Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2016

D. Wessels
Verisign
W. Kumari
Google

March 9, 2016

The EDNS Key Tag Option
draft-ietf-dnsop-edns-key-tag-01

Abstract

The DNS Security Extensions (DNSSEC) were developed to provide origin authentication and integrity protection for DNS data by using digital signatures. These digital signatures can be verified by building a chain-of-trust starting from a trust anchor and proceeding down to a particular node in the DNS. This document specifies a way for validating end-system resolvers to signal to a server which keys are referenced in their chain-of-trust. The extensions allow zone administrators to monitor the progress of rollovers in a DNSSEC-signed zone.

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 September 10, 2016.

Table of Contents

1. Introduction

The DNS Security Extensions (DNSSEC) [RFC4033], [RFC4034] and [RFC4035] were developed to provide origin authentication and integrity protection for DNS data by using digital signatures. DNSSEC uses Key Tags to efficiently match signatures to the keys from which they are generated. The Key Tag is a 16-bit value computed from the RDATA portion of a DNSKEY RR using a formula not unlike a ones-complement checksum. RRSIG RRs contain a Key Tag field whose value is equal to the Key Tag of the DNSKEY RR that validates the signature.

Likewise, Delegation Signer (DS) RRs also contain a Key Tag field whose value is equal to the Key Tag of the DNSKEY RR to which it refers.

This draft sets out to specify a way for validating end-system resolvers to tell a server in a DNS query which DNSSEC key(s) they would use to validate the expected response. This is done using the new EDNS option specified below in Section 4 for use in the OPT meta-RR [RFC6891]. This new EDNS option code is OPTIONAL to implement and use.

This proposed EDNS option serves to measure the acceptance and use of new trust anchors and key signing keys (KSKs). This signaling option can be used by zone administrators as a gauge to measure the successful deployment of new keys. This is of particular interest for the DNS root zone in the event of key and/or algorithm rollovers that rely on [RFC5011] to automatically update a validating end-system's trust anchor.

[FOR WG DISCUSSION: There is some reluctance within the working group to use EDNS0 to convey the key tags upstream. In particular there is a concern that middleboxes might block messages with unknown option codes. Also since EDNS0 is hop-by-hop, middleboxes and un-upgraded recursives won't necessarily know whether or not the edns-key-tag options should be forwarded. RFC6891 says: "OPTION-CODE values not understood by a responder or requestor MUST be ignored." draft-wkumari-dnsop-trust-management proposed encoding this information in query names, but sufficient issues with this approach were discovered that the authors of the above decided to abandon this work. The authors of draft-ietf-dnsop-edns-key-tag are willing to consider this alternative if so guided by the working group.]

This draft does not seek to introduce another process for rolling keys or updating trust anchors. Rather, this document specifies a means by which a client query can signal the set of keys that a client uses for DNSSEC validation.

2. Requirements 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 [RFC2119].

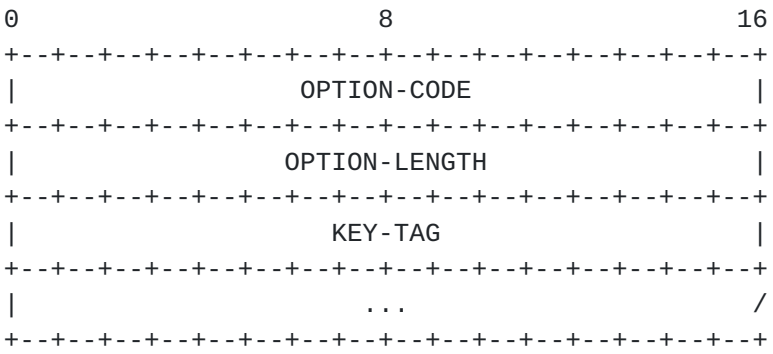
3. Terminology

Trust Anchor: A configured DNSKEY RR or DS RR hash of a DNSKEY RR. A validating security-aware resolver uses this public key or hash as a starting point for building the authentication chain to a signed DNS response. In general, a validating resolver will have to obtain the initial values of its trust anchors via some secure or trusted means outside the DNS protocol. Presence of a trust anchor also implies that the resolver should expect the zone to which the trust anchor points to be signed. (quoted from [RFC4033] Section 2)

Key Tag: A 16-bit integer that identifies and enables efficient selection of DNSSEC public keys. A Key Tag value can be computed over the RDATA of a DNSKEY RR. The Key Tag field in the RRSIG and DS records can be used to help select the corresponding DNSKEY RR efficiently when more than one candidate DNSKEY RR is available. For most algorithms the Key Tag is a simple 16-bit modular sum of the DNSKEY RDATA. See [RFC4034] Appendix B.

4. Option Format

The edns-key-tag option is encoded as follows:



where:

OPTION-CODE: The EDNS0 option code assigned to edns-key-tag, [TBD].

OPTION-LENGTH: The value 2 x number of key-tag values present.

KEY-TAG: One or more 16-bit Key Tag values ([RFC4034], Appendix B).

5. Use By Queriers

A validating end-system resolver sets the edns-key-tag option in the OPT meta-RR when sending a DNSKEY query. The validating end-system resolver SHOULD also set the DNSSEC OK bit [RFC4034] to indicate that it wishes to receive DNSSEC RRs in the response.

A DNS client MUST NOT include the edns-key-tag option for non-DNSKEY queries.

The KEY-TAG value(s) included in the edns-key-tag option represent the Key Tag of the Trust Anchor or DNSKEY RR that will be used to validate the expected response. When the client sends a DNSKEY query, the edns-key-tag option represents the Key Tag(s) of the KSK(s) of the zone for which the server is authoritative. A validating end-system resolver learns the Key Tag(s) of the KSK(s) from the zone's DS record(s) (found in the parent), or from a configured trust anchor.

A DNS client SHOULD include the edns-key-tag option when issuing a DNSKEY query for a zone corresponding to a configured Trust Anchor.

A DNS client MAY include the edns-key-tag option when issuing a DNSKEY query for a non-Trust Anchor zone (i.e., Key Tags learned via DS records). Since some DNSSEC validators implement bottom-up validation, non-Trust Anchor Key Tags zone might not be known at the time of the query. Such a validator can include the edns-key-tag option based on previously cached data.

A DNS client MUST NOT include Key Tag(s) for keys which are not learned via either configured Trust Anchor or DS records.

Since the edns-key-tag option is only set in the query, if a client sees these options in the response, no action needs to be taken and the client MUST ignore the option values.

5.1. Stub Resolvers

Typically, stub resolvers rely on an upstream recursive server (or cache) to provide a response. Optimal setting of the edns-key-tag option depends on whether the stub resolver elects to perform its own

validation.

5.1.1. Validating Stub Resolvers

A validating stub resolver sets the DNSSEC OK (DO) bit [RFC4034] to indicate that it wishes to receive additional DNSSEC RRs (i.e., RRSIG RRs) in the response. Such validating resolvers SHOULD include the edns-key-tag option in the OPT RR when sending a DNSKEY query.

5.1.2. Non-validating Stub Resolvers

The edns-key-tag option MUST NOT be included by non-validating stub resolvers.

5.2. Recursive Resolvers

5.2.1. Validating Recursive Resolvers

A validating recursive resolver is, by definition, configured with at least one trust anchor. Thus, a recursive resolver SHOULD include the edns-key-tag option in its DNSKEY queries as described above.

In addition, the clients of a validating recursive resolver might be configured to do their own validation, with their own trust anchor(s). When a validating recursive resolver receives a query that includes the edns-key-tag option with a Key Tag list that differs from its own, it SHOULD forward both the client's Key Tag list as well as its own. When doing so, the recursive resolver SHOULD transmit the two Key Tag lists using separate instances of the edns-key-tag option code in the OPT meta-RR. For example, if the recursive resolver's Key Tag list is (19036, 12345) and the stub/client's list is (19036, 34567), the recursive would include the edns-key-tag option twice: Once with values (19036, 12345) and once with values (19036, 34567).

A validating recursive resolver MAY combine stub/client Key Tag values from multiple incoming queries into a single outgoing query. It is RECOMMENDED that implementations place reasonable limits on the number of Key Tags to include in the outgoing edns-key-tag option.

If the client included the DO and Checking Disabled (CD) bits, but did not include the edns-key-tag option in the query, the validating recursive resolver MAY include the option with its own Key Tag values in full.

Validating recursive resolvers MUST NOT set the edns-key-tag option in the final response to the stub client.

5.2.2. Non-validating Recursive Resolvers

Recursive resolvers that do not validate responses SHOULD copy the edns-key-tag option seen in received queries, as they represent the wishes of the validating downstream resolver that issued the original query.

6. Use By Responders

An authoritative name server receiving queries with the edns-key-tag option MAY log or otherwise collect the Key Tag values to provide information to the zone operator.

A responder MUST NOT include the edns-key-tag option in any DNS response.

7. IANA Considerations

The IANA is directed to assign an EDNS0 option code for the edns-key-tag option from the DNS EDNS0 Option Codes (OPT) registry as follows:

Value	Name	Status	Reference
[TBA]	edns-key-tag	Optional	[This document]

8. Security Considerations

This document specifies a way for a client to signal its trust anchor knowledge to a cache or server. The signals are optional codes contained in the OPT meta-RR used with EDNS. The goal of these options is to signal new trust anchor uptake in clients to allow zone administrators to know when it is possible to complete a key rollover in a DNSSEC-signed zone.

There is a possibility that an eavesdropper or server could infer the validator in use by a client by the Key Tag list seen in queries. This may allow an attacker to find validators using old, possibly broken, keys. It could also be used to identify the validator or narrow down the possible validator implementations in use by a client, which could have a known vulnerability that could be exploited by the attacker.

Consumers of data collected from the edns-key-tag option are advised

that provided Key Tag values might be "made up" by some DNS clients with malicious or at least mischievous intentions. For example, an attacker with sufficient resources might try to generate large numbers of queries including only old Key Tag values, with the intention of delaying the completion of a key rollover.

DNSSEC does not require keys in a zone to have unique Key Tags. During a rollover there is a small possibility that an old key and a new key will have identical Key Tag values. Zone operators relying on the edns-key-tag mechanism SHOULD take care to ensure that new keys have unique Key Tag values.

9. Privacy Considerations

This proposal adds additional, optional "signaling" to DNS queries in the form of Key Tag values. While Key Tag values themselves are not considered private information, it may be possible for an eavesdropper to use Key Tag values as a fingerprinting technique to identify particular DNS validating clients. This may be especially true if the validator is configured with trust anchor for zones in addition to the root zone.

A validating end-system resolver need not transmit the edns-key-tag option in every applicable query. Due to privacy concerns, such a resolver MAY choose to transmit the edns-key-tag option for a subset of queries (e.g., every 25th time), or by random chance with a certain probability (e.g., 5%).

Implementations of this specification MAY be administratively configured to only transmit the edns-key-tag option for certain zones. For example, the software's configuration file may specify a list of zones for which use of the option is allowed or denied. Since the primary motivation for this specification is to provide operational measurement data for root zone key rollovers, it is RECOMMENDED that implementations at least include the edns-key-tag option for root zone DNSKEY queries.

10. Acknowledgments

This document was inspired by and borrows heavily from [RFC6975] by Scott Rose and Steve Crocker. The authors would like to thank Casey Deccio, Burt Kalisky, Bob Harold, Tim Wicinski, Suzanne Woolf, and other members of the dnsop working group for their input.

11. References

11.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <http://www.rfc-editor.org/info/rfc1034>.

[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>.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <http://www.rfc-editor.org/info/rfc6891>.

11.2. Informative References

[RFC5011] StJohns, M., "Automated Updates of DNS Security (DNSSEC) Trust Anchors", STD 74, RFC 5011, DOI 10.17487/RFC5011, September 2007, <http://www.rfc-editor.org/info/rfc5011>.

[RFC6975] Crocker, S. and S. Rose, "Signaling Cryptographic Algorithm Understanding in DNS Security Extensions (DNSSEC)", RFC 6975, DOI 10.17487/RFC6975, July 2013, <http://www.rfc-editor.org/info/rfc6975>.

Appendix A. Changes / Author Notes.

[RFC Editor: Please remove this section before publication]

From -00 to -01:

- Changed how a recursive should combine a stub's key tag values with its own. Previously it was to be a union of key tag values. Now it is a separate instance of the option code for recursive and stub.

- Added Warren as coauthor.

Authors' Addresses

Duane Wessels
Verisign
12061 Bluemont Way
Reston, VA 20190
United States

Phone: +1 703 948-3200
Email: dwessels@verisign.com
URI: http://verisigninc.com

Warren Kumari
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
United States

Email: warren@kumari.net