Network Working Group
Internet-Draft
Expires: November 30, 2004

R. Arends
Telematica Instituut

June 2004

DNSSEC Non-Repudiation Resource Record
draft-arends-dnsnr-00

Status of this Memo

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, and any of which I become aware will be disclosed, in accordance with RFC 3668.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on November 30, 2004.

Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

Abstract

This document describes the DNSNR Resource Record (RR) for the Non-Repudiation (NR) of Existence service in the context of the DNS Security Extensions (DNSSEC). The DNSNR is an alternative to NSEC or "Authenticated Denial of Existence" Resource Records.

A signed DNSNR RR protects security-aware DNS components against false denial of existence of RRsets by providing the RR types that exist for its ownername, which optionally includes a non-authoritative delegation point NS RR type. Labels in the ownername and the RDATA may be a hash-value as a defense against zone

traversal.

The DNSNR is an alternative to current NSEC or "Authenticated Denial of Existence" records.

Table of Contents

1. Introduction 3
  1.1 Rationale 3
  1.2 Reserved Words 3
2. The DNSNR Resource Record 4
  2.1 DNSNR RDATA Wire Format 4
    2.1.1 The Authoritative Only Flag Field 4
    2.1.2 The Hash Function Field 5
    2.1.3 The Salt Field 5
    2.1.4 The Next Ownername Field 5
    2.1.5 The Type Bit Maps Field 6
    2.1.6 Inclusion of Wildcard Names in DNSNR RDATA 7
  2.2 The DNSNR RR Presentation Format 8
  2.3 DNSNR RR Example 8
3. Special Considerations 10
  3.1 Delegation points 10
    3.1.1 Unsigned Delegations 10
    3.1.2 Salting 11
  3.2 Hash Collision 11
    3.2.1 Avoiding Hash Collisions during generation 11
    3.2.2 Second Preimage Requirement Analysis 11
    3.2.3 Possible Hash Value Truncation Method 12
  3.3 Future Hash Functions 12
4. Acknowledgements 13
5. References 14
5.1 Normative References 14
5.2 Informative References 14
  Author's Address 14
  Intellectual Property and Copyright Statements 15

1. Introduction

The DNS Security Extensions (DNSSEC) introduced the NSEC Resource Record (RR) for authenticated denial of existence. This document introduces a new RR as an alternative to NSEC that allows for gradual expansion of delegation-centric zones and provides measures against zone traversal.

1.1 Rationale

The DNS Security Extensions included the NSEC RR to provide authenticated denial of existence. Though the NSEC RR meets the requirements for authenticated denial of Existence, it introduced a side-effect in that the contents of a zone can be enumerated. This side-effect may introduce undesired policy issues.

A second requirement was that the existence of all record types in a zone -including delegation point NS record types- can be accounted for, despite the fact that delegation point NS RRsets are not authoritative and not signed. This requirement has a side-effect that the overhead of delegation centric signed zones is not related to the increase in security of subzones. This requirement does not allow delegation centric zones size to grow in relation to the grow of signed subzones.

In the past, solutions have been proposed as a measure against these side effects but at the time were regarded as secondary over the need to have a stable DNSSEC specification. With (draft-vixie-dnssec-ter) a graceful transition path to future enhancements is introduced, while current DNSSEC deployment can continue. This document accumulates measures against the side effects introduced by NSEC, and presents the DNS Non-Repudiation Record.

The reader is assumed to be familiar with the basic DNS concepts described in RFC1034 [RFC1034], RFC1035 [RFC1035] and subsequent RFCs that update them: RFC2136 [RFC2136], RFC2181 [RFC2181] and RFC2308 [RFC2308].

1.2 Reserved Words

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].

2. The DNSNR Resource Record

The DNSNR RR provides Non-repudiation of existence of DNS Resource Record Sets.

The DNSNR resource record lists RR types present at the DNSNR RR's ownername. It includes the ownername of the next authoritative, optionally hashed, ownername in the canonical ordering of the zone. A set of DNSNR RRs in a zone indicate which authoritative RRsets exist, while delegation point NS records can optionally be included to proof existence of an unsigned delegation. To provide protection against zone traversal, the labels used in the DNSNR RR may be a cryptographic hash-value.

The type value for the DNSNR RR is XX.

The DNSNR RR is class independent.

The DNSNR RR has no special TTL requirements.

2.1 DNSNR RDATA Wire Format

The RDATA of the DNSNR RR is as shown below:

```
                     1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|A|Hash Function|                       Salt                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                       Next Ownername                          /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                        Type Bit Maps                          /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

2.1.1 The Authoritative Only Flag Field

The Authoritative Only Flag field indicates whether the Type Bit Maps include delegation point NS record types.

If the flag is set to 0, the NS RR type bit for a delegation point ownername SHOULD be clear when the DNSNR RR is generated. The NS RR type bit MUST be ignored during processing of the DNSNR RR. The NS RR type bit has no meaning in this context (it is not authoritative), hence the DNSNR does not contest the existence of a NS RR type record for this ownername. When a delegation is not secured, there exist no DS RR type nor any other authoritative types for this delegation, hence the unsecured delegation has no DNSNR record associated.

Please see the Special Consideration section for implications for unsigned delegations

If the flag is set to 1, the NS RR type bit for a delegation point ownername MUST be set if the DNSNR covers a delegation, eventhough the NS RR itself is not authoritative. This implies that all delegations, signed or unsigned have a DNSNR record associated. This behaviour is identical to NSEC behaviour with regards to interprating the Type Bit Maps

2.1.2 The Hash Function Field

The Hash Funtion field identifies the cryptographic hash function used to construct the hash-value.

If the Hash Function field has value 0, no hash function was used, and the ownername is a set of plain labels.

If the Hash Function field has a value other the 0, a cryptographic hash function has been used to create a hash-value for each individual label under the apex ownername, prepended with the Salt field value, and is presented by a base32 value.

The hashed label of a DNSNR RR associated with the apex is a hash of the salt field value. Logically, the salt field is prepended to non-existent label and the hashed result is a label prepended to the apex.

This document defines Value 0 (no Hash Function) and Value 1 (SHA-1). All other values are reserved.

On reception, a resolver MUST discard DNSNR RR with an unknown Hash Function value.

2.1.3 The Salt Field

When the Hash Function field value is not zero, a hash funtion is used. In that case, the label is prepended with the salt field value before hashing in order to defend against precalculated dictionary attacks. The Salt field value must be the same for every DNSNR generated in the zone.

When the Hash Function field value is zero, the Salt field SHOULD have all bits set to zero, and MUST be ignored during processing.

2.1.4 The Next Ownername Field

If the Hash Function field has value 0, the Next Ownername field

contains the ownername of the next authoritative RRset in the canonical ordering of the zone; see ... for an explanation of canonical ordering. The value of the Next Owner Name field in the last DNSNR record in the zone is the name of the zone apex (the ownername of the zone's SOA RR).

If the Hash Function field has a value other then 0, the Next Ownername field contains the next hash-value of an ownername of an authoritative RRset in the canonical ordering of hash values for ownernames. The value of the Next Ownername field in the last DNSNR record in the zone is the value of the first hash value in canonical ordering of the zone.

A sender MUST NOT use DNS name compression on the Next Ownername field when transmitting an DNSNR RR.

A receiver which receives an DNSNR RR containing a compressed Next Ownername field SHOULD decompress the field value.

Ownernames of RRsets not authoritative for the given zone (such as glue records) MUST NOT be listed in the Next Ownername unless either authoritative RRset exists at the same ownername, or the Authortiative Only flag is clear and the ownername is an unsigned delegation point.

2.1.5 The Type Bit Maps Field

The Type Bit Maps field identifies the RRset types which exist at the DNSNR RR's ownername.

The Type bit for the DNSNR and RRSIG MUST be set during generation, and MUST be ignored during processing.

The RR type space is split into 256 window blocks, each representing the low-order 8 bits of the 16-bit RR type space. Each block that has at least one active RR type is encoded using a single octet window number (from 0 to 255), a single octet bitmap length (from 1 to 32) indicating the number of octets used for the window block's bitmap, and up to 32 octets (256 bits) of bitmap.

Blocks are present in the DNSNR RR RDATA in increasing numerical order.

Type Bit Maps field = (Window Block # | Bitmap Length | Bitmap)+

where "|" denotes concatenation.

Each bitmap encodes the low-order 8 bits of RR types within the

window block, in network bit order. The corresponds to RR type 1 (A), bit 2 corresponds to RR type 2 (NS), and so forth. For window block 1, bit 1 corresponds to RR type 257, bit 2 to RR type 258. If a bit is set to 1, it indicates that an RRset of that type is present for the DNSNR RR's ownername. If a bit is set to 0, it indicates that no RRset of that type is present for the DNSNR RR's ownername.

The RR type 2 (NS) is authoritative at the apex of a zone and is not authoritative at a delegation point. If the Authoritative Only Flag is clear, the delegation point NS RR type MUST NOT be included in the type bit maps. If the Authoritative Only Flag is set, the NS RR type at a delegation point MUST be included in the type bit maps.

Since bit 0 in window block 0 refers to the non-existent RR type 0, it MUST be set to 0. After verification, the validator MUST ignore the value of bit 0 in window block 0.

Bits representing pseudo-types MUST be set to 0, since they do not appear in zone data. If encountered, they MUST be ignored upon reading.

Blocks with no types present MUST NOT be included. Trailing zero octets in the bitmap MUST be omitted. The length of each block's bitmap is determined by the type code with the largest numerical value, within that block, among the set of RR types present at the DNSNR RR's ownername. Trailing zero octets not specified MUST be interpreted as zero octets.

A zone MAY include a DNSNR RR for ownernames at unsigned delegation points with Authoritative Only flag set to 1.

A zone MUST include a DNSNR RR for ownernames at unsigned delegation points with Authoritative Only flag set to 0.

Signed delegation points have an authoritative DS record for the ownername, and has therefor always a DNSNR record associated with the ownername.

Signed delegation point DNSNR records may have the NS type bit set in the type bit maps, depending on the MNR bit 0.

A zone MUST NOT generate an DNSNR RR for any domain name that only holds glue records.

2.1.6 Inclusion of Wildcard Names in DNSNR RDATA

If a wildcard ownername appears in a zone, the wildcard label ("*") is treated as a literal symbol and is treated the same as any other

ownername for purposes of generating DNSNR RRs. Wildcard owner names appear in the Next Ownername field without any wildcard expansion.

2.2 The DNSNR RR Presentation Format

The presentation format of the RDATA portion is as follows:

The NRM field is represented as a unsigned decimal integer. The value has a maximum of 3.

The Hash Function field is represented as an unsigned decimal integer. The value has a maximum of 127.

The Salt field is represented as a sequence of case-insensitive hexadecimal digits. Whitespace is allowed within the hexadecimal text.

The Next Ownername field is represented as a domain name.

The Type Bit Maps field is represented as a sequence of RR type mnemonics. When the mnemonic is not known, the TYPE representation as described in ... MUST be used.

2.3 DNSNR RR Example

The following DNSNR RR identifies the RRsets associated with alfa.example.com. and identifies the next authoritative name after alfa.example.com.

```
alfa.example.com. 86400 IN DNSNR 1 0 0 host.example.com. (
                                A MX RRSIG DNSNR TYPE1234 )
```

The first four text fields specify the name, TTL, Class, and RR type (DNSNR). The first RDATA value (1) indicate that the DNSNR RR includes non-authoritative NS types. The second value (0) indicate that the labels in the ownernames are not hash-values. The third value (0) is the Salt value and is ignored, since no Hash Function has been indicated. The entry host.example.com. is the next authoritative name after alfa.example.com. in canonical order. The A, MX, RRSIG and DNSNR mnemonics indicate there are A, MX, RRSIG, DNSNR, and TYPE1234 RRsets associated with the name alfa.example.com.

The RDATA section of the DNSNR RR above would be encoded as:

```
0x00 0x80 0x00 0x00 0x00
0x04 'h'  'o'  's'  't'
0x07 'e'  'x'  'a'  'm'  'p'  'l'  'e'
0x03 'c'  'o'  'm'  0x00
0x00 0x06 0x40 0x01 0x00 0x00 0x00 0x03
0x04 0x1b 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x20
```

Assuming that the resolver can authenticate this DNSNR record, it could be used to prove that beta.example.com does not exist, or could be used to prove there is no AAAA record associated with alfa.example.com.

3. Special Considerations

The following paragraphs clarify specific behaviour explain special considerations for implementations.

3.1 Delegation points

This proposal introduces the Authoritative Only Flag which indicates wether non authoritative delegation point NS records are included in the type bit Maps. As discussed in paragraph 2.1.1, a flag value of 1 indicates that the interpration of the type bit maps is identical to NSEC records

The following subsections describe behaviour when the flag value is 0.

3.1.1 Unsigned Delegations

Unsigned Delegation point NS records are not authoritative. They are authoritative in the delegated zone. There exist also no other data at the ownername of an unsigned delegation point.

Since no authoritative data exist at this ownername, it is excluded from the DNSNR chain. This is an optimisation since it relieves the zone of including an DNSNR record and its associated signature for this name.

A DNSNR that denies existence of ownernames between X and X' with the Authoritative Only Flag clear can not be used to proof presence nor absence of the delegation point NS records in the interval X, X'. The Authoritative Only Flag effectively states No Contest on the presence of delegation point NS resource records

Since proof is absent, there exist a new attack vector. Unsigned delegation point NS records can be deleted during a man in the middle attack, effectively dnying existence of the delegation. This is a form of Denial of Service, where the victim has no information it is under attack, since all signatures are valid and the fabricated response form is a known type of response.

The only possible mittigation is to either not use this method, hence proving existence or absence of unsigned delegations, or signing the delegated zone, changing the unsigned delegation into a signed delegation.

A second attack vector exists in that an adversary is able to succesfully fabricate a response claiming a not existant delegation to exist, though unsigned.

The only possible mittigation is to either not use this method, hence proving absence of unsigned delegations.

3.1.2 Salting

Augmenting labels with salt before hashing increases the cost of a dictionary of pre-generated hash-values. For every bit of salt, the cost of the dictionary doubles. The DNSNR RR uses 24 bits of salt, multiplying the cost with 2^24.

The salt value for each DNSNR RR MUST be equal for a single version of the zone.

3.2 Hash Collision

This section is relevant when the Hash Function field value is not zero, which indicates the use of a hash function.

Hash collisions occur when different messages have the same hash value. The probability that this happens during the generation of hash values is for SHA1 about 1 in 2^80 on average. Though this probability is low, the following paragraphs deal with avoiding Collisions and assessing possible damage in the event of an attack using Hash collisions.

3.2.1 Avoiding Hash Collisions during generation

During generation of DNSNR RRs, hash values are supposedly unique. In the (academic) case a collision does occur, an alternative salt SHOULD be chosen and all hash values SHOULD be regenerated.

In case hash values are not regenerated on collision, the DNSNR RR MUST list all authoritative RR types that exist for both messages, to avoid a replay attack, spoofing an existing type as non-existent.

3.2.2 Second Preimage Requirement Analysis

A collision resistant hash function has a second-preimage resistance property. The second-preimage resistance property means that it is computationally infeasible to find another message with the same hash value as a given message, i.e, given x, to find a second-preimage X' <> X such that hash(X) = hash(X'). The probability of finding a second preimage is 1 in 2^160 for SHA-1 on average. To mount an attack using an existing DNSNR RR, an adversary needs to find a second preimage.

Assuming an adversary is capable of mounting such an extreme, the actual damage is that a response message can be generated which

claims that a certain QNAME (i.e. the second pre-image) does exist, while in reality QNAME does not exist (aka false positive), which will either cause a security aware resolver to re-query for the non-existent name, or to fail the initial query. Note that the adversary can't mount this attack on an existing name, solely on a name that the adversary can't choose and does not yet exist.

3.2.3 Possible Hash Value Truncation Method

The previous sections outlined the low probability and low impact of a second-preimage attack. When impact and probability are low, while space in a DNS message is costly, truncation is tempting. Truncation might be considered to allow for shorter ownernames and rdata in case of labels with hash values. The author seeks comfirmation on the assumption that truncation decreases resilliance on colliding hash values though the overall security model does not significantly deteriorate.

An extreme hash value truncation would be truncating to the shortest possible unique label value. Considering that hash values are presented in base32, which represents 5 bits per label character, truncation must be done on a 5 bit boundary.

Though the mentioned truncation can be maximised to a certain extreme, the probability of collision increases exponentially for every truncated bit. Given the low impact of hash value collisions and limited space in DNS messages, the balance between truncation profit and collision damage may be determined by local policy (but see first paragraph where 'the author seeks').

3.3 Future Hash Functions

4. Acknowledgements

Thanks to Mark Kosters, David Blacka, Simon Josefsson, Paul Vixie and Ben Laurie for their time, review and input.

5. References

5.1 Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

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

[RFC2136] Vixie, P., Thomson, S., Rekhter, Y. and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, April 1997.

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, July 1997.

[RFC2308] Andrews, M., "Negative Caching of DNS Queries (DNS NCACHE)", RFC 2308, March 1998.

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

5.2 Informative References

Author's Address

Roy Arends
Telematica Instituut
Drienerlolaan 5
7522 NB Enschede
NL

EMail: roy.arends@telin.nl

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgment

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