Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: January 04, 2018

S. Huque
Salesforce
H. Shulman
Fraunhofer Institute

July 03, 2017

Algorithm Negotiation in DNSSEC
draft-huque-dnssec-alg-nego-00

Abstract

This document specifies a DNS extension that allows a DNS client to specify a list of DNSSEC algorithms, in preference order, that the client desires to use. A DNS server upon receipt of this extension can choose to selectively respond with DNSSEC signatures using the most preferred algorithm they support. This mechanism may make it easier for DNS zone operators to support signing zone data simultaneously with multiple DNSSEC algorithms, without significantly increasing the size of DNS responses. It will also allow an easier way to transition to new algorithms while still retaining support for older DNS validators that do not yet support the new algorithms.

Table of Contents

1. Introduction

The DNS Security Extensions (DNSSEC) specifications [RFC4033] [RFC4034] [RFC4035] support multiple signature algorithms. A DNS zone can be signed simultaneously with multiple algorithms, but there is no provision in the current specifications to negotiate the selective delivery of signatures of a specific algorithm in DNS responses.

In contrast, many other security protocols, like TLS, IKE, SSH and others, support an algorithm or cipher suite negotiation mechanism to enable the client and server to select the "best" algorithm they jointly support.

This means that DNS servers have to send responses with signatures of all algorithms that the requested data are signed with, which can result in significantly large responses. Not only is this inefficient in terms of the additional communication and processing overhead, but it often causes a variety of operational problems. Most DNS queries and responses utilize UDP transport today. While the EDNS0 specification can support very large DNS over UDP payload sizes, once they exceed the common Internet Path MTU (typically about 1,500 octets), they need to be fragmented at the IP layer. Many

studies [add citations] have shown that IP fragmentation does not work reliably on today's Internet, because fragments are often blocked by network security devices.

DNS can run over other transports that can obviate the IP fragmentation problem, such as TCP (with Path MTU discovery or a suitably configured Maximum Segment Size) and TLS. In fact, some operators are known to truncate a DNS payload in preference to emitting a a response that is likely to be fragmented, instructing the client to re-query over TCP. However, these alternative transports have not been widely deployed in the field, and there is some reluctance by operators to make wide use of TCP or TLS because of their added processing and performing costs. This situation may change over time, but at least today, the dominant transport for DNS query and response remains UDP.

The response size issue is also a significant barrier to the introduction of new algorithms in DNSSEC. As can be readily seen from the RSA to ECDSA transition, very few zones have transitioned from RSA to ECDSA, and furthermore, very few have been willing to sign their zones with multiple algorithms. Newer DNSSEC algorithms have already appeared or are being proposed: EdDSA [RFC8080], NSEC5 [nsec5], and some time time in the near there will be post quantum algorithms. These will likely require zone operators to deploy multiple algorithms, and support older algorithms for an extended period of time until the population of validators have upgraded themselves to support the newer algorithms.

This document proposes a new mechanism by which a DNS client when sending a query can indicate an ordered list of DNSSEC signature algorithms it desires to use. The DNS server can use this information to selectively construct a response with only the signatures using the most preferred algorithm that it supports.

2. DNSSEC Preferred Algorithms Option

The EDNS0 specification outlined in [RFC6891] defines a way to include new options using a standardized mechanism. These options are contained in the RDATA of the OPT meta resource record. This document defines a new EDNS0 option called "DNSSEC Preferred Algorithms" used by a client to indicate an ordered list of DNSSEC signature algorithms that it supports and prefers. This option can appear only once in an OPT RR.

```
0                       8                      16
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                  OPTION-CODE                  |
```

```
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                  LIST-LENGTH                  |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|       ALG-CODE        |        ...            |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

OPTION-CODE is the code (TBD) assigned by IANA for this EDNS0 option.

LIST-LENGTH is the length of the list of digital signatures or hash algorithm codes in octets. Each algorithm code occupies a single octet.

ALG-CODE is the list of assigned values of DNSSEC zone signature algorithms that the client prefers to be used. The algorithms are listed in the order preferred by the client.

3. Why not use RFC 6975 for Algorithm Signaling?

The new EDNS0 option described in this document is very similar to the DNSSEC Algorithm Understood (DAU) option defined in [RFC6975] (Signaling Cryptographic Algorithm Understanding). That specification has not seen much adoption or even implementation, and it has been suggested that it could be repurposed to implement the algorithm negotiation mechanism described in this document.

This document proposes a new option instead, because the RFC6975 option could not be reused without significantly revising its semantics. For example, it currently says that the list of algorithm codes is unordered, and that the server must not infer any ordering or preferences from the list. Furthermore, it states that the option must not trigger any special processing on the server side.

4. Changes to Clients

A client is defined to be any DNS speaker that issues a query, e.g. a stub resolver, a resolver issuing outbound queries to authoritative servers, or to other resolvers etc.

A client implementing this specification and configured to use it, adds an EDNS0 DNSSEC Preferred Algorithms option to the OPT Pseudo Resource Record in the Additional Section of the query, listing its desired DNSSEC algorithm numbers in preferred order. It only makes sense to add this option if the client is requesting DNSSEC signatures, so the DNSSEC-OK bit in the EDNS Flags field MUST also be set.

As a general rule, to maximize security, the client should prefer stronger DNSSEC algorithms to weaker ones.

5. Changes to Servers

A server is defined to be any DNS speaker that sends DNS responses, e.g. an authoritative server, or a resolver when answering queries from downstream clients.

Upon receipt of a query with the DNSSEC-OK bit set, and the DNSSEC Preferred Algorithms EDNS0 option, an Authoritative Server SHOULD include in its response, DNSSEC signatures using only the most preferred algorithm that it supports. It also includes the Preferred Algorithms EDNS0 option in the response, to indicate that it recognizes the option, and should include the list of algorithms supported at the server.

If an Authoritative Server has no algorithms in common with the Preferred Algorithms list in the incoming query, it MUST send back a SERVFAIL response (Response Code 2). This response MUST contain the list of algorithms supported by the server in the EDNS0 Preferred Algorithms option.

If a resolver receives a query from a downstream validating client with a Preferred Algorithms list different from its own, then it should send outbound queries with the client's preferred list, and return answers appropriately.

6. Cache Considerations

A Validating Resolver answering queries with the DNSSEC-OK bit set from data in its cache needs to take a few additional steps. If the query does not include the Preferred Algorithms option, and the resolver has selectively cached signatures of a subset of algorithms supported by the zone containing the query domain name, then it MUST re-send outbound queries to the authoritative server without the Preferred Algorithms option in order to retrieve the entire set of signatures for the query. If the query includes the Preferred Algorithms option, but prefers algorithms known to be supported for the name, but different from what has been cached, the resolver MUST again send outbound queries to retrieve answers with signatures the client prefers, by copying the client's Preferred Algorithms option into the outbound query.

7. Preventing Downgrade Attacks

There is no cryptographic integrity protection of EDNS0 options. In theory, Transaction Signatures [RFC2845] could be deployed to

integrity protect the entire query message with per-client keys in closed populations of DNS speakers, but this is not a viable mechanism in the general case of arbitrary DNS clients and servers on the Internet.

Hence an active man-in-the-middle attacker could strip out stronger algorithms from the client's supported algorithms list and force the server to send back signatures with a weaker algorithm than it might have otherwise sent.

In order to detect such attacks, the client SHOULD compare the zone signing algorithms listed in the zone's authenticated DNSKEY RRset, and the preferred list in the query that it sent, to the algorithms seen in the response signatures. If signatures by the most preferred algorithm they have in common have not been sent, this may indicate an algorithm downgrade attack.

QUESTION: The server may have its own algorithm ranking policy, that might differ from the client. Should we allow the server to select its highest ranked algorithm that it shares in common with the client's list, regardless of the client's preference? This is how some other security protocols do it. But it will likely make it harder for the (DNS) client to reliably detect downgrade attacks, unless there is a common notion of ranking. One way of addressing this is to define a new zone apex resource record that lists the zone's preferred order of algorithm numbers. This could be queried by resolvers in parallel with DNSKEY RRset queries as part of the iterative resolution process, and similarly cached and refreshed.

8. Dealing with Proxies and Middleboxes

EDNS is a hop-by-hop mechanism. Hence all DNS speakers in the path from the querier invoking this option to the responding server need to support this mechanism for it to work correctly. DNS proxies along the path that transparently relay requests and responses, and largely comply with the implementation guidelines described in [RFC5625] should not be a problem. But more complicated proxies, middleboxes, forwarding resolvers, etc that actively interpret DNS messages, but do not understand this new option, will likely strip off the unrecognized option in their outbound queries. The result will be that the responding server will send back signatures made with the full set of algorithms.

There is always a danger that a misbehaving middlebox might block or drop a DNS packet with an unrecognized EDNS option, but this is a threat that applies to almost all DNS extension proposals. Deployment of new DNS options provides an opportunity to identify and remove or fix such misbehaving devices.

An alternative end-to-end mechanism is described in [dnssec-nego] to workaround DNS speaking middleboxes that haven't been upgraded to recognize this option. It involves the client encoding the ordered list of algorithms in a sequence of labels prepended to the query name, and the addition of a new DNSKEY RR (with a new algorithm number) at the authoritative server to signal to clients that the server recognizes these specially constructed query names. No further details are provided in this document, but could be incorporated in future revisions if there is interest in developing that solution.

9. Acknowledgements

This specification builds on earlier work on DNSSEC algorithm negotiation by Amir Herzberg and Haya Shulman in [dnssec-nego].

10. Security Considerations

[TODO]

11. IANA Considerations

This specification requires the registration of a new value in the DNS EDNS0 Option Code Registry, maintained by IANA.

12. References

12.1. Normative References

[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.

[RFC5625] Bellis, R., "DNS Proxy Implementation Guidelines", BCP 152, RFC 5625, DOI 10.17487/RFC5625, August 2009, <http://www.rfc-editor.org/info/rfc5625>.

[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>.

[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>.

12.2. Informative References

[RFC2845] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, DOI 10.17487/RFC2845, May 2000, <http://www.rfc-editor.org/info/rfc2845>.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

[RFC8080] Sury, O. and R. Edmonds, "Edwards-Curve Digital Security Algorithm (EdDSA) for DNSSEC", RFC 8080, DOI 10.17487/RFC8080, February 2017, <http://www.rfc-editor.org/info/rfc8080>.

[dnssec-nego] Herzberg, A. and H. Shulman, "Cipher-Suite Negotiation for DNSSEC: Hop-by-Hop or End-to-End?", in IEEE Internet Computing, February 2015, <http://ieeexplore.ieee.org/document/7031814/>.

[nsec5] Vcelak, J., Goldberg, S., Papadopoulos, D., Huque, S., and D. Lawrence, "NSEC5, DNSSEC Authenticated Denial of Existence", , <https://tools.ietf.org/html/draft-vcelak-nsec5>.

Authors' Addresses

Shumon Huque
Salesforce

Email: shuque@gmail.com

Haya Shulman
Fraunhofer Institute

Email: haya.shulman@gmail.com