Network Working Group
Internet-Draft
Expires: June 22, 2005

D. Blacka
Verisign, Inc.
December 22, 2004

DNSSEC Experiments
draft-blacka-dnssec-experiments-00

Status of this Memo

Copyright Notice

Abstract

In the long history of the development of the DNS security [1] extensions, a number of alternate methodologies and modifications have been proposed and rejected for practical, rather than strictly technical, reasons. There is a desire to be able to experiment with these alternate methods in the public DNS. This document describes a methodology for deploying alternate, non-backwards-compatible, DNSSEC methodologies in an experimental fashion without disrupting the deployment of standard DNSSEC.

Table of Contents

1. Definitions and Terminology

Throughout this document, familiarity with the DNS system (RFC 1035 [4]) and the DNS security extensions ([1], [2], and [3].

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 [5].

2. Overview

Historically, experimentation with DNSSEC alternatives has been a problematic endeavor. There has typically been a desire to both introduce non-backwards-compatible changes to DNSSEC, and to try these changes on real zones in the public DNS. This creates a problem when the change to DNSSEC would make all or part of the zone using those changes appear bogus or otherwise broken to existing DNSSEC-aware resolvers.

This document describes a standard methodology for setting up public DNSSEC experiments. This methodology addresses the issue of co-existence with standard DNSSEC and DNS by using unknown algorithm identifiers to hide the experimental DNSSEC protocol modifications from standard DNSSEC-aware resolvers.

3. Experiments

When discussing DNSSEC experiments, it is necessary to classify these experiments into two broad categories:

Backwards-Compatible: describes experimental changes that, while not strictly adhering to the DNSSEC standard, are nonetheless interoperable with clients and server that do implement the DNSSEC standard.

Non-Backwards-Compatible: describes experiments that would cause a standard DNSSEC-aware resolver to (incorrectly) determine that all or part of a zone is bogus, or to otherwise not interoperable with standard DNSSEC clients and servers.

Not included in these terms are experiments with the core DNS protocol itself.

The methodology described in this document is not necessary for backwards-compatible experiments, although it certainly could be used if desired.

Note that, in essence, this metholodolgy would also be used to introduce a new DNSSEC algorithm, independently from any DNSSEC experimental protocol change.

4. Method

The core of the methodology is the use of only "unknown" algorithms to sign the experimental zone, and more importantly, having only unknown algorithm DS records for the delegation to the zone at the parent.

This technique works because of the way DNSSEC-compliant validators are expected to work in the presence of a DS set with only unknown algorithms. From [3], Section 5.2:

> If the validator does not support any of the algorithms listed in an authenticated DS RRset, then the resolver has no supported authentication path leading from the parent to the child. The resolver should treat this case as it would the case of an authenticated NSEC RRset proving that no DS RRset exists, as described above.

And further:

> If the resolver does not support any of the algorithms listed in an authenticated DS RRset, then the resolver will not be able to verify the authentication path to the child zone. In this case, the resolver SHOULD treat the child zone as if it were unsigned.

While this behavior isn't strictly mandatory (as marked by MUST), it is unlikely that a validator would not implement the behavior, or, more to the point, it will not violate this behavior in an unsafe way (see below (Section 6).)

Because we are talking about experiments, it is recommended that private algorithm numbers be used (see [2], appendix A.1.1 [Comment.1].) Normally, instead of actually inventing new signing algorithms, the recommended path is to create alternate algorithm identifiers that are aliases for the existing, known algorithms. While, strictly speaking, it is only necessary to create an alternate identifier for the mandatory algorithms (currently, this is only algorithm 5, RSASHA1), it is RECOMMENDED that all OPTIONAL defined algorithms as well.

It is RECOMMENDED that for a particular DNSSEC experiment, a particular domain name base is chosen for all new algorithms, then the algorithm number (or name) is prepended to it. For example, for experiment A, the base name of "dnssec-experiment-a.example.com" is chosen. Then, aliases for algorithms 3 (DSA) and 5 (RSASHA1) are defined to be "3.dnssec-experiment-a.example.com" and "5.dnssec-experiment-a.example.com". However, any unique identifier will suffice.

Using this method, resolvers (or, more specificially, DNSSEC validators) essentially indicate their ability to understand the DNSSEC experiment's semantics by understanding what the new algorithm identifiers signify.

This method creates two classes of DNSSEC-aware servers and resolvers: servers and resolvers that are aware of the experiment (and thus recognize the experiments algorithm identifiers and experimental semantics), and servers and resolvers that are unware of the experiment.

5. Defining an Experiment

The DNSSEC experiment must define the particular set of (previously unknown) algorithms that identify the experiment, and define what each unknown algorithm identifier means. Typically, unless the experiment is actually experimenting with a new DNSSEC algorithm, this will be a mapping of private algorithm identifiers to existing, known algorithms.

Typically, the experiment will choose a DNS name as the algorithm identifier base. This DNS name SHOULD be under the control of the authors of the experiment. Then the experiment will define a mapping between known mandatory and optional algorithms into this private algorithm identifier space. Alternately, the experiment MAY use the OID private algorithm space instead (using algorithm number 254), or may choose non-private algorithm numbers, although this would require an IANA allocation (see below (Section 9).)

For example, an experiment might specify in its description the DNS name "dnssec-experiment-a.example.com" as the base name, and provide the mapping of "3.dnssec-experiment-a.example.com" is an alias of DNSSEC algorithm 3 (DSA), and "5.dnssec-experiment-a.example.com" is an alias of DNSSEC algorithm 5 (RSASHA1).

Resolvers MUST then only recognize the experiment's semantics when present in a zone signed by one or more of these private algorithms.

In general, however, resolvers involved in the experiment are expected to understand both standard DNSSEC and the defined experimental DNSSEC protocol, although this isn't, strictly speaking, required.

6. Considerations

There are a number of considerations with using this methodology.

1. Under some circumstances, it may be that the experiment will not be sufficiently masked by this technique and may cause resolution problem for resolvers not aware of the experiment. For instance, the resolver may look at the not validatable response and conclude that the response is bogus, either due to local policy or implementation details. This is not expected to be the common case, however.
2. It will, in general, not be possible for DNSSEC-aware resolvers not aware of the experiment to build a chain of trust through an experimental zone.

7. Transitions

If an experiment is successful, there may be a desire to move the experiment to a standards-track extension. One way to do so would be to move from private algorithm numbers to IANA allocated algorithm numbers, with otherwise the same meaning. This would still leave a divide between resolvers that understood the extension versus resolvers that did not. It would, in essence, create an additional version of DNSSEC.

An alternate technique might be to do a typecode rollover, thus actually creating a definitive new version of DNSSEC. There may be other transition techniques available, as well.

8. Security Considerations

Zones using this methodology will be considered insecure by all resolvers except those aware of the experiment. It is not generally possible to create a secure delegation from an experimental zone that will be followed by resolvers unaware of the experiment.

9. IANA Considerations

IANA may need to allocate new DNSSEC algorithm numbers if that transition approach is taken, or the experiment decides to use allocated numbers to begin with. No IANA action is required to deploy an experiment using private algorithm identifiers.

10. References

10.1 Normative References

[1] Arends, R., Austein, R., Massey, D., Larson, M. and S. Rose, "DNS Security Introduction and Requirements", draft-ietf-dnsext-dnssec-intro-13 (work in progress), October 2004.

[2] Arends, R., "Resource Records for the DNS Security Extensions", draft-ietf-dnsext-dnssec-records-11 (work in progress), October 2004.

[3] Arends, R., "Protocol Modifications for the DNS Security Extensions", draft-ietf-dnsext-dnssec-protocol-09 (work in progress), October 2004.

10.2 Informative References

[4] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

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

Editorial Comments

[Comment.1] Note: the DS record does not appear to be defined to handle private algorithm identifiers in a way consistent with DNSKEY and RRSIG, so it is possible that building in support for private algorithms in a DNSSEC validator is not feasible

Author's Address

David Blacka
Verisign, Inc.
21355 Ridgetop Circle
Dulles, VA 20166
US

Phone: +1 703 948 3200
EMail: davidb@verisign.com
URI: http://www.verisignlabs.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.