

**DNS query name minimisation to improve privacy
draft-ietf-dnsop-qname-minimisation-07**

Abstract

This document describes one of the techniques that could be used 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 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 April 14, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction and background	2
2.	QNAME minimisation	3
3.	Possible issues	4
4.	Protocol and compatibility discussion	5
5.	Operational considerations	5
6.	Performance considerations	5
7.	IANA Considerations	6
8.	Security Considerations	6
9.	Implementation status - RFC EDITOR: REMOVE BEFORE PUBLICATION	7
10.	Acknowledgments	7
11.	References	8
11.1.	Normative References	8
11.2.	Informative References	8
11.3.	URIs	9
Appendix A.	An algorithm to find the zone cut	9
Appendix B.	Alternatives	10
Author's Address	11

[1.](#) Introduction and background

The problem statement is exposed 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.

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

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 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.

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 one which maximises privacy benefits (other alternatives are discussed in appendixes).

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

the QTYPE NS,

the QNAME which 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 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 [[RFC2181](#)] ([section 6](#)). 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 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 it out, it will query the .example name servers for the NS records for bar.example. It will get a NODATA response, indicating 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 details.)

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 just above).

3. Possible issues

QNAME minimisation is legal, since the original DNS RFC do not mandate sending the full QNAME. So, in theory, it should work without any problems. However, in practice, some problems may occur (see an analysis in [[huque-qnamemin](#)] and an interesting discussion in [[huque-qnamestorify](#)]).

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 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.

Another way to deal with such broken name servers would be to try with QTYPE=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.

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. 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.

Other strange and non-conformant 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).

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 an 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 the [RFC1034] ([section 5.3.3](#)) or [RFC1035] ([section 7.2](#)). As said in [Section 1](#), the current method, sending the full QNAME, is not mandated by the DNS protocol.

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.

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 or C.example could exist. Thus in this common case the total number of upstream queries under QNAME minimisation would be counter-intuitively less than 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.

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 which knows only the name servers of `.example`. Without QNAME minimisation, it would send these `.example` nameservers 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 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-qnamemin](#)]. Such deep domains are specially common under `ip6.arpa`.

7. IANA Considerations

This document has no actions for IANA.

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 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 depend 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 performances. They must not be used if you want the maximum privacy.

9. 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, no production resolver implements QNAME minimisation but it has been publically announced for the future Knot DNS resolver [\[1\]](#). For Unbound, see ticket 648 [\[2\]](#) and for PowerDNS [<https://github.com/PowerDNS/pdns/issues/2311>](https://github.com/PowerDNS/pdns/issues/2311).

The algorithm to find the zone cuts described in [Appendix A](#) is implemented with QNAME minimisation in the sample code `zonecut.go` [\[3\]](#). It is also implemented, for a much longer time, in an option of `dig`, "`dig +trace`", but without QNAME minimisation.

Another implementation was done by Shumon Huque for testing, and is described in [\[huque-qnamemin\]](#).

10. Acknowledgments

Thanks to Olaf Kolkman for the original idea during a KLM flight from Amsterdam to Vancouver, although the concept is probably much older [\[4\]](#). Thanks for Shumon Huque and Marek Vavrusa for implementation and testing. 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](#) and for discussion of the algorithm. 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.

[11.](#) References

[11.1.](#) 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>>.

[11.2.](#) Informative References

- [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>>.
- [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.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-dnsexst-resimprove]

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

[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>.

[huque-qnamemin]

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

[huque-qnamestorify]

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

11.3. URIs

- [1] <https://ripe70.ripe.net/presentations/121-knot-resolver-ripe70.pdf>
- [2] https://www.nlnetlabs.nl/bugs-script/show_bug.cgi?id=648
- [3] <https://github.com/bortzmeyer/my-IETF-work/blob/master/draft-ietf-dnsop-QNAME-minimisation/zonecut.go>
- [4] <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.

[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. The one in [Section 2](#) is the suggested one. 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), which 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 [[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.)

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. This will decrease the privacy but will guarantee no degradation of performance.

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/>

