

DNS PRIVate Exchange (dprive) Working Group
Internet-Draft
Intended status: Informational
Expires: August 23, 2015

S. Bortzmeyer
AFNIC
February 19, 2015

**DNS privacy considerations
draft-ietf-dprive-problem-statement-02**

Abstract

This document describes the privacy issues associated with the use of the DNS by Internet users. It is intended to be an analysis of the present situation and does not prescribe solutions.

Discussions of the document should take place on the DPRIVE working group mailing list [dprive].

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 August 23, 2015.

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	2
2.	Risks	4
2.1.	The alleged public nature of DNS data	4
2.2.	Data in the DNS request	5
2.3.	Cache snooping	6
2.4.	On the wire	6
2.5.	In the servers	8
2.5.1.	In the recursive resolvers	8
2.5.2.	In the authoritative name servers	8
2.5.3.	Rogue servers	10
3.	Actual "attacks"	10
4.	Legalities	10
5.	Security considerations	11
6.	Acknowledgments	11
7.	IANA considerations	11
8.	References	11
8.1.	Normative References	11
8.2.	Informative References	12
8.3.	URIs	15
	Author's Address	15

[1.](#) Introduction

This document is an analysis of the DNS privacy issues, in the spirit of [section 8 of \[RFC6973\]](#).

The Domain Name System is specified in [\[RFC1034\]](#) and [\[RFC1035\]](#). It is one of the most important infrastructure components of the Internet and often ignored or misunderstood. Almost every activity on the Internet starts with a DNS query (and often several). Its use has many privacy implications and this is an attempt at a comprehensive and accurate list.

Let us begin with a simplified reminder of how the DNS works. (REMOVE BEFORE PUBLICATION: We hope that the document [\[I-D.hoffman-dns-terminology\]](#) will be published as a RFC so most of this section could be replaced by a reference to it.) A client, the stub resolver, issues a DNS query to a server, called the recursive resolver (also called caching resolver or full resolver or recursive name server). Let's use the query "What are the AAAA records for www.example.com?" as an example. AAAA is the qtype (Query Type), and www.example.com is the qname (Query Name). The recursive resolver will first query the root nameservers. In most cases, the root

nameservers will send a referral. In this example, the referral will be to the .com nameservers. The resolver repeats the query to one of the .com nameservers. The .com nameservers, in turn, will refer to the example.com nameservers. The example.com nameserver will then return the answer. The root name servers, the name servers of .com and the name servers of example.com are called authoritative name servers. It is important, when analyzing the privacy issues, to remember that the question asked to all these name servers is always the original question, not a derived question. The question sent to the root name servers is "What are the AAAA records for www.example.com?", not "What are the name servers of .com?". By repeating the full question, instead of just the relevant part of the question to the next in line, the DNS provides more information than necessary to the nameserver.

Because DNS relies on caching heavily, not all questions are sent to the authoritative name servers. If a few seconds later the stub resolver asks to the recursive resolver, "What are the SRV records of _xmpp-server._tcp.example.com?", the recursive resolver will remember that it knows the name servers of example.com and will just query them, bypassing the root and .com. Because there is typically no caching in the stub resolver, the recursive resolver, unlike the authoritative servers, sees all the DNS traffic. (Applications, like Web browsers, may have some form of caching, which do not follow DNS rules. So, the recursive resolver does not see all the name resolution activity.)

It should be noted that DNS recursive resolvers sometimes forward requests to other recursive resolvers, typically bigger machines, with a larger and more shared cache (and the query hierarchy can be even deeper, with more than two levels of recursive resolvers). From the point of view of privacy, these forwarders are like resolvers, except that they do not see all of the requests being made (due to caching in the first resolver).

All this DNS traffic is currently sent in clear (unencrypted), except a few cases when the IP traffic is protected, for instance in an IPsec VPN.

Today, almost all DNS queries are sent over UDP. This has practical consequences when considering encryption of the traffic as a possible privacy technique. Some encryption solutions are only designed for TCP, not UDP.

Another important point to keep in mind when analyzing the privacy issues of DNS is the mix of many sort of DNS requests received by a server. Let's assume an eavesdropper wants to know which Web page is

viewed by an user. For a typical Web page, there are three sorts of DNS requests being issued:

Primary request: this is the domain name in the URL that the user typed, selected from a bookmark or chose by clicking on an hyperlink. Presumably, this is what is of interest for the eavesdropper.

Secondary requests: these are the additional requests performed by the user agent (here, the Web browser) without any direct involvement or knowledge of the user. For the Web, they are triggered by embedded content, CSS sheets, JavaScript code, embedded images, etc. In some cases, there can be dozens of domain names in different contexts on a single Web page.

Tertiary requests: these are the additional requests performed by the DNS system itself. For instance, if the answer to a query is a referral to a set of name servers, and the glue records are not returned, the resolver will have to do additional requests to turn name servers' names into IP addresses. Similarly, even if glue records are returned, a careful recursive server will do tertiary requests to verify the IP addresses of those records.

It can be noted also that, in the case of a typical Web browser, more DNS requests are sent, for instance to prefetch resources that the user may query later, or when autocompleting the URL in the address bar (which obviously is a big privacy concern).

For privacy-related terms, we will use here the terminology of [\[RFC6973\]](#).

2. Risks

This document focuses mostly on the study of privacy risks for the end-user (the one performing DNS requests). We consider the risks of pervasive surveillance ([\[RFC7258\]](#)) as well as risks coming from a more focused surveillance. Privacy risks for the holder of a zone (the risk that someone gets the data) are discussed in [\[RFC5936\]](#) and [\[RFC5155\]](#). Non-privacy risks (such as cache poisoning) are out of scope.

2.1. The alleged public nature of DNS data

It has long been claimed that "the data in the DNS is public". While this sentence makes sense for an Internet-wide lookup system, there are multiple facets to the data and metadata involved that deserve a more detailed look. First, access control lists and private namespaces notwithstanding, the DNS operates under the assumption

that public facing authoritative name servers will respond to "usual" DNS queries for any zone they are authoritative for without further authentication or authorization of the client (resolver). Due to the lack of search capabilities, only a given qname will reveal the resource records associated with that name (or that name's non-existence). In other words: one needs to know what to ask for, in order to receive a response. The zone transfer qtype [[RFC5936](#)] is often blocked or restricted to authenticated/authorized access to enforce this difference (and maybe for other, more dubious reasons).

Another differentiation to be considered is between the DNS data itself and a particular transaction (i.e., a DNS name lookup). DNS data and the results of a DNS query are public, within the boundaries described above, and may not have any confidentiality requirements. However, the same is not true of a single transaction or sequence of transactions; that transaction is not/should not be public. A typical example from outside the DNS world is: the Web site of Alcoholics Anonymous is public; the fact that you visit it should not be.

[2.2.](#) Data in the DNS request

The DNS request includes many fields but two of them seem particularly relevant for the privacy issues: the qname and the source IP address. "source IP address" is used in a loose sense of "source IP address + maybe source port", because the port is also in the request and can be used to sort out several users sharing an IP address (behind a CGN for instance [[RFC6269](#)]).

The qname is the full name sent by the user. It gives information about what the user does ("What are the MX records of example.net?" means he probably wants to send email to someone at example.net, which may be a domain used by only a few persons and therefore very revealing about communication relationships). Some qnames are more sensitive than others. For instance, querying the A record of google-analytics.com reveals very little (everybody visits Web sites which use Google Analytics) but querying the A record of www.verybad.example where verybad.example is the domain of an illegal or very offensive organization may create more problems for the user. Also, sometimes, the qname embeds the software one uses, which could be a privacy issue. For instance, _ldap._tcp.Default-First-Site-Name._sites.gc._msdcs.example.org. There are also some BitTorrent clients that query a SRV record for _bittorrent-tracker._tcp.domain.example.

Another important thing about the privacy of the qname is the future usages. Today, the lack of privacy is an obstacle to putting potentially sensitive or personally identifiable data in the DNS. At

the moment your DNS traffic might reveal that you are doing email but not with whom. If your MUA starts looking up PGP keys in the DNS [[I-D.wouters-dane-openpgp](#)] then privacy becomes a lot more important. And email is just an example; there would be other really interesting uses for a more privacy-friendly DNS.

For the communication between the stub resolver and the recursive resolver, the source IP address is the address of the user's machine. Therefore, all the issues and warnings about collection of IP addresses apply here. For the communication between the recursive resolver and the authoritative name servers, the source IP address has a different meaning; it does not have the same status as the source address in a HTTP connection. It is now the IP address of the recursive resolver which, in a way "hides" the real user. However, hiding does not always work. Sometimes [[I-D.ietf-dnsop-edns-client-subnet](#)] is used (see its privacy analysis in [[denis-edns-client-subnet](#)]). Sometimes the end user has a personal recursive resolver on her machine. In both cases, the IP address is as sensitive as it is for HTTP.

A note about IP addresses: there is currently no IETF document which describes in detail all the privacy issues around IP addressing. In the meantime, the discussion here is intended to include both IPv4 and IPv6 source addresses. For a number of reasons their assignment and utilization characteristics are different, which may have implications for details of information leakage associated with the collection of source addresses. (For example, a specific IPv6 source address seen on the public Internet is less likely than an IPv4 address to originate behind a CGN or other NAT.) However, for both IPv4 and IPv6 addresses, it's important to note that source addresses are propagated with queries and comprise metadata about the host, user, or application that originated them.

[2.3.](#) Cache snooping

The content of recursive resolvers' caches can reveal data about the clients using it (the privacy risks depend on the number of clients). This information can sometimes be examined by sending DNS queries with RD=0 to inspect cache content, particularly looking at the DNS TTLs. Since this also is a reconnaissance technique for subsequent cache poisoning attacks, some counter measures have already been developed and deployed.

[2.4.](#) On the wire

DNS traffic can be seen by an eavesdropper like any other traffic. It is typically not encrypted. (DNSSEC, specified in [[RFC4033](#)] explicitly excludes confidentiality from its goals.) So, if an

initiator starts a HTTPS communication with a recipient, while the HTTP traffic will be encrypted, the DNS exchange prior to it will not be. When other protocols will become more and more privacy-aware and secured against surveillance, the DNS risks to become "the weakest link" in privacy.

An important specificity of the DNS traffic is that it may take a different path than the communication between the initiator and the recipient. For instance, an eavesdropper may be unable to tap the wire between the initiator and the recipient but may have access to the wire going to the recursive resolver, or to the authoritative name servers.

The best place to tap, from an eavesdropper's point of view, is clearly between the stub resolvers and the recursive resolvers, because traffic is not limited by DNS caching.

The attack surface between the stub resolver and the rest of the world can vary widely depending upon how the end user's computer is configured. By order of increasing attack surface:

The recursive resolver can be on the end user's computer. In (currently) a small number of cases, individuals may choose to operate their own DNS resolver on their local machine. In this case the attack surface for the connection between the stub resolver and the caching resolver is limited to that single machine.

The recursive resolver may be at the local network edge. For many/most enterprise networks and for some residential users the caching resolver may exist on a server at the edge of the local network. In this case the attack surface is the local network. Note that in large enterprise networks the DNS resolver may not be located at the edge of the local network but rather at the edge of the overall enterprise network. In this case the enterprise network could be thought of as similar to the IAP network referenced below.

The recursive resolver can be in the IAP (Internet Access Provider) premises. For most residential users and potentially other networks the typical case is for the end user's computer to be configured (typically automatically through DHCP) with the addresses of the DNS recursive resolvers at the IAP. The attack surface for on-the-wire attacks is therefore from the end user system across the local network and across the IAP network to the IAP's recursive resolvers.

The recursive resolver can be a public DNS service. Some machines may be configured to use public DNS resolvers such as those operated by Google Public DNS or OpenDNS. The end user may have configured their machine to use these DNS recursive resolvers themselves - or

their IAP may have chosen to use the public DNS resolvers rather than operating their own resolvers. In this case the attack surface is the entire public Internet between the end user's connection and the public DNS service.

2.5. In the servers

Using the terminology of [[RFC6973](#)], the DNS servers (recursive resolvers and authoritative servers) are enablers: they facilitate communication between an initiator and a recipient without being directly in the communications path. As a result, they are often forgotten in risk analysis. But, to quote again [[RFC6973](#)], "Although [...] enablers may not generally be considered as attackers, they may all pose privacy threats (depending on the context) because they are able to observe, collect, process, and transfer privacy-relevant data." In [[RFC6973](#)] parlance, enablers become observers when they start collecting data.

Many programs exist to collect and analyze DNS data at the servers. From the "query log" of some programs like BIND, to tcpdump and more sophisticated programs like PacketQ [[packetq](#)] and DNSmezzo [[dnsmezzo](#)]. The organization managing the DNS server can use these data itself or it can be part of a surveillance program like PRISM [[prism](#)] and pass data to an outside observer.

Sometimes, these data are kept for a long time and/or distributed to third parties, for research purposes [[ditl](#)] [[day-at-root](#)], for security analysis, or for surveillance tasks. These uses are sometimes under some sort of contract, with various limitations, for instance on redistribution, giving the sensitive nature of the data. Also, there are observation points in the network which gather DNS data and then make it accessible to third-parties for research or security purposes ("passive DNS [[passive-dns](#)]").

2.5.1. In the recursive resolvers

Recursive Resolvers see all the traffic since there is typically no caching before them. To summarize: your recursive resolver knows a lot about you. The resolver of a large IAP, or a large public resolver can collect data from many users. You may get an idea of the data collected by reading the privacy policy of a big public resolver [[1](#)].

2.5.2. In the authoritative name servers

Unlike what happens for recursive resolvers, observation capabilities of authoritative name servers are limited by caching; they see only the requests for which the answer was not in the cache. For

aggregated statistics ("What is the percentage of LOC queries?"), this is sufficient; but it prevents an observer from seeing everything. Still, the authoritative name servers see a part of the traffic, and this subset may be sufficient to violate some privacy expectations.

Also, the end user has typically some legal/contractual link with the recursive resolver (he has chosen the IAP, or he has chosen to use a given public resolver), while having no control and perhaps no awareness of the role of the authoritative name servers and their observation abilities.

As noted before, using a local resolver or a resolver close to the machine decreases the attack surface for an on-the-wire eavesdropper. But it may decrease privacy against an observer located on an authoritative name server. This authoritative name server will see the IP address of the end client, instead of the address of a big recursive resolver shared by many users.

This "protection", when using a large resolver with many clients, is no longer present if [[I-D.ietf-dnsop-edns-client-subnet](#)] is used because, in this case, the authoritative name server sees the original IP address (or prefix, depending on the setup).

As of today, all the instances of one root name server, L-root, receive together around 20,000 queries per second. While most of it is junk (errors on the TLD name), it gives an idea of the amount of big data which pours into name servers.

Many domains, including TLDs, are partially hosted by third-party servers, sometimes in a different country. The contracts between the domain manager and these servers may or may not take privacy into account. Whatever the contract, the third-party hoster may be honest or not but, in any case, it will have to follow its local laws. It may be surprising for an end-user that requests to a given ccTLD may go to servers managed by organisations outside of the country.

Also, it seems [[aeris-dns](#)] that there is a strong concentration of authoritative name servers among "popular" domains (such as the Alexa Top N list). For instance, among the Alexa Top 100k, one DNS provider hosts today 10 % of the domains. The ten most important DNS providers host together one third of the domains. With the control (or the ability to sniff the traffic) of a few name servers, you can gather a lot of information.

2.5.3. Rogue servers

A rogue DHCP server, or a trusted DHCP server that has had its configuration altered by malicious parties, can direct you to a rogue recursive resolver. Most of the time, it seems to be done to divert traffic, by providing lies for some domain names. But it could be used just to capture the traffic and gather information about you. Same thing for malware like DNSChanger[dnschanger] which changes the recursive resolver in the machine's configuration, or with transparent DNS proxies in the network that will divert the traffic intended for a legitimate DNS server (for instance [[turkey-googledns](#)]).

3. Actual "attacks"

A very quick examination of DNS traffic may lead to the false conclusion that extracting the needle from the haystack is difficult. "Interesting" primary DNS requests are mixed with useless (for the eavesdropper) secondary and tertiary requests (see the terminology in [Section 1](#)). But, in this time of "big data" processing, powerful techniques now exist to get from the raw data to what the eavesdropper is actually interested in.

Many research papers about malware detection use DNS traffic to detect "abnormal" behaviour that can be traced back to the activity of malware on infected machines. Yes, this research was done for the good; but, technically, it is a privacy attack and it demonstrates the power of the observation of DNS traffic. See [[dns-footprint](#)], [[dagon-malware](#)] and [[darkreading-dns](#)].

Passive DNS systems [[passive-dns](#)] allow reconstruction of the data of sometimes an entire zone. They are used for many reasons, some good, some bad. Well-known passive DNS systems keep only the DNS responses, and not the source IP address of the client, precisely for privacy reasons. Other passive DNS systems may not be so careful. And there is still the potential problems with revealing qnames.

The revelations (from the Edward Snowden documents, leaked from the NSA) of the MORECOWBELL surveillance program [[morecowbell](#)], which uses the DNS, both passively and actively, to gather surreptitiously information about the users, is another good example that the lack of privacy protections in the DNS is actively exploited.

4. Legalities

To our knowledge, there are no specific privacy laws for DNS data. Interpreting general privacy laws like [[data-protection-directive](#)]

(European Union) in the context of DNS traffic data is not an easy task and it seems there is no court precedent here.

5. Security considerations

This document is entirely about security, more precisely privacy. It just lays out the problem, it does not try to set requirements (with the choices and compromises they imply), much less to define solutions. A possible document on requirements for DNS privacy is [[I-D.hallambaker-dnse](#)]. Possible solutions to the issues described here are discussed in other documents (currently too many to all be mentioned), see for instance [[I-D.ietf-dnsop-qname-minimisation](#)] for the minimisation of data, or [[I-D.hzhwm-start-tls-for-dns](#)] about encryption.

6. Acknowledgments

Thanks to Nathalie Boulevard and to the CENTR members for the original work which led to this document. Thanks to Ondrej Sury for the interesting discussions. Thanks to Mohsen Souissi and John Heidemann for proofreading, to Paul Hoffman, Marcos Sanz, Tim Wicinski, Allison Mankin and Warren Kumari for proofreading, technical remarks, and many readability improvements. Thanks to Dan York, Suzanne Woolf, Tony Finch, Peter Koch and Frank Denis for good written contributions.

7. IANA considerations

This document has no actions for IANA.

8. References

8.1. Normative References

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, [RFC 1034](#), November 1987.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, [RFC 1035](#), November 1987.
- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", [RFC 6973](#), July 2013.
- [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", [BCP 188](#), [RFC 7258](#), May 2014.

8.2. Informative References

- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", [RFC 4033](#), March 2005.
- [RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", [RFC 5155](#), March 2008.
- [RFC5936] Lewis, E. and A. Hoenes, "DNS Zone Transfer Protocol (AXFR)", [RFC 5936](#), June 2010.
- [RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", [RFC 6269](#), June 2011.
- [I-D.ietf-dnsop-edns-client-subnet]
Contavalli, C., Gaast, W., Lawrence, D., and W. Kumari, "Client Subnet in DNS Requests", [draft-ietf-dnsop-edns-client-subnet-00](#) (work in progress), November 2014.
- [I-D.hallambaker-dnse]
Hallam-Baker, P., "DNS Privacy and Censorship: Use Cases and Requirements.", [draft-hallambaker-dnse-02](#) (work in progress), November 2014.
- [I-D.wouters-dane-openpgp]
Wouters, P., "Using DANE to Associate OpenPGP public keys with email addresses", [draft-wouters-dane-openpgp-02](#) (work in progress), February 2014.
- [I-D.hzhwm-start-tls-for-dns]
Zi, Z., Zhu, L., Heidemann, J., Mankin, A., and D. Wessels, "Starting TLS over DNS", [draft-hzhwm-start-tls-for-dns-01](#) (work in progress), July 2014.
- [I-D.ietf-dnsop-qname-minimisation]
Bortzmeyer, S., "DNS query name minimisation to improve privacy", [draft-ietf-dnsop-qname-minimisation-01](#) (work in progress), February 2015.
- [I-D.hoffman-dns-terminology]
Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", [draft-hoffman-dns-terminology-01](#) (work in progress), January 2015.

- [dprive] IETF, DPRIVE., "The DPRIVE working group", March 2014, <<http://www.ietf.org/mail-archive/web/dns-privacy/>>.
- [denis-edns-client-subnet] Denis, F., "Security and privacy issues of edns-client-subnet", August 2013, <<https://00f.net/2013/08/07/edns-client-subnet/>>.
- [dagon-malware] Dagon, D., "Corrupted DNS Resolution Paths: The Rise of a Malicious Resolution Authority", 2007, <<https://www.dns-oarc.net/files/workshop-2007/Dagon-Resolution-corruption.pdf>>.
- [dns-footprint] Stoner, E., "DNS footprint of malware", October 2010, <<https://www.dns-oarc.net/files/workshop-201010/OARC-ers-20101012.pdf>>.
- [morecowbell] Grothoff, C., Wachs, M., Ermert, M., and J. Appelbaum, "NSA's MORECOWBELL: Knell for DNS", January 2015, <<https://gnunet.org/morecowbell>>.
- [darkreading-dns] Lemos, R., "Got Malware? Three Signs Revealed In DNS Traffic", May 2013, <<http://www.darkreading.com/monitoring/got-malware-three-signs-revealed-in-dns/240154181>>.
- [dnschanger] Wikipedia, , "DNSChanger", November 2011, <<http://en.wikipedia.org/wiki/DNSChanger>>.
- [packetq] Dot SE, , "PacketQ, a simple tool to make SQL-queries against PCAP-files", 2011, <<https://github.com/dotse/packetq/wiki>>.
- [dnsmezzo] Bortzmeyer, S., "DNSmezzo", 2009, <<http://www.dnsmezzo.net/>>.
- [prism] NSA, , "PRISM", 2007, <http://en.wikipedia.org/wiki/PRISM_%28surveillance_program%29>.
- [ditl] CAIDA, , "A Day in the Life of the Internet (DITL)", 2002, <<http://www.caida.org/projects/ditl/>>.

[day-at-root]

Castro, S., Wessels, D., Fomenkov, M., and K. Claffy, "A Day at the Root of the Internet", 2008,
<<http://www.sigcomm.org/sites/default/files/ccr/papers/2008/October/1452335-1452341.pdf>>.

[turkey-googledns]

Bortzmeyer, S., "Hijacking of public DNS servers in Turkey, through routing", 2014,
<<http://www.bortzmeyer.org/dns-routing-hijack-turkey.html>>.

[data-protection-directive]

Europe, , "European directive 95/46/EC on the protection of individuals with regard to the processing of personal data and on the free movement of such data", November 1995, <<http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:EN:HTML>>.

[passive-dns]

Weimer, F., "Passive DNS Replication", April 2005,
<<http://www.enyo.de/fw/software/dnslogger/#2>>.

[tor-leak]

Tor, , "DNS leaks in Tor", 2013,
<<https://trac.torproject.org/projects/tor/wiki/doc/TorFAQ#IkeepseeingthesewarningsaboutSOCKSandDNSandinformationleaks.ShouldIworry>>.

[yanbin-tsudik]

Yanbin, L. and G. Tsudik, "Towards Plugging Privacy Leaks in the Domain Name System", 2009,
<<http://arxiv.org/abs/0910.2472>>.

[castillo-garcia]

Castillo-Perez, S. and J. Garcia-Alfaro, "Anonymous Resolution of DNS Queries", 2008,
<<http://deic.uab.es/~joaquin/papers/is08.pdf>>.

[fangming-hori-sakurai]

Fangming, , Hori, Y., and K. Sakurai, "Analysis of Privacy Disclosure in DNS Query", 2007,
<<http://dl.acm.org/citation.cfm?id=1262690.1262986>>.

[federrath-fuchs-herrmann-piosecn]

Federrath, H., Fuchs, K., Herrmann, D., and C. Piosecny,
"Privacy-Preserving DNS: Analysis of Broadcast, Range
Queries and Mix-Based Protection Methods", 2011,
<https://svs.informatik.uni-hamburg.de/publications/2011/2011-09-14_FFHP_PrivacyPreservingDNS_ESORICS2011.pdf>.

[aeris-dns]

Vinot, N., "[In French] Vie privee : et le DNS alors ?",
2015, <<https://blog.imirhil.fr/vie-privee-et-le-dns-alors.html>>.

8.3. URIs

[1] <https://developers.google.com/speed/public-dns/privacy>

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/>

