The ISP Column
A monthly column on things Internet

September 2020
Geoff Huston
Joao Damas

DNS Query Privacy Revisited

This article was first written in August 2019, and it ended with the comment: "It's likely that we will return to this measurement of the use of Qname minimisation in a year or so to see if anything has changed from the picture today." A year has passed and it's time to relook at this topic and see what has changed in the DNS resolution environment over the past 12 months.

Much has been said and written in recent times about the use of the DNS as a means of looking at the behaviour of end systems and inferring user behaviours. Almost every transaction starts with a DNS query, and if one were to assemble the complete set of DNS queries generated by an Internet user it would be possible to assemble a relatively complete picture of their online activity. For many years this aspect of the DNS as a means of observation into the activities of others received little attention from the mainstream, but the more recent sensitivities over state and private digital surveillance has brought significant attention to the overall topic of DNS privacy. Another reason for all this attention is that in terms of privacy DNS resolution protocol has been sadly lacking in some basic privacy protections. The DNS name resolution protocol was not designed with privacy as the foremost consideration. The queries and responses are unencrypted, which makes them prone to hostile man-in-the-middle manipulation and they leak superfluous information to third party onlookers.

There are two major approaches to try and mitigate the DNS privacy issues. The first approach is to make it harder to eavesdrop on DNS queries by using encryption for DNS transactions. The issues around encryption and the efforts with DNS-over-TLS (DoT) and DNS-over-HTTP (DoH) are a current topic of very high interest in the DNS world. The second approach is to reduce the information leakage by reducing the amount of information in each DNS query. The IETF published an approach to achieve this using a technique called "Query Name Minimisation" (*Qname Minimisation* or *Qmin*), described in an Experimental RFC document (RFC 7816, "DNS Query Name Minimisation to Improve Privacy" by Stephan Bortzmeyer, March 2016).

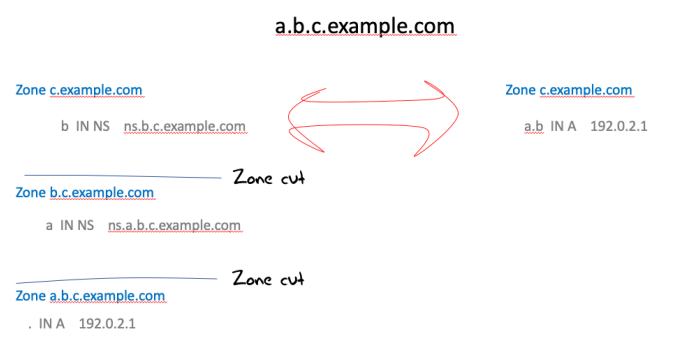
In this article we will re-look at Qname Minimisation in a little more detail and present some results of our measurement of the current level of use of this resolver query technique in today's Internet.

Query Name Minimisation

The technique described in RFC 7816 is query management approach based on a principle described in RFC 6973, "Privacy Considerations for Internet Protocols" (July 2013), which could be summarized as: the less data you emit the fewer privacy issues you are likely to encounter.

The DNS has conventionally optimised its behaviour for simplicity and performance. The underlying factor in the DNS name resolution protocol is that a DNS recursive resolver does not necessarily know in advance which servers are authoritative for a given zone, so it has to discover this information. Also, if a name that has a number of labels then the resolver does not necessarily know where the zone cuts occur between labels.

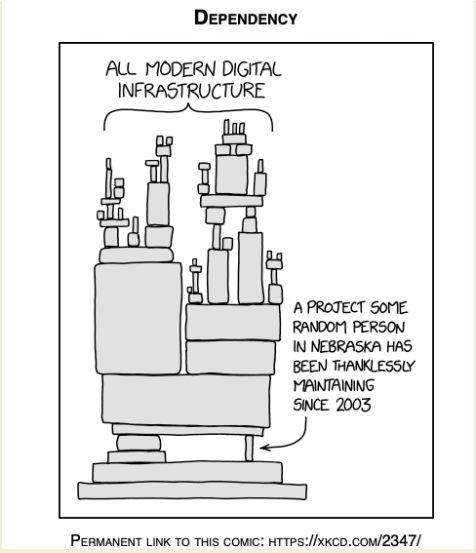
For example, the zone **`c.example.com`** may contain a delegation record for the label **`b`**. The zone **`b.c.example.com`** might contain a delegation record for the label **`a`**. The zone **`a.b.c.example.com`** might contain a resource record for this name. In this case there are *zone cuts* between **`b`** and **`c`** and between **`a`** and **`b`** in the domain name **`a.b.c.example.com`**. Alternatively, the zone **`c.example.com`** might contain a resource record for the label **`a.b`**, in which care there are no further *zone cuts* below **`c.example.com`**.



The point here is that a resolver cannot determine in advance just by examining the domain name where the *zone cuts* might be. They are discovered by the resolver during the name resolution process.

The *zone cuts* may relate to a desire on the part of a zone administrator to split the zone management role to different servers. They may also relate to a change in administrative control where the name space below the cut is delegated to a different administrative entity. The IETF chartered the dbound Working Group in 2014 to work on a way to make these points of delegation that relate to a change of administrative control explicit. The idea was to allow the DNS to self-describe these points of change of administrative control within the DNS name hierarchy itself. The effort was directed to replacing the volunteer-operated *Public Suffix List* (https://publicsuffix.org), which has some concerning shortcomings (https://www.icann.org/en/system/files/files/sac-070-en.pdf). The dbound Working Group worked on a problem statement for a couple of years (https://datatracker.ietf.org/doc/draft-sullivan-dbound-problem-statement/). The group was shut down in early 2017 without publishing any of its work as an RFC. Instead, we continue to rely on the volunteer efforts of the Mozilla Foundation to maintain the public suffix list, and we currently have no alternative in mind. Perhaps this reliance on volunteer effort may resurface as an issue in the coming months due to the current adverse economic conditions impacting Mozilla (https://bit.ly/32c6QKl).

This implicit reliance on the erstwhile work of volunteers was a feature of the early Internet, and perhaps one of its core strengths at the time. These days the Internet is a trillion-dollar sector and it seems a little uncomfortable to still find critical dependencies on volunteer effort when that effort underpins large amounts of the Internet's commercially operated service infrastructure.



In the absence of this meta-information about the structure of the namespace, a DNS recursive resolver uses the full query name in all queries as it descends the name hierarchy looking for the lowest level authoritative name server, as this iterative technique will expose the zone cuts and the name servers for each zone.

To expand on this a little, DNS resolution occurs in a 'top down' manner, and when an authoritative server for a zone receives a query for a name that is only resolvable in a delegated subordinate zone (i.e. at a level in the zone hierarchy that is lower in the name hierarchy than zone served by this authoritative server) it returns a NOERROR code and no *Answer Section* in its response (a "NODATA" response). The response includes the name of the next lower level delegated zone and its name servers, as enumerated in the delegation record (the point of the zone cut), in the *Authority Section* of the response and the IP addresses of these name servers in the *Additional Section* of the response, assuming that these addresses are known to the authoritative server.

The DNS is a strict hierarchal namespace, so each server is only aware of immediately delegated zones. The name resolution process will iterate down through the hierarchy to either reach the server that can provide an authoritative response for this query name or obtain a response indicating that the name does not exist in the DNS. This process is illustrated in Figure 1.

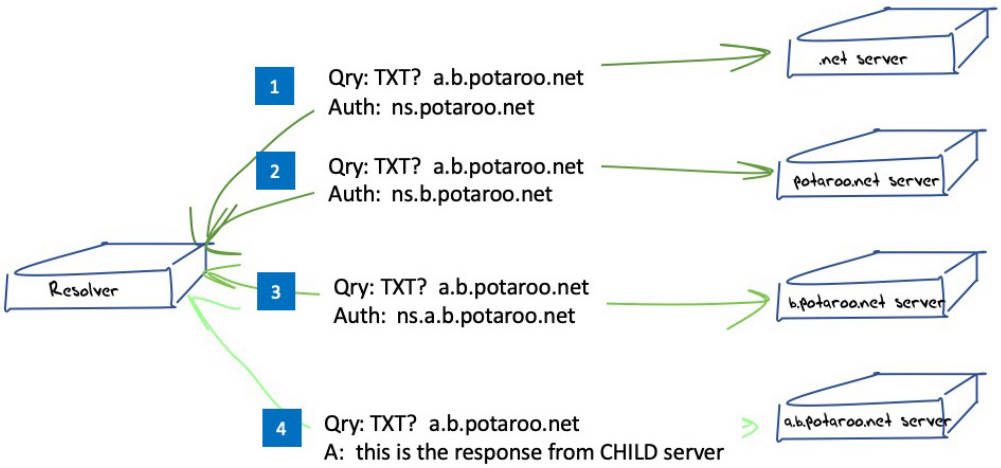


Figure 1 – DNS "discovery" process in Name Resolution

Of course, an efficient recursive resolver will use cached information whenever it can, so the process is typically nowhere near as exhaustive and slow as it may appear from this informal description.

This description is not exactly the case in all situations. A server may be an authoritative server for both a 'parent' zone and some or all of its delegated 'child' zone or zones. The query does not specify which zone is the intended subject of its query, allowing the server to answer the query using data from the most specific served zone in the name hierarchy that partially matches the query name.

How does Query Name Minimisation alter this behaviour? To quote from RFC 7816:

> 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

A resolver using Qname Minimisation implicitly assumes that each label in the query name corresponds to a zone cut. The resolver queries a parent zone server, using an abbreviated query name that is truncated after the name of the immediate child label, and uses a query type of NS. This altered resolution process is illustrated in Figure 2.

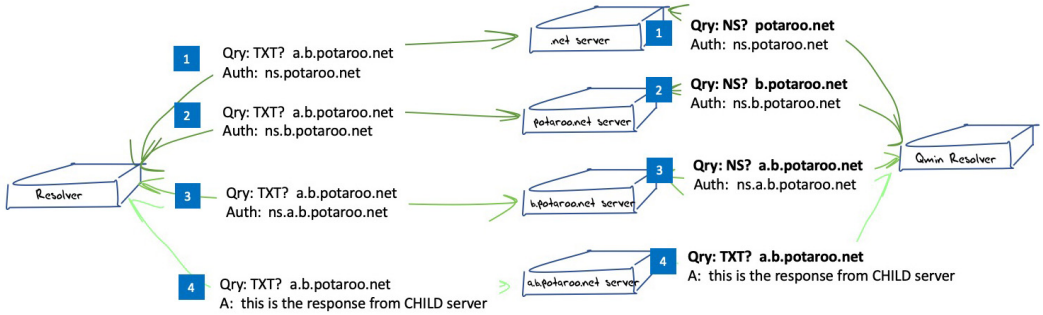


Figure 2 – DNS "discovery" process in Name Resolution using Query Name Minimisation

Let's look at the query sequence in two cases to illustrate the difference between full name queries and minimised name queries. In the case of a full name query for the name **myspecialname.me.example.com** the query name has been exposed to a **root** server, a **.com** server, an **example.com** server and a **me.example.com** server. If the query logs from any of these servers were to be inspected my "interest" in the name **myspecialname.me.example.com** would be evident. In the minimised case the information 'leak' has been trimmed considerably. The **root** server only sees a query for the .**com** label, the com servers only see a query for **example.com**, and so on.

In terms of an improvement to DNS privacy, this technique sounds like a great step forward. Nothing changes for authoritative servers and it's only the recursive resolvers that change their behaviour to trim the query name and alter the query type to a 'neutral' query for the NS record rather than expose the intended query type to these servers. Only at the target zone is the full query name used with the original query type. This approach essentially removes superfluous information from the DNS query stream. The approach can be deployed incrementally, and the benefits are immediately available to those recursive resolvers, and their user population, that use this Qname Minimisation technique.

In short, it seems like an ideal technology change, where current incumbent service providers need to do nothing to allow those who want to adopt this approach to proceed. The benefit for those who do this is that they cease to broadcast their actions and intent to a larger circle of potential DNS onlookers.

Query Name Minimisation Considerations

Why hasn't this technique been deployed in all resolvers already? Why isn't this the default mode of operation of the DNS? Assuming that the concerns relating to DNS privacy aren't just the products of

the fevered imagination of a few activists in the IETF, but a reflection of a larger set of very real user community concerns over obsessive levels of DNS surveillance, then surely there would be a clear consumer preference for services that use such techniques to improve DNS privacy. Why have vendors not identified this consumer preference and deployed product to meet this incipient demand? If all this is so simple and easy, and is deployable in a piecemeal and uncoordinated manner then what's stopping us for doing this?

The Qname Minimisation picture is nowhere as simple as you might think at this juncture. There are a number of DNS structures that need to be considered, and three such cases are considered here.

Empty Non-Terminal (ENT) zones

What if the query name does not exist?

The simple response is that whenever the Qname minimising resolver receives an NXDOMAIN response then it should stop and return NXDOMAIN as the response to the querier. NXDOMAIN is a very particular form of response indicating that this name does not exist in any form in the DNS, not even as a delegation point. NXDOMAIN means that this name, and any name that shares this name as a common suffix, does not exist.

Only in theory do theory and practice coincide. In practice they don't.

The practice of the DNS is filled with odd behaviours and stupid DNS tricks that tend to assume a particular mode of resolver behaviour. As Shumon Huque has pointed out in a recent OARC meeting (https://indico.dns-oarc.net/event/21/contributions/298/attachments/267/487/qname-min.pdf), some common Content Data Networks (CDNs) host content by using CNAME records to map a client's name into their CDN name space and then assume that subsequent queries into the CDN zone's name space always contain the full query name. Rather than assuming that every name needs to be "discoverable" as a top-down hierarchical search, they assume that their part of the DNS is an exact match lookup.

A common CDN hosting technique is to map a hosted content name into the content provider's managed name space through a CNAME DNS alias record.

For example, if the CDN provider uses the common DNS suffix such as **`hosted-service.cdn`** then the service name **`www.example.com`** would be mapped into the hosted service by placing a CNAME record for ***`www.example.com`***, aliasing this name to **`www.example.com.hosted-service.cdn`**.

The strict definition of a CNAME record is that the recursive resolver follows the CNAME record and re-commences name resolution for this alias name.

In this example, recursive resolver would then use the query name **`www.example.com.hosted-service.cdn`** to query the DNS. When the server for **`hosted-service.cdn`** is queried for this name it will then return the provider's hosting point for the client **`www.example.com`**.

The service provider is not hosting **`example.com`**, nor **`.com`**, so rather than synthesizing a delegation hierarchy that includes empty non-terminal zones for **`com.hosted-service.cdn`** and **`example.com.hosted-service.cdn`**, the service provider often uses a zone structure that emulates a flattened enumerated name space. In other words, the **`hosted-service.cdn`** zone server behaves in a manner that is consistent with have a zone file that has an entry for **`www.example.com.hosted-service.cdn`**. In this light, it is not inconsistent for the server to respond with NXDOMAIN for all name queries in **`hosted-service.cdn`** apart from precisely those names that are mapped to hosted content.

The result: If a partial form of these mapped names is passed to the CDN's authoritative server, then an NXDOMAIN may be generated by the server, which will confuse a Qmin recursive resolver.

These are instances of so-called "empty non-terminal" (ENT) zones, where the zone exists in the DNS hierarchy, but aside from a delegation record it has no other record. The expected response when an ENT is queried is NODATA (response code 0 (NOERROR) and an empty Answer Section). The NXDOMAIN is an overclaim in this case as NXDOMAIN is intended to be interpreted as "this name does not exist and there are no delegated names in the name hierarchy below this name."

As long as the recursive resolver used the full query name this anomalous use of NXDOMAIN does not have any visible impact. Qname Minimisation exposes this anomaly as it expects queries for all shortened name forms of a defined query name to return the names for the servers of the delegated zone.

NS vs A Query types

RFC 7816 points out some issues that have been encountered with DNS load distributors, where the response to a NS query is the somewhat unhelpful response code of REFUSED. The specification suggests that a possible workaround is to use an A Query Type with the minimised query name.

Don't forget that a Qname Minimising resolver asks the parent zone server about the child zone name, so this A Query Type is analogous to asking for the NS record, and the anticipated response to the A query type is a NODATA response with the details of the name servers of the child zone in the Authority Section. This is the same information to that provided if the NS Query Type was correctly handled. Don't forget that the parent zone is not authoritative for the child zone, so the NS query to the parent can only generate a NODATA response, rather than an authoritative answer.

If the only reasons to use NS queries is to mask the intended query type for intermediate queries, then it can be argued that an A Query Type is so common that in itself it gives out even less information than the NS Query Type. Our measurement show that this is the conclusion reached by resolver vendors and the predominate query type in Qname Minimising resolvers is for an A record, not a NS record.

DNS Zone Server Misconfiguration

As has been said many times the DNS is nowhere near as simple as it looks. Configuring authoritative servers for zones can be prone to all kinds of subtle errors. A server for a delegated zone does not necessarily know that it is a "properly" delegated server.

For example, a DNS server can be set up to serve the zone **`b.c.example.com`**, but it is not explicitly aware whether or not the server for **`c.example.com`** has listed this server as a delegated nameserver for the zone. The server will still answer all queries for names in **`b.c.example.com`** if it is asked. If the zone was DNSSEC-signed, then DNSSEC validation would expose any attempt pass off false data in this manner, but for unsigned domain names or non-validating resolvers, this can have unintended consequences.

Most of the time it's not a problem, as it is difficult for the DNS to discover this rogue server. A top-down conventional name server discovery process will use the parent zone delegation details to find the child zone's name servers, and so on. As the parent zone's delegation records do not point to the rogue server, the server will not be discovered in the normal course of events.

However, consider the case where a server is a duly delegated server for both the parent zone and is also an undelegated server for a child zone.

Continuing our example, if our server (an undelegated server for **`b.c.example.com`**) was also a duly delegated server for **`c.example.com`**, and this zone contained a delegation record for **`b.c.example.com`** that pointed to an entirely different server. When a recursive resolver passes a query to this server for the name **`a.b.c.example.com`** it does so because it has been told that this is an authoritative server for the zone **`c.example.com`**. However, the query does not contain any such information about intention, and the server will use the most specific served zone, in this case the undelegated **`b.c.example.com`** zone, to answer the query.

Qname Minimisation imposes a stricter regime on this situation. A Qname Minimising resolver will use the query name **`b.c.example.com`** when querying this server and will correctly follow the zone delegation directions to the duly delegated server for this zone.

An illustration of the difference between these two cases is shown in Figure 3.

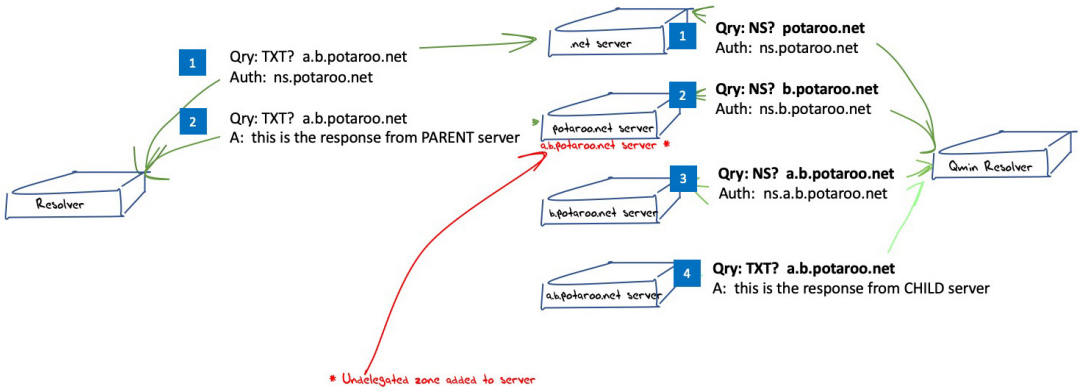


Figure 3 – DNS resolution anomalies between full name and minimised name queries

This form of DNS configuration, where a single server is configured to serve both zones and direct or indirect ancestors of these zones, is not uncommon in the DNS. As long as all servers of a zone are kept in sync with each other and serve the same information then this DNS server situation will be largely unnoticed. However, two tools will explicitly follow the full delegation path and will not 'short cut' across zone cuts, namely DNSSEC validation and Qname Minimisation.

Status of Recursive Resolvers and Qname Minimisation

There are a small set of recursive resolver implementation in use in the Internet today. This small set of DNS resolvers includes ISC's *Bind 9* (https://www.isc.org/bind/), NLnet Lab's *Unbound* (https://nlnetlabs.nl/projects/unbound/about/), CZ.nic's *Knot* (https://www.knot-resolver.cz) and the Power DNS *Recursor* (https://www.powerdns.com/recursor.html).

In *Bind 9*, Qname Minimisation is on by default since version 9.14.0. The configuration option is called `qname-minimization` and it can be set to `off`, `relaxed` and `strict`. The `off` setting disables qname minimisation completely, `strict` proceeds with qname minimisation as described by RFC 7816, and `relaxed` first tries Qname Minimisation, but falls back to regular resolution if it fails (presumably through the ENT issues described previously. The default setting is `relaxed`, although that may change in future releases of *Bind*.

In *Unbound* Qname Minimisation has been included since release 1.7.2. This setting is on by default. There are two directives: `qname-minimisation`: which is either `yes` or `no`, and `qname-minimisation-strict:` which determines fallback behaviour if the name fails to resolve. Strict mode `yes` turns off this fallback behaviour. The default in *Unbound* is not to use strict mode.

In *Knot* Qname Minimisation is enabled by default. In the struct `kr_qflags` the member `NO_MINIMIZE` can be turned on to disable this behaviour.

In *Power DNS Recusor* Qname Minimisation was implemented in 4.3.0-alpha1 and enabled by default since 4.3.0-beta1.

Measuring Qname Minimisation

Let's now turn to the measurement results. We want to understand the extent of deployment of Qname Minimisation in the DNS today, both as a count of the number of visible resolvers that ask authoritative servers and as a count of the proportion of users who send their queries to Qname Minimising resolvers.

As usual, when attempting to measure the DNS we need to take into consideration the conventional caching behaviour of resolvers, so in order to expose the queries being made by resolvers we use a pair of unique dynamically generated labels in the test scenario. The labels were served by DNS servers that are operated as part of the measurement experiment and the query logs were analysed to determine the extent to which resolvers were performing Qname minimisation.

We ran this test from the 6th February 2019 until the 24th July 2019. In that period the we saw 644,406 "visible" resolvers (recursive resolvers that query authoritative servers). Of this set of visible resolvers some 69,869 resolvers queried for the intermediate name form, indicating that they were performing some form of Qname minimisation.

Resolvers	**Qmin**	**Query Type**			
		NS	**A**	**AAAA**	
644,406	69,869	14,523	55,360	16	
	11%	2%	9%	0%	**% of all resolvers**
		21%	79%	0%	**% of Qmin resolvers**

Table 1a – Per Resolver QMin Counts: 2019

We reran the test from the 15th August 2020 until the 7th September. In this 24-day period we observed 240,287 unique IP addresses of visible DNS resolvers (Table 1b).

Resolvers	**Qmin**	**Query Type**			
		NS	**A**	**AAAA**	
240,287	27,969	1,032	26,935	0	
	11%	0%	11%	0%	**% of all resolvers**
		3%	96%	0%	**% of Qmin resolvers**

Table 1b – Per Resolver QMin Counts: 2020

In the shorter observation period in 2020 (Table 1b) we saw approximately one third the number of DNS resolvers, but the same relative proportion, namely 11% of these resolvers, used Query Minimisation. So not much has changed here. There is one significant change in the Query Type. In 2019 there was still some use of the NS query type, this has all but vanished in 2020, and the query type is now an A record.

This figure of 11% of all visible resolvers does not show to what extent Qname Minimisation is being used in today's DNS. For that we need to count relative use, and one way of doing this is to count the query load.

Queries	**NON Qmin**	**Qmin**	**Query Type**			
			NS	**A**	**AAAA**	
1,107,728,866	1,087,081,329	20,647,552	4,651,599	15,993,284	2,654	
	98%	2%	0%	1%	0%	**% of all queries**
			23%	77%	0%	**% of Qmin queries**

Table 2a – Query Counts: 2019

In 2019 Some 2% of all queries were using QMin, and of these queries some three quarters of these Qname Minimised queries used the A query type, not the NS type.

Queries	NON Qmin	Qmin	Query Type			
			NS	A	AAAA	
756,582,911	670,281,376	80,364,153	5,937,379	80364153	0	
	88%	11%	0%	10%	0%	**% of all queries**
			6%	93%	0%	**% of Qmin queries**

Table 2b – Query Counts: 2020

Table 2b shows a distinct change since 2019. Now some 11% of all seen queries are for the minimised name and the overall majority of such queries use the NS query type. There has been an appreciable increase in the use of Qname minimisation over the past 12 months.

We can break this down a little further, looking at the query patterns for each individual experiment. Table 3a shows the results from the 2019 measurement.

Experiments	Qmin	Query Type			
		NS	A	AAAA	
429,773,288	11,089,823	2,811,053	8,336,008	1,721	
	3%	1%	2%	0%	**% of all experiments**
		25%	75%	0%	**% of Qmin experiments**

Table 3a – Experiment Counts: 2019

The number of users that we observed using Qmin resolvers in 2019 quite small: some 3% of users send their queries through QMin resolvers. This is the measurement that has changed significantly over 12 months. We now see some 18% of users using resolvers that support Qname minimisation.

Experiments	Qmin	Query Type			
		NS	A	AAAA	
165,955,865	30,166,162	2,064,499	28,214,158	0	
	18%	1%	17%	0%	% of all experiments
		6%	93%	0%	% of Qmin experiments

Table 3b – Experiment Counts: 2020

Where are these users? Table 4a lists those economies where we collected more than 20,000 sample points over the duration of the 2019 measurement period, and where 10% or more of the users in these economies used a recursive resolver that performed Query name minimisation.

CC	Samples	Qmin %	Name
MG	105,216	73%	Madagascar
IQ	283,585	43%	Iraq
NP	278,585	43%	Nepal
NE	19,244	32%	Niger
BY	214,911	30%	Belarus
AO	268,288	29%	Angola
NZ	135,714	25%	New Zealand
PT	199,847	23%	Portugal
ZA	817,385	21%	South Africa
MM	23,940	14%	Myanmar
MY	349,914	12%	Malaysia
AM	23,083	12%	Armenia
UA	291,953	12%	Ukraine
IR	550,999	11%	Iran
CZ	115,284	10%	Czech Republic

Table 4a – Qname Minimisation Query rates per economy: 2019

What a curious collection of economies! It is unclear whether service providers in these economies have enabled Qname minimisation deliberately, or whether this is an outcome of using a recursive resolver such as the recent version of the Bind 9 resolver or the Knot resolver, where this functionality has been enabled by default.

The list has changed somewhat over the past 12 months, notably in India where more than half of the user population there passes their queries through Qname Minimising resolvers.

CC	Samples	Qmin Ratio	CName
CY	43,466	54%	Cyprus
IR	2,550,719	53%	Iran
NE	192,211	52%	Niger
IN	24,238,563	51%	India
BW	42,773	50%	Botswana
NP	293,496	49%	Nepal
MG	192,516	47%	Madagascar
AF	219,688	43%	Afghanistan
IQ	1,483,950	42%	Iraq
ZW	195,691	41%	Zimbabwe
DE	2,808,832	41%	Germany
GM	22,185	38%	Gambia
PT	323,984	36%	Portugal
GE	111,948	36%	Georgia
SI	70,623	36%	Slovenia

Table 4b – Qname Minimisation Query rates per economy: 2020

Two economies of interest are not listed in Table 4: China, which has seen a growth of 4% to 14% of users over the past 12 months and the United States where the growth is from under 1% to 5% of users.

Open DNS Resolvers

Just a little under one third of all users in the Internet today have Open DNS resolvers in the DNS resolver set that that use.

In terms of the set of Oopen DNS resolvers deployed in the Internet, Google's public DNS server does not appear to support Qname Minimisation (which is the most popularly used DNS Open resolver) Within the collection of the 10 most popularly used Open DNS resolver services, Cloudflare's 1.1.1.1 service, Quad9, and the OpenDNS service resolve their queries using Qname Minimisation.

Table 5 shows the current measurements of the use of Qname Minimisation for the major Open DNS resolvers in 2020.

Open DNS resolver	Qmin Ratio	Experiments	Qmin Experiments
Google DNS	0%	99,461,612	1,397
114 DNS	6%	16,010,343	887,461
Cloudflare	50%	9,202,176	4,634,222
Open DNS	69%	8,049,990	5,532,784
dnspai	5%	7,817,329	417,661
onedns	11%	6,017,193	644,946
Verisign	0%	1,122,718	0
Quad 9	70%	1,077,202	758,760
Level3	0%	913,193	0
Yandex	0%	569,829	11
Neustar	59%	563,892	331,570
dnswatch	57%	235,789	133,229
dyn	58%	157,767	92,279
cnnic	0%	152,308	0
greenteamdns	0%	86,057	51
he	97%	57,389	55,888

Table 5 – Qname Minimisation Query rates for Open DNS Resolvers: 2020

These measurements are interesting in that only one open DNS resolver (Hurricane Electric's Open DNS service) has a 97% Qname Minimisation ratio. The Open Resolvers services that record ratios of between 50% and 70% raise a question as to what is happening here? Are the individual resolver engines used by the service at different levels of support for Qname Minimisation? Or is some other DNS query pattern causing only some queries to be handled using Qname Minimisation?

> It is unclear to me whether Qname Minimisation in a very heavily used public DNS resolver provides any substantive beneficial privacy outcome for the users of this service. In many ways each user is "hiding in a crowd" and their individual queries are lost in the volume of queries being made by such recursive resolvers in the first place. It would also be expected that the open resolver's caches would be heavily populated so the full query name would be unlikely to be passed to the servers at the higher levels of the DNS name hierarchy in any case. Yes, the recursive resolver is privy to each user's DNS activity, but that is part of the direct consequences of using such a service in the first place and is unrelated to the Qname Minimisation aspect of the resolver's behaviour.
>
> The story changes completely when using a small volume DNS resolver, such as a resolver in a home network. The small client pool means that the resolver can be linked to end users, particularly if the resolver's clients share an IP address subnet with the resolver. A small volume recursive resolver may not have a continually refreshed local cache, so the full query names are more likely to be passed across to DNS servers at all levels in the DNS hierarchy.
>
> As ever, all privacy bets are off when Explicit Client Subnet (ECS) attributes are attached to the query! But the true horror of ECS is best left as a story for another article!

Qname Minimisation

Our measurements indicate that in mid-2020 some 18% of users pass their queries through resolvers that actively work to minimize the extent of leakage of superfluous information in DNS queries. This is a significant increase from the 3% of users seen some 12 months ago. Hopefully this will rise to upward of 90% in the coming 12 months!

Further Reading

Making the DNS More Private with QNAME Minimisation, Wouter de Vries, RIPE Labs Blog, April 2019. https://labs.ripe.net/Members/wouter_de_vries/make-dns-a-bit-more-private-with-qname-minimisation

Qname Minimisation and your Privacy, Vicky Risk, ISC. https://www.isc.org/blogs/qname-minimization-and-privacy/

DNS Privacy Project https://dnsprivacy.org

Disclaimer

The above views do not necessarily represent the views or positions of the Asia Pacific Network Information Centre.

Author

Geoff Huston B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region.

www.potaroo.net