                DNS Query Name Minimisation to Improve Privacy

Abstract

   This document describes a technique to improve DNS privacy, a
   technique called "QNAME minimisation", where the DNS resolver no
   longer sends the full original QNAME to the upstream name server.

Status of This Memo

   This document is not an Internet Standards Track specification; it is
   published for examination, experimental implementation, and
   evaluation.

   This document defines an Experimental Protocol for the Internet
   community. This document is a product of the Internet Engineering
   Task Force (IETF). It represents the consensus of the IETF
   community. It has received public review and has been approved for
   publication by the Internet Engineering Steering Group (IESG). Not
   all documents approved by the IESG are a candidate for any level of
   Internet Standard; see Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc7816.

Table of Contents

1. Introduction and Background

   The problem statement is described in [RFC7626]. 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.

   QNAME minimisation follows the principle explained in Section 6.1 of
   [RFC6973]: the less data you send out, the fewer privacy problems
   you have.

   Currently, 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 the
   full QNAME to the authoritative name server is a tradition, not a
   protocol requirement. In a conversation with the author in
   January 2015, Paul Mockapetris explained that this tradition comes
   from a desire to optimise the number of requests, when the same
   name server is authoritative for many zones in a given name
   (something that 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 that is strongly
   discouraged now). Whatever the merits of this choice at this time,
   the DNS is quite different now.

2. QNAME Minimisation

   The idea is to minimise the amount of data sent from the DNS resolver
   to the authoritative name server. In the example in the previous
   section, sending "What are the NS records for .com?" would have been
   sufficient (since it will be the answer from the root anyway). The
   rest of this section describes the recommended way to do QNAME
   minimisation -- the way that maximises privacy benefits (other
   alternatives are discussed in the appendices).

   Instead of sending the full QNAME and the original QTYPE upstream, a
   resolver that implements QNAME minimisation and does not already have
   the answer in its cache sends a request to the name server
   authoritative for the closest known ancestor of the original QNAME.
   The request is done with:

   o the QTYPE NS

   o the QNAME that is the original QNAME, stripped to just one label
      more than the zone for which the server is authoritative

   For example, a resolver receives a request to resolve
   foo.bar.baz.example. Let's assume that it already knows that
   ns1.nic.example is authoritative for .example and the resolver does
   not know a more specific authoritative name server. It will send the
   query QTYPE=NS,QNAME=baz.example to ns1.nic.example.

   The minimising resolver works perfectly when it knows the zone cut
   (zone cuts are described in Section 6 of [RFC2181]). But 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 that 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 where the zone cut
   will be. To find the zone cut, it will query the .example
   name servers for the NS records for bar.example. It will get a
   NODATA response, indicating that there is no zone cut at that point,
   so it has to query the .example name servers again with one more
   label, and so on. (Appendix A describes this algorithm in deeper
   detail.)

   Since the information about the zone cuts will be stored in the
   resolver's cache, the performance cost is probably reasonable.
   Section 6 discusses this performance discrepancy further.

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

3. Possible Issues

QNAME minimisation is legal, since the original DNS RFCs do not
mandate sending the full QNAME. So, in theory, it should work
without any problems. However, in practice, some problems may occur
(see [Huque-QNAME-Min] for an analysis and [Huque-QNAME-storify] for
an interesting discussion on this topic).

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.
This behaviour is a protocol violation, and there is no need to stop
improving the DNS because of such behaviour. However, QNAME
minimisation may still work with such domains, since they are only
leaf domains (no need to send them NS requests). Such a 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.

Another way to deal with such incorrect name servers would be to try
with QTYPE=A requests (A being chosen because it is the most common
and hence a QTYPE that will always be 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.

A problem can also appear when a name server does not react properly
to ENTs (Empty Non-Terminals). If ent.example.com has no resource
records but foobar.ent.example.com does, then ent.example.com is an
ENT. Whatever the QTYPE, a query for ent.example.com must return
NODATA (NOERROR / ANSWER: 0). However, some name servers incorrectly
return NXDOMAIN for ENTs. 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 an NXDOMAIN.
See also Section 3 of [DNS-Res-Improve] for the other bad
consequences of this bad behaviour.

A possible solution, currently implemented in Knot, is to retry with
the full query when you receive an NXDOMAIN. It works, but it is not
ideal for privacy.

Other practices that do not conform to the DNS protocol standards may
pose a problem: there is a common DNS trick used by some 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

(They could just wildcard all of "*.", which would be sufficient. We
don't know why they don't do it.)

This lets them have many web-hosting customers without having to
configure thousands of individual zones on their name servers. They
just tell the prospective customer to point their NS records at the
hoster's name servers, 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 not give the right result, which may
endanger QNAME minimisation (it will be a problem for DNSSEC, too).

4. Protocol and Compatibility Discussion

   QNAME minimisation 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 it is a
   unilateral change, resolver implementers may do QNAME minimisation in
   slightly different ways; see the appendices for examples.)

   One should note that the behaviour suggested here (minimising the
   amount of data sent in QNAMEs from the resolver) is NOT forbidden by
   Section 5.3.3 of [RFC1034] or Section 7.2 of [RFC1035]. As stated in
   Section 1, the current method, sending the full QNAME, is not
   mandated by the DNS protocol.

   One may notice that many documents that explain the DNS and that are
   intended for a wide audience incorrectly describe the resolution
   process as using QNAME minimisation (e.g., by showing a request going
   to the root, with just the TLD in the query). As a result, these
   documents may confuse readers that use them for privacy analysis.

5. 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) [Kaliski-Minimum].

   DNS administrators are reminded that the data on DNS requests that
   they store may have legal consequences, depending on your
   jurisdiction (check with your local lawyer).

6. Performance Considerations

   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.example
   followed by B.example followed by C.example, the result will be three
   NXDOMAINs, since .example does not exist in the root zone. Under
   query name minimisation, the root name servers would hear only one
   question (for .example 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.example nor C.example
   could exist. Thus, in this common case the total number of upstream
   queries under QNAME minimisation would be counterintuitively less
   than the number of queries under the traditional iteration (as
   described in the DNS standard).

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

   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). Let's assume a resolver that knows only the
   name servers of .example. Without QNAME minimisation, it would send
   these .example name servers a query for
   www.host.group.department.example.com and immediately get a specific
   referral or an answer, without the need for more queries to probe for
   the zone cut. For such a name, a cold resolver with QNAME
   minimisation will, depending on how QNAME minimisation is
   implemented, send more queries, one per label. Once the cache is
   warm, there will be no difference with a traditional resolver.
   Actual testing is described in [Huque-QNAME-Min]. Such deep domains
   are especially common under ip6.arpa.

7. On the Experimentation

   This document has status "Experimental". Since the beginning of time
   (or DNS), the fully qualified host name was always sent to the
   authoritative name servers. There was a concern that changing this
   behaviour may engage the Law of Unintended Consequences -- hence this
   status.

   The idea behind the experiment is to observe QNAME minimisation in
   action with multiple resolvers, various authoritative name servers,
   etc.

8. 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 as 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. Also, the effect of QNAME minimisation on
   wire sniffers depends on whether the sniffer is on the DNS path.)

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

   All the alternatives mentioned in Appendix B decrease privacy in the
   hope of improving performance. They must not be used if you want
   maximum privacy.

9. References

9.1. 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>.

   [RFC1035] Mockapetris, P., "Domain names - implementation and
              specification", STD 13, RFC 1035, DOI 10.17487/RFC1035,
              November 1987, <http://www.rfc-editor.org/info/rfc1035>.

   [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J.,
              Morris, J., Hansen, M., and R. Smith, "Privacy
              Considerations for Internet Protocols", RFC 6973,
              DOI 10.17487/RFC6973, July 2013,
              <http://www.rfc-editor.org/info/rfc6973>.

   [RFC7626] Bortzmeyer, S., "DNS Privacy Considerations", RFC 7626,
              DOI 10.17487/RFC7626, August 2015,
              <http://www.rfc-editor.org/info/rfc7626>.

9.2. Informative References

   [DNS-Res-Improve]
             Vixie, P., Joffe, R., and F. Neves, "Improvements to DNS
             Resolvers for Resiliency, Robustness, and Responsiveness",
             Work in Progress, draft-vixie-dnsext-resimprove-00,
             June 2010.

   [HAMMER]   Kumari, W., Arends, R., Woolf, S., and D. Migault, "Highly
             Automated Method for Maintaining Expiring Records", Work
             in Progress, draft-wkumari-dnsop-hammer-01, July 2014.

   [Huque-QNAME-Min]
             Huque, S., "Query name minimization and authoritative
             server behavior", May 2015,
             <https://indico.dns-oarc.net/event/21/contribution/9>.

   [Huque-QNAME-storify]
             Huque, S., "Qname Minimization @ DNS-OARC", May 2015,
             <https://storify.com/shuque/qname-minimization-dns-oarc>.

   [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>.

   [RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS
             Specification", RFC 2181, DOI 10.17487/RFC2181, July 1997,
             <http://www.rfc-editor.org/info/rfc2181>.

Appendix A. An Algorithm to Perform QNAME Minimisation

   This algorithm performs name resolution with QNAME minimisation in
   the presence of zone cuts that are not yet known.

   Although a validating resolver already has the logic to find the
   zone cuts, implementers of other resolvers may want to use this
   algorithm to locate the cuts. This is just a possible aid for
   implementers; it is not intended to be normative:

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

   (1) Find the 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 ANCESTOR.

   (2) Initialise CHILD to the same as ANCESTOR.

   (3) If CHILD is the same as the QNAME, resolve the original query
       using ANCESTOR'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 ANCESTOR'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.

Appendix B. Alternatives

   Remember that QNAME minimisation is unilateral, so a resolver is not
   forced to implement it exactly as described here.

   There are several ways to perform QNAME minimisation. See Section 2
   for the suggested way. It can be called the aggressive algorithm,
   since 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.
   Another possible algorithm, not fully studied at this time, could be
   to "piggyback" on the traditional resolution code. At startup, 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 could require fewer
   changes in the existing resolver codebase.

   In the above specification, the original QTYPE is replaced by NS (or
   may be A, if too many servers react incorrectly to NS requests); this
   is the best approach to preserve privacy. But this erases
   information about the relative use of the various QTYPEs, which may
   be interesting for researchers (for instance, if they try to follow
   IPv6 deployment by counting the percentage of AAAA vs. A queries). A
   variant of QNAME minimisation would be to keep the original QTYPE.

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

   To address the "number of queries" issue described in Section 6, a
   possible solution is to always use the traditional algorithm when the
   cache is cold and then to move to QNAME minimisation (precisely
   defining what is "hot" or "cold" is left to the implementer). This
   will decrease the privacy but will guarantee no degradation of
   performance.

Acknowledgments

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/