Domain Name System Operations (dnsop) Working Group
Internet-Draft
Intended status: Experimental
Expires: September 5, 2015

S. Bortzmeyer
AFNIC

March 4, 2015

DNS query name minimisation to improve privacy
draft-ietf-dnsop-qname-minimisation-02

Abstract

This document describes one of the techniques that could be used to improve DNS privacy (see [I-D.ietf-dprive-problem-statement]), a technique called "qname minimisation".

REMOVE BEFORE PUBLICATION Discussions of the document should take place on the DNSOP working group mailing list [dnsop].

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 5, 2015.

Copyright Notice

Table of Contents

1. Introduction and background

The problem statement is exposed in [I-D.ietf-dprive-problem-statement] TODO: add a reference to the specific section when ietf-dprive-problem-statement will be published as RFC. The terminology ("qname", "resolver", etc) is also defined in this companion document. This specific solution is not intended to fully solve the DNS privacy problem; instead, it should be viewed as one tool amongst many.

It follows the principle explained in section 6.1 of [RFC6973]: the less data you send out, the fewer privacy problems you'll get.

2. Qname minimisation

The idea is to minimise the amount of data sent from the DNS resolver. Under current practice, when a resolver receives the query "What is the AAAA record for www.example.com?", it sends to the root (assuming a cold resolver, whose cache is empty) the very same question. Sending "What are the NS records for .com?" would be sufficient (since it will be the answer from the root anyway). This is compatible with the current DNS system and therefore can easily be deployed; since it is a unilateral change to the resolver, it does not change the protocol. Because of that, resolver implementers may do qname minmisation in slightly different ways.

To do such minimisation, the resolver needs to know the zone cut [RFC2181]. Zone cuts do not necessarily exist at every label boundary. If we take the name www.foo.bar.example, it is possible

that there is a zone cut between "foo" and "bar" but not between "bar" and "example". So, assuming the resolver already knows the name servers of .example, when it receives the query "What is the AAAA record of www.foo.bar.example", it does not always know whether the request should be sent to the name servers of bar.example or to those of example. [RFC2181] suggests a method to find the zone cut (section 6), so resolvers may try it.

Note that DNSSEC-validating resolvers already have access to this information, since they have to find the zone cut (the DNSKEY record set is just below, the DS record set just above).

One should note that the behaviour suggested here (minimising the amount of data sent in qnames from the resolver) is NOT forbidden by the [RFC1034] (section 5.3.3) or [RFC1035] (section 7.2). Sending the full qname to the authoritative name server is a tradition, not a protocol requirement. This tradition comes[mockapetris-history] from a desire to optimize the number of requests, when the same name server is authoritative for many zones in a given name (something which was more common in the old days, where the same name servers served .com and the root) or when the same name server is both recursive and authoritative (something which is strongly discouraged now). Whatever the merits of this choice at this time, the DNS is quite different now.

It may be noticed that many documents explaining the DNS and intended for a wide audience, incorrectly describe the resolution process as using qname minimisation, for instance by showing a request going to the root, with just the TLD in the query. As a result, these documents may confuse the privacy analysis of the users who see them.

As mentioned before, there are several ways to implement qname minimisation. Two main strategies are the aggressive one and the lazy one. In the aggressive one, the resolver only sends NS queries as long as it does not know the zone cuts. This is the safest, from a privacy point of view. The lazy way "piggybacks" on the traditional resolution code. It sends traditional full qnames and learns the zone cuts from the referrals received, then switches to NS queries asking only for the minimum domain name. This leaks more data but probably requires fewer changes in the existing resolver codebase.

3. Operational considerations

The administrators of the forwarders, and of the authoritative name servers, will get less data, which will reduce the utility of the statistics they can produce (such as the percentage of the various qtypes). On the other hand, it may decrease their legal

responsibility in some jurisdictions. (TODO: do we keep any mention of legal issues? We're not lawyers.)

Some broken name servers do not react properly to qtype=NS requests. For instance, some authoritative name servers embedded in load balancers reply properly to A queries but send REFUSED to NS queries. REMOVE THIS SENTENCE BEFORE PUBLICATION: As an example of today, look at www.ratp.fr (not ratp.fr). This behaviour is a gross protocol violation, and there is no need to stop improving the DNS because of such brokenness. However, qname minimisation may still work with such domains since they are only leaf domains (no need to send them NS requests). Such setup breaks more than just qname minimisation. It breaks negative answers, since the servers don't return the correct SOA, and it also breaks anything dependent upon NS and SOA records existing at the top of the zone.

A problem can also appear when a name server does not react properly to ENT (Empty Non-Terminals). If ent.example.com has no resource records but foobar.ent.example.com does, then ent.example.com is an ENT. A query, whatever the qtype, for ent.example.com must return NODATA (NOERROR / ANSWER: 0). However, some broken name servers return NXDOMAIN for ENTs. REMOVE THIS SENTENCE BEFORE PUBLICATION: As an example of today, look at com.akadns.net or www.upenn.edu with its delegations to Akamai. If a resolver queries only foobar.ent.example.com, everything will be OK but, if it implements qname minimisation, it may query ent.example.com and get a NXDOMAIN. See also section 3 of [I-D.vixie-dnsext-resimprove] for the other bad consequences of this brokenness.

Another way to deal with such broken name servers would be to try with A requests (A being chosen because it is the most common and hence a qtype which will be always accepted, while a qtype NS may ruffle the feathers of some middleboxes). Instead of querying name servers with a query "NS example.com", we could use "A _.example.com" and see if we get a referral.

Other strange and illegal practices may pose a problem: there is a common DNS anti-pattern used by low-end web hosters that also do DNS hosting that exploits the fact that the DNS protocol (pre-DNSSEC) allows certain serious misconfigurations, such as parent and child zones disagreeing on the location of a zone cut. Basically, they have a single zone with wildcards for each TLD like:

```
*.example.          60  IN  A   192.0.2.6
```

(It is not known why they don't just wildcard all of "*." and be done with it.)

This lets them turn up many web hosting customers without having to configure thousands of individual zones on their nameservers. They just tell the prospective customer to point their NS records at the hoster's nameservers, and the Web hoster doesn't have to provision anything in order to make the customer's domain resolve. NS queries to the hoster will therefore do not give the right result, which may endanger qname minimisation (it will be a problem for DNSSEC, too).

Qname minimisation can decrease performance in some cases, for instance for a deep domain name (like www.host.group.department.example.com where host.group.department.example.com is hosted on example.com's name servers). For such a name, a cold resolver will, depending how qname minimisation is implemented, send more queries. Once the cache is warm, there will be no difference with a traditional resolver. A possible solution is to always use the traditional algorithm when the cache is cold and then to move to qname minimisation. This will decrease the privacy a bit but will guarantee no degradation of performance.

Another useful optimisation may be, in the spirit of the HAMMER idea [I-D.wkumari-dnsop-hammer] to probe in advance for the introduction of zone cuts where none previously existed (i.e. confirm their continued absence, or discover them.)

4. Performance implications

The main goal of qname minimisation is to improve privacy by sending less data. However, it may have other advantages. For instance, if a root name server receives a query from some resolver for A.CORP followed by B.CORP followed by C.CORP, the result will be three NXDOMAINs, since .CORP does not exist in the root zone. Under query name minimisation, the root name servers would hear only one question (for .CORP itself) to which they could answer NXDOMAIN, thus opening up a negative caching opportunity in which the full resolver could know a priori that neither B.CORP or C.CORP could exist. Thus in this common case the total number of upstream queries under qname minimisation would be counter-intuitively inferior to the number of queries under the traditional iteration (as described in the DNS standard).

Qname minimisation may also improve look-up performance for TLD operators. For a typical TLD, delegation-only, and with delegations just under the TLD, a 2-label QNAME query is optimal for finding the delegation owner name.

5. Security considerations

Qname minimisation's benefits are clear in the case where you want to decrease exposure to the authoritative name server. But minimising the amount of data sent also, in part, addresses the case of a wire sniffer as well the case of privacy invasion by the servers. (Encryption is of course a better defense against wire sniffers but, unlike qname minimisation, it changes the protocol and cannot be deployed unilaterally.)

Qname minimisation offers zero protection against the recursive resolver, which still sees the full request coming from the stub resolver.

At this stage, this document does not recommend one of the two qname minimisation approaches (aggressive or lazy) against the other.

No security consequence (besides privacy improvment) is known at this time.

6. Implementation status - 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, no production resolver implements qname minimisation. For Unbound, see ticket 648 [1].

The algorithm to find the zone cuts described in Appendix A is implemented with qname minimisation in the sample code zonecut.go

[2]. It is also implemented, for a much longer time, in an option of dig, "dig +trace", but without qname minimisation.

7. Acknowledgments

Thanks to Olaf Kolkman for the original idea although the concept is probably much older [3]. Thanks to Mark Andrews and Francis Dupont for the interesting discussions. Thanks to Brian Dickson, Warren Kumari, Evan Hunt and David Conrad for remarks and suggestions. Thanks to Mohsen Souissi for proofreading. Thanks to Tony Finch for the zone cut algorithm in Appendix A. Thanks to Paul Vixie for pointing out that there are practical advantages (besides privacy) to qname minimisation. Thanks to Phillip Hallam-Baker for the fallback on A queries, to deal with broken servers. Thanks to Robert Edmonds for an interesting anti-pattern.

8. References

8.1. 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.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, July 2013.

[I-D.ietf-dprive-problem-statement] Bortzmeyer, S., "DNS privacy considerations", draft-ietf-dprive-problem-statement-01 (work in progress), January 2015.

8.2. Informative References

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

[RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", RFC 6982, July 2013.

[I-D.wkumari-dnsop-hammer]
Kumari, W., Arends, R., Woolf, S., and D. Migault, "Highly Automated Method for Maintaining Expiring Records", draft-wkumari-dnsop-hammer-01 (work in progress), July 2014.

[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.

[dnsop] IETF, , "The DNSOP working group of IETF", March 2014, <https://datatracker.ietf.org/wg/dnsop/charter/>.

[mockapetris-history]
Mockapetris, P., "Private discussion", January 2015.

[kaliski-minimum]
Kaliski, B., "Minimum Disclosure: What Information Does a Name Server Need to Do Its Job?", March 2015, <http://blogs.verisigninc.com/blog/entry/minimum_disclosure_what_information_does>.

8.3. URIs

[1] https://www.nlnetlabs.nl/bugs-script/show_bug.cgi?id=648

[2] https://github.com/bortzmeyer/my-IETF-work/blob/master/draft-ietf-dnsop-qname-minimisation/zonecut.go

[3] https://lists.dns-oarc.net/pipermail/dns-operations/2010-February/005003.html

Appendix A. An algorithm to find the zone cut

Although a validating resolver already has the logic to find the zone cut, other resolvers may be interested by this algorithm to follow in order to locate this cut:

(0) If the query can be answered from the cache, do so, otherwise iterate as follows:

(1) Find closest enclosing NS RRset in your cache. The owner of this NS RRset will be a suffix of the QNAME - the longest suffix of any NS RRset in the cache. Call this PARENT.

(2) Initialize CHILD to the same as PARENT.

(3) If CHILD is the same as the QNAME, resolve the original query using PARENT's name servers, and finish.

(4) Otherwise, add a label from the QNAME to the start of CHILD.

(5) If you have a negative cache entry for the NS RRset at CHILD, go back to step 3.

(6) Query for CHILD IN NS using PARENT's name servers. The response can be:

(6a) A referral. Cache the NS RRset from the authority section and go back to step 1.

(6b) An authoritative answer. Cache the NS RRset from the answer section and go back to step 1.

(6c) An NXDOMAIN answer. Return an NXDOMAIN answer in response to the original query and stop.

(6d) A NOERROR/NODATA answer. Cache this negative answer and go back to step 3.

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/