DNSEXT Working Group
INTERNET-DRAFT

Brian Wellington (Nominum)

October 2000

<draft-ietf-dnsext-signing-auth-02.txt>

Updates: RFC 2535

Domain Name System Security (DNSSEC) Signing Authority

Status of this Memo

Abstract

This document proposes a revised model of Domain Name System Security (DNSSEC) Signing Authority. The revised model is designed to clarify earlier documents and add additional restrictions to simplify the

secure resolution process. Specifically, this affects the authorization of keys to sign sets of records.

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 RFC 2119 [RFC2119].

1 - Introduction

This document defines additional restrictions on DNSSEC signatures (SIG) records relating to their authority to sign associated data. The intent is to establish a standard policy followed by a secure resolver; this policy can be augmented by local rules. This builds upon [RFC2535], updating section 2.3.6 of that document.

The most significant change is that in a secure zone, zone data is required to be signed by the zone key.

Familiarity with the DNS system [RFC1034, RFC1035] and the DNS security extensions [RFC2535] is assumed.

2 - The SIG Record

A SIG record is normally associated with an RRset, and ``covers'' (that is, demonstrates the authenticity and integrity of) the RRset. This is referred to as a ``data SIG''. Note that there can be multiple SIG records covering an RRset, and the same validation process should be repeated for each of them. Some data SIGs are considered ``material'', that is, relevant to a DNSSEC capable resolver, and some are ``immaterial'' or ``extra-DNSSEC'', as they are not relevant to DNSSEC validation. Immaterial SIGs may have application defined roles. SIG records may exist which are not bound to any RRset; these are also considered immaterial. The validation process determines which SIGs are material; once a SIG is shown to be immaterial, no other validation is necessary.

SIGs may also be used for transaction security. In this case, a SIG record with a type covered field of 0 is attached to a message, and is used to protect message integrity. This is referred to as a SIG(0) [RFC2535, RFC2931].

The following sections define requirements for all of the fields of a

SIG record. These requirements MUST be met in order for a DNSSEC capable resolver to process this signature. If any of these requirements are not met, the SIG cannot be further processed. Additionally, once a KEY has been identified as having generated this SIG, there are requirements that it MUST meet.

2.1 - Type Covered

For a data SIG, the type covered MUST be the same as the type of data in the associated RRset. For a SIG(0), the type covered MUST be 0.

2.2 - Algorithm Number

The algorithm specified in a SIG MUST be recognized by the client, and it MUST be an algorithm that has a defined SIG rdata format.

2.3 - Labels

The labels count MUST be less than or equal to the number of labels in the SIG owner name, as specified in [RFC2535, section 4.1.3].

2.4 - Original TTL

The original TTL MUST be greater than or equal to the TTL of the SIG record itself, since the TTL cannot be increased by intermediate servers. This field can be ignored for SIG(0) records.

2.5 - Signature Expiration and Inception

The current time at the time of validation MUST lie within the validity period bounded by the inception and expiration times.

2.6 - Key Tag

There are no restrictions on the Key Tag field, although it is possible that future algorithms will impose contraints.

2.7 - Signer's Name

The signer's name field of a data SIG MUST contain the name of the zone to which the data and signature belong. The combination of signer's name, key tag, and algorithm MUST identify a zone key if the SIG is to be considered material. The only exception that the signer's name field in a SIG KEY at a zone apex SHOULD contain the parent zone's name, unless the KEY set is self-signed. This document defines a standard policy for DNSSEC validation; local policy may override the standard policy.

There are no restrictions on the signer field of a SIG(0) record. The combination of signer's name, key tag, and algorithm MUST identify a key if this SIG(0) is to be processed.

2.8 - Signature

There are no restrictions on the signature field. The signature will be verified at some point, but does not need to be examined prior to verification unless a future algorithm imposes constraints.

3 - The Signing KEY Record

Once a signature has been examined and its fields validated (but before the signature has been verified), the resolver attempts to locate a KEY that matches the signer name, key tag, and algorithm fields in the SIG. If one is not found, the SIG cannot be verified and is considered immaterial. If KEYs are found, several fields of the KEY record MUST have specific values if the SIG is to be considered material and authorized. If there are multiple KEYs, the following checks are performed on all of them, as there is no way to determine which one generated the signature until the verification is performed.

3.1 - Type Flags

The signing KEY record MUST have a flags value of 00 or 01 (authentication allowed, confidentiality optional) [RFC2535, 3.1.2]. A DNSSEC resolver MUST only trust signatures generated by keys that are

permitted to authenticate data.

3.2 - Name Flags

The interpretation of this field is considerably different for data SIGs and SIG(0) records.

3.2.1 - Data SIG

If the SIG record covers an RRset, the name type of the associated KEY MUST be 01 (zone) [RFC2535, 3.1.2]. This updates RFC 2535, section 2.3.6. The DNSSEC validation process performed by a resolver MUST ignore all keys that are not zone keys unless local policy dictates otherwise.

The primary reason that RFC 2535 allows host and user keys to generate material DNSSEC signatures is to allow dynamic update without online

zone keys; that is, avoid storing private keys in an online server. The desire to avoid online signing keys cannot be achieved, though, because they are necessary to sign NXT and SOA sets [SSU]. These online zone keys can sign any incoming data. Removing the goal of having no online keys removes the reason to allow host and user keys to generate material signatures.

Limiting material signatures to zone keys simplifies the validation process. The length of the verification chain is bounded by the name's label depth. The authority of a key is clearly defined; a resolver does not need to make a potentially complicated decision to determine whether a key can sign data. amount of work to determine if all such keys have the proper authority.

Finally, there is no additional flexibility granted by allowing host/user key generated material signatures. As long as users and hosts have the ability to authenticate update requests to the primary zone server, signatures by zone keys are sufficient to protect the integrity of the data to the world at large.

3.2.2 - SIG(0)

If the SIG record is a SIG(0) protecting a message, the name type of the associated KEY SHOULD be 00 (user) or 10 (host/entity). Transactions are initiated by a host or user, not a zone, so zone keys SHOULD not generate SIG(0) records.

A client is either explicitly executed by a user or on behalf of a host, therefore the name type of a SIG(0) generated by a client SHOULD be either user or host. A nameserver is associated with a host, and its use of SIG(0) is not associated with a particular zone, so the name type of a SIG(0) generated by a nameserver SHOULD be host.

3.3 - Signatory Flags

This document does not assign any values to the signatory field, nor require any values to be present.

3.4 - Protocol

The signing KEY record MUST have a protocol value of 3 (DNSSEC) or 255 (ALL). If a key is not specified for use with DNSSEC, a DNSSEC resolver MUST NOT trust any signature that it generates.

3.5 - Algorithm Number

The algorithm field MUST be identical to that of the generated SIG record, and MUST meet all requirements for an algorithm value in a SIG record.

4 - Security considerations

This document defines a standard baseline for a DNSSEC capable resolver. This is necessary for a thorough security analysis of DNSSEC, if one is to be done.

Specifically, this document places additional restrictions on SIG records that a resolver must validate before the signature can be considered worthy of DNSSEC trust. This simplifies the protocol, making

it more robust and able to withstand scrutiny by the security community.

5 - Acknowledgements

The author would like to thank the following people for review and informative comments (in alphabetical order):

Olafur Gudmundsson
Ed Lewis

6 - References

[RFC1034] P. Mockapetris, ``Domain Names - Concepts and Facilities,'' RFC 1034, ISI, November 1987.

[RFC1035] P. Mockapetris, ``Domain Names - Implementation and Specification,'' RFC 1035, ISI, November 1987.

[RFC2119] S. Bradner, ``Key words for use in RFCs to Indicate Requirement Levels,'' BCP 14, RFC 2119, Harvard, March 1997.

[RFC2136] P. Vixie (Ed.), S. Thomson, Y. Rekhter, J. Bound ``Dynamic Updates in the Domain Name System,'' RFC 2136, ISC & Bellcore & Cisco & DEC, April 1997.

[RFC2535] D. Eastlake, ``Domain Name System Security Extensions,'' RFC 2535, IBM, March 1999.

[RFC2931] D. Eastlake, ``DNS Request and Transaction Signatures (SIG(0)s),'' RFC 2931, Motorola, September 2000.

[SSU] B. Wellington, ``Simple Secure Domain Name System (DNS) Dynamic Update,'' draft-ietf-dnsext-simple-secure-update-02.txt, Nominum, October 2000.

7 - Author's Address

Brian Wellington

Nominum, Inc.
950 Charter Street
Redwood City, CA 94063
+1 650 779 6022
<Brian.Wellington@nominum.com>

8 - Full Copyright Statement