DNS Operations
Internet-Draft
Expires: May 16, 2008

M. Larson
VeriSign
O. Gudmundsson
OGUD Consulting LLC
November 13, 2007

DNSSEC Trust Anchor Configuration and Maintenance
draft-larson-dnsop-trust-anchor-02

Status of this Memo

Copyright Notice

Abstract

This document recommends a preferred format for specifying trust anchors in DNSSEC validating security-aware resolvers and describes how such a resolver should initialize trust anchors for use. This document also describes different mechanisms for keeping trust anchors up to date over time.

Table of Contents

1. Introduction 3
2. Trust Anchor Format 4
3. Trust Anchor Priming 5
4. Trust Anchor Maintenance 7
5. Acknowledgments 8
6. Security considerations 9
7. IANA considerations 10
8. Internationalization considerations 11
9. References 12
9.1. Normative References 12
9.2. Informative References 12
Authors' Addresses 13
Intellectual Property and Copyright Statements 14

1. Introduction

The DNSSEC standards documents ([2], [3] and [4]) describe the need for trust anchors and how they are used. A validating security-aware resolver (subsequently referred to as a "validating resolver") needs to be configured with one or more trust anchors, which specify the public keys of signed zones. To authenticate DNS data, a validating resolver builds a chain of trust from a configured trust anchor to that data.

In a widespread public DNSSEC deployment, the DNS root zone would be signed and a validating resolver would need to be configured with at least the root zone's trust anchor. A validating resolver might need additional trust anchors configured to accommodate islands of security. (An island of security is a signed, delegated zone that does not have an authentication chain from its delegating parent.) For example, consider the situation where the root zone is signed but a given top-level domain (TLD) zone is not. Various second-level zones under this unsigned TLD might be signed and resolver operators might want to validate responses from those zones, requiring a validating resolver to be configured with those zones' trust anchors.

Because many validating resolvers would be configured with trust anchors in a widespread DNSSEC deployment, there is a benefit to creating a common trust anchor format. A similar situation has occurred with the "root hints", the list of root name server names and IP addresses: this information is distributed in standard master file format and many resolver implementations support this common format.

To simplify this trust anchor configuration process that will occur on a large number of resolvers, this document offers guidance to validating resolver implementers by specifying a standardized format for describing trust anchors. The document also describes how a validating resolver should initialize or "prime" trust anchors before first use. Finally, the document lists options for keeping trust

anchor information current over time.

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 [1].

2. Trust Anchor Format

A trust anchor is a DNSSEC public key configured in a validating resolver. A validating resolver's configuration MUST allow one or more trust anchors to be specified. According to the definition in Section 2 of RFC 4033 [2], a trust anchor can be specified as either a DNSKEY resource record (RR) or a DS RR, which contains the hash of the specific DNSKEY RR. (DS records are defined in Section 5 of RFC 4034 [3].)

This document RECOMMENDS that a trust anchor be specified as a DS RR. A DS RR used to specify a trust anchor in this manner SHOULD use a digest algorithm of SHA-256 [5], which is DS digest type 2. DS RRs using SHA-1 (DS digest type 1) to specify trust anchors are NOT RECOMMENDED: RFC 4509 encourages the use of DS RRs using SHA-256 over those using SHA-1.

Specifying a trust anchor using a DS RR instead of a DNSKEY RR offers a slight advantage because it forces the resolver to make a DNS query to obtain the trust anchor's complete DNSKEY RRSet during a priming operation (described below). If only a DNSKEY record were specified, a resolver implementers could conceivably avoid priming the trust anchor. But priming is desirable because it causes the resolver to retrieve an up-to-date version of a zone's DNSKEY RRSet from one of the zone's authoritative servers. It should be noted that in practice, priming is almost always required because data in the trust anchor zone will usually be signed with a different key than the one configured as the trust anchor, thus requiring the validating resolver to obtain all keys in the DNSKEY RRSet.

Using a DS RR is also recommended because it is smaller than the DNSKEY RR and is easier to enter manually, either by typing or cutting and pasting.

Another advantage of configuring a trust anchor using a DS RR is that the entire hash of the public key in the DS RDATA need not necessarily be specified. A validating resolver MAY support configuration using a truncated DS hash value as a human-factors convenience: shorter strings are easier to type and less prone to error when entered manually. Even with a truncated hash configured, a validating resolver can still verify that the corresponding DNSKEY is present in the trust anchor zone's apex DNSKEY RRSet.

3. Trust Anchor Priming

A validating resolver needs to obtain and validate the DNSKEY RRSet corresponding to a configured DS RR for that trust anchor to be usable in DNSSEC validation. This process is called "priming" the trust anchor. Priming can occur when the validating resolver starts, but a validating resolver SHOULD defer priming of individual trust anchors until each is first needed for verification. This priming on demand is especially important when a validating resolver is configured with a large number of trust anchors to avoid sending a large number of DNS queries on start-up. This section adds additional details to the discussion of trust anchors in Section 5 of RFC 4035 [4].

Following are the steps a validating resolver SHOULD take to prime a configured trust anchor:

1. Read the trust anchor's DS RR from the validating resolver's configuration (e.g., a text file).

2. Look up the DNSKEY RRSet corresponding to the owner name of the DS RR. (The validating resolver can either perform iterative

resolution or request recursive service from a recursive name server, depending on its capabilities.)

3. Verify that the DNSKEY RR corresponding to the configured DS RR (i.e., the DNSKEY whose hash appears in the DS record) appears in the DNSKEY RRSet and that the DNSKEY RR has the Zone Key Flag (DNSKEY RDATA bit 7) set.

4. Verify that the DNSKEY RRSet is signed by one of the DNSKEYs found in the previous step, i.e., that there exists a valid RRSIG (cryptographically and temporally) for the DNSKEY RRSet generated with the private key corresponding to the DNSKEY found in the previous step.

If the validating resolver can successfully complete the steps above, all DNSKEY RRs in the RRSet ought to be considered authenticated and used authenticate RRSets at or below the trust anchor.

If any of the steps above result in an error, the validating resolver SHOULD log them.

If there are multiple trust anchors configured for a zone, any one of them is sufficient to validate data in the zone. For this reason, old trust anchors SHOULD be removed from a validating resolver's trust anchor list soon after the corresponding keys are no longer used by the zone. A validating resolver should remove a trust anchor

that has been revoked as indicated by the REVOKE bit in the corresponding DNSKEY record as described in RFC 5011. RFC5011 [6]

If a validating resolver is unable to to retrieve a signed DNSKEY RRSet corresponding to a trust anchor (i.e., prime the trust anchor), it SHOULD log this condition as an error. Inability to prime a zone's trust anchor will likely result in the validating resolver's inability to validate data from the corresponding zone and cause the resolver to return an error in response to the original DNS query.

4. Trust Anchor Maintenance

Trust anchors correspond to zones' key signing keys and these keys do change in the course of normal operation. Validating resolver operators MUST ensure that configured trust anchor information remains current and does not go stale: each configured trust anchor DS RR SHOULD correspond to a DNSKEY RR in the trust anchor zone's apex DNSKEY RRSet. This process is called trust anchor maintenance.

(Initial trust anchor configuration requires human intervention to verify the trust anchor's authenticity using out-of-band means and is outside the scope of this document.)

This section provides a brief overview of some possible mechanisms to keep trust anchor information current:

Manual configuration: The validating resolver operator MAY choose to maintain trust anchor information completely manually. In this case, the operator assumes responsibility for noticing stale trust anchor information (i.e., DS records that no longer point to a corresponding DNSKEY RR in the trust anchor zone's apex DNSKEY RRSet) and updating that information. This process MAY require the operator to use the same out-of-band verification mechanism used to initial configuration to ensure that the new trust anchor DS RR is trustworthy. Because manual maintenance is burdensome and prone to error, and because other automated trust anchor maintenance processes either exist or are in development, manual trust anchor maintenance is NOT RECOMMENDED.

DNSSEC In-band Update: The IETF DNS Extensions Working Group has developed a protocol to automatically update DNSSEC trust anchors, which is described in RFC 5011. RFC5011 [6] This protocol relies on a small DNSSEC protocol change (an additional flag in the DNSKEY record) and can be implemented either in a validating resolver itself or in an external program with access to the validating resolver's trust anchor configuration data.

Trusted update mechanism: Updated trust anchor information MAY be obtained via a trusted non-DNS update mechanism. One possibility is the operating system update mechanism provided by most software vendors. Operators already place considerable trust in this mechanism, so it is reasonable to extend this trust to allow distribution and update of DNSSEC public key material. Another possibility is to obtain trust anchor configuration directly from the validating resolver software vendor. This mechanism is realistically only feasible for updating a small number of trust anchors, such as for the top-level domains. In a public DNSSEC deployment, the root zone would be signed and only the root's trust anchor would need updating.

5. Acknowledgments

This work was undertaken at the suggestion of the DNSSEC Deployment working group (www.dnssec-deployment.org).

6. Security considerations

This document proposes a standard format for documenting DNSSEC trust anchors. Configuration of trust anchors, especially those obtained from third parties as part of an automated process, is a critical security operation. The procedures described above describe the minimal checks that should be performed and reporting that should be done when configuring trust anchors.

In a widespread DNSSEC deployment, the root zone and many TLD zones would be signed, thus greatly reducing the number trust anchors that validating resolvers would need to store and keep track of.

7. IANA considerations

This document does not have any IANA actions.

8. Internationalization considerations

There are no new internationalization considerations introduced by this memo.

9. References

9.1. Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[3] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

[4] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

[5] Hardaker, W., "Use of SHA-256 in DNSSEC Delegation Signer (DS) Resource Records (RRs)", RFC 4509, May 2006.

[6] StJohns, M., "Automated Updates of DNS Security (DNSSEC) Trust Anchors", RFC 5011, September 2007.

9.2. Informative References

Authors' Addresses

Matt Larson
VeriSign, Inc.
21345 Ridgetop Circle
Dulles, VA 20166-6503
USA

Email: mlarson@verisign.com

Olafur Gudmundsson
OGUD Consulting LLC
3821 Village Park Drive
Chevy Chase, MD 20815
USA

Email: ogud@ogud.com

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).