Workgroup: Network Working Group
Internet-Draft: draft-dickson-dnsop-ds-hack-02
Published: 19 September 2021
Intended Status: Informational
Expires: 23 March 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 on the parental side of a zone cut by encoding them in DS records based on a new DNSKEY algorithm.

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

Notably, support for updating DS records in a parent zone is already present (by necessity) in the Registry-Registrar-Registrant (RRR) provisioning system, EPP. Thus, no changes to the EPP protocol are needed, and no changes to registry database or publication systems upstream of the DNS zones published by top level domains (TLDs).

This NS validation mechanism is beneficial if the name server *names* need to be validated prior to use.

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 23 March 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this 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

Currently, any query for delegation 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 because the parent NS records are being authoritative, and thus do not have RRSIGs. The child NS records with the same owner name are authoritave, but the parent NS records are what get used for delegations.

There is new privacy work that relies on the name server names in the delgation RDATA. Unsigned records are vulnerable to modification by on-path attackers and to cache poisoning by off-path attackers. That privacy work uses the name for TLS validation, and the only source of the name server name is the NS record in the delgation.

This document is about protecting the RDATA of NS record, not the privacy issues per se.

Note that the use of an encrypted trasport (such as DoT [RFC7858] to the parent would be an alternative approach, but in the absence of encrypted transport, the current approach is recommended.

If an attacker alters the NS records returned, or poisons the resolver's cache for the unsigned delegation NS, the recursive resolver could be directed to a server operated by an attacker.

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 parent side of the zone cut has records needed for DNS resolution which are not signed and not validatable.

This has no security (data validation) 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.

3.1. Attack Example

Suppose a resolver queries for the NS records for "example.com", at the name servers for the "com" TLD. Suppose this domain has been published in the com zone as "example.com NS ns1.example.net".

The response is not protected against MITM attacks. An on-path attacker can substitute its own name, "ns1.attacker.example". The resolver would then send its queries to the attacker.

Note that this vulnerability to MITM is present even if the domain "example.net" (the domain serving records for "ns1.example.net") is

DNSSEC signed, and the resolver intends to use TLS to make queries for names within the child zone, "example.com".

Substituting the name server name is sufficient to prevent the resolver from validating the TLS connection. It can validate the received TLS certificate, but would do expect the certificate to be for "ns1.attacker.example".

4. New DNSKEY Algorithm

This new DNSKEY algorithm conforms to the structure requirements from [RFC4034], but is not itself used as actual DNSKEY algorithm. It is assigned a value from the DNSKEY algorithm table. No DNSKEY records are published in the child zone using this algorithm.

This DNSKEY is used only as the input to the corresponding DS hashs published in the parent zone.

Note that this method is orthogonal to the specific choice of DS hashes. Examples here refer to the what is published currently in the IANA tables for recommended DNSSEC parameters, including recommended choices. Any valid supported hash for DS records MAY be used.

4.1. Algorithm {TBD1}

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

The original NS records are canonicalized according to the DNSSEC signing process [RFC4034] section 6, including removing any label compression, and normalizing the character cases to lower case. The RDATA field of the record is hashed using the selected digest algorithm(s), e.g. SHA2-256 for DS digest algorithm 2.

Note that only the RDATA from the wire format of the original NS record is used in constructing the DS record.

4.1.1. Example

Consider the delegation in the COM zone:

```
example.com NS ns1.Example.Net
example.com NS ns2.Example.Net
```

The input to the digest for each NS record is the uncompressed wire format of their respective RVALUEs.

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

The resulting combination of NS and DS records are:

```
example.com NS ns1.Example.Net
example.com NS ns2.Example.Net
; example.com DS KeyTag=FOO Algorithm={TBD1}
;   DigestType=2 Digest=sha2-256(wireformat("ns1.example.net"))
example.com DS KeyTag=FOO Algorithm={TBD1} DigestType=2 Digest=...
; example.com DS KeyTag=FOO Algorithm={TBD1}
;   DigestType=2 Digest=sha2-256(wireformat("ns2.example.net"))
example.com DS KeyTag=FOO Algorithm={TBD1} DigestType=2 Digest=...
```

5. Validation Using These DS Records

These new DS records are used to validate corresponding delegation records and glue. Each NS record must have a matching DS record. The expected DS record RDATA is constructed, and a matching DS record with identical RDATA MUST be present. Any NS record without matching valid DS record MUST be ignored.

- *NS records are validated using {TBD1}. The RDATA consists of only the RDATA from the NS record.

6. Protection of glue records

For the issue of glue records (parent side A/AAAA records which are not signed), please see the proposal [I-D.dickson-dnsop-glueless].

7. Security Considerations

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

The target domain containing each name server name is presumed (and required) to be DNSSEC signed.

8. IANA Considerations

This document has no IANA actions. (FIXME - update this doc to specify the required IANA actions - add TBD1 to the DNSKEY algorithm table)

9. Normative References

[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>.

10. Informative References

[I-D.dickson-dnsop-glueless]
Dickson, B., "Operating a Glueless DNS Authority Server", Work in Progress, Internet-Draft, draft-dickson-dnsop-glueless-00, 17 September 2021, <https://datatracker.ietf.org/doc/html/draft-dickson-dnsop-glueless-00>.

[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>.

[RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <https://www.rfc-editor.org/info/rfc7858>.

Appendix A. 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 (specifically mmark) 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