Workgroup: DNSOP Working Group
Internet-Draft: draft-dickson-dnsop-ds-hack-00
Published: 11 August 2021
Intended Status: Informational
Expires: 12 February 2022
Authors: . B Dickson
GoDaddy

DS Algorithms for Securing NS and Glue

Abstract

This Internet Draft proposes a mechanism to encode relevant data for NS records (and optionally A and AAAA records) on the parental side of a zone cut, by encoding them in new DS algorithms.

Since DS records are signed by the parent, this creates a method for validation of the otherwise unsigned delegation and glue records.

This is beneficial if the name server *names* are 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 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 12 February 2022.

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

There are new privacy goals and DNS server capability discovery goals, which cannot be met without the ability to validate the name of the name servers for a given domain at the delegation point.

Specifically, a query for NS records over an unprotected transport path returns results which do not have protection from tampering by an active on-path attacker, or against successful cache poisoning attackes.

This is true regardless of the DNSSEC status of the domain containing the authoritative information for the name servers for the queried domain.

For example, querying for the NS records for "example.com", at the name servers for the "com" TLD, where the published com zone has "example.com NS ns1.example.net", is not protected against MITM attacks, even if the domain "example.net" (the domain serving records for "ns1.example.net") is DNSSEC signed.

More infomation can be found in [I-D.nottingham-for-the-users]. (An exmple of an informative reference to a draft in the middle of text. Note that referencing an Internet draft involves replacing "draft-" in the name with "I-D.")

2. Conventions and Definitions

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Background

The methods developed for adding security to the Domain Name System, collectively refered to as DNSSEC, had as a primary requirement that they be backward compatible. The original specifications for DNS used the same Resourc Record Type (RRTYPE) on both the parent and child side of a zone cut (the NS record). The main goal of DNSSEC was to ensure data integrity by using cryptographic signatures. However, owing to this overlap in the NS record type where the records above and below the zone cut have the same owner name created an inherent conflict, as only the child zone is authoritative for these records.

The result is that the parental side of the zone cut has records needed for DNS resolution which are not signed and not validatable.

This has no impact on DNS zones which are fully DNSSEC signed (anchored at the IANA DNS Trust Anchor), but does impact unsigned zones regardless of where the transition from secure to insecure occurs.

4. New DNSKEY Algorithms

These new DNSKEY algorithms conform to the structure requirements from [RFC4034], but are not themselves used as actual DNSKEY algorithms. They are assigned values from the DNSKEY algorithm table. No DNSKEY records are published with these algorithms.

They are used only as the input to the corresponding DS hashes published in the parent zone.

4.1. Algorithm {TBD1}

This algorithm is used to validate the NS records of the delegation for the owner name.

The NS records are canonicalized and sorted according to the DNSSEC signing process [RFC4034] section 6, including removing any label compression, and normalizing the character cases to lower case. The RDATA fields of the records are concatenated, and the result is hashed using the selected digest algorithm(s), e.g. SHA2-256 for DS digest algorithm 1.

4.1.1. Example

Consider the delegation in the COM zone: example.com NS ns1.example.net example.com NS ns2.example.net

These two records have RDATA, which after canonicalization and sorting, would be ns1.example.net ns2.example.net

The input to the digest is the concatenation of those values in wire format. For example, if the NS set's RDATA are "ns1.example.net" and "ns2.example.net", the wire format would be

```
0x03
ns1
0x07
example
0x03
net
0x00
0x03
ns2
0x07
example
0x03
net
0x00
```

The Key Tag is calculated per [RFC4034] using this value as the RDATA.

The resulting DS record is

```
example.com DS KeyTag=0 Algorithm={TBD1} DigestType=2 \
Digest=sha2-256()
```

4.2. Algorithm {TBD2}

This algorithm is used to validate the glue A records required as glue for the delegation NS set associated with the owner name.

The glue A records are canonicalized and sorted according to the DNSSEC signing process [RFC4034], including removing any label compression, and normalizing the character cases. The entirety of the records are concatenated, and the result is hashed using the selected hash type(s), e.g. SHA2-256 for DS type 2.

4.2.1. Example

For example, if the original "glue" (unsigned) A records are:

```
ns1.example.net IN 3600 A standard-example-ip-1
ns2.example.net IN 3600 A standard-example-ip-2
```

There would be one DS record for each of the glue "A" records, with the canonicalized wire format of the entire record provided as input to the hash function.

```
FIXME replace 0xffffffx with real example IP addresses
(per IANA table of example IPs)
First A record's DS record:
wire_format(ns1.example.net) 0x01 0x01 3600 0xfffffff0
Second A record's DS record:
wire_format(ns2.example.net) 0x01 0x01 3600 0xfffffff1
```

Then the resulting DS record is

```
FIXME - who is the right owner to use here?
(The glue owner name, or the zone owner name (bailiwick only)?)
example.net DS KeyTag=0 Algorithm={TBD2} DigestType=2 \
Digest=sha2-256()
example.net DS KeyTag=0 Algorithm={TBD2} DigestType=2 \
Digest=sha2-256()
```

4.3. Algorithm {TBD3}

This algorithm is used to validate the glue AAAA records required as glue for the delegation NS set associated with the owner name.

The glue AAAA records are canonicalized and sorted according to the DNSSEC signing process [RFC4034], including removing any label compression, and normalizing the character cases. The entirety of the records are concatenated, and the result is hashed using the selected hash type(s), e.g. SHA2-256 for DS type 2.

4.3.1. Example

For example, if the original "glue" (unsigned) AAAA records are:

```
ns1.example.net IN 3600 AAAA standard-example-ip6-1
ns2.example.net IN 3600 AAAA standard-example-ip6-2
```

There would be one DS record for each of the glue "A" records, with the canonicalized wire format of the entire record provided as input to the hash function.

FIXME replace 0xffffffx with real example IP addresses
(per IANA table of example IPs)
First A record's DS record:
wire_format(ns1.example.net) 0x01 0xXX 3600 0x32-hex-digits
Second A record's DS record:
wire_format(ns2.example.net) 0x01 0xXX 3600 0x32-hex-digits

Then the resulting DS record is

FIXME - who is the right owner to use here?
(The glue owner name, or the zone owner name (bailiwick only)?)
example.net DS KeyTag=0 Algorithm={TBD2} DigestType=2 \
Digest=sha2-256()
example.net DS KeyTag=0 Algorithm={TBD2} DigestType=2 \
Digest=sha2-256()

5. Validation Using These DS Records

These new DS records are used to validate corresponding delegation records and glue, as follows: - NS records are validated using {TBD1} - Glue A records (if present) are validated using {TBD2} - Glue AAAA records (if present) are validated using {TBD3}

The same method used for constructing the DS records, is used to validate their contents. The algorithm is replicated with the corresponding inputs, and the hash compared to the published DS record(s).

6. Security Considerations

As outlined above, there could be security issues in various use cases.

7. IANA Considerations

This document has no IANA actions. (Well, actually, TBD1, TBD2, and TBD3 need to be assigned from the DNSSEC DNSKEY Algorithm Table.)

8. References

8.1. Normative References

[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>.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

8.2. Informative References

[I-D.nottingham-for-the-users]
Nottingham, M., "The Internet is for End Users", Work in Progress, Internet-Draft, draft-nottingham-for-the-users-09, 22 July 2019, <https://www.ietf.org/archive/id/draft-nottingham-for-the-users-09.txt>.

Acknowledgments

Thanks to everyone who helped create the tools that let everyone use Markdown to create Internet Drafts, and the RFC Editor for xml2rfc.

Thanks to Dan York for his Tutorial on using Markdown for writing IETF drafts.

Thanks to YOUR NAME HERE for contributions, reviews, etc.

Author's Address

Brian Dickson
GoDaddy

Email: brian.peter.dickson@gmail.com