-is, or encrypt the message before delivering. Each of these programs should ensure that the security of an email message is never downgraded, and that an unencrypted received message will be encrypted whenever possible.

Organisations that require to be able to read everyone's encrypted email should publish the escrow key as the OPENPGPKEY record. Upon receipt, such mail servers can optionally re-encrypt the message to the individual's OpenPGP key.

4.1. Email address information leak

DNS zones that are signed with DNSSEC using NSEC for denial of existence are susceptible to zone-walking, a mechanism that allows someone to enumerate all the names in the zone. Someone who wanted to collect email addresses from a zone that uses OPENPGPKEY might use such a mechanism. DNSSEC-signed zones using NSEC3 for denial of existence are significantly less susceptible to zone-walking. Someone could still attempt a dictionary attack on the zone to find OPENPGPKEY records, just as they can use dictionary attacks on an SMTP server or grab the entire contents of existing PGP key servers to see which addresses are valid.

4.2. OpenPGP security and DNSSEC

DNSSEC key sizes are chosen based on the fact that these keys can be rolled with next to no requirement for security in the future. If one doubts the strength or security of the DNSSEC key for whatever reason, one simply rolls to a new DNSSEC key with a stronger algorithm or larger key size.

This effectively means that anyone who can obtain a DNSSEC private key of a domain name via coercion, theft or brute force calculations, can replace any OPENPGPKEY record in that zone and all of the delegated child zones, irrespective of the key length strength of the OpenPGP keypair.

Therefore, DNSSEC is not an alternative for the "web of trust" or for manual fingerprint verification by humans. It is a solution aimed to ease obtaining someone's public key, and without manual verification should be treated as "better than plaintext" only. While this thwarts all passive attacks that simply capture and log all plaintext email content, it is not a security measure against active attacks.

4.3. MTA behaviour

An MTA could be operating in a stand-alone mode, without access to the sender's OpenPGP public keyring, or in a way where it can access the user's OpenPGP public keyring. Regardless, the MTA MUST NOT modify the user's OpenPGP keyring.

An MTA sending an email MUST NOT add the public key obtained from an OPENPGPKEY resource record to a permanent public keyring for future use beyond the TTL.

If the obtained public key is revoked, the MTA MUST NOT use the key for encryption, even if that would result in sending the message in plaintext.

If a message is already encrypted, the MTA SHOULD NOT re-encrypt the message, even if different encryption schemes or different encryption keys were used.

If an OPENPGPKEY resource record is received without DNSSEC protection, it MAY still be used for encryption.

If the DNS request for an OPENPGPKEY record returned an "indeterminate" or "bogus" answer, the MTA MUST NOT send the message and queue the plaintext message for delivery at a later time. If the problem persists, the email should be returned via the regular bounce methods.

If multiple non-revoked OPENPGPKEY resource records are found, the MTA SHOULD pick the most secure RR based on its local policy. [or should it encrypt to both?]

4.4. MUA behaviour

If the public key for a recipient obtained from the locally stored sender's public keyring differs from the recipient's OPENPGPKEY RR, the MUA MUST NOT accept the message for delivery.

If the public key for a recipient obtained from the locally stored sender's public keyring contains contradicting properties for the same key obtained from an OPENPGPKEY RR, the MUA SHOULD NOT accept the message for delivery.

If multiple non-revoked OPENPGPKEY resource records are found, the MUA SHOULD pick the most secure OpenPGP public key based on its local policy.

4.5. Email client behaviour

Email clients should adhere to the above listed MUA behaviour. Additionally, an email client MAY interact with the user to resolve any conflicts between locally stored keyrings and OPENPGPKEY RRdata.

An email client that is encrypting a message SHOULD clearly indicate to the user the difference between encrypting to a locally stored and humanly verified public key and encrypting to an unverified (by the human sender) public key obtained via an OPENPGPKEY resource record.

5. References

5.1. Normative References

- [OPENPGPKEY] Wouters, P., "DANE for OpenPGP public keys", draft-wouters-dane-openpgp (work in progress), February 2014.
- [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.
- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.
- [RFC4880] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, November 2007.
- [RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, August 2010.

5.2. Informative References

- [RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, July 1997.
- [RFC2822] Resnick, P., "Internet Message Format", RFC 2822, April 2001.
- [RFC4255] Schlyter, J. and W. Griffin, "Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints", RFC 4255, January 2006.
- [RFC6530] Klensin, J. and Y. Ko, "Overview and Framework for Internationalized Email", RFC 6530, February 2012.

[RFC6672] Rose, S. and W. Wijngaards, "DNAME Redirection in the DNS", RFC 6672, June 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.

Author's Address

Paul Wouters
Red Hat

Email: pwouters@redhat.com