Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 19, 2016

K. Fujiwara
JPRS
A. Kato
Keio/WIDE
March 18, 2016

Aggressive use of NSEC/NSEC3
draft-fujiwara-dnsop-nsec-aggressiveuse-03

Abstract

While DNS highly depends on cache, its cache usage of non-existence information has been limited to exact matching. This draft proposes the aggressive use of a NSEC/NSEC3 resource record, which is able to express non-existence of a range of names authoritatively. With this proposal, it is expected that shorter latency to many of negative responses as well as some level of mitigation of random sub-domain attacks (referred to as "Water Torture" attacks). It is also expected that non-existent TLD queries to Root DNS servers will decrease.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on September 19, 2016.

Copyright Notice

Table of Contents

1. Introduction

While negative (non-existence) information of DNS caching mechanism has been known as DNS negative cache [RFC2308], it requires exact matching in most cases. Assume that "example.com" zone doesn't have names such as "a.example.com" and "b.example.com". When a full-service resolver receives a query "a.example.com" , it performs a DNS

resolution process, and eventually gets NXDOMAIN and stores it into its negative cache. When the full-service resolver receives another query "b.example.com", it doesn't match with "a.example.com". So it will send a query to one of the authoritative servers of "example.com". This was because the NXDOMAIN response just says there is no such name "a.example.com" and it doesn't tell anything for "b.example.com".

Section 5 of [RFC2308] seems to show that negative answers should be cached only for the exact query name, and not (necessarily) for anything below it.

Recently, DNSSEC [RFC4035] [RFC5155] has been practically deployed. Two types of resource record (NSEC and NSEC3) along with their RRSIG records represent authentic non-existence. For a zone signed with NSEC, it would be possible to use the information carried in NSEC resource records to indicate that a range of names where no valid name exists. Such use is discouraged by Section 4.5 of RFC 4035, however.

This document proposes to make a minor change to RFC 4035 and a full-service resolver can use NSEC/NSEC3 resource records aggressively so that the resolver responds with NXDOMAIN immediately if the name in question falls into a range expressed by a NSEC/NSEC3 resource record.

Aggressive Negative Caching was first proposed in Section 6 of DNSSEC Lookaside Validation (DLV) [RFC5074] in order to find covering NSEC records efficiently. Unbound [UNBOUND] has aggressive negative caching code in its DLV validator. Unbound TODO file contains "NSEC/NSEC3 aggressive negative caching".

Section 3 of [I-D.vixie-dnsext-resimprove] ("Stopping Downward Cache Search on NXDOMAIN") proposed another approach to use NXDOMAIN information effectively.

2. Terminology

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].

Many of the specialized terms used in this specification are defined in DNS Terminology [RFC7719].

3. Problem Statement

Random sub-domain attacks (referred to as "Water Torture" attacks or NXDomain attacks) send many non-existent queries to full-service resolvers. Their query names consist of random prefixes and a target domain name. The negative cache does not work well and target full-service resolvers result in sending queries to authoritative DNS servers of the target domain name.

When number of queries is large, the full-service resolvers drop queries from both legitimate users and attackers as their outstanding queues are filled up.

For example, BIND 9.10.2 [BIND9] full-service resolvers answer SERVFAIL and Unbound 1.5.2 full-service resolvers drop most of queries under 10,000 queries per second attack.

The countermeasures implemented at this moment are rate limiting and disabling name resolution of target domain names in ad-hoc manner.

4. Proposed Solution

4.1. Aggressive Negative Caching

If the target domain names are DNSSEC signed, aggressive use of NSEC/NSEC3 resource records mitigates the problem.

Section 4.5 of [RFC4035] shows that "In theory, a resolver could use wildcards or NSEC RRs to generate positive and negative responses

(respectively) until the TTL or signatures on the records in question expire. However, it seems prudent for resolvers to avoid blocking new authoritative data or synthesizing new data on their own. Resolvers that follow this recommendation will have a more consistent view of the namespace".

To reduce non-existent queries sent to authoritative DNS servers, it is suggested to relax this restriction as follows:

```
+-------------------------------------------------------------+
|  DNSSEC enabled full-service resolvers MAY use              |
|  NSEC/NSEC3 resource records to generate negative responses |
|  until their effective TTLs or signatures on the records    |
|  in question expire.                                        |
+-------------------------------------------------------------+
```

If the full-service resolver's cache have enough information to validate the query, the full-service resolver MAY use NSEC/NSEC3/

wildcard records aggressively. Otherwise, the full-service resolver MUST fall back to send the query to the authoritative DNS servers.

Necessary information to validate are matching/covering NSEC/NSEC3 of the wildcards which may match the query name, matching/covering NSEC/NSEC3 of non-terminals which derive from the query name and matching/covering NSEC/NSEC3 of the query name.

If the query name has the matching NSEC/NSEC3 RR and it shows the query type does not exist, the full-service resolver is possible to respond with NODATA (empty) answer.

4.2. NSEC

A full-service resolver implementation SHOULD support aggressive use of NSEC and enable it by default. It SHOULD provide a configuration knob to disable aggressive use of NSEC.

The validating resolver need to check the existence of matching wildcards which derive from the query name, covering NSEC RRs of the matching wildcards and covering NSEC RR of the query name.

If the full-service resolver's cache contains covering NSEC RRs of matching wildcards and the covering NSEC RR of the query name, the full-service resolver is possible to respond with NXDOMAIN error immediately.

4.3. NSEC3

NSEC3 aggressive negative caching is more difficult. If the zone is signed with NSEC3, the validating resolver need to check the existence of non-terminals and wildcards which derive from query names.

If the full-service resolver's cache contains covering NSEC3 RRs of matching wildcards, the covering NSEC3 RRs of the non-terminals and the covering NSEC3 RR of the query name, the full-service resolver is possible to respond with NXDOMAIN error immediately.

If the validating resolver proves the non-exisence of the non-terminal domain name of the query name, the query name does not exist.

To identify signing types of the zone, validating resolvers need to build separated cache of NSEC and NSEC3 resource records for each signer domain name.

When a query name is not in the regular cache, find closest enclosing NS RRset in the regular cache. The owner of the closest enclosing NS RRset may be the longest signer domain name of the query name. If there is no entry in the NSEC/NSEC3 cache of the signer domain name, aggressive negative caching is not possible at this moment. Otherwise, there is at least one NSEC or NSEC3 resource records. The record shows the signing type.

A full-service resolver implementation MAY support aggressive use of NSEC3. It SHOULD provide a configuration knob to disable aggressive use NSEC3 in this case.

4.4. NSEC3 Opt-Out

If the zone is signed with NSEC3 and with Opt-Out flag set to 1, the

aggressive negative caching is not possible at the zone.

4.5. Wildcard

Even if a wildcard is cached, it is necessary to send a query to an authoritative server to ensure that the name in question doesn't exist as long as the name is not in the negative cache.

When aggressive use is enabled, regardless of description of Section 4.5 of [RFC4035], it is possible to send a positive response immediately when the name in question matches a NSEC/NSEC3 RRs in the negative cache.

4.6. Consideration on TTL

This function needs care on the TTL value of negative information because newly added domain names cannot be used while the negative information is effective. RFC 2308 states the maximum number of negative cache TTL value is 10800 (3 hours). So the full-service resolver SHOULD limit the maximum effective TTL value of negative responses (NSEC/NSEC3 RRs) to 10800 (3 hours). It is reasonably small but still effective for the purpose of this document as it can eliminate significant amount of DNS attacks with randomly generated names.

5. Additional Considerations

5.1. The CD Bit

The CD bit disables signature validation. It is one of the basic functions of DNSSEC protocol and it SHOULD NOT be changed. However, attackers may set the CD bit to their attack queries and the aggressive negative caching will be of no use.

Ignoring the CD bit function may break the DNSSEC protocol.

This draft proposes that the CD bit may be ignored to support aggressive negative caching when the full-service resolver is under attacks with CD bit set.

5.2. Detecting random subdomain attacks

Full-service resolvers should detect conditions under random subdomain attacks. When they are under attacks, their outstanding queries increase. If there are some destination addresses whose outstanding queries are many, they may contain attack target domain names. Existing countermeasures may implement attack detection.

6. Possible side effect

Aggressive use of NSEC/NSEC3 resource records may decrease queries to Root DNS servers.

People may generate many typos in TLD, and they will result in unnecessary DNS queries. Some implementations leak non-existent TLD queries whose second level domain are different each other. Well observed TLDs are ".local" and ".belkin". With this proposal, it is possible to return NXDOMAIN immediately to such queries without further DNS recursive resolution process. It may reduces round trip time, as well as reduces the DNS queries to corresponding authoritative servers, including Root DNS servers.

7. Additional proposals

There are additional proposals to the aggressive negative caching.

7.1. Partial implementation

It is possible to implement aggressive negative caching partially.

DLV aggressive negative caching [RFC5074] is an implementation of NSEC aggressive negative caching which targets DLV domain names.

NSEC only aggressive negative caching is easier to implement NSEC/NSEC3 aggressive negative caching (full implantation) because NSEC3 handling is hard to implement.

Root only aggressive negative caching is possible. It uses NSEC and RRSIG resource records whose signer domain name is root.

An implementation without detecting attacks is possible. It cannot ignore the CD bit and the effectiveness may be limited.

7.2. Aggressive negative caching without DNSSEC validation

Aggressive negative caching may be applicable to full-service resolvers without DNSSEC validation. They can set DNSSEC OK bit in query packets to obtain corresponding NSEC/NSEC3 resource records. While the full-service resolvers SHOULD validate the NSEC/NSEC3 resource records, they MAY use the records to respond NXDOMAIN error immediately without DNSSEC validation.

However, it is highly recommended to apply DNSSEC validation.

7.3. Aggressive negative caching flag idea

Authoritative DNS servers that dynamically generate NSEC records normally generate minimally covering NSEC Records [RFC4470]. Aggressive negative caching does not work with minimally covering NSEC records. Most of DNS operators don't want zone enumeration and zone information leaks. They prefer NSEC resource records with narrow ranges. When there is a flag that show a full-service resolver support the aggressive negative caching and a query have the aggressive negative caching flag, authoritative DNS servers can generate NSEC resource records with wider range under random subdomain attacks.

However, changing range of minimally covering NSEC Records may be implemented by detecting attacks. Authoritative DNS servers can answer any range of minimally covering NSEC Records.

8. IANA Considerations

This document has no IANA actions.

9. Security Considerations

Newly registered resource records may not be used immediately. However, choosing suitable TTL value will mitigate the problem and it is not a security problem.

It is also suggested to limit the maximum TTL value of NSEC resource records in the negative cache to, for example, 10800 seconds (3hrs), to mitigate the issue. Implementations which comply with this proposal is suggested to have a configurable maximum value of NSEC RRs in the negative cache.

Aggressive use of NSEC/NSEC3 resource records without DNSSEC validation may cause security problems.

10. Implementation Status

Unbound has aggressive negative caching code in its DLV validator. The author implemented NSEC aggressive caching using Unbound and its DLV validator code.

11. Acknowledgments

The authors gratefully acknowledge DLV [RFC5074] author Samuel Weiler and Unbound developers. Olafur Gudmundsson and Pieter Lexis proposed aggressive negative caching flag idea. Valuable comments were provided by Bob Harold, Tatuya JINMEI, Shumon Huque, Mark Andrews, and Casey Deccio.

12. Change History

This section is used for tracking the update of this document. Will be removed after finalize.

12.1. Version 01

- Added reference to DLV [RFC5074] and imported some sentences.
- Added Aggressive Negative Caching Flag idea.
- Added detailed algorithms.

12.2. Version 02

- Added reference to [I-D.vixie-dnsext-resimprove]
- Added considerations for the CD bit
- Updated detailed algorithms.
- Moved Aggressive Negative Caching Flag idea into Additional Proposals

12.3. Version 03

- Added "Partial implementation"
- Section 4,5,6 reorganized for better representation
- Added NODATA answer in Section 4

- Trivial updates

- Updated pseudo code

13. References

13.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2308] Andrews, M., "Negative Caching of DNS Queries (DNS NCACHE)", RFC 2308, DOI 10.17487/RFC2308, March 1998, <http://www.rfc-editor.org/info/rfc2308>.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <http://www.rfc-editor.org/info/rfc4035>.

[RFC4470] Weiler, S. and J. Ihren, "Minimally Covering NSEC Records and DNSSEC On-line Signing", RFC 4470, DOI 10.17487/RFC4470, April 2006, <http://www.rfc-editor.org/info/rfc4470>.

[RFC5074] Weiler, S., "DNSSEC Lookaside Validation (DLV)", RFC 5074, DOI 10.17487/RFC5074, November 2007, <http://www.rfc-editor.org/info/rfc5074>.

[RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, DOI 10.17487/RFC5155, March 2008, <http://www.rfc-editor.org/info/rfc5155>.

[RFC7719] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", RFC 7719, DOI 10.17487/RFC7719, December 2015, <http://www.rfc-editor.org/info/rfc7719>.

13.2. Informative References

[BIND9] Internet Systems Consortium, Inc., "Name Server Software", 2000, <https://www.isc.org/downloads/bind/>.

[I-D.vixie-dnsext-resimprove]
Vixie, P., Joffe, R., and F. Neves, "Improvements to DNS Resolvers for Resiliency, Robustness, and Responsiveness", draft-vixie-dnsext-resimprove-00 (work in progress), June 2010.

[UNBOUND] NLnet Labs, "Unbound DNS validating resolver", 2006, <http://www.unbound.net/>.

Appendix A. Aggressive negative caching from RFC 5074

Imported from Section 6 of [RFC5074].

Previously, cached negative responses were indexed by QNAME, QCLASS, QTYPE, and the setting of the CD bit (see RFC 4035, Section 4.7), and only queries matching the index key would be answered from the cache. With aggressive negative caching, the validator, in addition to checking to see if the answer is in its cache before sending a query, checks to see whether any cached and validated NSEC record denies the existence of the sought record(s).

Using aggressive negative caching, a validator will not make queries for any name covered by a cached and validated NSEC record. Furthermore, a validator answering queries from clients will synthesize a negative answer whenever it has an applicable validated NSEC in its cache unless the CD bit was set on the incoming query.

Imported from Section 6.1 of [RFC5074].

Implementing aggressive negative caching suggests that a validator will need to build an ordered data structure of NSEC records in order to efficiently find covering NSEC records. Only NSEC records from DLV domains need to be included in this data structure.

Appendix B. Detailed implementation idea

Section 6.1 of [RFC5074] is expanded as follows.

Implementing aggressive negative caching suggests that a validator will need to build an ordered data structure of NSEC and NSEC3 records for each signer domain name of NSEC / NSEC3 records in order to efficiently find covering NSEC / NSEC3 records. Call the table as NSEC_TABLE.

The aggressive negative caching may be inserted at the cache lookup part of the full-service resolvers.

If errors happen in aggressive negative caching algorithm, resolvers MUST fall back to resolve the query as usual. "Resolve the query as usual" means that the full-resolver resolve the query in Recursive-mode as if the full-service resolver does not implement aggressive negative caching.

To implement aggressive negative caching, resolver algorithm near cache lookup will be changed as follows:

```
QNAME = the query name;
QTYPE = the query type;
if ({QNAME,QTYPE} entry exists in the cache) {
    // the resolver responds the RRSet from the cache
    resolve the query as usual;
}

// if NSEC* exists, QTYPE existence is proved by type bitmap
if (matching NSEC/NSEC3 of QNAME exists in the cache) {
    if (QTYPE exists in type bitmap of NSEC/NSEC3 of QNAME) {
        // the entry exists, however, it is not in the cache.
        // need to iterate QNAME/QTYPE.
        resolve the query as usual;
    } else {
        // QNAME exists, QTYPE does not exist.
        the resolver can generate NODATA response;
    }
}

// Find closest enclosing NS RRset in the cache.
// The owner of this NS RRset will be a suffix of the QNAME
//    - the longest suffix of any NS RRset in the cache.
```

```
SIGNER = closest enclosing NS RRSet of QNAME in the cache;

// Check the SOA RR of the SIGNER
if (SOA RR of SIGNER does not exist in the cache
    or SIGNER zone is not signed or not validated) {
   Resolve the query as usual;
}

if (SIGNER zone does not have NSEC_TABLE) {
    Resolve the query as usual;
}

if (SIGNER zone is signed with NSEC) { // NSEC mode

    // Check the non-existence of QNAME
    CoveringNSEC = Find the covering NSEC of QNAME;
    if (Covering NSEC doesn't exist in the cache) {
        Resolve the query as usual.
    }

    // Select the longest existing name of QNAME from covering NSEC
    LongestExistName = common part of both owner name and
                       next domain name of CoveringNSEC;
```



```
    if (*.LongestExistName entry exists in the cache) {
        the resolver can generate positive response
        // synthesize the wildcard *.TEST
    }
    if covering NSEC RR of "*.LongestExistName" at SIGNER zone exists
         in the cache {
        the resolver can generate negative response;
    }
    //*.LongestExistName may exist. cannot generate negative response
    Resolve the query as usual.

} else
if (SIGNER zone is signed with NSEC3 and does not use Opt-Out) {
    // NSEC3 mode

    TEST = SIGNER;
    while (TEST != QNAME) {
        // if any error happens in this loop, break this loop
```

```
            UPPER = TEST;
            add a label from the QNAME to the start of TEST;
              // TEST = label.UPPER
            if (TEST name entry exist in the cache
                || matching NSEC3 of TEST exist in the cache) {
                // TEST exist
                continue; // need to check rest of QNAME
            }
            if (covering NSEC3 of TEST exist in the cache) {
                // (non-)terminal name TEST does not exist
                if (*.UPPER name entry exist in the cache) {
                    // TEST does not exist and *.UPPER exist
                    the resolver can generate positive response;
                } else
                if (covering NSEC3 of *.UPPER exist in the cache) {
                    // TEST does not exist and *.UPPER does not exist
                    the resolver can generate negative response;
                }
                break; // Lack of information (No *.UPPER information)
            }
            break; // Lack of information (No TEST information)
        }
        // no matching/covering NSEC3 of QNAME information
        Resolve the query as usual
    }
```


Authors' Addresses

Kazunori Fujiwara
Japan Registry Services Co., Ltd.
Chiyoda First Bldg. East 13F, 3-8-1 Nishi-Kanda
Chiyoda-ku, Tokyo 101-0065
Japan

Phone: +81 3 5215 8451
Email: fujiwara@jprs.co.jp

Akira Kato
Keio University/WIDE Project
Graduate School of Media Design, 4-1-1 Hiyoshi
Kohoku, Yokohama 223-8526
Japan

Phone: +81 45 564 2490
Email: kato@wide.ad.jp