Internet Draft
Expires: October 2005

G. Lozano
NIC Mexico

April 7, 2005

Random Boundary NSEC
draft-lozano-nsec-random-01.txt

Status of this Memo

This Internet-Draft will expire on October 7, 2005.

Copyright Notice

Abstract

The purpose of this memo is to introduce the RNXD and RNXRR resource records and the necessary modifications to the DNS protocol that permit secure denial of existence without the zone enumeration problems found in DNSSEC [RFC 4033, 4034 and 4035].

Conventions used in this document

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].

Table of Contents

1. Introduction

DNSSEC NSEC resource record is used to securely indicate that a name or a resource record type for a name does not exist in a zone by using ranges of non existence. The names in the zone are used as boundaries for the non existence ranges. An attacker can follow the chain created by the boundaries and enumerate the zone.

In today's Internet, it's difficult to obtain the zone data by sending queries when zone transfer policies have been properly configured.

This memo address the enumeration problems found in DNSSEC NSEC by the usage of ranges of non existence with random boundaries.

This memo is a NSEC++ proposal.

The author of this memo believes that the NSEC RR and the NSEC++ RR could coexist, because there are zones in which zone walking is not a problem, for example: registries which allow zone transfers, telephone number series, reverse zone data, and more.

The zone maintainer would be responsible for choosing the RR type for denial of existence that is better suited for a specific zone and situation.

2. Random Boundaries

In order to securely prove the non existence of a name in a zone, DNSSEC NSEC [RFC 4033, 4034 and 4035] creates ranges of non existence.

Consider the following:

```
b.example.com  A  10.10.10.1
e.example.com  A  10.10.10.2
h.example.com  A  10.10.10.2
```

Following DNSSEC, NSEC RRsets are created:

```
b.example.com NSEC e.example.com (A RRSIG NSEC)
e.example.com NSEC h.example.com (A RRSIG NSEC)
...
```

The ranges of non existence are:

```
From b - e
From e - h
...
```

Using random boundaries with the same example, a random name generation function is used to create three random names: a.example.com, d.example.com and f.example.com.

Three random ranges of non existence are created:

From a - d
From d - f
From f - a

Each range of non existence covers one or more of the original names found in the zone.

If two or more names which are immediate successors and antecessors between them are found in a zone then two random names would be created that cover them.

3. The RNXD Resource Record

The RNXD RR is used to deny the existence of a name.

The RNXD RR lists: the number of original names found in the random range of non existence, the algorithm used to produce the hashes of the original names, the next random generated owner name and the hash of each of the original names found in the random range of non existence.

The type value for the RNXD RR is ?.

The RNXD RR is class independent.

The RNXD RR SHOULD have the same TTL value as the SOA minimum TTL field. This is in the spirit of negative caching [RFC2308].

3.1. The RNXD wire format

The RDATA of a RNXD RR consists of: 1 octet Number of Original Names Field, 1 octet Algorithm Field, the Next Random Name field and the Original Hash Name Field.

```
                        1 1 1 1 1 1
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |Original Names |Hash Algorithm |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   /       Next Random Name        /
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   /      Original Hash Name       /
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.1.1. Number of Original Names Field

The Number of Original Names field specifies the number of original names that were covered in the random range of non existence.

3.1.2. Hash Algorithm Field

The Hash Algorithm field lists the algorithm function number used in the hash function to produce the hashes of the original names.

See Appendix A for a list of hash algorithm numbers.

3.1.3. Next Random Name Field

The Next Random Named field lists the next random generated name (in the canonical ordering of the zone).

A sender MUST NOT use DNS name compression on the Next Random Name field when transmitting a RNXD RR.

3.1.4. Original Hash Name Field

The Original Hash Name field lists the hashes of the original names covered by the random generated non existence range.

3.2. RNXD protocol modifications

Original name in the zone:

```
e.example.com  A  10.10.10.1
```

Random generated names that cover the original name:

```
d.example.com and f.example.com
```

RNXD RR:

```
d.example.com RNXD 1 1 f.example.com (Hash(e.example.com))
```

A resolver asking for esomething.example.com will receive:

d.example.com RNXD 1 1 f.example.com (eb391a1462ef3dda8fdba242cf55cdab843422bb) and the associated RRSIG RR.

The resolver MUST check the integrity of the response by validating the signature of the RRSIG record and by validating that the asked name is covered by the answer.

The hashes of the original names in the RNXD response are used to protect from man in the middle attacks. The resolver MUST verify that the denied of existence name is not an original name of the zone. The resolver computes the hash of the asked name and compares it with the hashes in the RNXD response. A matching hash will indicate a malfunctioning server or a man in the middle attack. The resolver SHOULD report a server error.

4. The RNRR Resource Record

The RNRR RR is used to deny the existence of a type of an existent name.

The RNRR RR lists: the types found in an original RRset.

Random generated names MUST NOT have RNRR RR associated with them.

The type value for the RNRR RR is ?.

The RNRR RR is class independent.

The RNRR RR SHOULD have the same TTL value as the SOA minimum TTL field. This is in the spirit of negative caching [RFC2308].

(Pending section)

The RDATA of a RNRR RR is similar to the Type Bit Maps field found in DNSSECbis [RFC4033, RFC4034 and RFC4035].

5. Wildcards

Wildcards will be covered in a newer version of this draft.

6. IANA Considerations

A registry for Hash Algorithm numbers need to be created.

7. Security Considerations

An attacker can analyze the random generated names in order to obtain some data of the original name by analyzing the response and extracting the equal prefix.

Consider the following zone as example:

```
research-america-almaden.ibm.com (original name)
research-asia-aaaa.ibm.com (random name)
research-asia-china.ibm.com (original name)
research-asia-dddd.ibm.com (random name)
research-asia-japan.ibm.com (original name)
```

An attacker asking form research-asia-cccc.ibm.com will obtain a RNXD in the form of:

```
research-asia-aaaa.ibm.com RNXD ..... research-asia-dddd.ibm.com.
```

The attacker will discover that research-asia is probably part of the original name. Note that the attack requires a previous knowledge about research-asia, and the attacker still requires a substantial number of queries to discover the original name. In the actual DNS an attacker can obtain a good number of names by using a dictionary, business directory and other listings, which requires less work than the attack previously mentioned.

An attacker can do a zone walking using the RNXD RRs and obtain the number of records in the zone. An attacker will need the same amount of work as in actual DNS to obtain the types of the original records.

It's possible to construct a zone with names that are immediate successors making RNXD RRs that cover more than one original name. The size of this kind of RNXD RRset will require the use EDNS-0 or TCP. It's unlikely to find immediate successors in the real world considering the limit of 255 bytes per name.

Hash collisions only affect original names that are covered by a random range of non existence.

APPENDIX A: Hash Algorithms

The Hash Algorithm field lists the algorithm number used to produce the hash of the original names

The following algorithms are assigned.

VALUE	Algorithm	STATUS
0	Reserved	-
1	SHA-1	MANDATORY
2	SHA-512	OPTIONAL
3-255	Unassigned	-

8. References

8.1. Normative References

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

[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.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for DNS Security Extensions", RFC 4034, March 2005.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

8.2. Informative References

Author's Addresses

Gustavo Lozano
NIC Mexico
Av. Eugenio Garza Sada #427 Sur Col. Altavista
Monterrey, NL
Mexico

Email: glozano@nic.mx

Intellectual Property Statement

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

Acknowledgment

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