January 2019 – Version 1.0

Networks & Trust

Introduction to DNS Privacy

Author
Fernando Gont

Internet Society

Abstract

Almost every time we use an Internet application, it starts with a Domain Name System (DNS) transaction to map a human-friendly domain name into a set of IP addresses that can be used to deliver packets over the Internet. DNS transactions can therefore be correlated to the applications we use, the web sites we visit, and sometimes even the people we communicate with.

While the information being transferred in DNS transactions is public, the information about set of transactions performed by each host on the Internet is not. Unfortunately, the DNS does not employ any mechanisms to provide confidentiality for DNS transactions, and the corresponding information can therefore easily be logged by the operators of DNS resolvers and nameservers, and eavesdropped by rogue entities and groups.

This document raises awareness of the privacy implications of the DNS and discusses a number of mechanisms that have been developed to improve DNS privacy, along with their limitations.

1. An Introduction to the Domain Name System (DNS)

The Domain Name System (DNS) [RFC1034] is a distributed database that is mainly used to map human-friendly domain names to IP addresses that can be used to deliver IP packets over the Internet. A simplified model of how the DNS operates is depicted in the following diagram:

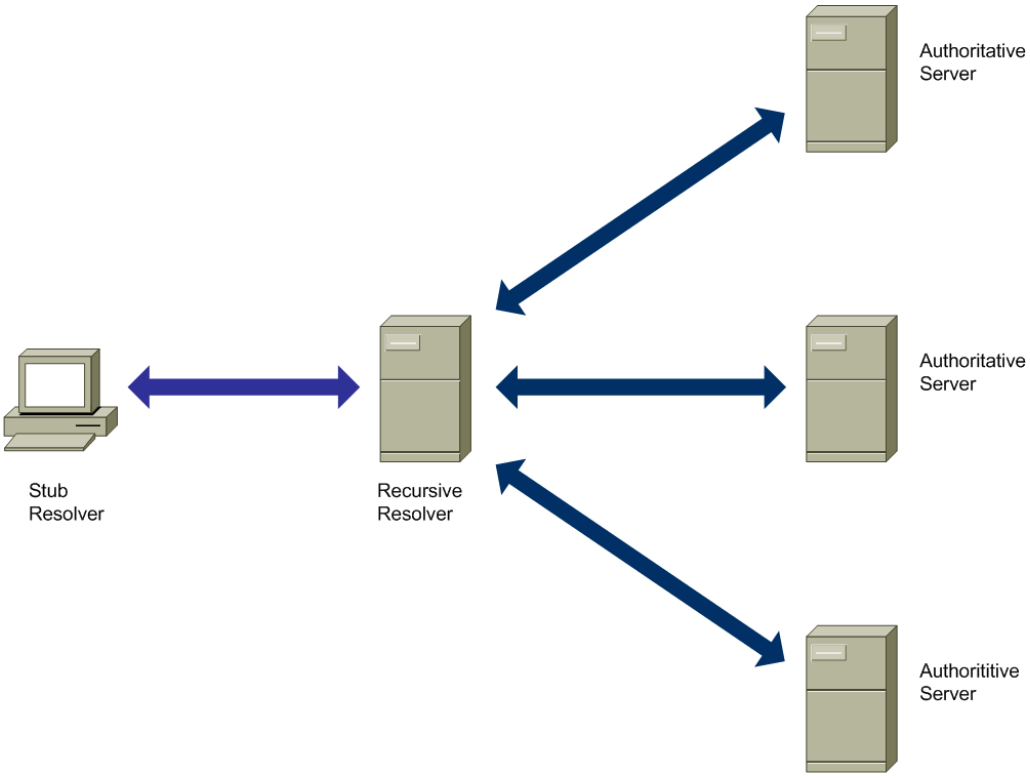

Figure 1: The Domain Name System (DNS)

There are three main different types of entities in this diagram:

- A stub resolver
- A recursive resolver (or “caching DNS server”)
- A number of authoritative name servers

The DNS stub resolver is a component of the DNS that is accessed by application programs when using the DNS for e.g. resolving domain names to IP addresses. The stub resolver simply serves as an intermediary between the application requiring DNS resolution, and a recursive DNS resolver.

The recursive resolver typically performs a number of successive queries to the DNS, to obtain the answer to the query sent by the stub resolver. While it is possible for hosts to implement a recursive resolver and on their own, it is common for hosts to throw the burden of name resolution to a recursive resolver, for at least two reasons:

1. Simplicity of code
2. Performance

For obvious reasons, if the bulk of the work to perform name resolution is done by a system other than the one running the application program, the code required at such system will be simpler (and smaller). While this is unlikely to be an important factor for modern desktop or mobile systems, it may be of value for embedded systems. Additionally, if all systems on a given network employ the same caching DNS server, the server in question might benefit from “caching” the results for recent queries, so that if the same information is required by any subsequent queries, the cached information can be employed to respond in a timelier manner.

Finally, authoritative name servers are responsible for maintaining information about domain names in a given DNS zone.

Typically, DNS resolution involves querying authoritative name servers recursively, starting from the “root zone” of the DNS, and walking the DNS hierarchy until an authoritative name server can finally provide a response to the original query. For example, the following diagram illustrates one possible scenario to resolve the domain name “www.example.com” to a set of IP addresses:

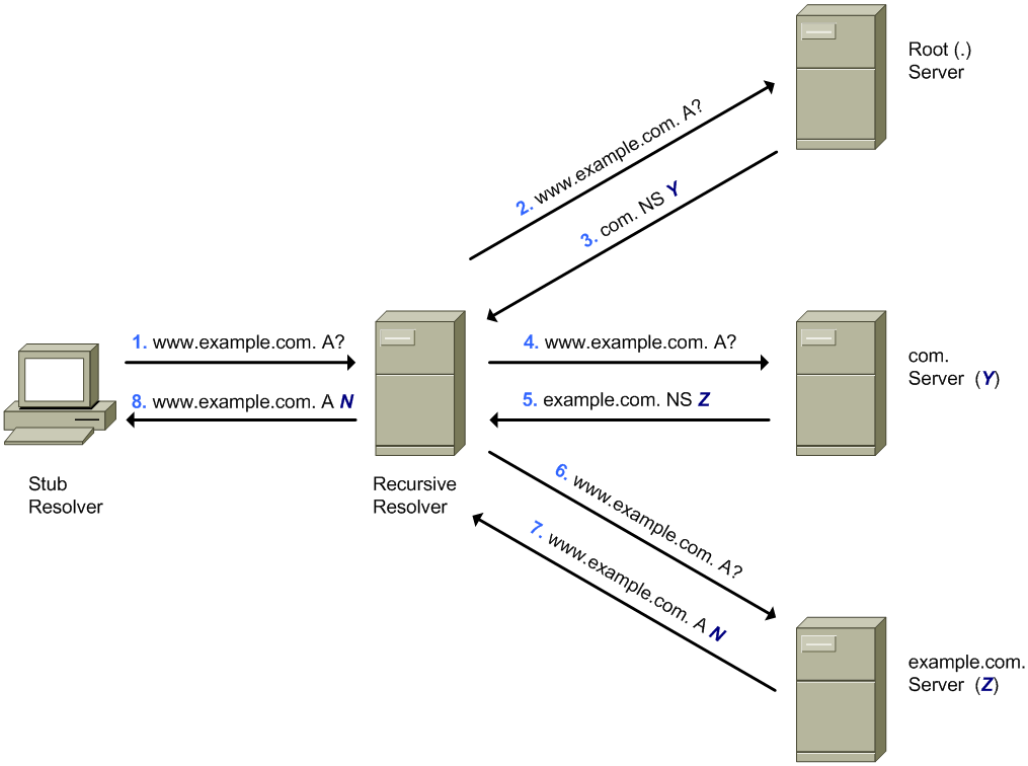

Figure 2: DNS Resolution Example

The first transaction involves the query (step 1) from the stub resolver to the DNS recursive resolver -- that will only respond (in step 8) once it has obtained the answer to the query as a result of multiple queries to a number of authoritative DNS servers.

The recursive resolver will typically resend the query (step 2) to one of the authoritative DNS servers for the root zone (“.”), which will respond (step 3) with the list of authoritative name servers for the “.com” zone.

The recursive resolver will then resend the query (step 4) to one of the authoritative DNS servers for the “.com” zone, which will respond (step 5) with a list of authoritative DNS servers for the “example.com” zone.

Finally, the recursive resolver will resend the query (step 6) to one of such authoritative DNS servers, that will respond (step 7) with the answer to the original query – that is, a list of IP addresses (the single fictional “N” address in our case) corresponding to the domain name “www.example.com”.

It is important to note that our previous example shows a simplified model of how the DNS functions. In some scenarios there might be multiple levels of caching DNS servers, such that the caching DNS server interacting with the authoritative name servers can cache and share and reuse more information, while the caching DNS servers closer to the stub resolver can provide a timelier response.

2. Information Leakages in the DNS

Since DNS resolution involves querying a public database (the DNS), one might be tempted to assume that there are no privacy implications arising from the use of the DNS. However, while the information being queried is public, the identity of the nodes querying the DNS, and the specific information being queried, are not.

For example, the DNS might publicly maintain information about the IP addresses corresponding to the domain name "www.example.com". However, the list of hosts actively querying for such information (possibly to visit the corresponding web site) is certainly not public.

Our discussion about the privacy implications of the DNS is closely related to the extent to which such information might become readily available to other (possibly rogue) entities.

The following diagram illustrates the two main "types" of interactions that typically take place as part of DNS resolution, along with the systems involved in such interactions:

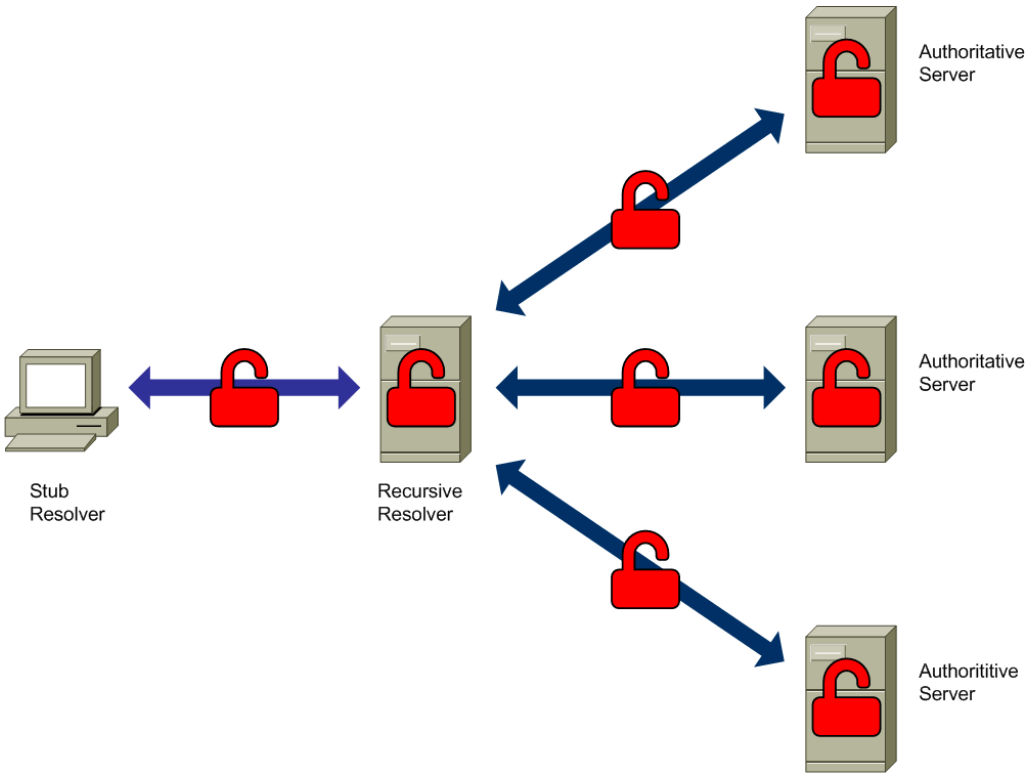

Figure 3: Interactions between DNS entities

The figure above identifies two kinds of interactions:

- Interaction between the stub resolver and the recursive resolver
- Interaction between the recursive resolver and the authoritative DNS servers

In both cases, the interaction between the involved systems occurs in plain-text – that is, there is nothing in the DNS protocol that prevents an eavesdropper from looking at the content of DNS queries and responses, along with the addresses of the systems participating in the DNS

"transactions". Additionally, the systems processing the queries, the recursive resolver and the authoritative nameservers, will have access to the queries themselves and possibly to other associated information.

Revisiting the DNS resolution procedure from a privacy standpoint may help to identify where and when DNS information leakages might occur. Based on our previous analysis of the DNS resolution process, one might assume that DNS information might leak at:

- the communications links and devices between the stub resolver and the recursive resolver;
- the recursive resolver;
- the communications links and devices between the recursive resolver and the authoritative DNS servers, and;
- the authoritative nameservers

Any entity with access to the communications links or devices between the stub resolver and the recursive resolver, or the recursive resolver and the authoritative servers, could passively collect all or part of the DNS transactions. Alternatively, an attacker that does not have access to such communications links or devices might still be able to eavesdrop on such DNS transactions – such as by attacking the routing system to divert traffic to a communication link the attacker can eavesdrop.

Additionally, it should be obvious that the recursive resolver is in a privileged position to log all of the DNS queries and responses – as after all, DNS queries are explicitly sent to the recursive resolver by the stub resolver. For example, an Internet Service Provider (ISP) operating a recursive resolver, or a company operating a public recursive resolver, could log DNS queries, and sell the corresponding log to a third party that might employ it for marketing purposes, or to profile users for political reasons. Similarly, a government agency could tap one of the communicating links to eavesdrop on DNS queries for profiling Internet users without being overseen by any other party.

The placement of the recursive resolver has concrete implications on the privacy of users, as the closer the recursive resolver is to the application, the easier it is for authoritative servers to correlate DNS queries to users. A host that implements a recursive resolver will normally query authoritative DNS servers directly, exposing its own IP address(es), thus allowing correlation of DNS queries to hosts. Similarly, a host that employs a recursive resolver at a Customer-Premises Equipment (CPE) will cause the CPE's IP address(es) to be exposed – possibly allowing the correlation of DNS activity to the set of nodes in the local network connected by the CPE.

On the other hand, use of external recursive resolvers normally "masquerades" the identities of the hosts issuing the queries from the authoritative servers – that is, the authoritative servers will only see the IP address(es) of the recursive resolvers, but not those of the stub resolver. Thus, the farther the recursive resolver is from the hosts performing DNS resolution, the more difficult it will normally be for authoritative servers to correlate incoming queries to any specific host.

While the above might seem to favour the use of recursive nameservers that are far (in terms of network topology) from the stub resolver, such network setups also have drawbacks. The further the recursive resolver is from the stub resolver, the larger the area of the network where the IP address(es) of the stub resolver are leaked along with the DNS queries. That is, queries would have to traverse a larger portion of the network, exposing the IP address(es) of the stub resolver, until they get to the recursive resolver where the identities of the hosts issuing the queries are finally "masqueraded".

Therefore, from a privacy standpoint, the placement of the recursive resolver represents a tradeoff between the amount of information exposed to the authoritative servers, and the amount of information potentially exposed to eavesdroppers that might capture traffic between the stub resolver and the recursive resolver.

Finally, authoritative nameservers might be able to log at least part of the DNS queries. One important aspect to highlight is that, during the recursive DNS resolution process, the same full DNS query (in our case, a query of "A" records for "www.example.com") is re-sent to each authoritative server as part of "walking" the authority hierarchy. This means that all authoritative servers that are queried as part of the DNS resolution process, will receive the full domain name to be resolved (even when most of them will simply refer to another authoritative nameserver, rather than provide the answer to the query). Root servers and authoritative nameservers for popular Top-Level Domains (TLDs) are therefore in a privileged position to log queries.

3. Possible DNS Privacy Improvements and Limitations

Mitigating the privacy implications of the DNS implies introducing improvements in each of the areas where information leakages could take place, namely:

- the communications links and devices between the stub resolver and the recursive resolver,
- the recursive resolver;
- the communications links and devices between the recursive resolver and the authoritative DNS servers, and;
- the authoritative nameservers

In order to prevent eavesdropping, DNS transactions should be encrypted and authenticated, but the scale of the two types of interactions (stub resolver with recursive resolver, and recursive resolver with authoritative servers) is very different.

Securing the DNS transactions between a stub resolver and a recursive resolver requires only one trust relationship between two systems, and thus the bootstrapping procedure (e.g. configuring secret keys or certificates at the stub resolver) is simple. However, securing the transactions between a recursive resolver and all authoritative nameservers requires a large number of trust relationships (one between each recursive resolver and each authoritative nameserver), and thus requires more complex solutions and a more coordinated effort for the solution to be deployed (e.g. a Public Key Infrastructure).

Encrypting transactions between recursive resolvers and authoritative servers would prevent eavesdroppers that have access to communications links between recursive resolvers and authoritative nameservers to collect information about DNS transactions. However, as noted before, mitigating these possible information leakages is harder since it requires the adoption of mitigations in all authoritative servers, so that all transactions between recursive resolvers and authoritative nameservers are encrypted and authenticated.

Furthermore, the additional resources needed to perform an encrypted transaction between a resolver and authoritative server might substantially increase the computing and network requirements of the authoritative servers, thus degrading or even limiting the quality of the DNS services – especially on the heavily trafficked root and TLD servers. This is the main reason why most of the current improvements for DNS privacy try to improve the privacy of transactions between the stub resolver

and the recursive resolver but leave the privacy of transactions between the recursive resolvers and the authoritative nameservers unaddressed. However, the DPRIVE working group of the IETF has recently been re-chartered to also address this aspect (see [DPRIVE-WG]).

Note here that DNSSEC only allows authentication of DNS responses, and does not provide confidentiality for the DNS transactions, since this was never amongst the goals of DNSSEC.

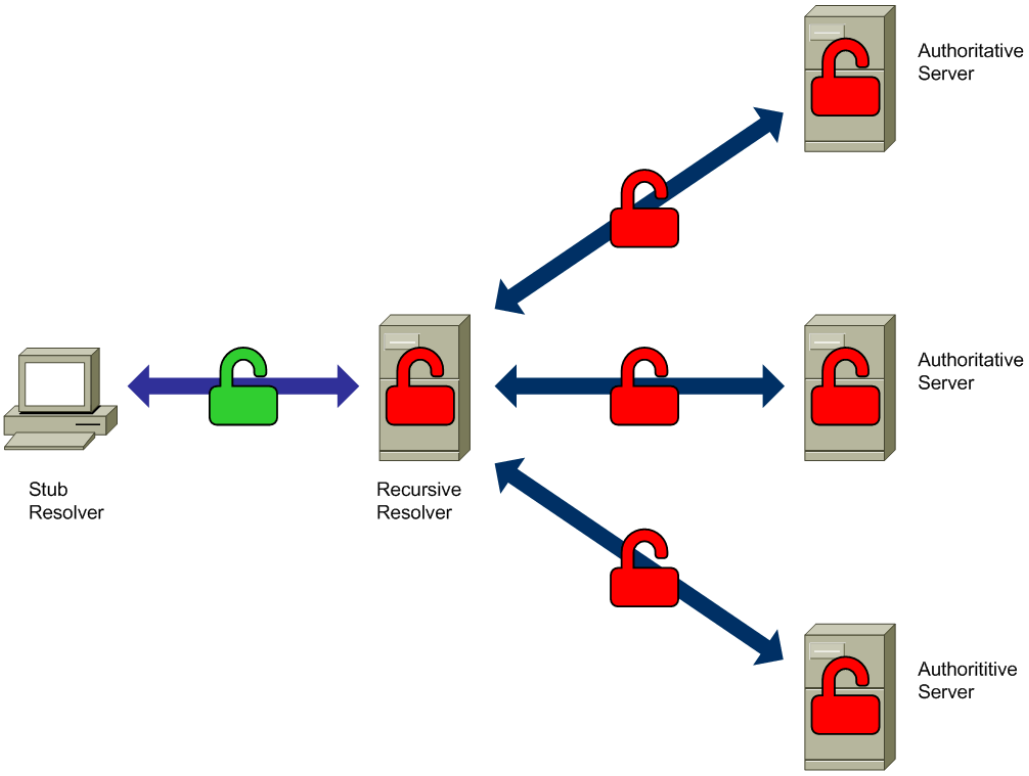

Figure 4: Improving the Privacy of DNS Resolution

As previously mentioned, one of the elements where DNS privacy might be affected is the recursive resolver. Since many hosts will normally employ the same recursive resolver, a recursive resolver is a privileged point to collect information about DNS queries -- in fact, queries from a number of hosts will be explicitly directed to the recursive server (e.g. the one provided by the ISP) and therefore DNS data collection becomes trivial. So, when using an external recursive resolver, there is implicit trust in the organization operating the recursive resolver.

In the typical case where the recursive resolver is provided by the ISP, the user implicitly trusts the ISP not to collect information about DNS queries or, at the very least, not share the collected data with any third parties. On the other hand, when users employ third-party recursive resolvers (e.g. Google, Cloudflare, Quad9, OpenDNS, etc.) to avoid using their ISP's recursive resolvers, trust is switched to the organization operating the third-party recursive resolver. Put another way, external recursive resolvers remain one strategic point where information about DNS transactions may be collected, no matter which organization operates the recursive resolver

The selection and preference of e.g. a third-party "privacy-enhanced" recursive resolver over an ISP-provided recursive resolver simply shifts trust from one organization to another, and does not really eliminate the possible information leakage at the recursive resolver (i.e. the ability of the organization

operating the resolver to collect DNS information). While there seems to be a tendency to assume that it is preferable to employ a privacy-enhanced recursive resolver over the ones provided by ISPs, the choice of the recursive resolver should be based on threat modeling. Please see "5. On the Use of Technologies such as DoT or DoH" for further details.

One might be also tempted to believe that the privacy implications of recursive resolvers might be eliminated by avoiding the use of an external recursive resolver in the first place, running a local recursive resolver in each host. However, external recursive resolvers also introduce benefits for DNS privacy, since they normally masquerade the identity of the clients during the resolution process, preventing authoritative servers from correlating DNS queries to specific stub resolvers and clients.

If hosts were to incorporate a recursive resolver, they would expose the host IP address when communicating with authoritative servers, possibly enabling both authoritative servers and eavesdroppers to correlate each query to the actual host that originated it. Additionally, when an external recursive resolver is employed, some or all the queried information might be readily available in the DNS cache, relieving the recursive resolver of the need to query the corresponding authoritative servers and therefore "hiding" some of the DNS information needed by the hosts.

Finally, as illustrated in the example from Figure 2, recursive resolvers normally re-send the same original query in each step of the iterative resolution [RFC1035]. While this is common and standard practice, it unnecessarily discloses complete query to be resolved to each authoritative nameserver that participates in some part of the resolution process (and possibly to any eavesdroppers). This is another area where recent work has helped to improve DNS privacy.

4. Standardization Work in DNS Privacy

Several efforts have undertaken at the IETF and elsewhere to mitigate some of the privacy implications discussed in this document. The following sub-sections describe developments to improve DNS privacy properties in the following areas:

- Query Name Minimisation
- Encruption of transactions between stub resolvers and recursive resolvers

The work on these two areas is rather orthogonal as Query Name Minimisation aims at mitigating information leaks that happen when the recursive resolver resends the original query multiple times during the recursive DNS resolution process. On the other hand, a number of alternative efforts aim at improving the confidentiality of DNS transactions between stub resolvers and recursive resolvers.

4.1. DNS Query Name Minimisation

QNAME Minimisation is an experimental proposal specified in [RFC7816] that aims to minimize the amount of information sent in DNS queries. Rather than resending the same DNS query to each authoritative name server, the QNAME Minimisation argues that the recursive resolver should walk the authority hierarchy of a domain name by querying for NS records of domain names, starting with the TLD, and increasing one level in the domain depth for each subsequent query.

To illustrate the algorithm in action, our example from Figure 2 (Section 1) would look as follows if QNAME minimisation were employed:

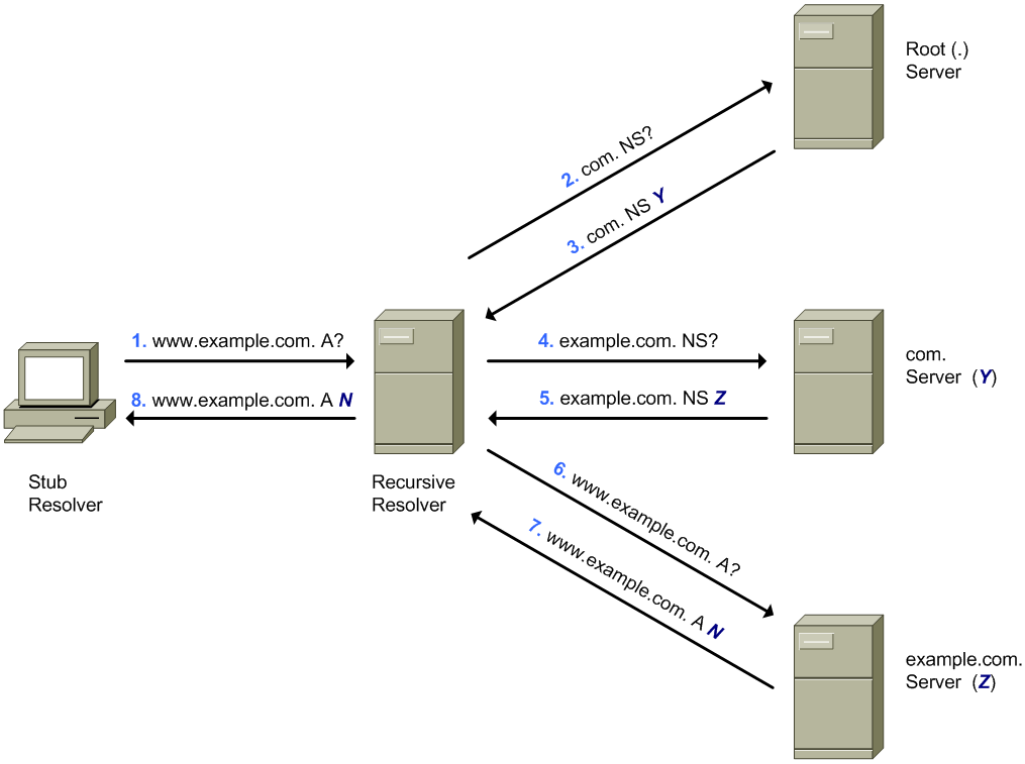

Figure 5: DNS Query Name (QNAME) Minimisation

While QNAME minimisation may be seen as a modification in how queries are sent, there is no explicit requirement in any of the DNS standards that the same full original query be resent to each authoritative server during DNS resolution. Unfortunately, some non-compliant DNS software may respond erroneously when QNAME minimisation is employed, and in some DNS setups the algorithm illustrated above might not work as expected.

In order to address these issues, [RFC7816] specifies a slightly modified version of this algorithm, such that the aforementioned corner cases and network scenarios are addressed.

QNAME minimisation is already implemented in a number of popular recursive resolvers. See [DNS-IMPL] for further details. At the time of this writing, there is ongoing work at the DPRIVE working group of the IETF to publish QNAME minimisation on Standards Track [QNAME-S].

4.2. Encryption of Transactions between Stub Resolvers and Recursive Resolvers

The following technologies have been developed to improve the privacy properties of the transactions between stub and recursive resolvers.

4.2.1. DNS over TLS (DoT)

[RFC7858] specifies how to communicate with a recursive resolver over a TLS-secured connection. However, it also has the potential for improving the privacy properties of transactions between recursive resolvers and authoritative nameservers (see e.g. [FB-DOT]).

The service employs a separate port number, TCP port 853, rather than the existing DNS service port (53). The recursive resolvers may be authenticated by means of a Subject Public Key Info (SPKI) Fingerprint (please see Section 3.2 and Section 4 of [RFC7858] for details).

There are multiple implementations of DoT, and there are a number of public recursive resolvers that support DoT. [DOT-SYSD] describes how to configure the DoT implementation in Linux's systemd. [DOT-STBY] describes how to implement a DoT stub resolver and a DoT recursive resolver with Stubby. For more details about DoT implementations, please see [DNS-IMPL].

4.2.2. DNS over DTLS (DoD)

[RFC8094] specifies how to communicate with a recursive resolver over a DTLS-secured "connection". DoD employs the UDP service port 853 in a similar manner to DoT.

While DoD has a number of benefits, it suffers from the same issues as UDP-based DNS: it cannot operate as a stand-alone mechanism but needs a fallback mechanism for cases where the payload is too big, and the packet must be truncated.

At the time of writing, there are not any known implementations of DoD.

4.2.3. DNS over HTTPS (DoH)

[RFC8484] specifies how to send and receive DNS queries over HTTPS. Server configuration is performed out of band, and the connection with the resolver is secured as any other HTTPS traffic. DoH is mostly targeted at web browsers and does **not** have the potential for improving the privacy properties of transactions between recursive resolvers and authoritative nameservers.

One possible benefit from a user standpoint is that since DNS queries can be intermingled with normal web traffic, DoH might prove more difficult to block than DoD and DoT – or at the very least would require blocking the IP addresses of well-known DoH servers, rather than simply the corresponding DoH service port.

There are multiple implementations of this protocol, as well as multiple public recursive resolvers with support for DoH. Please see [DNS-IMPL] for further details.

4.2.4. DNSCrypt

The DNSCrypt protocol pioneered DNS privacy and was developed outside of formal standardization bodies such as the IETF. As with the other protocols such as DoT, it is meant to provide confidentiality to the DNS transactions between stub resolvers and recursive resolvers.

There are several implementations of DNSCrypt that can be easily integrated with popular DNS software, and there are also a number of public recursive resolvers with support for DNSCrypt. Please see [DNS-CRYPT] for more information.

5. On the Use of Technologies such as DoT or DoH

While there is a tendency to assume that it is preferable to employ a "privacy-enhanced" recursive resolver over the one advertised on the local network (e.g. the one provided by local ISP), the choice of the recursive resolver should be based on an actual threat model. For example, if the adversary is expected to operate closer to the local ISP (or is assumed to be the ISP itself), encrypting all queries towards a third-party recursive resolver might help improve privacy. However, if the main adversary is assumed to operate on external networks, then using a third-party recursive resolver might actually have a negative impact on privacy. Additionally, employing a third-party recursive resolver might imply that DNS traffic become handled by an organization with a different legal jurisdiction, may result in the use of recursive resolvers that are shared by a larger number of users (thus concentrating trust of many hosts on a small set of recursive resolvers which become more attractive to rogue actors), etc.

On the other hand, when communicating with the recursive resolver of choice, it is generally preferable to employ mechanisms that encrypt DNS transactions (such as DoT) as opposed to traditional plain-text DNS transactions. For example, if an ISP-provided recursive resolver is to be employed, and the recursive resolver implements DoT, using DoT will prevent eavesdroppers on the local network from collecting information about the DNS queries performed by local users.

6. Conclusions

The DNS was originally developed without any kind of considerations for user privacy and may therefore leak information about DNS queries and responses that can be correlated to specific network activity (e.g. applications employed, web sites visited, people communicated with, etc.).

Since pervasive monitoring became a major concern in protocol development [RFC7258], a number of efforts have aimed improve the privacy properties of important Internet protocols such as the DNS. Most efforts on DNS privacy have tackled only part of the problem space – namely privacy in communications between stub rand recursive resolvers. However, there is ongoing work to also improve privacy in communications between recursive resolvers and authoritative nameservers (see [DPRIVE-WG]).

Most of the recent work on DNS privacy is associated with the ability of users to override the default recursive resolver provided by the local ISP, with a (typically public) privacy-enhanced recursive resolver, that provides privacy for the DNS transactions between the stub resolver and the recursive resolver. Depending on the specific threat model that applies to each user, use of third-party recursive resolvers via encrypted traffic may (or may not) help improve user privacy.

In addition to these considerations, it should be stressed that many protocols leak information that may endanger user privacy. For instance, the Server Name Identification (SNI) TLS extension includes the web server name being visited in plain-text, and leaks information about visited web sites even when employing HTTPS.

The mechanisms described in this document should be seen as ways to improve, in specific scenarios, certain aspects of network privacy, but not as replacements for other privacy mechanisms such as Virtual Private Networks (VPNs) or implementations of Onion routing such as that in the Tor software [TOR].

7. Acknowledgements

Stéphane Bortzmeyer, Kevin Meynell, Hugo Salgado, Dan York, and Jan Žorž provided valuable comments on this document.

8. References

[DNS-CRYPT] DNSCrypt web site
https://dnscrypt.info/.

[DNS-IMPL] "DNS Privacy Implementation Status", last updated September 11, 2018
https://dnsprivacy.org/wiki/display/DP/DNS+Privacy+Implementation+Status

[DOH-FFOX] Gont, F., "DNS-over-HTTPS (DoH) Support in Mozilla Firefox". December 2018.
https://www.internetsociety.org/blog/2018/12/dns-privacy-support-in-mozilla-firefox/

[DOT-SYSD] Gont, F. "DNS-over-TLS in Linux (systemd)". December 2018.
https://www.internetsociety.org/blog/2018/12/dns-privacy-in-linux-systemd/

[DOT-STBY] Žorž, J., "DNS over TLS: experience from the Go6lab". September 2018.
https://www.internetsociety.org/blog/2017/09/dns-tls-experience-go6lab/

[DPRIVE-WG] DNS PRIVate Exchange (dprive) working group.
https://datatracker.ietf.org/wg/dprive/about/

[FB-DOT] "Encrypting DNS end-to-end", Facebook Engineering Blog. December 21, 2018.
https://code.fb.com/security/dns-over-tls/

[QNAME-S] "DNS Query Name Minimisation to Improve Privacy", IETF Internet-Draft (work in progress), draft-ietf-dnsop-rfc7816bis-01
https://tools.ietf.org/html/draft-ietf-dnsop-rfc7816bis

[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

[RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014
https://www.rfc-editor.org/info/rfc7258

[RFC7626] Bortzmeyer, S., "DNS Privacy Considerations", RFC 7626, DOI 10.17487/RFC7626, August 2015

https://www.rfc-editor.org/info/rfc7626>

[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

[RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016
https://www.rfc-editor.org/info/rfc7858

[RFC8094] Reddy, T., Wing, D., and P. Patil, "DNS over Datagram Transport Layer Security (DTLS)", RFC 8094, DOI 10.17487/RFC8094, February 2017
https://www.rfc-editor.org/info/rfc8094

[RFC8484] Hoffman, P., and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018
https://www.rfc-editor.org/info/rfc8484

[TOR] Tor Project
https://www.torproject.org/