INTERNET-DRAFT   Samuel Weiler
Expires: June 2004   December 15, 2003
Updates: RFC 2535, [DS]

Legacy Resolver Compatibility for Delegation Signer
draft-ietf-dnsext-dnssec-2535typecode-change-06.txt

Status of this Memo

Comments should be sent to the author or to the DNSEXT WG mailing list: namedroppers@ops.ietf.org

Abstract

As the DNS Security (DNSSEC) specifications have evolved, the syntax and semantics of the DNSSEC resource records (RRs) have changed. Many deployed nameservers understand variants of these semantics. Dangerous interactions can occur when a resolver that understands an earlier version of these semantics queries an authoritative server that understands the new delegation signer semantics, including at least one failure scenario that will cause an unsecured zone to be unresolvable. This document changes the type codes and mnemonics of the DNSSEC RRs (SIG, KEY, and NXT) to avoid those interactions.

Changes between 05 and 06:

Signifigantly reworked the IANA section -- went back to one algorithm registry.

Removed Diffie-Hellman from the list of zone-signing algorithms (leaving only DSA, RSA/SHA-1, and private algorithms).

Added a DNSKEY flags field registry.

Changes between 04 and 05:

IESG approved publication.

Cleaned up an internal reference in the acknowledgements section.

Retained KEY and SIG for TKEY, too. Added TKEY (2930) reference.

Changed the names of both new registries. Added algorithm mnemonics to the new zone signing algorithm registry. Minor rewording in the IANA section for clarity.

Cleaned up formatting of references. Replaced unknown-rr draft references with RFC3597. Bumped DS version number.

Changes between 03 and 04:

Clarified that RRSIG(0) may be defined by standards action.

Created a new algorithm registry and renamed the old algorithm registry for SIG(0) only. Added references to the appropriate crypto algorithm and format specifications.

Several minor rephrasings.

Changes between 02 and 03:

KEY (as well as SIG) retained for SIG(0) use only.

Changes between 01 and 02:

SIG(0) still uses SIG, not RRSIG. Added 2931 reference.

Domain names embedded in NSECs and RRSIGs are not compressible and are not downcased. Added unknown-rrs reference (as informative).

Simplified the last paragraph of section 3 (NSEC doesn't always signal a negative answer).

Changed the suggested type code assignments.

Added 2119 reference.

Added definitions of "unsecure delegation" and "unsecure referral", since they're not clearly defined elsewhere.

Moved 2065 to informative references, not normative.

1. Introduction

The DNSSEC protocol has been through many iterations whose syntax and semantics are not completely compatible. This has occurred as part of the ordinary process of proposing a protocol, implementing it, testing it in the increasingly complex and diverse environment of the Internet, and refining the definitions of the initial Proposed Standard. In the case of DNSSEC, the process has been complicated by DNS's criticality and wide deployment and the need to add security while minimizing daily operational complexity.

A weak area for previous DNS specifications has been lack of detail in specifying resolver behavior, leaving implementors largely on their own to determine many details of resolver function. This, combined with the number of iterations the DNSSEC spec has been through, has resulted in fielded code with a wide variety of behaviors. This variety makes it difficult to predict how a protocol change will be handled by all deployed resolvers. The risk that a change will cause unacceptable or even catastrophic failures makes it difficult to design and deploy a protocol change. One strategy for managing that risk is to structure protocol changes so that existing resolvers can completely ignore input that might confuse them or trigger undesirable failure modes.

This document addresses a specific problem caused by Delegation Signer's [DS] introduction of new semantics for the NXT RR that are incompatible with the semantics in RFC 2535 [RFC2535]. Answers provided by DS-aware servers can trigger an unacceptable failure mode in some resolvers that implement RFC 2535, which provides a great disincentive to sign zones with DS. The changes defined in this document allow for the incremental deployment of DS.

1.1 Terminology

In this document, the term "unsecure delegation" means any delegation for which no DS record appears at the parent. An "unsecure referral" is an answer from the parent containing an NS RRset and a proof that no DS record exists for that name.

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 [RFC2119].

1.2 The Problem

Delegation Signer introduces new semantics for the NXT RR that are incompatible with the semantics in RFC 2535. In RFC 2535, NXT records were only required to be returned as part of a non-existence proof. With DS, an unsecure referral returns, in addition to the NS, a proof of non-existence of a DS RR in the form of an NXT and SIG(NXT). RFC 2535 didn't specify how a resolver was to interpret a response with both an NS and an NXT in the authority section, RCODE=0, and AA=0. Some widely deployed 2535-aware resolvers interpret any answer with an NXT as a proof of non-existence of the requested record. This results in unsecure delegations being invisible to 2535-aware resolvers and violates the basic architectural principle that DNSSEC must do no harm -- the signing of zones must not prevent the resolution of unsecured delegations.

2. Possible Solutions

This section presents several solutions that were considered. Section 3 describes the one selected.

2.1. Change SIG, KEY, and NXT type codes

To avoid the problem described above, legacy (RFC2535-aware) resolvers need to be kept from seeing unsecure referrals that include NXT records in the authority section. The simplest way to do that is to change the type codes for SIG, KEY, and NXT.

The obvious drawback to this is that new resolvers will not be able to validate zones signed with the old RRs. This problem already exists, however, because of the changes made by DS, and resolvers that understand the old RRs (and have compatibility issues with DS) are far more prevalent than 2535-signed zones.

2.2. Change a subset of type codes

The observed problem with unsecure referrals could be addressed by changing only the NXT type code or another subset of the type codes that includes NXT. This has the virtue of apparent simplicity, but it risks introducing new problems or not going far enough. It's quite possible that more incompatibilities exist between DS and earlier semantics. Legacy resolvers may also be confused by seeing records they recognize (SIG and KEY) while being unable to find NXTs. Although it may seem unnecessary to fix that which is not obviously broken, it's far cleaner to change all of the type codes at once. This will leave legacy resolvers and tools completely blinded to DNSSEC -- they will see only unknown RRs.

2.3. Replace the DO bit

Another way to keep legacy resolvers from ever seeing DNSSEC records with DS semantics is to have authoritative servers only send that data to DS-aware resolvers. It's been proposed that assigning a new EDNS0 flag bit to signal DS-awareness (tentatively called "DA"), and having authoritative servers send DNSSEC data only in response to queries with the DA bit set, would accomplish this. This bit would presumably supplant the DO bit described in RFC 3225.

This solution is sufficient only if all 2535-aware resolvers zero out EDNS0 flags that they don't understand. If one passed through the DA bit unchanged, it would still see the new semantics, and it would probably fail to see unsecure delegations. Since it's impractical to know how every DNS implementation handles unknown EDNS0 flags, this is not a universal solution. It could, though, be considered in addition to changing the RR type codes.

2.4. Increment the EDNS version

Another possible solution is to increment the EDNS version number as defined in RFC 2671 [RFC2671], on the assumption that all existing implementations will reject higher versions than they support, and retain the DO bit as the signal for DNSSEC awareness. This approach has not been tested.

2.5. Do nothing

There is a large deployed base of DNS resolvers that understand DNSSEC as defined by the standards track RFC 2535 and RFC 2065 and, due to under specification in those documents, interpret any answer with an NXT as a non-existence proof. So long as that is the case, zone owners will have a strong incentive to not sign any zones that contain unsecure delegations, lest those delegations be invisible to such a large installed base. This will dramatically slow DNSSEC adoption.

Unfortunately, without signed zones there's no clear incentive for operators of resolvers to upgrade their software to support the new version of DNSSEC, as defined in [DS]. Historical data suggests that resolvers are rarely upgraded, and that old nameserver code never dies.

Rather than wait years for resolvers to be upgraded through natural processes before signing zones with unsecure delegations, addressing this problem with a protocol change will immediately remove the disincentive for signing zones and allow widespread deployment of DNSSEC.

3. Protocol changes

This document changes the type codes of SIG, KEY, and NXT. This approach is the cleanest and safest of those discussed above, largely because the behavior of resolvers that receive unknown type codes is well understood. This approach has also received the most testing.

To avoid operational confusion, it's also necessary to change the mnemonics for these RRs. DNSKEY will be the replacement for KEY, with the mnemonic indicating that these keys are not for application use, per [RFC3445]. RRSIG (Resource Record SIGnature) will replace SIG, and NSEC (Next SECure) will replace NXT. These new types completely replace the old types, except that SIG(0)

[RFC2931] and TKEY [RFC2930] will continue to use SIG and KEY.

The new types will have exactly the same syntax and semantics as specified for SIG, KEY, and NXT in RFC 2535 and [DS] except for the following:

1) Consistent with [RFC3597], domain names embedded in RRSIG and NSEC RRs MUST NOT be compressed,

2) Embedded domain names in RRSIG and NSEC RRs are not downcased for purposes of DNSSEC canonical form and ordering nor for equality comparison, and

3) An RRSIG with a type-covered field of zero has undefined semantics. The meaning of such a resource record may only be defined by IETF Standards Action.

If a resolver receives the old types, it SHOULD treat them as unknown RRs and SHOULD NOT assign any special meaning to them or give them any special treatment. It MUST NOT use them for DNSSEC validations or other DNS operational decision making. For example, a resolver MUST NOT use DNSKEYs to validate SIGs or use KEYs to validate RRSIGs. If SIG, KEY, or NXT RRs are included in a zone, they MUST NOT receive special treatment. As an example, if a SIG is included in a signed zone, there MUST be an RRSIG for it. Authoritative servers may wish to give error messages when loading zones containing SIG or NXT records (KEY records may be included for SIG(0) or TKEY).

As a clarification to previous documents, some positive responses, particularly wildcard proofs and unsecure referrals, will contain NSEC RRs. Resolvers MUST NOT treat answers with NSEC RRs as negative answers merely because they contain an NSEC.

4. IANA Considerations

4.1 DNS Resource Record Types

This document updates the IANA registry for DNS Resource Record Types by assigning types 46, 47, and 48 to the RRSIG, NSEC, and DNSKEY RRs, respectively.

Types 24 and 25 (SIG and KEY) are retained for SIG(0) [RFC2931] and TKEY [RFC2930] use only.

Type 30 (NXT) should be marked as Obsolete.

4.2 DNS Security Algorithm Numbers

To allow zone signing (DNSSEC) and transaction security mechanisms (SIG(0) and TKEY) to use different sets of algorithms, the existing "DNS Security Algorithm Numbers" registry is modified to include the applicability of each algorithm. Specifically, two new columns are added to the registry, showing whether each algorithm may be used for zone signing, transaction security mechanisms, or both. Only algorithms usable for zone signing may be used in DNSKEY, RRSIG, and DS RRs. Only algorithms usable for SIG(0) and/or TSIG may be used in SIG and KEY RRs.

All currently defined algorithms remain usable for transaction security mechanisms. Only RSA/SHA-1, DSA/SHA-1, and private algorithms (types 253 and 254) may be used for zone signing. Note that the registry does not contain the requirement level of each algorithm, only whether or not an algorithm may be used for the given purposes. For example, RSA/MD5, while allowed for transaction security mechanisms, is NOT RECOMMENDED, per RFC3110.

Additionally, the presentation format algorithm mnemonics from RFC2535 Section 7 are added to the registry. This document assigns RSA/SHA-1 the mnemonic RSASHA1.

As before, assignment of new algorithms in this registry requires IETF Standards Action. Additionally, modification of algorithm mnemonics or applicability requires IETF Standards Action. Documents defining a new algorithm must address the applicability of the algorithm and should assign a presentation mnemonic to the algorithm.

4.3 DNSKEY Flags

Like the KEY resource record, DNSKEY contains a 16-bit flags field. This document creates a new registry for the DNSKEY flags field.

Initially, this registry only contains an assignment for bit 7 (the ZONE bit). Bits 0-6 and 8-15 are available for assignment by IETF Standards Action.

4.4 DNSKEY Protocol Octet

Like the KEY resource record, DNSKEY contains an eight bit protocol field. The only defined value for this field is 3 (DNSSEC). No other values are allowed, hence no IANA registry is needed for this field.

5. Security Considerations

The changes introduced here do not materially affect security. The implications of trying to use both new and legacy types together are not well understood, and attempts to do so would probably lead to unintended and dangerous results.

Changing type codes will leave code paths in legacy resolvers that are never exercised. Unexercised code paths are a frequent source of security holes, largely because those code paths do not get frequent scrutiny.

Doing nothing, as described in section 2.5, will slow DNSSEC deployment. While this does not decrease security, it also fails to increase it.

6. Normative references

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

[DS] Gudmundsson, O., "Delegation Signer Resource Record", draft-ietf-dnsext-delegation-signer-15.txt, work in progress, June 2003.

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

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

[RFC2930] Eastlake, D., "Secret Key Establishment for DNS (TKEY RR)", RFC 2930, September 2000.

[RFC2536] Eastlake, D., "DSA KEYs and SIGs in the Domain Name System (DNS)", RFC 2436, March 1999.

[RFC2539] Eastlake, D., "Storage of Diffie-Hellman Keys in the Domain Name System (DNS)", RFC 2539, March 1999.

[RFC3110] Eastlake, D., "RSA/SHA-1 SIGs and RSA KEYs in the Domain Name System (DNS)", RFC 3110, May 2001.

7. Informative References

[RFC2065] Eastlake, D. and C. Kaufman, "Domain Name System Security Extensions", RFC 2065, January 1997.

[RFC2671] Vixie, P., "Extension Mechanisms for DNS (EDNS0)", RFC 2671, August 1999.

[RFC3225] Conrad, D., "Indicating Resolver Support of DNSSEC", RFC 3225, December 2001.

[RFC2929] Eastlake, D., E. Brunner-Williams, and B. Manning, "Domain Name System (DNS) IANA Considerations", BCP 42, RFC 2929, September 2000.

[RFC3445] Massey, D., and S. Rose, "Limiting the Scope of the KEY Resource Record (RR)", RFC 3445, December 2002.

[RFC3597] Gustafsson, A., "Handling of Unknown DNS Resource

Record (RR) Types", RFC 3597, September 2003.

8. Acknowledgments

The changes introduced here and the analysis of alternatives had many contributors. With apologies to anyone overlooked, those include: Micheal Graff, John Ihren, Olaf Kolkman, Mark Kosters, Ed Lewis, Bill Manning, and Suzanne Woolf.

Thanks to Jakob Schlyter and Mark Andrews for identifying the incompatibility described in section 1.2.

In addition to the above, the author would like to thank Scott Rose, Olafur Gudmundsson, and Sandra Murphy for their substantive comments.

9. Author's Address

Samuel Weiler
SPARTA, Inc.
7075 Samuel Morse Drive
Columbia, MD 21046
USA
weiler@tislabs.com