Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: November 18, 2018

D. Wessels
P. Barber
M. Weinberg
Verisign
W. Kumari
Google
W. Hardaker
USC/ISI

May 17, 2018

Message Digest for DNS Zones
draft-wessels-dns-zone-digest-01

Abstract

This document describes a protocol and DNS Resource Record used to provide a message digest over DNS zone data. In particular, it describes how to compute, sign, represent, and use the message digest to verify the contents of a zone for accuracy and completeness. The ZONEMD Resource Record type is introduced for conveying the message digest data.

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 https://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 November 18, 2018.

Table of Contents

1. Introduction

In the DNS, a zone is the collection of authoritative resource records (RRs) sharing a common origin ([RFC7719]), which can be distributed from primary to secondary name servers. Zones are often stored as files on disk in the so-called master file format [RFC1034]. Sometimes zones are distributed outside of the DNS, with such protocols as FTP, HTTP, rsync, and so on. While zone files are self-contained, currently there is no way to verify the authenticity of a stand-alone zone file without relying on an external checksum or signature.

This document introduces a new RR type that serves as a cryptographic message digest of the data in a zone file. It allows a receiver of the zone file to verify the zone file's authenticity, especially when used in combination with DNSSEC.

DNS transaction signatures (TSIG [RFC2845]) uses a message digest to protect individual query and response messages. TSIG is generally used to authenticate and validate UPDATE [RFC2136] AXFR [RFC5936], and IXFR [RFC1995] messages. However, TSIG's protections are ephemeral, existing only "on the wire," and are not retained after the transaction is complete. Additionally, TSIG utilizes shared secret keys, which are not available to third parties.

The technique described in this document makes the message digest a part of the zone file itself, allowing anything to verify the zone file as a whole, no matter how it is transmitted.

DNSSEC provides three strong security guarantees relevant to this protocol:

1. whether or not to expect DNSSEC records in the zone,

2. whether or not to expect a ZONEMD record in a signed zone, and

3. whether or not the ZONEMD record has been altered since it was signed.

FOR DISCUSSION: currently this document does not require DNSSEC. Should it be a prerequisite?

The motivation for including a message digest in a zone file comes largely from the DNS root zone. At the time of this writing, there is increased attention to the idea of widely distributing the root zone, beyond the root server system. [RFC7706] describes how a recursive resolver can serve the root zone via a loopback address. As the root zone spreads beyond its traditional deployment

boundaries, the need for verification of the completeness of the zone contents becomes increasingly important.

Nothing in this specification, however, is specific to the root zone. The zone digest is designed to work with any DNS zone.

This specification is OPTIONAL to implement by both publishers and consumers of zone file data.

2. 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].

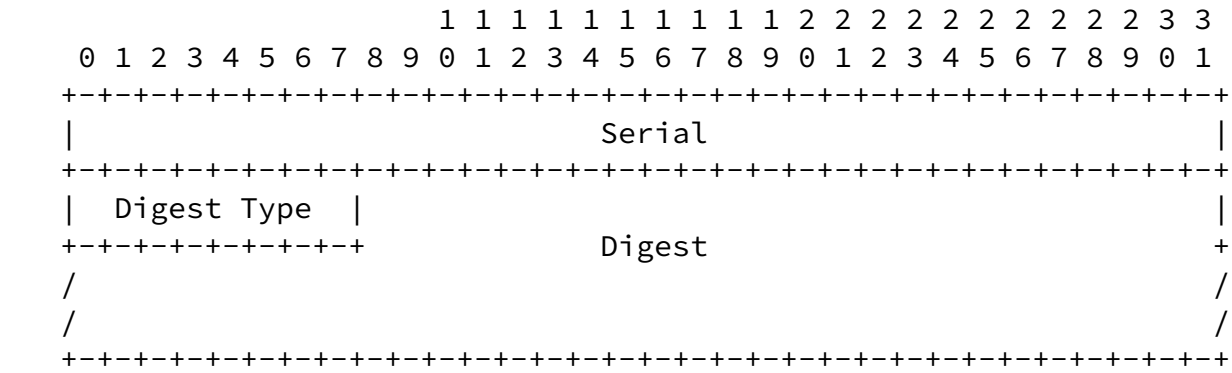
3. The ZONEMD Resource Record

This section describes the ZONEMD Resource Record, including its fields, wire format, and presentation format. The Type value for the ZONEMD RR is TBD. The ZONEMD RR is class independent. The RDATA of the resource record consists of three fields: Serial, Digest Type, and Digest.

FOR DISCUSSION: This document is currently written as though a zone MUST NOT contain more than one ZONEMD RR. Having exactly one ZONEMD record per zone simplifies this protocol and eliminates confusion around downgrade attacks, at the expense of algorithm agility.

3.1. ZONEMD RDATA Wire Format

The ZONEMD RDATA wire format is encoded as follows:



3.1.1. The Serial Field

The Serial field is a 32-bit unsigned integer in network order. It is equal to the serial number from the zone's SOA record ([RFC1035] section 3.3.13) for which the message digest was generated.

FOR DISCUSSION: the serial number is included in order to make DNS response messages of type ZONEMD meaningful. Without the serial number, a stand-alone ZONEMD digest has no association to any particular zone file. If there is agreement that ZONEMD responses are not useful, this field could be removed. See also the end of Security Considerations.

3.1.2. The Digest Type Field

The Digest Type field is an 8-bit unsigned integer, with meaning equivalent to the Digest Type of the DS resource record, as defined in section 5.1.3 of [RFC4034].

The status of ZONEMD digest types (e.g., mandatory, optional, deprecated) SHALL always match the status for DS records. This information can be found in the IANA protocol registry for DS digest types [iana-ds-digest-types].

At the time of this writing the following digest types are defined:

Value	Description	Status	Reference

1	SHA1	Mandatory	[RFC3658]
2	SHA256	Mandatory	[RFC4509]
3	GOST R 34.11-94	Optional	[RFC5933]
4	SHA384	Optional	[RFC6605]

Table 1: Digest Types

3.1.3. The Digest Field

The Digest field is a variable-length sequence of octets containing the message digest. Section 4 describes how to calculate the digest for a zone. Section 5 describes how to use the digest to verify the contents of a zone.

3.2. ZONEMD Presentation Format

The presentation format of the RDATA portion is as follows:

The Serial field MUST be represented as an unsigned decimal integer.

The Digest Type field MUST be represented as an unsigned decimal integer.

The Digest MUST be represented as a sequence of case-insensitive hexadecimal digits. Whitespace is allowed within the hexadecimal text.

3.3. ZONEMD Example

The following example shows a ZONEMD RR.

```
example.com. 86400 IN ZONEMD ( 2018031500 4 FEBE3D4CE2EC2FFA4BA9
                                            9D46CD69D6D29711E552
                                            17057BEE7EB1A7B641A4
                                            7BA7FED2DD5B97AE499F
                                            AFA4F22C6BD647DE )
```

4. Calculating the Digest

4.1. Canonical Format and Ordering

Calculation of the zone digest REQUIRES the RRs in a zone to be in a consistent format and ordering. Correct ordering of the zone depends on (1) ordering of owner names in the zone, (2) ordering of RRsets with the same owner name, and (3) ordering of RRs within an RRset.

This specification adopts DNSSEC's canonical ordering for names (Section 6.1 of [RFC4034]), and canonical ordering for RRs within an RRset (Section 6.3 of [RFC4034]). It also adopts DNSSEC's canonical RR form (Section 6.2 of [RFC4034]). However, since DNSSEC does not define a canonical ordering for RRsets having the same owner name, that ordering is defined here.

4.1.1. Order of RRsets Having the Same Owner Name

For the purposes of calculating the zone digest, RRsets having the same owner name MUST be numerically ordered by their numeric RR TYPE.

4.1.2. Special Considerations for SOA RRs

When AXFR is used to transfer zone data, the first and last records are always the SOA RR ([RFC5936] Section 2.2). Because of this, zone files on disk often contain two SOA RRs. When calculating the zone digest, the first SOA RR MUST be included and any subsequent SOA RRs MUST NOT be included.

Additionally, per established practices, the SOA record is generally the first record in a zone file. However, according to the requirement to sort RRsets with the same owner name by type, the SOA RR (type value 2) might not be first in the digest calculation. If

the zone has an A RR (type value 1) at the apex, it MUST be processed before the SOA RR.

4.2. Add ZONEMD Placeholder

In preparation for calculating the zone digest, any existing ZONEMD records MUST first be deleted from the zone.

Prior to calculation of the digest, and prior to signing with DNSSEC, a placeholder ZONEMD record MUST be added to the zone. This serves two purposes: (1) it allows the digest to cover the Serial and Digest Type field values, and (2) ensures that appropriate denial-of-existence (NSEC, NSEC3) records are created if the zone is signed with DNSSEC.

In the placeholder record, the Serial field MUST be set to the current SOA Serial. The Digest Type field MUST be set to the value for the chosen digest algorithm. The Digest field MUST be set to all zeroes and of length appropriate for the chosen digest algorithm.

4.3. Optionally Sign the Zone

Following addition of the placeholder record, the zone MAY be signed with DNSSEC. Note that when the digest calculation is complete, and the ZONEMD record is updated, the signature(s) for that record MUST be recalculated and updated as well. Therefore, the signer is not required to calculate a signature over the placeholder record at this step in the process, but it is harmless to do so.

4.4. Calculate the Digest

The zone digest is calculated by concatenating the canonical on-the-wire form of RRs in the zone, in the order described above, subject to the inclusion/exclusion rules described below, and then applying the digest algorithm:

digest = digest_algorithm(RR(1) | RR(2) | RR(3) | ...)

where "|" denotes concatenation, and

RR(i) = owner | type | class | TTL | RDATA length | RDATA

4.4.1. Inclusion/Exclusion Rules

When calculating the digest, the following inclusion/exclusion rules apply:

- More than one SOA MUST NOT be included.

- The placeholder ZONEMD RR MUST be included.

- If the zone is signed, DNSSEC RRs MUST be included, except:

- The RRSIG covering ZONEMD MUST NOT be included.

4.5. Update ZONEMD RR

Once the zone digest has been calculated, its value is then copied to the Digest field of the ZONEMD record.

If the zone is signed with DNSSEC, the appropriate RRSIG records covering the ZONEMD record MUST then be added. Because the ZONEMD placeholder was added prior to signing, the zone will already have the appropriate denial-of-existence (NSEC, NSEC3) records.

5. Verifying Zone Message Digest

The recipient of a zone that has a message digest record can verify the zone by calculating the digest as follows:

1. The verifier SHOULD first determine whether or not to expect DNSSEC records in the zone. This can be done by examining locally configured trust anchors, or querying for (and validating) DS RRs in the parent zone. For zones that are provably unsigned, digest validation continues at step 4 below.

2. For zones that are provably signed, the existence of the ZONEMD record MUST be verified. If the ZONEMD record provably does not exist, digest verification cannot be done. If the ZONEMD record does provably exist, but is not found in the zone, digest verification MUST NOT be considered successful.

3. For zones that are provably signed, the SOA RR and ZONEMD RR(set) MUST have valid signatures, chaining up to a trust anchor. If DNSSEC validation of the SOA or ZONEMD records fails, digest verification MUST NOT be considered successful.

4. The SOA Serial field MUST exactly match the ZONEMD Serial field. If the fields to not match, digest verification MUST NOT be considered successful.

5. The ZONEMD Digest Type field MUST be checked. If the verifier does not support the given digest type, it SHOULD report that the zone digest could not be verified due to an unsupported algorithm.

6. The zone digest is calculated using the algorithm described in Section 4.4. Note in particular that digested ZONEMD RRs MUST be placeholders and their RRSIGs are not included in the digest.

7. The calculated digest is compared to the received digest. If the two digest values match, verification is considered successful. Otherwise, verification MUST NOT be considered successful.

8. If the zone is to be served and transferred, the original (not placeholder) ZONEMD RRs MUST be sent to recipients so that downstream clients can verify the zone.

6. IANA Considerations

6.1. ZONEMD RRtype

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

6.2. ZONEMD Digest Type

The ZONEMD Digest Type field has the same semantics as the DS RR Digest Type field. Thus, it does not add new IANA protocol registry requirements.

7. Security Considerations

7.1. Attacks Against the Zone Digest

The zone digest allows the receiver to verify that the zone contents haven't been modified since the zone was generated/published. Verification is strongest when the zone is also signed with DNSSEC. An attacker, whose goal is to modify zone content before it is used by the victim, may consider a number of different approaches.

The attacker might perform a downgrade attack to an unsigned zone. This is why Section 5 RECOMMENDS that the verifier determine whether or not to expect DNSSEC signatures for the zone in step 1.

The attacker might perform a downgrade attack by removing the ZONEMD record. This is why Section 5 REQUIRES that the verifier checks DNSSEC denial-of-existence proofs in step 2.

The attacker might alter the Digest Type or Digest fields of the ZONEMD record. Such modifications are detectable only with DNSSEC

validation.

7.2. Attacks Utilizing the Zone Digest

Nothing in this specification prevents clients from making, and servers from responding to, ZONEMD queries. One might consider how well ZONEMD responses could be used in a distributed denial-of-service amplification attack.

The ZONEMD RR is moderately sized, much like the DS RR. A single ZONEMD RR contributes approximately 40 to 65 octets to a DNS response, for currently defined digest types. Certainly other query types result in larger amplification effects (i.e., DNSKEY).

FOR DISCUSSION: The primary purpose of the ZONEMD record is to verify a zone file prior to being loaded or served by a name server. We could allow a name server implementation to respond to ZONEMD queries with the REFUSED RCODE without loss of functionality. Note that refusal would prevent ensuring that a zone-walk is complete.

8. Privacy Considerations

This specification has no impacts on user privacy.

9. Acknowledgments

The authors wish to thank David Blacka, Scott Hollenbeck, and Rick Wilhelm for providing feedback on early drafts of this document.

10. Implementation Status

10.1. Authors' Implementation

The authors are currently working on an implementation in C, using the ldns library [ldns]. This implementation is able to perform the following functions:

- Read input zone file, output zone file with ZONEMD placeholder.
- Compute zone digest over signed zone file and update ZONEMD record.

- Re-compute DNSSEC signature over ZONEMD record.

- Verify zone digest from input zone file.

The authors expect to be able to release this implementation as open source following submission of this Internet-Draft.

11. Change Log

RFC Editor: Please remove this section.

This section lists substantial changes to the document as it is being worked on.

From -00 to -01:

- Removed requirement to sort by RR CLASS.

- Added Kumari and Hardaker as coauthors.

- Added Change Log section.

- Minor clarifications and grammatical edits.

12. References

12.1. Normative References

[iana-ds-digest-types]
IANA, "Delegation Signer (DS) Resource Record (RR) Type Digest Algorithms", April 2012, <https://www.iana.org/assignments/ds-rr-types/ds-rr-types.xhtml>.

[ldns] NLNet Labs, "The ldns Library", March 2018, <https://www.nlnetlabs.nl/projects/ldns/>.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987,

<https://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://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, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3658] Gudmundsson, O., "Delegation Signer (DS) Resource Record (RR)", RFC 3658, DOI 10.17487/RFC3658, December 2003, <https://www.rfc-editor.org/info/rfc3658>.

[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, <https://www.rfc-editor.org/info/rfc4034>.

[RFC4509] Hardaker, W., "Use of SHA-256 in DNSSEC Delegation Signer (DS) Resource Records (RRs)", RFC 4509, DOI 10.17487/RFC4509, May 2006, <https://www.rfc-editor.org/info/rfc4509>.

[RFC5933] Dolmatov, V., Ed., Chuprina, A., and I. Ustinov, "Use of GOST Signature Algorithms in DNSKEY and RRSIG Resource Records for DNSSEC", RFC 5933, DOI 10.17487/RFC5933, July 2010, <https://www.rfc-editor.org/info/rfc5933>.

[RFC6605] Hoffman, P. and W. Wijngaards, "Elliptic Curve Digital Signature Algorithm (DSA) for DNSSEC", RFC 6605, DOI 10.17487/RFC6605, April 2012, <https://www.rfc-editor.org/info/rfc6605>.

12.2. Informative References

[RFC1995] Ohta, M., "Incremental Zone Transfer in DNS", RFC 1995, DOI 10.17487/RFC1995, August 1996, <https://www.rfc-editor.org/info/rfc1995>.

[RFC2136] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, DOI 10.17487/RFC2136, April 1997, <https://www.rfc-editor.org/info/rfc2136>.

[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, <https://www.rfc-editor.org/info/rfc2845>.

[RFC5936] Lewis, E. and A. Hoenes, Ed., "DNS Zone Transfer Protocol (AXFR)", RFC 5936, DOI 10.17487/RFC5936, June 2010, <https://www.rfc-editor.org/info/rfc5936>.

[RFC7706] Kumari, W. and P. Hoffman, "Decreasing Access Time to Root Servers by Running One on Loopback", RFC 7706, DOI 10.17487/RFC7706, November 2015, <https://www.rfc-editor.org/info/rfc7706>.

[RFC7719] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", RFC 7719, DOI 10.17487/RFC7719, December 2015, <https://www.rfc-editor.org/info/rfc7719>.

Authors' Addresses

Duane Wessels
Verisign
12061 Bluemont Way
Reston, VA 20190

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

Piet Barber
Verisign
12061 Bluemont Way
Reston, VA 20190

Phone: +1 703 948-3200

Email: pbarber@verisign.com
URI: http://verisign.com

Matt Weinberg
Verisign
12061 Bluemont Way
Reston, VA 20190

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

Warren Kumari
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043

Email: warren@kumari.net

Wes Hardaker
USC/ISI
P.O. Box 382
Davis, CA 95617

Email: ietf@hardakers.net