Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 7, 2016

K. Fujiwara
JPRS
A. Kato
Keio/WIDE

July 06, 2015

Aggressive use of NSEC/NSEC3
draft-fujiwara-dnsop-nsec-aggressiveuse-01

Abstract

While DNS highly depends on cache, its cache usage of non-existence information was limited to exact matching. This draft proposes the aggressive use of a NSEC/NSEC3 resource record, which is able to express non-existence of range of names authoritatively. With this proposal, shorter latency to many of negative responses is expected 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 January 7, 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".

Recently, DNSSEC [RFC4035] [RFC5155] has been practically deployed. Two types of resource record (NSEC and NSEC3) are used for authentic non-existence. For a zone signed with NSEC, it may be possible to use the information carried in NSEC resource records to indicate that the 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 the full-service resolver can use NSEC/NSEC3 resource records aggressively.

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".

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 [I-D.ietf-dnsop-dns-terminology].

3. Problem Statement

Random sub-domain attacks (referred to as "Water Torture" 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.

4. Proposed Solution: 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 wildcards which match the query name, covering NSEC/NSEC3 of the wildcards, and covering NSEC/NSEC3 of (parts of) the query name.

If the zone has a wildcard and it is in the full-service resolver's cache, the full-service resolver MAY generate positive responses based on the information associated with the wildcard in the cache.

This approach is effective for DNSSEC signed zones with NSEC or NSEC3, except zones with NSEC3 Opt-Out. To identify signing types of the zone, validating resolvers need to build special cache of NSEC and NSEC3 resource records for each signer domain name. When a query name is not in the cache, find closest enclosing NS RRset in the cache. The owner of this NS RRset may be the longest signer domain name of the query name. If the NSEC/NSEC3 cache of the signer domain name is empty, the aggressive negative caching is not possible. Otherwise, there is at least one NSEC or NSEC3 resource records. The record shows the signing type. If the resource record is NSEC3 and with Opt-Out, the aggressive negative caching is not possible.

When the query name has a matching NSEC resource records in the cache and there is no wildcard in the zone which the query name matches with, the full-service resolver is allowed to respond with NXDOMAIN error immediately.

NSEC3 aggressive negative caching is more difficult. If the zone is signed with NSEC3, the validating resolver need to check the existence of each label from the query name. If a label is not exist in the zone, and there is no matching wildcard in the zone, the full-

service resolver is allowed to respond with NXDOMAIN error immediately.

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.

The same discussion is also applicable to wildcards. If a query name is covered by a NSEC or a NSEC3 resource record in the cache and there is a covering wildcard, the full-service resolver MAY use wildcards to generate positive responses while wildcard and NSEC/NSEC3 resource records in the cache are effective.

5. 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.

6. Another option

The proposed technique is applicable to zones where there is a NSEC record to each owner name in the zone even without DNSSEC signed. And it is also applicable to full-service resolvers without DNSSEC validation. Full-service resolvers can set DNSSEC OK bit in query packets and they will cache NSEC/NSEC3 resource records. They can apply aggressive use of NSEC/NSEC3 resource records without DNSSEC validation.

It is highly recommended to sign the zone, of course, and it is recommended to apply DNSSEC validation of NSEC record prior to cache it in the negative cache.

7. Aggressive negative caching flag

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. DNS operators don't want zone walking and zone information leaks. They prefer NSEC resource records with narrow ranges. When a query have the aggressive negative caching flag (AN flag), they can generate NSEC resource records with wider range under random subdomain attacks.

A full-service resolver which supports aggressive negative caching SHOULD set the AN flag when sending queries to authoritative DNS servers.

8. IANA Considerations

This document reserves one of these bits as the AN flag (Aggressive Negative Caching flag) in EDNS Header Flags defined in EDNS0 [RFC6891].

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 Considerations

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.

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.

13. References

13.1. Normative References

[I-D.ietf-dnsop-dns-terminology] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", draft-ietf-dnsop-dns-terminology-03 (work in progress), June 2015.

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

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

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

[RFC4470] Weiler, S. and J. Ihren, "Minimally Covering NSEC Records and DNSSEC On-line Signing", RFC 4470, April 2006.

[RFC5074] Weiler, S., "DNSSEC Lookaside Validation (DLV)", RFC 5074, November 2007.

[RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, March 2008.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, April 2013.

13.2. Informative References

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

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

Appendix A. Aggressive negative caching from RFC 5074

Imported from RFC 5074.

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.

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

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

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.

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

```
If the query name entry exists in the cache {
    resolve the query as usual.
```

```
        // if RRSet (query name and query type) exists in the cache,
        //     the resolver responds the RRSet from the cache
        // Otherwise, the resolver needs to iterate the query.
    }

    // 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 the QNAME in the cache.

    If SIGNER zone does not have a special NSEC/NSEC3 data structure {
        Resolve the query as usual.
    }

    if SIGNER zone is not signed or not validated {
       Resolve the query as usual
    }

    If SIGNER zone is signed with NSEC {
        // NSEC mode
        If covering NSEC RR of the query name at SIGNER zone
           doesn't exist in the cache {
            Resolve the query as usual.
        }

        TEST = Find the longest existing domain name of the query name
               from the covering NSEC RR

        if "*.TEST" exists in the cache {
            the resolver can generate positive response
            or resolve the query as usual.
        }

        if covering NSEC RR of "*.TEST" at SIGNER zone exists
             in the cache {
            the resolver can generate negative response.
        }

        // Lack of information, need to resolve the query as usual

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

        QNAME = the query name
        TEST = SIGNER
        do {
            UPPER = TEST
```

```
            add a label from the QNAME to the start of TEST
            If the covering NSEC3 of TEST exist in the cache {
                // non-terminal name TEST does not exist
                if *.UPPER or NSEC3 of *.UPPPER exist in the cache {
                    the resolver can generate negative response.
                } else {
                if the covering NSEC3 of *.UPPER exist in the cache {
                    the resolver can generate positive response.
                }
                // lack of information
                break
            } else
            if the NSEC3 of TEST does not exist {
                // lack of information
                break
            }
            // TEST label exist, then need to check the rest of the QNAME
        } while(TEST != QNAME)
        // lack of 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