Domain Name System Operations (dnsop) Working Group S. Bortzmeyer
Internet-Draft AFNIC
Intended status: Standards Track November 6, 2015
Expires: May 9, 2016

NXDOMAIN really means there is nothing underneath
draft-bortzmeyer-dnsop-nxdomain-cut-00

Abstract

This document states clearly that when a DNS resolver receives a response with status code NXDOMAIN, it means that the name in the question section AND ALL THE NAMES UNDER IT do not exist.

REMOVE BEFORE PUBLICATION: this document should be discussed in the IETF DNSOP (DNS Operations) group, through its mailing list. The source of the document, as well as a list of open issues, is currently kept on at Github [1].

Table of Contents

1. Introduction and background

In virtually all existing resolvers, a cached NXDOMAIN is not considered "proof" that there can be no child domains underneath. This is due to an ambiguity in [RFC1034] that failed to distinguish ENT (empty nonterminal domain names, [I-D.ietf-dnsop-dns-terminology]) from nonexistent names. For DNSSEC, the IETF had to distinguish this case ([RFC4034], section 3.1.3.2), but the implication on non-DNSSEC resolvers wasn't fully realized.

This document dictates that NXDOMAIN means NXDOMAIN for all child domains. Since the domain names are organized in a tree, it is a simple consequence of the tree structure: non-existence of a node implies non-existence of the entire sub-tree rooted at this node.

1.1. Requirements 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 [RFC2119].

2. Rules

When searching downward in its cache, an iterative caching DNS resolver SHOULD stop searching if it encounters a cached NXDOMAIN. The response to the triggering query should be NXDOMAIN.

When an iterative caching DNS resolver stores an NXDOMAIN in its cache, all names and RRsets at or below that node SHOULD be deleted since they will have become unreachable.

By implication, a stream of queries foo.example, then bar.foo.example, where foo.example does not exist would normally cause both queries to be forwarded to example's nameservers. Following this recommended practice of "NXDOMAIN cut", the second query and indeed any other query for names at or below foo.example would not be forwarded.

3. Benefits

The main benefit is a better efficiency of the caches. In the example above, we send only one query instead of two, the second one being answered from the cache.

The correct behavior (in [RFC1034] and made clearer in this document) is specially useful when combined with QNAME minimisation [I-D.ietf-dnsop-qname-minimisation] since it will allow to stop searching as soon as a NXDOMAIN is encountered.

NXDOMAIN cut may also help with random QNAME attacks [joost-dnsterror] [balakrichenan-dafa888]. In these attacks, queries are sent for a QNAME composed of a fixed suffix ("dafa888.wf" in one of the articles above), which is typically nonexistent, and a random prefix, different for each request. A resolver receiving these requests have to forward them to the authoritative servers. With NXDOMAIN cut, we would just have to send to the resolver a query for the fixed suffix, the resolver would get a NXDOMAIN and then would stop forwarding the queries. (It would be better if the SOA record in the NXDOMAIN response were sufficient to find the non-existing domain but this is more delicate, see Section 5.)

Since the principles set in this document are so great, why are the rules of Section 2 SHOULD and not MUST? This is because some resolver may have a cache which is NOT organized as a tree (but, for instance, as a dictionary) and therefore have a good reason to ignore this.

4. Possible issues

Let's assume the TLD example exists but foobar.example is not delegated (so the example's name servers will reply NXDOMAIN for a query about anything.foobar.example). A system administrator decides to name the internal machines of his organization under office.foobar.example and use a trick of his resolver to forward requests about this zone to his local authoritative name servers. NXDOMAIN cut would create problems here, since, depending on the order of requests to the resolver, it may have cached the NXDOMAIN from example and therefore "deleted" everything under. This document assumes that such setup is rare and does not need to be supported.

Another issue that may happen: today, we see broken authoritative name servers which reply to ENT ([I-D.ietf-dnsop-dns-terminology], section 6) with NXDOMAIN instead of the normal NODATA ([I-D.ietf-dnsop-dns-terminology], section 3).

RFC-EDITOR: REMOVE THE PARAGRAPH BEFORE PUBLICATION. An example today is mta2._domainkey.cbs.nl (which exists) where querying _domainkey.cbs.nl yields NXDOMAIN. Another example is www.upenn.edu, redirected to www.upenn.edu-dscg.edgesuite.net while a query for edu-dscg.edgesuite.net returns NXDOMAIN.

Such name servers are definitely broken and have always been. They MUST be fixed. Given the advantages of NXDOMAIN cuts, there is little reason to support this behavior.

5. Future work

In this document, we deduce the non-existence of a domain only for NXDOMAIN answers where the QNAME was this exact domain. If a resolver sends a query to the name servers of the TLD example, and asks the MX record for www.foobar.example, and receives a NXDOMAIN, it can only register the fact that www.foobar.example (and everything underneath) does not exist. Even if the accompanying SOA record is for example only, it may be dangerous to infer that foobar.example is nonexistent. TODO explain why.

In the future, deducing the non-existence of a node from the SOA in the NXDOMAIN reply may certainly help with random qnames attacks but this is out-of-scope for this document.

6. IANA Considerations

This document has no actions for IANA.

7. Security Considerations

The technique described here may help against a denial-of-service attack named "random qnames" and described in Section 3. Apart from that, it is believed to have no security consequences.

If a resolver does not validate the answers with DNSSEC, it can of course be poisoned with a false NXDOMAIN, thus "deleting" a part of the domain name tree. This denial-of-service attack is already possible with the rules of this document (but "NXDOMAIN cut" may increase its effects). The only solution is to use DNSSEC.

8. Implementation status - RFC EDITOR: REMOVE BEFORE PUBLICATION

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC6982]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC6982], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

As of today, all existing DNS resolvers are conservative: they consider a NXDOMAIN as only significant for the name itself, not for the names under. All current recursive servers will upstream a query for out-of-cache sub.example.com even if their cache contains an NXDOMAIN for example.com.

9. Acknowledgments

The text of this document was mostly copied from [I-D.vixie-dnsext-resimprove], section 3. Thanks to its authors, Paul Vixie, Roland Joffe and Frederico Neves.

10. References

10.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <http://www.rfc-editor.org/info/rfc1034>.

[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>.

[I-D.ietf-dnsop-dns-terminology] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", draft-ietf-dnsop-dns-terminology-05 (work in progress), September 2015.

10.2. Informative References

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <http://www.rfc-editor.org/info/rfc4034>.

[RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", RFC 6982, DOI 10.17487/RFC6982, July 2013, <http://www.rfc-editor.org/info/rfc6982>.

[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.

[I-D.ietf-dnsop-qname-minimisation] Bortzmeyer, S., "DNS query name minimisation to improve privacy", draft-ietf-dnsop-qname-minimisation-07 (work in progress), October 2015.

[joost-dnsterror] Joost, M., "About DNS Attacks and ICMP Destination Unreachable Reports", December 2014, <http://www.michael-joost.de/dnsterror.html>.

[balakrichenan-dafa888]
Balakrichenan, S., "Disturbance in the DNS - "Random qnames", the dafa888 DoS attack"", October 2014, <https://indico.dns-oarc.net/event/20/session/3/contribution/37>.

10.3. URIs

[1] https://github.com/bortzmeyer/ietf-dnsop-nxdomain

Author's Address

Stephane Bortzmeyer
AFNIC
1, rue Stephenson
Montigny-le-Bretonneux 78180
France

Phone: +33 1 39 30 83 46
Email: bortzmeyer+ietf@nic.fr
URI: http://www.afnic.fr/