

Network Working Group
Internet-Draft
Obsoletes: [7816](#) (if approved)
Intended status: Standards Track
Expires: September 25, 2019

S. Bortzmeyer
AFNIC
P. Hoffman
ICANN
March 24, 2019

DNS Query Name Minimisation to Improve Privacy
draft-ietf-dnsop-rfc7816bis-02

Abstract

This document describes techniques called "QNAME minimisation" to improve DNS privacy, where the DNS resolver no longer always sends the full original QNAME to the upstream name server. This document obsoletes [RFC 7816](#).

This document is part of the IETF DNSOP (DNS Operations) Working Group. The source of the document, as well as a list of open issues, is at <https://framagit.org/bortzmeyer/rfc7816-bis>

NOTE FOR THE DNSOP WORKING GROUP: There is still much work to be done in this draft. Future versions of this draft will contain descriptions of different minimisation implementation choices that have been made since the [RFC 7816](#) first came out, as well as deployment experience.

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 <https://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 25, 2019.

Copyright Notice

Copyright (c) 2019 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 (<https://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
1.1.	Terminology	3
2.	General Description of QNAME Minimisation	3
2.1.	Algorithm to Perform Aggressive Method QNAME Minimisation	5
3.	Operational Considerations	6
4.	Performance Considerations	8
5.	Alternative Methods for QNAME Minimisation	9
6.	Results of the Experimentation	9
7.	Security Considerations	9
8.	Implementation Status	10
9.	References	11
9.1.	Normative References	11
9.2.	Informative References	11
	Acknowledgments	13
	Changes from RFC 7816	13
	Authors' Addresses	13

[1.](#) Introduction and Background

The problem statement for this document and its predecessor [[RFC7816](#)] is described in [[I-D.bortzmeyer-dprive-rfc7626-bis](#)]. The terminology ("QNAME", "resolver", etc.) is defined in [[I-D.ietf-dnsop-terminology-bis](#)]. 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.

Before QNAME minimisation, when a resolver received the query "What is the AAAA record for www.example.com?", it sent to the root (assuming a resolver whose cache is empty) the very same question. Sending the full QNAME to the authoritative name server was 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.

QNAME minimisation is compatible with the current DNS system and therefore can easily be deployed. Because it is only a change to the way that the resolver operates, it does not change the protocol. The behaviour suggested here (minimising the amount of data sent in QNAMEs from the resolver) is allowed by [Section 5.3.3 of \[RFC1034\]](#) or [Section 7.2 of \[RFC1035\]](#).

1.1. Terminology

A "cold" cache is one that is empty, having literally no entries in it. A "warm" cache is one that has some entries in it.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP 14 \[RFC2119\]](#) [\[RFC8174\]](#) when, and only when, they appear in all capitals, as shown here.

2. General Description of QNAME Minimisation

The idea behind QNAME minimisation is to minimise the amount of data sent from the DNS resolver to the authoritative name server. This section describes the RECOMMENDED way to do QNAME minimisation -- the way that maximises privacy benefits. That algorithm is summarized in [Section 2.1](#).

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

This method is called the "aggressive method" in this document because the resolver only sends NS queries until it knows the nameserver responsible for the desired name. This method is the safest from a privacy point of view, and is thus the RECOMMENDED method for this document. Other methods are described in [Section 5](#).

For example, a resolver receives a request to resolve `foo.bar.baz.example`. Assume that the resolver already knows that `ns1.nic.example` is authoritative for `.example`, and that 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. In 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. ([Section 2.1](#) describes this algorithm in deeper detail.)

Here are more detailed examples of queries with the aggressive method of QNAME minimisation:

Cold cache, aggressive method, request for `www.isc.org`:

QTYPE	QNAME	TARGET	NOTE
NS	org	root nameserver	
NS	isc.org	.org nameserver	
NS	www.isc.org	isc.org nameserver	"www" may be delegated
A	www.isc.org	isc.org nameserver	

Cold cache, lazy algorithm (for a cold cache, it is the same algorithm as now), request for `www.isc.org`:

QTYPE	QNAME	TARGET	NOTE
A	www.isc.org	root nameserver	
A	www.isc.org	.org nameserver	
A	www.isc.org	isc.org nameserver	

Warm cache (all NS RRsets are known), both algorithms, request for `www.isc.org`:

QTYPE	QNAME	TARGET	NOTE
A	<code>www.isc.org</code>	<code>isc.org nameserver</code>	

Warm cache with only `isc.org`, (`example.org`'s NS RRset is not known), aggressive method, request for `www.example.org`:

QTYPE	QNAME	TARGET	NOTE
NS	<code>example.org</code>	<code>.org nameserver</code>	
NS	<code>www.example.org</code>	<code>.example nameserver</code>	
A	<code>www.example.org</code>	<code>.example nameserver</code>	

Since the information about the zone cuts will be stored in the resolver's cache, the performance overhead for using the aggressive method is probably reasonable. [Section 4](#) discusses this performance discrepancy further.

Note that DNSSEC-validating resolvers already have access to the zone cut information because the DNSKEY record set is just below a zone cut and the DS record set is just above it.

2.1. Algorithm to Perform Aggressive Method QNAME Minimisation

This algorithm performs name resolution with aggressive method 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 zone cuts.

- (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.

3. Operational Considerations

TODO may be remove the whole section now that it is no longer experimental?

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-Discuss](#)] for an interesting discussion on this topic).

Note that the aggressive method described in this document prevents authoritative servers other than the server for a full name from seeing information about the relative use of the various QTYPES. That information may be interesting for researchers (for instance, if they try to follow IPv6 deployment by counting the percentage of AAAA vs. A queries).

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", a resolver could use "A __.example.com" and see if it gets a referral. TODO this is what Unbound does

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 foo.bar.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 foo.bar.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 or Unbound, 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 hosts 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. It is impossible to tell 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).

TODO report by Akamai about why they return erroneous responses

<https://mailarchive.ietf.org/arch/msg/dnsop/XIX16DCe2ln3ZnZai723v32ZIjE>

TODO what to do if the resolver forwards? Unbound disables QNAME minimisation in that case, since the forwarder will see everything, anyway. What should a minimising resolver do when forwarding the request to a forwarder, not to an authoritative name server? Send

the full qname? Minimises? (But how since the resolver does not know the zone cut?)

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).

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).

4. 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 resolver sends a root name server queries 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. When using QNAME minimisation, the resolver would send 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 could be counterintuitively less than the number of queries under the traditional iteration (as described in the DNS standard). TODO mention [\[RFC8020\]](#)? And [\[RFC8198\]](#), the latter depending on DNSSEC?

QNAME minimisation may also improve lookup performance for TLD operators. For a TLD that is delegation-only, a two-label QNAME query may be optimal for finding the delegation owner name, depending on the way domain matching is implemented.

QNAME minimisation can decrease performance in some cases, most notably for domain names with many labels (like `www.host.group.department.example.com`, where `host.group.department.example.com` is hosted on `example.com`'s name servers). Assume a resolver that knows only the name servers of `example.com`. Without QNAME minimisation, it would send these `example.com` 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.

5. Alternative Methods for QNAME Minimisation

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

To reduce the number of queries (an issue described in [Section 4](#)), a resolver could always use full name queries when the cache is cold and then to move to the aggressive method of QNAME minimisation when the cache is warm. (Precisely defining what is "warm" or "cold" is left to the implementer). This will decrease the privacy for initial queries but will guarantee no degradation of performance.

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.

6. Results of the Experimentation

TODO various experiences from actual deployments, problems heard.
TODO the Knot bug #339 <https://gitlab.labs.nic.cz/knot/knot-resolver/issues/339?> TODO Problems with AWS <https://forums.aws.amazon.com/thread.jspa?threadID=269116?>

7. 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 [Section 5](#) decrease privacy in the hope of improving performance. They must not be used if you want maximum privacy.

8. Implementation Status

\[\[Note to RFC Editor: Remove this entire section, and the reference to [RFC 7942](#), before publication. \]\]

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 [\[RFC7942\]](#). 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 [\[RFC7942\]](#), "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".

Unbound has had a QNAME minimisation feature since version 1.5.7, December 2015, (see [\[Dolmans-Unbound\]](#)) and it has had QNAME minimisation turned default since version 1.7.2, June 2018. It has two modes set by the "qname-minimisation-strict" configuration option. In strict mode (option set to "yes"), there is no workaround for broken authoritative name servers. In lax mode, Unbound retries when there is a NXDOMAIN response from the minimized query. Since November 2016, Unbound uses only queries for the A RRtype and not the NS RRtype.

Knot Resolver has had a QNAME minimisation feature since version 1.0.0, May 2016, and it is activated by default.

BIND has had a QNAME minimisation feature since unstable development version 9.13.2, July 2018. It currently has several modes, with or without workarounds for broken authoritative name servers.

The Cloudflare's public resolver at IP address 1.1.1.1 has QNAME minimisation. (It currently uses Knot.)

Testing with one thousand RIPE Atlas probes [[atlas-qname-min](#)], one can see that QNAME minimisation is now common:

```
% blaeu-resolve --requested 1000 --type TXT qnamemintest.internet.nl
["no - qname minimisation is not enabled on your resolver :("] : 888
occurrences
["hooray - qname minimisation is enabled on your resolver :)!"] : 105
occurrences
[ERROR: SERVFAIL] : 3 occurrences
Test #16113243 done at 2018-09-14T13:01:47Z
```

10 % of the probes have a resolver with QNAME minimisation (it is not possible to infer the percentage of users having QNAME minimisation).

9. References

9.1. Normative References

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, [RFC 1034](#), DOI 10.17487/RFC1034, November 1987, <<https://www.rfc-editor.org/info/rfc1034>>.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, [RFC 1035](#), DOI 10.17487/RFC1035, November 1987, <<https://www.rfc-editor.org/info/rfc1035>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [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, <<https://www.rfc-editor.org/info/rfc6973>>.
- [RFC7816] Bortzmeyer, S., "DNS Query Name Minimisation to Improve Privacy", [RFC 7816](#), DOI 10.17487/RFC7816, March 2016, <<https://www.rfc-editor.org/info/rfc7816>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

9.2. Informative References

[atlas-qname-min]

Bortzmeyer, S., "DNS resolution of qnamemintest.internet.nl/TXT on RIPE Atlas probes", September 2018, <<https://atlas.ripe.net/measurements/16113243/>>.

[DNS-Res-Improve]

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

[Dolmans-Unbound]

Dolmans, R., "Unbound QNAME minimisation @ DNS-OARC", March 2016, <https://indico.dns-oarc.net/event/22/contributions/332/attachments/310/542/unbound_qnamemin_oarc24.pdf>.

[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-Discuss]

Huque, S., "Qname Minimization @ DNS-OARC", May 2015, <<https://www.huque.com/2015/05/16/qname-min.html>>.

[Huque-QNAME-Min]

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

[I-D.bortzmeyer-dprive-rfc7626-bis]

Bortzmeyer, S. and S. Dickinson, "DNS Privacy Considerations", [draft-bortzmeyer-dprive-rfc7626-bis-02](#) (work in progress), January 2019.

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

Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", [draft-ietf-dnsop-terminology-bis-14](#) (work in progress), September 2018.

[RFC2181]

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

- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", [BCP 205](#), [RFC 7942](#), DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.
- [RFC8020] Bortzmeyer, S. and S. Huque, "NXDOMAIN: There Really Is Nothing Underneath", [RFC 8020](#), DOI 10.17487/RFC8020, November 2016, <<https://www.rfc-editor.org/info/rfc8020>>.
- [RFC8198] Fujiwara, K., Kato, A., and W. Kumari, "Aggressive Use of DNSSEC-Validated Cache", [RFC 8198](#), DOI 10.17487/RFC8198, July 2017, <<https://www.rfc-editor.org/info/rfc8198>>.

Acknowledgments

TODO (refer to 7816)

Changes from [RFC 7816](#)

- o Made changes to deal with errata #4644
- o Changed status to be on standards track
- o Major reorganization

Authors' Addresses

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

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

Paul Hoffman
ICANN

Email: paul.hoffman@icann.org

