Network Working Group
Internet-Draft
Expires: December 18, 2006

B. Laurie
Nominet
R. Loomis
SAIC
June 16, 2006

Requirements related to DNSSEC Signed Proof of Non-Existence
draft-ietf-dnsext-signed-nonexistence-requirements-03

Status of this Memo

Copyright Notice

Abstract

DNSSEC-bis uses the NSEC record to provide authenticated denial of existence of RRsets. NSEC also has the side-effect of permitting zone enumeration, even if zone transfers have been forbidden. Because some see this as a problem, this document has been assembled to detail the possible requirements for denial of existence A/K/A signed proof of non-existence.

Table of Contents

1. Introduction

NSEC records allow trivial enumeration of zones - a situation that has existed for several years but which has recently been raised as a significant concern for DNSSECbis deployment in several zones. Alternate proposals have been made that make zone enumeration more difficult, and some previous proposals to modify DNSSEC had related requirements/desirements that are relevant to the discussion. In addition the original designs for NSEC/NXT records were based on working group discussions and the choices made were not always documented with context and requirements-- so some of those choices may need to be restated as requirements. Overall, the working group needs to better understand the requirements for denial of existence (and certain other requirements related to DNSSECbis deployment) in order to evaluate the proposals that may replace NSEC. The -01 version of this document was presented at IETF61 on 10 November 2004 along with a re-categorization of the then-current list of potential requirements. This version of the document formalizes that re-categorization of the requirements, and is intended to serve as the basis for further discussions and evaluation of potential solutions.

2. Terminology

In the remainder of this document, "NSEC++" is used as shorthand for "a denial of existence proof that will replace NSEC". "NSECbis" has also been used as shorthand for this, but we avoid that usage since NSECbis will not be part of DNSSECbis and therefore there might be some confusion. We also use the term "DNSSEC-TNG" A/K/A "DNSSECter". This is meant to indicate the current DNSSECbis plus whatever changes are required as part of NSEC++. We expect that DNSSECter will likely still include the current NSEC record as well.

3. Non-purposes

This document does not currently document the reasons why zone enumeration might be "bad" from a privacy, security, business, or other perspective--except insofar as those reasons result in requirements. Once the list of requirements is complete and vaguely coherent, the trade-offs (reducing zone enumeration will have X cost, while providing Y benefit) may be revisited. The editors of this compendium received inputs on the potential reasons why zone enumeration is bad (and there was significant discussion on the DNSEXT WG mailing list) but that information fell outside the scope of this document.

Note also that this document does not assume that NSEC *must* be

replaced with NSEC++, if the requirements can be met through other methods (e.g., "white lies" with the current NSEC). As is stated above, this document is focused on requirements collection and (ideally) prioritization rather than on the actual implementation.

4. Group 1 - Zone Enumeration and exposure

Comprised of previous requirements numbered as 3, 4, 5, 6, 10, and 26

The editors suggest that these boil down to: "DNSSECter should not make it easier to enumerate zones than it is with plain DNS."

We believe that this is a high-priority requirement.

Threshold requirement: Enumeration is at least non-trivial (where current NSEC provides a linked list that is considered trivial to walk).

A concrete test might be that a random zone is infeasible to fully enumerate--this also reflects the "goal requirement"

Contributor: various

5. Group 2 - Zone Size

Requirement: NSEC++ should make it possible to take precautions against trivial zone size estimates. Since not all zone owners care about others estimation of the size of a zone, it is not always necessary to prohibit trivial estimation of the size of the zone but NSEC++ should allow such measures.

We believe that this is a "nice to have" item and not a true requirement, and recommend weighting it appropriately when considering solutions.

Additional Discussion: Even with proposals based on obfuscating names with hashes it is trivial to give very good estimates of the number of domains in a certain zone. Just send 10 random queries and look at the range between the two hash values returned in each NSEC++. As hash output can be assumed to follow a rectangular random distribution, using the mean difference between the two values, you can estimate the total number of records. It is probably sufficient to look at even one NSEC++, since the two hash values should follow a (I believe) Poisson distribution.

The concern is motivated by some wording remembered from NSEC, which

stated that NSEC MUST only be present for existing owner names in the zone, and MUST NOT be present for non-existing owner names. If similar wording were carried over to NSEC++, introducing bogus owner names in the hash chain (an otherwise simple solution to guard against trivial estimates of zone size) wouldn't be allowed.

One simple attempt at solving this is to describe in the specifications how zone signer tools can add a number of random "junk" records.

Editor's comment: it is interesting that obfuscating names might actually make it easier to estimate zone size.

Contributor: Simon Josefsson.

6. Group 3 - Compatibility and Transition

Comprised of requirements previously numbered as 8, 20, 21, 22, 23, 24

Editor comments: The editors suggest that these boil down to, "Current deployment of DNSSECbis with NSEC, by those who care not about zone enumeration, should not be affected by future NSEC++ deployment."

We believe that this is a high priority requirement.

NOTE: Requirement 8 is no longer truly applicable, because it would have mandated a change to the draft DNSSECbis documents that was not made before they were submitted for IESG review.

Contributor: Various

7. Group 4 - Empty Non-terminals

Goal: Empty-non-terminals (ENT) should remain empty. In other words, adding NSEC++ records to an existing DNS structure should not cause the creation of NSEC++ records (or related records) at points that are otherwise ENT.

Editor comments: We believe that this is a low-priority desire and not a strict requirement, and we recommend that it be weighted appropriately when comparing possible solutions.

Additional discussion: Currently NSEC complies with ENT requirement: b.example.com NSEC a.c.example.com implies the existence of an ENT

with ownername c.example.com. NSEC2 breaks that requirement, since the ownername is entirely hashed causing the structure to disappear. This is why EXIST was introduced. But EXIST causes ENT to be non-empty-terminals. Next to the dissappearance of ENT, it causes (some) overhead since an EXIST record needs a SIG, NSEC2 and SIG(NSEC2). DNSNR honours this requirement by hashing individual labels instead of ownernames. However this causes very long labels. Truncation is a measure against very long ownernames, but that is controversial. There is a fair discussion of the validity of truncation in the DNSNR draft, but that hasn't got proper review yet.

Contributor: Roy Arends.

(Editor comment: it is not clear to us that an EXIST record needs an NSEC2 record, since it is a special purpose record only used for denial of existence)

8. Group 5 - DNSSEC-Adoption and Zone-Growth Relationship

Background: Currently with NSEC, when a delegation centric zone deploys DNSSEC, the zone-size multiplies by a non-trivial factor even when the DNSSEC-adoption rate of the subzones remains low--because each delegation point creates at least one NSEC record and corresponding signature in the parent even if the child is not signed.

Goal/Requirements: A delegation-only (or delegation-mostly) zone that is signed but which has no signed child zones should initially need only to add SIG(SOA), DNSKEY, and SIG(DNSKEY) at the apex, along with some minimal set of NSEC++ records to cover zone contents. Further, during the transition of a delegation-only zone from 0% signed children to 100% signed children, the growth in the delegation-only zone should be roughly proportional to the percentage of signed child zones.

Editor comments: We believe that this is a medium-priority goal or desire and should be considered. Because of the similarity of this item to the older "opt-in signed zones" proposal, we recognize that consideration of this item may bog down the DNSEXT WG and that a decision must be made by the WG chairs.

Additional Discussion: This is why DNSNR has the Authoritative Only bit. This is similar to opt-in for delegations only. This (bit) is currently the only method to help delegation-centric zone cope with zone-growth due to DNSSEC adoption. As an example, A delegation only zone which deploys DNSSEC with the help of this bit, needs to add SIG(SOA), DNSKEY, SIG(DNSKEY), DNSNR, SIG(DNSNR) at the apex. No

more than that.

Contributor: Roy Arends.

9. Group 6 - Non-overlap of denial records with possible zone records

Goal: NSEC++ records should in some way be differentiated from regular zone records, so that there is no possibility that a record in the zone could be duplicated by a non-existence proof (NSEC++) record.

Editor comment: We are not sure that this is a valid concern much less a requirement. Even if there is an apparent conflict or overlap, the "conflicting" NSEC2 name _only_ appears in NSEC2 records, and the other name _never_ appears in NSEC2 records. Protocols cannot protect against all possible silly or foolish actions, and should a randomly chosen salt produce an NSEC2 record that matches a zone entry (either current or future) then the solution will be to pick a new salt and re-sign the zone.

Additional discussion: This requirement is derived from a discussion on the DNSEXT mailing list related to copyrights and domain names. As was outlined there, one solution would be to put NSEC++ records in a separate namespace, e.g.: $ORIGIN co.uk. 873bcdba87401b485022b8dcd4190e3e IN NS jim.rfc1035.com ; your delegation 873bcdba87401b485022b8dcd4190e3e._no IN NSEC++ 881345... ; for amazon.co.uk. However, it is not obvious that this separate namespace is useful.

Contributor: various

10. Group 7 - Exposure of Private Keys

Private keys associated with the public keys in the DNS should be exposed as little as possible. It is highly undesirable for private keys to be distributed to nameservers, or to otherwise be available in the run-time environment of nameservers.

We believe that this is a medium priority desire. For some organizations the use of online keys may be an acceptable trade-off if it allows the prevention of zone enumeration. On the other hand, there are some organizations which may be concerned about zone enumeration and for whom online storage/availability of keys on the authoritative servers may be unacceptable.

Contributors: Nominet, Olaf Kolkman, Ed Lewis

11. Group 8 - Minimisation of Zone Signing Cost

The additional cost of creating an NSEC++ signed zone should not significantly exceed the cost of creating an ordinary signed zone. Furthermore, DNSSEC++ should not make incremental signing of existing zones any "harder" (in terms of computational or administrative resources) than it currently is with DNSSECbis/NSEC.

We believe that this is a medium-priority desire.

Contributor: Nominet

12. Group 9 - DoS prevention/symmetric cost

NSEC++ should not make Denial of Service (DoS) attacks significantly more effective than plain DNSSECbis. Any increase in real-time cost at the name server (to respond) should correspond to a proportional increase in real-time cost to generate the original query.

Editor comment: We believe that this is a low-priority desire. In general DNSSEC makes DoS attacks against both authoritative and recursive DNS servers so much easier that the answer will be to increase available server CPU resources. Further, we are not sure that this a realistic requirement given the other requirements for NSEC++. In the end, we recommend that this be considered along with other factors when reviewing potential solutions.

Contributor: Nominet

13. Group 10 - Acceptable Complexity

Caching, wildcards, CNAMEs, and DNAMEs should continue to work without significant increases in complexity at the client side--where complexity specifically includes complexity of operational usage and complexity of validator implementation.

We believe that this is a medium priority desire.

Contributor: Olaf Kolkman

14. Group 11 - Completeness

There should not be a proof of nonexistence possible for any valid data in the zone. NOTE: This has a much different meaning than the way in which this requirement was stated in the -01 version of this

document, based on further discussions with the original contributor.

This requirement now appears to conflict with Group 5 above and has been given the same priority as Group 5 (previously requirement 11). The WG will need to resolve the conflict and remove one of the two goals/requirements from consideration.

Contributor: Olaf Kolkman

15. Group 12 - Purity of Namespace

The name space should not be muddied with fake names or data sets.

Editor comment: After further discussion with the contributor, this appears to be more of an awareness issue than a true requirement, and one that may be possible to address on the implementation side. See also Group 6, which appears to be based on similar concerns (although the similarity was not identified during discussions at IETF 61).

Contributor: Ed Lewis

16. Group 13 - Replay Attacks

Requirement: NSEC++ should not allow replay attacks that are any more effective than those which currently exist in DNSSECbis.

Editor comment: This is a high-priority requirement. The requirement was reworded based on further discussion with the original contributor and other WG members. The basis for the rewording is that DNSSECbis by design does not allow replay attacks that deny a record which already exists. However, attacks against a record which has been added will succeed (until the signature expires on the relevant NSEC)

Contributor: Ed Lewis

17. Group 14 - Security during Zone Transition

Requirement: It should be possible to switch between NSEC and NSEC++ without any zone data appearing to be unsigned, insecure, or "partly secure" during the transition, taking into account externally cached data.

Additional discussion: We never want an end-user to see "inconsistently signed" data. Both positive and negative answers

should still be able to be validated.

Editor comment: This is a newly identified requirement. This is at least highly desirable and perhaps a hard-and-fast requirement.

18. Group 15a - Universal Signing

Editor comment: The 15 a/b/c nomenclature is used in this version for consistency with the presentation made to DNSEXT by the editors during IETF 61 in DC. This should probably be fixed in some way for the next version of this document...hopefully in a way that minimizes confusion.

Requirement: "Every zone that can be signed with DNSSECbis can also be signed with DNSSECter." (We believe that this is all zones, but do not want to prove it nor raise the bar for DNSSECter.)

Additional discussion: This is a newly-identified, hard-and-fast requirement.

19. Group 15b - Universal Signing

Requirement: It should be possible to sign all zones with DNSSECter.

Additional discussion: Newly identified requirement. We rate this as highly desirable.

20. Group 15c - Universal Signing

Requirement: If it is not possible to sign all zones with NSEC++, then it should be clearly defined which zones cannot be signed.

This is a newly identified, hard-and-fast requirement.

21. Group 16 - NSEC++ as seen by NSEC-only resolver

Requirement: An NSEC++ (only) zone, regardless of whether parent uses NSEC or NSEC++, should appear as consistently unsigned when seen by an NSEC-only resolver.

Basis: We never want an end-user to see "inconsistently signed" data.

This is a newly-identified requirement. This is at least highly desirable and perhaps a hard-and-fast requirement.

22. Prioritization

Clearly not all of these requirements can be met. Therefore the editors have attempted to prioritize the requirements as they understand the relevant impacts and needs. The following lists give details as to the prioritization. The order of listing within each priority level is also intended to show which requirements should be given higher priority if a "tie-breaker" is needed. Further, there are likely some potential DNSSEC users who would assign different priorities to specific requirement sets--these are meant to be an overall list that best serves the wider community.

High priority: Group 1 (Zone enumeration and exposure), group 3 (compatibility and transition), group 13 (replay), group 15a (universal signing), and group 15c (universal signing).

Medium priority: Group 14 (security during transition), group 15b (universal signing), Group 16 (NSEC-only resolver results), group 5 (adoption and zone growth), group 11 (completeness), group 7 (exposure of signing keys), group 10 (complexity), group 12 (DNS "purity"), group 8 (zone signing cost)

Low priority: Group 9 (DoS prevention), group 2 (zone size), group 4 (Empty non-terminals), group 6 (non-overlap in namespace)

23. Non-requirements

Explicit non-requirement: Prevent enumeration of RR types for a given name.

Even if it is notionally possible to provide this capability, we expect a steep cost and little benefit.

Future provision of this capability is not prevented, if warranted.

24. Acknowledgements

25. Requirements notation

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].

26. Security Considerations

There are only very limited security considerations called out in this draft, primarily related to questions of whether some of the methods for avoiding zone enumeration might require a message to be cryptographically signed "on the fly", which would imply that private keys must in some way be accessible on authoritative nameservers.

There will be security considerations in the choice of which requirements will be implemented, but there are no specific security requirements during the requirements collection process.

27. References

27.1. Normative References

[dnssecbis-protocol]
"DNSSECbis Protocol Definitions", BCP XX, RFC XXXX, Some Month 2004.

27.2. Informative References

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

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

[RFC2418] Bradner, S., "IETF Working Group Guidelines and Procedures", BCP 25, RFC 2418, September 1998.

Authors' Addresses

Ben Laurie
Nominet
17 Perryn Road
London W3 7LR
England

Phone: +44 (20) 8735 0686
Email: ben@algroup.co.uk

Rip Loomis
Science Applications International Corporation
7125 Columbia Gateway Drive, Suite 300
Columbia, MD 21046
US

Email: gilbert.r.loomis@saic.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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