Network Working Group
Internet-Draft
Updates: 4034, 4035 (if approved)
Intended status: Standards Track
Expires: December 30, 2017

P. Hoffman
M. Larson
ICANN

June 28, 2017

Session-baseed Authentication for DNS: DNSSEC-S
draft-hoffman-dnssec-s-00

Abstract

DNSSEC as defined in RFCs 4033, 4034, and 4035 is based on authenticated messages. That design has allowed DNSSEC to be deployed at the upper levels of the DNS tree, but operational issues with message-based authentication has caused lower levels fo the DNS tree to mostly forego DNSSEC. This document extends DNSSEC with a second type of authentication, based on session authentication from TLS, that is easier to deploy by some (but certainly not all) authoritative DNS servers. The goal is to have many more zones be DNSSEC-enabled.

Note that this document does _not_ replace current DNSSEC. A validating resolver needs to implement all of traditional DNSSEC, and might also implement the protocol defined here. A server might protect the contents of DNS zones for which it is authoritative with traditional DNSSEC, with the protocol defined here, or both. The protocol defined here is only useful for some authoritative servers, and is explicitly not useful for others.

*** Notice for -00 ***

This -00 draft is meant to engender discussion, particularly to find out if there is a good use case for this proposal. This draft is definitely not considered ready for consideration in an IETF WG.

Status of This Memo

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 December 30, 2017.

1. Introduction

As stated in [RFC4033], "The Domain Name System Security Extensions (DNSSEC) add data origin authentication and data integrity to the Domain Name System". The protocol described in [RFC4033], [RFC4034], and [RFC4035] provide those services by adding new resource records and specifying how to cryptographically validate the contents of those records. In this document, DNSSEC as defined in [RFC4033], [RFC4034], [RFC4035] and all RFCs that update those three is called "DNSSEC-M". The designation "-M" means "message": those RFCs define secure message-based authentication for DNS messages.

This document defines a second type of DNSSEC that can be used alongside DNSSEC-M. This second type uses secure session-based authentication using TLS. It is called "DNSSEC-S", where "-S" means "session". In short, DNSSEC-S allows the validating resolver to authenticate the origin of the DNS data because it comes directly from the authoritative server during a TLS session; the TLS session also provides data integrity. DNSSEC-S clients and servers MUST use TLS 1.3 [I-D.ietf-tls-tls13].

The protocol described in this document provides a mechanism could encourage many more organizations, particularly large organizations that already run secure web services, to enable DNSSEC on their zones.

1.1. Use Cases

Deploying DNSSEC-M has proven successful in some environments, but not in others. At the time this document is published, the root of the DNS tree is secured with DNSSEC-M, as are many of the TLDs in the root. However, only a few major content providers have signed their zones with DNSSEC-M.

When asked why they don't sign their zones, these content providers give various reasons, but one major reason stands out. The software that is used to sign zones for DNSSEC-M is often described as "complicated" and "fragile". If a zone is not properly signed before it is published by the authoritative server, parts or all of the zone may become unavailable to recursive resolvers that perform DNSSEC-M validation. Further, the signing software must be run periodically and the results must be correct; otherwise, the zone becomes unavailable and may stay that way for days even after the problem is discovered.

Content providers say that the risk of their zone becoming unavailable due to the above problems with DNSSEC-M (as well as other issues) is not worth the positive attributes of DNSSEC-M, and therefore leave their zones unsigned.

If a system can definitively eliminate this risk while introducing only much smaller risks, some content providers might consider authenticating their zones with DNSSEC. DNSSEC-S proposes to be such a system.

1.2. Use Cases Not Considered Here

DNSSEC-S uses TLS for session security, and TLS also provides encryption of sessions. However, the encrypted aspect of DNSSEC-S sessions is explicitly not considered a use case for DNSSEC-S. DNSSEC-S is not appropriate for all authoritative servers, so the encryption that comes as part of DNSSEC-S should only be considered a useful side-benefit, not a primary use case. The DPRIVE Working Group in the IETF is currently considering mechanisms to encrypt communications with authoritative servers.

The security of DNSSEC-S relies on keeping the TLS private key secret. Because of this, DNSSEC-S is not appropriate for DNSSEC-protected zones where the nameservers are run by multiple organizations that have different abilities to protect a private key. As an obvious example, the DNS root is currently served by a dozen different organizations. Because it is impossible to serve the current DNS using just DNSSEC-S, this document mandates that

validating resolvers MUST support validating with DNSSEC-M if they also support validating with DNSSEC-S.

1.3. 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, BCP 14 [RFC2119] and indicate requirement levels for compliant CBOR implementations.

This document creates two new terms, "DNSSEC-M" and "DNSSEC-S". See Section 1 for their definition. It also creates associated terms, "DS hash set" in Section 2.3 and "certificate hash set" in Section 2.5.

A great deal of other DNSSEC-related terminology can be found in [I-D.ietf-dnsop-terminology-bis].

2. Protocol Description

2.1. Overview

A DNS client that wants authenticated answers to queries (such as a Security-Aware Recursive Name Server, a Security-Aware Resolver, or a Security-Aware Stub Resolver) can detect that an authoritative server is using DNSSEC-S when it gets the DS records for that authoritative server from the parent zone. If one or more of those DS records have a digest type of DS-DIGEST-TBD, then the client can assume that the authoritative server speaks DNS over TLS as defined in this document, and that the TLS session can be authenticated with the contents of the DS record.

After a TLS session is established, the DNS client and the server speak the normal DNS protocol (using the two-octet length extension for TCP) described in [RFC1035], except that the messages go in the TLS connection, not over UDP or TCP. The DNS client treats all authoritative responses from the server as authenticated because they are received in the TLS session.

DNSSEC-S does not have the problems listed for DNSSEC-M listed in Section 1.1. There is no DNS-specific cryptography in DNSSEC-S; instead, it relies on TLS cryptography which is already well understood. There is no need to sign anything in a zone or to maintain the zone signing keys.

2.2. Service Discovery and Authentication Through a New DS Type

A DNSSEC-S client discovers that an authoritative server supports DNSSEC-S by querying the authoritative server's parent for the DS record set, and noting if any of the DS records are in the format given here. There is a DNSSEC-S DS record in the parent for each TLS certificate that the authoritative server might present when serving DNSSEC-S.

The DS record in the parent zone for DNSSEC-S has the same format as other DS records, but the values in some of the fields are different. For DNSSEC-S, the values MUST be:

- Key Tag - 0
- Algorithm - The signing algorithm of the TLS certificate
- Digest Type - DS-DIGEST-TBD
- Digest - The SHA-256 [SHA256] hash of the authenticating public key in the TLS certificate

The display format is unchanged from Section 5.3 of [RFC4034].

2.3. Client Preparation Before Establishing a TLS Connection

The client validates the DS resource record set in the parent for a zone; if the resource record set does not validate, the client MUST NOT use DNSSEC-S to authenticate values in the zone.

The client creates a data structure called the "DS hash set". Each member of the data structure consists of a hash value obtained from the DS records that have Digest Type DS-DIGEST-TBD. DS records with Digest Type other than DS-DIGEST-TBD MUST NOT be used to create the DS hash set.

If the DS hash set is empty, the client MUST NOT attempt to establish a TLS connection.

2.4. Establishing the TLS Session

The client chooses a record from the DS hash set and starts a TLS session on port 443 with the server at one of the IP addresses from the record. Both client and server MUST use TLS 1.3 [I-D.ietf-tls-tls13].

***

There are two proposals for how to connect to the TLS server: on port 443 using ALPN [RFC7301] and ALPN identifier ALPN-TBD, or on port PORT-TBD (a new port). Both proposals have the same cryptographic properties, and almost identical properties for middleboxes. One or the other needs to be chosen during the discussion of this protocol.

***

Depending on the choice made, one of the following statements will be made in the protocol.

- The client MUST identify the session with ALPN using ALPN-TBD as an identifier to establish the TLS session.

- The client MUST use port PORT-TBD to establish the TLS session.

The registration for one or the other appear in Section 6.

2.5. Authenticating the TLS Session

The client receives the TLS Certificates message from the server. Note that the Certificate message might contain one or more raw public keys (described in [RFC7250]. For each certificate in the Certificates message, the client hashes the public key in the certificate with SHA-256 and adds it to a data structure called the "certificate hash set".

The client then compares the the first element from the DS hash set to the elements in the certificate hash set. If there is a match, the certificate associated with that hash is considered authenticated for TLS. If there is no match, the client compares each of the remaining elements from the DS hash set, searching for a match.

If there is no match for any element in the DS hash set in the certificate hash set, the client MUST terminate the connection with an authentication failure as described in [I-D.ietf-tls-tls13]. In this case, the client MUST NOT use DNSSEC-S with this server for this session, and SHOULD NOT try again to use DNSSEC-S with that server until the signature on the DS resource record set has expired.

2.6. Continuing the Authenticated Session

After a TLS session is established, the DNS client and the server speak the normal DNS protocol (using the two-octet length extension for TCP) described in [RFC1035], except that the messages go in the TLS connection, not over UDP or TCP. The DNS client treats all authoritative responses from the server as authenticated because they are received in the TLS session.

3. Updates to RFC 4033, 4034, and 4035

*** This section will list specific textual updates to the base DNSSEC-M RFCs to allow for DNSSEC-S. The section might get long; if so, it will be broken into sub-sections for easier commenting. This section is a placeholder for later drafts if this work is adopted. ***

- Any authoritative answer from an authoritative server is considered validated if it was obtained with DNSSEC-S. [RFC4035] Section 5 (and maybe 4?). Non-authoritative answers are not considered validated. This will need a careful list of restrictions (Answer section, AA bit set, ...).

- A DS record with Digest Type DS-DIGEST-TBD MUST NOT be used for chaining with DNSKEY records. [RFC4034] Section 5.

- There are probably other updates.

4. Additional Considerations

4.1. Effect of DNSSEC-S on Resolvers that Only Validate with DNSSEC-M

DNSSEC-S is designed to not affect resolvers that only validate with DNSSEC-M. The choice of using DS-DIGEST-TBD in the DS record in the parent will cause a resolver that only knows about DNSSEC-M to think that the child zone is unsigned. Section 5.2 of [RFC4035] says that the following must hold:

"The Algorithm and Key Tag in the DS RR match the Algorithm field and the key tag of a DNSKEY RR in the child zone's apex DNSKEY RRset, and, when the DNSKEY RR's owner name and RDATA are hashed using the digest algorithm specified in the DS RR's Digest Type field, the resulting digest value matches the Digest field of the DS RR."

A validator that only knows DNSSEC-M will not know how to hash "using the digest algorithm specified in the DS RR's Digest Type field" because it will not know the algorithm for DS-DIGEST-TBD.

4.2. Simultaneous Use of Both DNSSEC-M and DNSSEC-S

An authoritative server that uses DNSSEC-S for authentication can also use DNSSEC-M if it wishes. That is, such a server can still answer queries with the DO bit set just as it would have if the queries came over UDP or TCP instead of over TLS. In order to do this, the authoritative server would need at least two DS records in its parent's zone, one for DNSSEC-M and one for DNSSEC-S.

Similarly, a DNS client that is using DNSSEC-S can still request DNSSEC-M records (using the DO bit) and process them as it would if those records were received over UDP or TCP (or out of band). A DNS client may successfully establishes a DNSSEC-S session and receive DNSSEC-M responses that do not validate; the result of such a situation is explicitly undefined in this document.

4.3. Standby Keys

An authoritative server can deploy standby keys whose private keys are not in production. To do this, they simply add NS records whose DNSSEC NS Label is the hash of the not-yet-operational public key, and the A or AAAA records for that name point to the same servers as the operational NS records.

This scheme will make validators that use DNSSEC-S to be slightly slower because they need to perform one extra decoding per standby key, and possibly one extra SHA-256 hash execution per comparison with certificates from the TLS server.

4.4. DoS Attacks on DNSSEC-S

All TLS servers are susceptible to CPU-exhaustion attacks from attackers who can generate traffic that requires the server to do asymmetric computations. Zones that use DNSSEC-S are as susceptible to these denial-of-service attacks as web servers that use HTTPS and mail servers that use SMTPS. TLS 1.3 will be less susceptible to such attacks than earlier versions of TLS, but some attacks are still possible.

Note that DNSSEC-M with online signing of DNSSEC records are also susceptible to DoS attacks. This area has not been heavily studied, but it is possible that DNSSEC-S servers will be less susceptible to CPU exhaustion attacks than DNSSEC-M servers that use online signing.

5. Security Considerations

*** This section will clearly be much longer before the document is finished. Proposed additions are welcome. ***

The security of DNSSEC-S is completely dependent on the ability of DNSSEC-S clients to correctly match the public keys in the TLS certificate messages with the values in the DNSSEC-S NS Labels. A DNSSEC-S client that mis-implements the rules in this protocol is capable of being fooled into validating answers that it should not.

An authoritative server that supports both DNSSEC-S and DNSSEC-M, and sends DNSSEC-M records that cause the authoritative answers to not

validate in DNSSEC-M, will likely have nondeterministic results when queried; see Section 4.2.

DNSSEC-S is susceptible to denial-of-service attacks that DNSSEC-M using offline signing is not; see Section 4.4.

6. IANA Considerations

*** If ALPN is chosen, a template for registering ALPN-TBD will go here. However, if a new port is chosen, a template for registering port PORT-TBD will go here. Only one of these two registrations will appear here. ***

*** The template for registering DS-DIGEST-TBD will go here. ***

--back

7. References

7.1. Normative References

[I-D.ietf-tls-tls13]
Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", draft-ietf-tls-tls13-20 (work in progress), April 2017.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <http://www.rfc-editor.org/info/rfc1035>.

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

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <http://www.rfc-editor.org/info/rfc4033>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <http://www.rfc-editor.org/info/rfc4034>.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <http://www.rfc-editor.org/info/rfc4035>.

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

[RFC7301] Friedl, S., Popov, A., Langley, A., and E. Stephan, "Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension", RFC 7301, DOI 10.17487/RFC7301, July 2014, <http://www.rfc-editor.org/info/rfc7301>.

[SHA256] National Institute of Standards and Technology, "Secure Hash Standard (SHS), FIPS 180-4", August 2015, <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>.

7.2. Informative References

[I-D.ietf-dnsop-terminology-bis] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", draft-ietf-dnsop-terminology-bis-05 (work in progress), March 2017.

Authors' Addresses

Paul Hoffman
ICANN

Email: paul.hoffman@icann.org

Matt Larson
ICANN

Email: matt.larson@icann.org