

dprive
Internet-Draft
Intended status: Experimental
Expires: January 3, 2019

A. Edmundson
P. Schmitt
N. Feamster
Princeton University
A. Mankin
Salesforce
July 2, 2018

Oblivious DNS - Strong Privacy for DNS Queries
draft-annee-dprive-oblivious-dns-00

Abstract

Recognizing the privacy vulnerabilities associated with DNS queries, a number of standards have been developed and services deployed that that encrypt a user's DNS queries to the recursive resolver and thus obscure them from some network observers and from the user's Internet service provider. However, these systems merely transfer trust to a third party. We argue that no single party should be able to associate DNS queries with a client IP address that issues those queries. To this end, this document specifies Oblivious DNS (ODNS), which introduces an additional layer of obfuscation between clients and their queries. To accomplish this, ODNS uses its own authoritative namespace; the authoritative servers for the ODNS namespace act as recursive resolvers for the DNS queries that they receive, but they never see the IP addresses for the clients that initiated these queries. The ODNS experimental protocol is compatible with existing DNS infrastructure.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 3, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Terminology	2
2. Introduction	3
3. ODNS Overview	4
4. Sending and Receiving ODNS Queries	5
5. Replication and Privacy-Preserving Key Distribution	6
5.1. Scalability and Performance Using Anycast	6
5.2. Key Distribution	6
5.3. QNAME Length	7
6. Backward Compatibility	7
7. IANA considerations	8
8. Security considerations	8
9. Acknowledgements	8
10. Contributors	8
11. Changelog	8
12. References	8
12.1. Normative References	8
12.2. Informative References	9
Authors' Addresses	10

1. Terminology

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

Privacy terminology is as described in Section 3 of [RFC6973].

DNS terminology is as described in [I-D.ietf-dnsop-terminology-bis] with one modification: we use the definition of Privacy-enabling DNS server taken from [RFC8310]:

2. Introduction

Recognizing the privacy vulnerabilities associated with DNS queries, a number of specifications and services have been developed that encrypt a user's DNS queries to the recursive resolver and thus obscure them from some network observers and from the user's Internet service provider. However, these systems merely transfer trust to a third party. We argue that no single party should be able to associate DNS queries with a client IP address that issues those queries, that there should be obfuscation between the client and its queries.

DNS queries can reveal significant information about the Internet destinations that a user or device is communicating with. For example, the domain names themselves may reveal the websites that a user is visiting. In the case of smart-home Internet of Things (IoT) devices, the DNS queries may reveal the types of devices in user homes. Previous work has also demonstrated that DNS lookups can identify the websites that a user is visiting, even when they are using an anonymizing service such as Tor [Tor-DNS]. The operator of a DNS resolver may also retain information about DNS queries and responses---including the IP addresses that query the domains and the DNS names that are queried.

Other approaches have layered encryption on top of the DNS query stream. For example, DNS-over-TLS [RFC7858], DNS-over-DTLS [RFC8094], and DNS-over-HTTPS [I-D.ietf-doh-dns-over-https] all send DNS queries over an encrypted channel, which prevents an eavesdropper from learning the contents of a DNS lookup but does not prevent the operator of the recursive resolver from linking queries and IP addresses. DNSCurve (ref to be added) uses elliptic curve cryptography to encrypt DNS requests and responses; it also authenticates all DNS responses and eliminates any forged responses. DNSCrypt (ref to be added) encrypts and authenticates DNS traffic between a client and a recursive resolver. None of the approaches prevent the recursive resolver from observing DNS queries and responses. Note: a new draft is under development, targeted to for BCP, that would offer a policy and best-practices approach to the problem of recursive resolvers's observation of this data.

ODNS (1) obfuscates the queries that a recursive resolver sees from the clients that issue DNS queries; and (2) obfuscates the client's IP address from upper levels of the DNS hierarchy that ultimately resolve the query (that is, the authoritative servers). ODNS

operates in the context of the existing DNS protocol, allowing the existing deployed infrastructure to remain unchanged. A client sends an encrypted query to a recursive resolver, which then forwards the query to an authoritative DNS server that can resolve ODNS queries. The recursive resolver never sees the DNS domain that the client queries, and the ODNS server never sees the IP address of the client.

ODNS requires a modified client stub resolver, and a modified authoritative DNS server. The stub resolver must take an existing DNS name, encrypt it, and append the ODNS domain to ensure that the query is forwarded to the ODNS authoritative DNS server. The authoritative DNS server for ODNS must also act as a recursive DNS resolver; it must not only reply for the ODNS namespace but also ultimately retrieve the DNS record that corresponds to the client's initial query.

3. ODNS Overview

ODNS operates similarly to conventional DNS, but adds two components: (1) each client runs a modified stub resolver; and (2) ODNS runs an authoritative name server that also acts as a recursive DNS resolver for the original DNS query:

- o The client's stub resolver obfuscates the domain that the client is requesting (via symmetric encryption), resulting in the client's configured recursive resolver being unaware of the requested domain.
- o The authoritative name server for ODNS separates the clients' identities from their corresponding DNS requests, such that the name servers cannot learn who is requesting specific domains.

As detailed in [RFC7626], operators of recursive DNS resolvers see individual IP addresses along with the fully qualified domain name those IPs request. Operators of authoritative resolvers may also be able to learn information about the client by using one of the extensions to DNS, notably EDNS0 Client Subnet (ECS) [RFC7871]. ECS can reveal information about the user's IP address or subnet to authoritative DNS servers higher in the DNS hierarchy (not only recursive DNS resolvers). ODNS hides a client's IP address from the authoritative name servers at different levels of the DNS hierarchy.

The configured (non-ODNS) recursive DNS resolver knows the client IP address but never sees the domain that it queries. ODNS requires the client to use a custom local stub resolver, which hides the requested domain from the recursive resolver. The ODNS stub resolver, which runs at the client, encrypts the original DNS query for the ODNS authoritative DNS server before it appends the domain for the ODNS

namespace to the query, which causes the recursive resolver to forward the encrypted domain name on to the ODNS authoritative server. NOTE: for simplicity, we sometimes say that this authoritative server is for .odns, but any authoritative DNS domain can run an ODNS server. Even if there was a TLD, there would be leakage of information, because the IP addresses of clients and recursive resolvers would be seen at the root. Experiments can be done to avoid leakage about queries of this nature through adaptation of [RFC7706].

When an ODNS authoritative DNS server receives a DNS query, it removes any client information from the request (e.g., the client IP address, EDNS0 client subnet information) before performing additional DNS lookups. The ODNS name server then switches to acting as a recursive resolver. The authoritative server forwards any response to the original recursive DNS resolver, which in turn sends the response to the client.

The recursive DNS resolver receives the request from the client, but cannot identify the genuine domain. It parses the TLD (.odns) and forwards the request onto the .odns authoritative server. Because the session key was originally encrypted with the authoritative server's public key, the authoritative server can decrypt the session key with its private key, and subsequently decrypt the domain with the session key. The authoritative server then acts as a recursive resolver and contacts the necessary name servers to resolve the domain. Once an answer is obtained, the authoritative server encrypts the domain with the session key, appends the .odns TLD and forwards the response to the recursive DNS resolver. As explained by the use of session keys, the recursive resolver cannot learn the domains a client requests, despite being able to learn who the client is.

TODO (in -01 or later): Create an ASCII diagram form of Figure 1 from odns.cs.princeton.edu

4. Sending and Receiving ODNS Queries

TODO (in -01 or later): Create an ASCII diagram form of Figure 2 from odns.cs.princeton.edu

- o When a client generates a DNS request, the local stub resolver generates a symmetric session key, encrypts the domain name with the session key, encrypts the session key with the authoritative server's public key, and appends the .odns TLD to the encrypted domain. (www.example.com_k.odns.) The stub also appends the session key encrypted under the ODNS authoritative server's public key k_PK)

- o The client sends the query in the Additional Information portion of the DNS query to the recursive resolver, which then sends it to the authoritative name server for ODNS.
- o The authoritative server for ODNS queries decrypts the session key, which it uses to decrypt the domain in the query.
- o The authoritative server forwards a recursive DNS request to the appropriate name server for the original domain, which then returns the answer to the ODNS server.
- o The ODNS server returns the answer to the client's recursive resolver.

Other authoritative DNS servers see incoming DNS requests, but these only see the IP address of the ODNS authoritative resolver, which effectively proxies the DNS request for the original client. The client's original recursive resolver can learn the client's IP address, but cannot learn the domain names in the client's DNS queries.

5. Replication and Privacy-Preserving Key Distribution

5.1. Scalability and Performance Using Anycast

To achieve scalability the authoritative server is replicated in a variety of geographical locations and all replicas are assigned to both an anycast IP address as well as a unique unicast IP address. Using anycast, all servers that share the IP address are able to answer a query. When a recursive sends a DNS query to the ODNS authoritative server, the query will be routed by BGP to the ``nearest'' authoritative server. And because the recursive resolver (an open resolver) is also anycast, both the recursive and the ODNS authoritative server should be the optimal choices based on the client's network connectivity {\it without revealing the client's location}. This results in maximizing the performance of ODNS by minimizing the network path that queries must traverse.

5.2. Key Distribution

Use of anycast and multiple authoritative replicas introduce a key distribution challenge for ODNS. The ODNS stub server uses the public key of the authoritative server to encrypt session keys in ODNS queries. Based on best practices, we cannot share public / private keypairs across all of the replicated authoritative servers. Likewise, in order to preserve user identity privacy the key distribution must be done in a way that the authoritative server never learns the identity (i.e., IP address) of a stub. This

disqualifies out-of-band key exchange as in EncDNS. Instead, we leverage the DNS infrastructure itself to distribute keys while maintaining privacy. We have defined a ``special'' query (e.g., special.odns) that we use to select a specific authoritative server as well as distribute the appropriate public key.

The client's stub resolver sends a special ODNS query to the recursive resolver, which will in turn use the anycast address to locate the nearest authoritative server. The authoritative that receives the query responds with an OPT record that includes a self-certifying name (e.g., ABC.odns), such that the name of the server is derived from the public key itself and is associated with an instance of the authoritative nameserver listening on the unique unicast IP address, and the authoritative server's public key; this response is returned to the client's stub resolver via the recursive. Subsequent ODNS queries at the stub append the unique name of the authoritative that responded to the special query, which means that the requests will all reach the same server and the client encrypt using the appropriate public key.

5.3. QNAME Length

In principle, a query could include the encrypted query and / or session key in a special Resource Record (RR) in the ``Additional Information'' section of a DNS message (known as an OPT), but we discovered that, in practice, most open resolvers strip all OPT records before forwarding the query on to the authoritative nameserver. In this case, ODNS cannot simply use an OPT to communicate the session key. ODNS overcomes this challenge by placing the encrypted key in the QNAME field of the DNS message; the QNAME field consists of 4 sets of 63 bytes, which limits both the key size and encryption scheme used. For this reason, ODNS uses 16-byte AES session keys and encrypts the session keys using the Elliptic Curve Integrated Encryption Scheme (ECIES)~. Once the session key is encrypted, the resulting value takes up 44 bytes of the QNAME field. In the future, we envision an ODNS-specific OPT code that would cause recursive resolvers to maintain and forward the ODNS OPT record intact to the authoritative nameserver. Such a mechanism allows for the use of larger encryption keys as OPT records can be much larger (typically 4096 bytes) than the space allotted for QNAMEs.

6. Backward Compatibility

For a new extension to DNS such as ODNS to be widely adopted it must be backward-compatible with existing infrastructure, as changes to the DNS system occur over long time scales. Our design must not rely upon changes made at recursive resolvers, root nameservers, or TLD nameservers. We engineer the ODNS stub and authoritative

functionality with this in mind as these two locations in the DNS hierarchy are readily controlled.

7. IANA considerations

For initial experimental deployment of this protocol, the name `obliviousdns.com` has been registered. Its length is a drawback, for the reasons discussed in Section 5.3 and a shorter privately registered name may be chosen for future larger-scale experimentation. An infrastructure related zone would be more advantageous choice. Therefore discussion should resolve the appropriateness and conditions of a request for a special use domain name, e.g. `odns.arpa`. This falls under the considerations in [RFC3172]. Notes: because of restrictions on TLD registration, following the example of `.onion` [RFC7686] is infeasible. Traffic for ODNS traverses normal Internet paths, therefore the IANA special use registry recently established for Locally-Served DNS Zones, in which `home.arpa` has recently been registered [RFC8375], is also not a model for IANA considerations for the ODNS Namespace.

8. Security considerations

TODO (some questions to consider): what are residual risks in the ODNS scheme and additional mitigations? Is there any increase in attack surface for the users and operators in ODNS? Are systems depending on ODNS vulnerable to DoS in specific ways that should be mitigated?

9. Acknowledgements

10. Contributors

The following contributed significantly to the document:

11. Changelog

`draft-annee-dprive-oblivious-dns-00`

- o Initial commit

12. References

12.1. Normative References

[I-D.ietf-dnsop-terminology-bis]
Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", draft-ietf-dnsop-terminology-bis-10 (work in progress), April 2018.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3172] Huston, G., Ed., "Management Guidelines & Operational Requirements for the Address and Routing Parameter Area Domain ("arpa")", BCP 52, RFC 3172, DOI 10.17487/RFC3172, September 2001, <<https://www.rfc-editor.org/info/rfc3172>>.
- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <<https://www.rfc-editor.org/info/rfc6973>>.
- [RFC7871] Contavalli, C., van der Gaast, W., Lawrence, D., and W. Kumari, "Client Subnet in DNS Queries", RFC 7871, DOI 10.17487/RFC7871, May 2016, <<https://www.rfc-editor.org/info/rfc7871>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8310] Dickinson, S., Gillmor, D., and T. Reddy, "Usage Profiles for DNS over TLS and DNS over DTLS", RFC 8310, DOI 10.17487/RFC8310, March 2018, <<https://www.rfc-editor.org/info/rfc8310>>.

12.2. Informative References

- [I-D.ietf-doh-dns-over-https] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", draft-ietf-doh-dns-over-https-12 (work in progress), June 2018.
- [RFC7626] Bortzmeyer, S., "DNS Privacy Considerations", RFC 7626, DOI 10.17487/RFC7626, August 2015, <<https://www.rfc-editor.org/info/rfc7626>>.
- [RFC7686] Appelbaum, J. and A. Muffett, "The ".onion" Special-Use Domain Name", RFC 7686, DOI 10.17487/RFC7686, October 2015, <<https://www.rfc-editor.org/info/rfc7686>>.

- [RFC7706] Kumari, W. and P. Hoffman, "Decreasing Access Time to Root Servers by Running One on Loopback", RFC 7706, DOI 10.17487/RFC7706, November 2015, <<https://www.rfc-editor.org/info/rfc7706>>.
- [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>>.
- [RFC8375] Pfister, P. and T. Lemon, "Special-Use Domain 'home.arpa.'", RFC 8375, DOI 10.17487/RFC8375, May 2018, <<https://www.rfc-editor.org/info/rfc8375>>.
- [Tor-DNS] Reschbach, G., Pulls, B., Roberts, L., Winter, P., and N. Feamster, "The Effect of DNS on Tor's Anonymity", 2016.

Authors' Addresses

Annie Edmundson
Princeton University
Princeton, NJ
United States

Email: annee@cs.princeton.edu

Paul Schmitt
Princeton University
Princeton, NJ
United States

Email: pschmitt@cs.princeton.edu

Nick Feamster
Princeton University
Princeton, NJ
United States

Email: nfeamster@cs.princeton.edu

Allison Mankin
Salesforce

Email: allison.mankin@gmail.com

DNS Privacy (dprive) Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 21, 2018

S. Bortzmeyer
AFNIC
March 20, 2018

Encryption and authentication of the DNS resolver-to-authoritative
communication
draft-bortzmeyer-dprive-resolver-to-auth-01

Abstract

This document proposes a mechanism for securing (privacy-wise) the communication between the DNS resolver and the authoritative name server.

REMOVE BEFORE PUBLICATION: this document should be discussed in the IETF DPRIVE group, through its mailing list. The source of the document, as well as a list of open issues, is currently kept at Github [1].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 21, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction and background	2
2. Rules	3
3. Operational considerations	4
4. IANA Considerations	5
5. Security Considerations	5
6. References	5
6.1. Normative References	5
6.2. Informative References	6
6.3. URIs	7
Appendix A. Acknowledgments	7
Appendix B. Alternatives	7
Author's Address	7

1. Introduction and background

To improve the privacy of the DNS user ([RFC7626]), the standard solution is to encrypt the requests with TLS ([RFC7858]). We use this DNS-over-TLS solution as well here, since it is standardized, already implemented in many programs, and relies on a well-known security protocol (inventing a new security protocol is quite dangerous). But just encrypting, without authenticating the remote server, leaves the user's privacy vulnerable to active man-in-the-middle attacks. [RFC7858] and [I-D.ietf-dprive-dtls-and-tls-profiles] describe how to authenticate the DNS resolver, in the stub-to-resolver link. We describe here authentication of the authoritative name server, in the resolver-to-authoritative link.

A stub DNS resolver has only a few resolvers, and there is typically a pre-existing relationship. But a resolver speaks to many authoritative name servers, without any prior relationship. This means that, for instance, having a static key for the resolver makes sense while it would be clearly unrealistic for the authoritative server.

Instead, we rely on DANE ([RFC6698]). Authoritative name servers are known by name (obtained from zone delegation). The manager of the ns1.example.net name server adds a TLSA record under example.net. The client establishes the TLS session, then authenticates in the normal DANE way.

The original charter of the DPRIVE working group, in force at the time of this draft, says "The primary focus of this Working Group is to develop mechanisms that provide confidentiality between DNS Clients and Iterative Resolvers" and adds "but it may also later consider mechanisms that provide confidentiality between Iterative Resolvers and Authoritative Servers". This document is here for this second step, "between Iterative Resolvers and Authoritative Servers". It will probably require a rechartering of the group.

2. Rules

A DNS full-service resolver who needs to query an authoritative name server establishes a TLS-over-TCP session with this authoritative name server. If the DNS material to perform DANE authentication is sent in the TLS session ([I-D.ietf-tls-dnssec-chain-extension]), it uses it. Otherwise, the resolver queries TLSA records ([RFC6698]) for this name server and authenticates the key or certificate of the server this way. If the name server is ns1.example.net, the TLSA record to query is _853._tcp.ns1.example.net.

Note that the server MAY use raw public keys ([RFC7250]) and so there is not always a certificate. If the server uses raw public keys, the TLSA record's Selector field must be 1 (SPKI, SubjectPublicKeyInfo).

The recommended order is to try TLS before querying the TLSA records. True, DANE signals if the server is willing to make DNS-over-TLS (and can therefore save a TLS attempt) but cannot guarantee that it will work (for instance if a middlebox blocks port 853). Also, the DANE records may be transferred in the TLS session, not through the DNS.

If the TLS session establishment fails, or if the DANE authentication fails, the result depends on whether the resolver runs in strict or opportunistic mode ([I-D.ietf-dprive-dtls-and-tls-profiles]). In strict mode, the resolver MUST stop using this authoritative name server, and MUST try other servers of the DNS zone. In opportunistic mode, the resolver MUST use the authoritative name server despite the failure. It MAY try other name servers of the zone before, in the hope they will accept TLS and be authenticated. To avoid a chicken-and-egg problem, the resolver, even in strict mode, MAY use unsecure servers for the meta-queries (getting the TLSA records). More specifically:

- (0) The resolver remembers the keys of the authoritative name servers (in the same way it remembers the lowest RTT among an NS RRset),

(1) When the resolver needs to talk to a server (say ns2.example.net) for which it does not know the key, it does a TLSA request for _853._tcp.ns2.example.net,

(2) If the resolution of this request requires that we talk to the same server for which we're searching for the TLSA record, the resolver connects to this server with TLS to port 853, does not bother to authenticate, and sends the query. This step offers no authentication.

(See also [I-D.ietf-dprive-dtls-and-tls-profiles], section 5.) A resolver MAY use the knowledge of TLS authentication it has to choose an authoritative name server among a NS RRset.

As of this revision, we do not expect resolvers to use strict mode, since the encryption and authentication modes described in this document are not yet supported in authoritative name servers.

3. Operational considerations

DNS-over-TLS depends on TCP, and the resolver and the authoritative name server must therefore support persistent TCP connections ([RFC7766], specially section 6.2.1).

A resolver may have a lot of client-side state, when managing hundreds of connections to remote authoritative servers ([tdns]).

The latency when connecting to a authoritative name server is certainly an issue. TLS 1.3 and TCP Fast Open ([RFC7413]) may help.

Open question: do we require a minimum TLS version of 1.3? ([I-D.ietf-tls-tls13])

Because the resolver cannot know in advance if the TLS connection will work (even if there is a DANE record), using parallel attempts ("happy eyeballs", [RFC8305]) is important. A resolver working in opportunistic mode should try ports 53 and 853 in parallel.

An authoritative name server cannot know if the resolver authenticated it, nor how. In the future, it may be interesting to have an EDNS option to signal a successful authentication, or a failure, but this is out of scope currently.

If it is a concern that the same authoritative name servers are used for ordinary DNS and for encrypted DNS, there are several ways to address this concern. A server operator may use front-end systems dispatching requests to ports 53 and 853 to different servers.

A resolver must be configurable to operate in strict or opportunistic modes. Until the features described herein are widely supported, opportunistic mode should not be the default since strict mode would yield frequent failures. A resolver may have a configuration mechanism to be in strict mode only for some domains.

4. IANA Considerations

No action for IANA. This section can be deleted.

5. Security Considerations

The state to be kept in both the client and the server may make some denial-of-service attacks easier. Following the advice contained in section 10 of [RFC7766] is recommended.

In opportunistic mode, there is no guarantee to have a secure use of the DNS, or even a guarantee to be informed of a problem. Opportunistic mode is a "best effort" privacy service. Even in strict mode, some leaks may occur, through the DANE meta-queries, and through SNI indication ([I-D.ietf-tls-sni-encryption]) in the TLS session.

Neither transport encryption nor authentication protect DNS users from authentic servers which nonetheless abuse users' privacy once they've received their queries. These techniques must therefore be combined with data minimization techniques ([RFC7816]).

6. References

6.1. Normative References

- [I-D.ietf-dprive-dtls-and-tls-profiles]
Dickinson, S., Gillmor, D., and T. Reddy, "Usage and (D)TLS Profiles for DNS-over-(D)TLS", draft-ietf-dprive-dtls-and-tls-profiles-11 (work in progress), September 2017.
- [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <<https://www.rfc-editor.org/info/rfc6698>>.
- [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>>.

6.2. Informative References

- [I-D.bortzmeyer-dprive-step-2]
Bortzmeyer, S., "Next step for DPRIVE: resolver-to-auth link", draft-bortzmeyer-dprive-step-2-05 (work in progress), December 2016.
- [I-D.ietf-tls-dnssec-chain-extension]
Shore, M., Barnes, R., Huque, S., and W. Toorop, "A DANE Record and DNSSEC Authentication Chain Extension for TLS", draft-ietf-tls-dnssec-chain-extension-06 (work in progress), January 2018.
- [I-D.ietf-tls-sni-encryption]
Huitema, C. and E. Rescorla, "SNI Encryption in TLS Through Tunneling", draft-ietf-tls-sni-encryption-02 (work in progress), March 2018.
- [I-D.ietf-tls-tls13]
Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", draft-ietf-tls-tls13-27 (work in progress), March 2018.
- [RFC7250] Wouters, P., Ed., Tschofenig, H., Ed., Gilmore, J., Weiler, S., and T. Kivinen, "Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", RFC 7250, DOI 10.17487/RFC7250, June 2014, <<https://www.rfc-editor.org/info/rfc7250>>.
- [RFC7413] Cheng, Y., Chu, J., Radhakrishnan, S., and A. Jain, "TCP Fast Open", RFC 7413, DOI 10.17487/RFC7413, December 2014, <<https://www.rfc-editor.org/info/rfc7413>>.
- [RFC7626] Bortzmeyer, S., "DNS Privacy Considerations", RFC 7626, DOI 10.17487/RFC7626, August 2015, <<https://www.rfc-editor.org/info/rfc7626>>.
- [RFC7766] Dickinson, J., Dickinson, S., Bellis, R., Mankin, A., and D. Wessels, "DNS Transport over TCP - Implementation Requirements", RFC 7766, DOI 10.17487/RFC7766, March 2016, <<https://www.rfc-editor.org/info/rfc7766>>.
- [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>>.

- [RFC8305] Schinazi, D. and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency", RFC 8305, DOI 10.17487/RFC8305, December 2017, <<https://www.rfc-editor.org/info/rfc8305>>.
- [tdns] Liang, Z., Wessels, D., Zi, H., Heidemann, J., Mankin, A., and N. Somaiya, "T-DNS: Connection-Oriented DNS to Improve Privacy and Security; USC/ISI Technical Report ISI-TR-706", August 2014, <<http://www.isi.edu/~johnh/PAPERS/Zhu16b.pdf>>.

6.3. URIs

- [1] <https://github.com/bortzmeyer/ietf-dprive-step-2>

Appendix A. Acknowledgments

Thanks to Bill Woodcock for a detailed review.

Appendix B. Alternatives

A number of other possible solutions to this problem may be found in in [I-D.bortzmeyer-dprive-step-2].

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/>

dprive
Internet-Draft
Obsoletes: 7626 (if approved)
Intended status: Informational
Expires: July 19, 2019

S. Bortzmeyer
AFNIC
S. Dickinson
Sinodun IT
January 15, 2019

DNS Privacy Considerations
draft-bortzmeyer-dprive-rfc7626-bis-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. This document obsoletes RFC 7626.

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 July 19, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<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	5
2.2. Data in the DNS Request	5
2.2.1. Data in the DNS payload	7
2.3. Cache Snooping	7
2.4. On the Wire	7
2.4.1. Unencrypted Transports	7
2.4.2. Encrypted Transports	9
2.5. In the Servers	10
2.5.1. In the Recursive Resolvers	10
2.5.2. In the Authoritative Name Servers	12
2.5.3. Rogue Servers	13
2.5.4. Authentication of servers	13
2.5.5. Blocking of services	14
2.6. Re-identification and Other Inferences	14
2.7. More Information	15
3. Actual "Attacks"	15
4. Legalities	15
5. Security Considerations	16
6. Acknowledgments	16
7. Changelog	16
8. References	17
8.1. Normative References	17
8.2. Informative References	17
8.3. URIs	22
Authors' Addresses	22

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], [RFC1035], and many later RFCs, which have never been consolidated. It is one of the most important infrastructure components of the Internet and often ignored or misunderstood by Internet users (and even by many professionals). 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. (See also [RFC8499]) 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 description that follows assumes a cold cache, for instance, because the server just started.) The recursive resolver will first query the root name servers. In most cases, the root name servers will send a referral. In this example, the referral will be to the `.com` name servers. The resolver repeats the query to one of the `.com` name servers. The `.com` name servers, in turn, will refer to the `example.com` name servers. The `example.com` name server 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 name server.

Because DNS relies on caching heavily, the algorithm described just above is actually a bit more complicated, and not all questions are sent to the authoritative name servers. If a few seconds later the stub resolver asks 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 that does not follow DNS rules, for instance, because it may ignore the TTL. 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).

Almost all this DNS traffic is currently sent in clear (unencrypted). At the time of writing there is increasing deployment of DNS-over-TLS [RFC7858] and work underway on DoH [RFC8484]. There are a few cases where there is some alternative channel encryption, for instance, in an IPsec VPN, at least between the stub resolver and the resolver.

Today, almost all DNS queries are sent over UDP [thomas-ditl-tcp]. 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 fact that DNS requests received by a server are triggered by different reasons. Let's assume an eavesdropper wants to know which web page is viewed by a 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, Cascading Style Sheets (CSS), 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 the 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 than strictly necessary are sent, for instance, to prefetch resources that the user may query later or when autocompleting the URL in the address bar. Both are a big privacy concern since they may leak information even about non-explicit actions. For instance, just reading a local HTML page, even without selecting the hyperlinks, may trigger DNS requests.

For privacy-related terms, we will use the terminology from [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 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 a 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 differentiate between several users sharing an IP address (behind a Carrier-Grade NAT (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 is therefore very revealing about communication relationships). Some QNAMEs are more sensitive than others. For instance, querying the A record of a well-known web statistics domain reveals very little (everybody visits web sites that use this analytics service), but querying the A

record of `www.verybad.example` where `verybad.example` is the domain of an organization that some people find offensive or objectionable 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 an 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 Mail User Agent (MUA) starts looking up Pretty Good Privacy (PGP) keys in the DNS [RFC7929], 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 an HTTP connection. It is now the IP address of the recursive resolver that, in a way, "hides" the real user. However, hiding does not always work. Sometimes EDNS(0) Client subnet [RFC7871] 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 [sidn-entrada].

A note about IP addresses: there is currently no IETF document that 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.

2.2.1. Data in the DNS payload

At the time of writing there are no standardized client identifiers contained in the DNS payload itself (ECS [RFC7871] while widely used is only of Category Informational).

DNS Cookies [RFC7873] are a lightweight DNS transaction security mechanism that provides limited protection against a variety of increasingly common denial-of-service and amplification/forgery or cache poisoning attacks by off-path attackers. It is noted, however, that they are designed to just verify IP addresses (and should change once a client's IP address changes), they are not designed to actively track users (like HTTP cookies).

There are anecdotal accounts of MAC addresses [1] and even user names being inserted in non-standard EDNS(0) options for stub to resolver communications to support proprietary functionality implemented at the resolver (e.g. parental filtering).

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 [grangeia.snooping]. 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

2.4.1. Unencrypted Transports

For unencrypted transports, DNS traffic can be seen by an eavesdropper like any other traffic. (DNSSEC, specified in [RFC4033], explicitly excludes confidentiality from its goals.) So, if an initiator starts an 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 (e.g. [RFC8446], [I-D.ietf-quic-transport]), the use of unencrypted transports for DNS may become "the weakest link" in privacy. It is noted that there is on-going work attempting to encrypt the SNI in the TLS handshake but that this is a non-trivial problem [I-D.ietf-tls-sni-encryption].

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 Internet Access Provider (IAP) network referenced below.

The recursive resolver can be in the IAP 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 today 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.

2.4.2. Encrypted Transports

The use of encrypted transports directly mitigates passive surveillance of the DNS payload, however there are still some privacy attacks possible.

These are cases where user identification, fingerprinting or correlations may be possible due to the use of certain transport layers or clear text/observable features. These issues are not specific to DNS, but DNS traffic is susceptible to these attacks when using specific transports.

There are some general examples, for example, certain studies have highlighted that IP TTL or TCP Window sizes os-fingerprint [2] values can be used to fingerprint client OS's or that various techniques can be used to de-NAT DNS queries dns-de-nat [3].

The use of clear text transport options to decrease latency may also identify a user e.g. using TCP Fast Open [RFC7413].

More specifically, (since the deployment of encrypted transports is not widespread at the time of writing) users wishing to use encrypted transports for DNS may in practice be limited in the resolver services available. Given this, the choice of a user to configure a single resolver (or a fixed set of resolvers) and an encrypted transport to use in all network environments can actually serve to identify the user as one that desires privacy and can provide an added mechanism to track them as they move across network environments.

Users of encrypted transports are also highly likely to re-use sessions for multiple DNS queries to optimize performance (e.g. via DNS pipelining or HTTPS multiplexing). Certain configuration options for encrypted transports could also in principle fingerprint a user, for example session resumption, the maximum number of messages to send or a maximum connection time before closing a connections and re-opening.

Whilst there are known attacks on older versions of TLS the most recent recommendations [RFC7525] and developments [RFC8446] in this area largely mitigate those.

Traffic analysis of unpadded encrypted traffic is also possible [pitfalls-of-dns-encrption] because the sizes and timing of encrypted DNS requests and responses can be correlated to unencrypted DNS requests upstream of a recursive resolver.

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] [packetq-list] and DNSmezzo [dnsmezzo]. The organization managing the DNS server can use this data itself, or it can be part of a surveillance program like PRISM [prism] and pass data to an outside observer.

Sometimes, this data is kept for a long time and/or distributed to third parties for research purposes [ditl] [day-at-root], security analysis, or surveillance tasks. These uses are sometimes under some sort of contract, with various limitations, for instance, on redistribution, given the sensitive nature of the data. Also, there are observation points in the network that 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, e.g., <<https://developers.google.com/speed/public-dns/privacy>>.

2.5.1.1. Encrypted transports

Use of encrypted transports does not reduce the data available in the recursive resolver and ironically can actually expose more information about users to operators. As mentioned in Section 2.4 use of session based encrypted transports (TCP/TLS) can expose correlation data about users. Such concerns in the TCP/TLS layers apply equally to DNS-over-TLS and DoH which both use TLS as the underlying transport.

2.5.1.2. DoH vs DNS-over-TLS

The proposed specification for DoH [RFC8484] includes a Privacy Considerations section which highlights some of the differences between HTTP and DNS. As a deliberate design choice DoH inherits the privacy properties of the HTTPS stack and as a consequence introduces new privacy concerns when compared with DNS over UDP, TCP or TLS [RFC7858]. The rationale for this decision is that retaining the ability to leverage the full functionality of the HTTP ecosystem is more important than placing specific constraints on this new protocol based on privacy considerations (modulo limiting the use of HTTP cookies).

In analyzing the new issues introduced by DoH it is helpful to recognize that there exists a natural tension between

- o the wide practice in HTTP to use various headers to optimize HTTP connections, functionality and behaviour (which can facilitate user identification and tracking)
- o and the fact that the DNS payload is currently very tightly encoded and contains no standardized user identifiers.

DNS-over-TLS, for example, would normally contain no client identifiers above the TLS layer and a resolver would see only a stream of DNS query payloads originating within one or more connections from a client IP address. Whereas if DoH clients commonly include several headers in a DNS message (e.g. user-agent and accept-language) this could lead to the DoH server being able to identify the source of individual DNS requests not only to a specific end user device but to a specific application.

Additionally, depending on the client architecture, isolation of DoH queries from other HTTP traffic may or may not be feasible or desirable. Depending on the use case, isolation of DoH queries from other HTTP traffic may or may not increase privacy.

The picture for privacy considerations and user expectations for DoH with respect to what additional data may be available to the DoH server compared to DNS over UDP, TCP or TLS is complex and requires a detailed analysis for each use case. In particular the choice of HTTPS functionality vs privacy is specifically made an implementation choice in DoH and users may well have differing privacy expectations depending on the DoH use case and implementation.

At the extremes, there may be implementations that attempt to achieve parity with DNS-over-TLS from a privacy perspective at the cost of using no identifiable headers, there might be others that provide

feature rich data flows where the low-level origin of the DNS query is easily identifiable.

Privacy focussed users should be aware of the potential for additional client identifiers in DoH compared to DNS-over-TLS and may want to only use DoH implementations that provide clear guidance on what identifiers they add.

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 typically has 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 ECS [RFC7871] 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 50,000 queries per second. While most of it is "junk" (errors on the Top-Level Domain (TLD) name), it gives an idea of the amount of big data that pours into name servers. (And even "junk" can leak information; for instance, if there is a typing error in the TLD, the user will send data to a TLD that is not the usual one.)

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. So, requests to a given ccTLD may go to servers managed by organizations outside of the ccTLD's country. End users may not anticipate that, when doing a security analysis.

Also, it seems (see the survey described in [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

The previous paragraphs discussed DNS privacy, assuming that all the traffic was directed to the intended servers and that the potential attacker was purely passive. But, in reality, we can have active attackers redirecting the traffic, not to change it but just to observe it.

For instance, 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. Other attacks, besides using DHCP, are possible. The traffic from a DNS client to a DNS server can be intercepted along its way from originator to intended source, for instance, by transparent DNS proxies in the network that will divert the traffic intended for a legitimate DNS server. This rogue server can masquerade as the intended server and respond with data to the client. (Rogue servers that inject malicious data are possible, but it is a separate problem not relevant to privacy.) A rogue server may respond correctly for a long period of time, thereby foregoing detection. This may be done for what could be claimed to be good reasons, such as optimization or caching, but it leads to a reduction of privacy compared to if there was no attacker present. Also, malware like DNSChanger [dnschanger] can change the recursive resolver in the machine's configuration, or the routing itself can be subverted (for instance, [ripe-atlas-turkey]).

2.5.4. Authentication of servers

Both Strict mode for DNS-over-TLS and DoH require authentication of the server and therefore as long as the authentication credentials are obtained over a secure channel then using either of these

transports defeats the attack of re-directing traffic to rogue servers. Of course attacks on these secure channels are also possible, but out of the scope of this document.

2.5.5. Blocking of services

User privacy can also be at risk if there is blocking (by local network operators or more general mechanisms) of access to recursive servers that offer encrypted transports. For example active blocking of port 853 for DNS-over-TLS or of specific IP addresses (e.g. 1.1.1.1 or 2606:4700:4700::1111) could restrict the resolvers available to the client. Similarly attacks on such services e.g. DDoS could force users to switch to other services that do not offer encrypted transports for DNS.

2.6. Re-identification and Other Inferences

An observer has access not only to the data he/she directly collects but also to the results of various inferences about this data.

For instance, a user can be re-identified via DNS queries. If the adversary knows a user's identity and can watch their DNS queries for a period, then that same adversary may be able to re-identify the user solely based on their pattern of DNS queries later on regardless of the location from which the user makes those queries. For example, one study [herrmann-reidentification] found that such re-identification is possible so that "73.1% of all day-to-day links were correctly established, i.e. user u was either re-identified unambiguously (1) or the classifier correctly reported that u was not present on day t+1 any more (2)." While that study related to web browsing behavior, equally characteristic patterns may be produced even in machine-to-machine communications or without a user taking specific actions, e.g., at reboot time if a characteristic set of services are accessed by the device.

For instance, one could imagine that an intelligence agency identifies people going to a site by putting in a very long DNS name and looking for queries of a specific length. Such traffic analysis could weaken some privacy solutions.

The IAB privacy and security program also have a work in progress [RFC7624] that considers such inference-based attacks in a more general framework.

2.7. More Information

Useful background information can also be found in [tor-leak] (about the risk of privacy leak through DNS) and in a few academic papers: [yanbin-tsudik], [castillo-garcia], [fangming-hori-sakurai], and [federrath-fuchs-herrmann-piosecn].

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" behavior 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, which were leaked from the National Security Agency (NSA)) of the MORECOWBELL surveillance program [morecowbell], which uses the DNS, both passively and actively, to surreptitiously gather information about the users, is another good example showing that the lack of privacy protections in the DNS is actively exploited.

4. Legalties

To our knowledge, there are no specific privacy laws for DNS data, in any country. Interpreting general privacy laws like [data-protection-directive] or GDPR [4] applicable in the European Union in the context of DNS traffic data is not an easy task, and we do not know a court precedent here. See an interesting analysis in [sidn-entrada].

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 define solutions. Possible solutions to the issues described here are discussed in other documents (currently too many to all be mentioned); see, for instance, 'Recommendations for DNS Privacy Operators' [I-D.ietf-dprive-bcp-op].

6. Acknowledgments

Thanks to Nathalie Boulevard and to the CENTR members for the original work that led to this document. Thanks to Ondrej Sury for the interesting discussions. Thanks to Mohsen Souissi and John Heidemann for proofreading and to Paul Hoffman, Matthijs Mekking, Marcos Sanz, Tim Wicinski, Francis Dupont, Allison Mankin, and Warren Kumari for proofreading, providing technical remarks, and making many readability improvements. Thanks to Dan York, Suzanne Woolf, Tony Finch, Stephen Farrell, Peter Koch, Simon Josefsson, and Frank Denis for good written contributions. And thanks to the IESG members for the last remarks.

7. Changelog

draft-bortzmeyer-dprive-rfc7626-bis-02

- o Update various references and fix some nits.

draft-bortzmeyer-dprive-rfc7626-bis-01

- o Update reference for dickinson-bcp-op to draft-dickinson-dprive-bcp-op

draft-borztmeyer-dprive-rfc7626-bis-00:

Initial commit. Differences to RFC7626:

- o Update many references
- o Add discussions of encrypted transports including DNS-over-TLS and DoH
- o Add section on DNS payload
- o Add section on authentication of servers
- o Add section on blocking of services

8. References

8.1. Normative References

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <<https://www.rfc-editor.org/info/rfc1034>>.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<https://www.rfc-editor.org/info/rfc1035>>.
- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <<https://www.rfc-editor.org/info/rfc6973>>.
- [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>>.

8.2. Informative References

- [aeris-dns] Vinot, N., "Vie privée: et le DNS alors?", (In French), 2015, <<https://blog.imirhil.fr/vie-privee-et-le-dns-alors.html>>.
- [castillo-garcia] Castillo-Perez, S. and J. Garcia-Alfaro, "Anonymous Resolution of DNS Queries", 2008, <<http://deic.uab.es/~joaquin/papers/is08.pdf>>.
- [dagon-malware] Dagon, D., "Corrupted DNS Resolution Paths: The Rise of a Malicious Resolution Authority", ISC/OARC Workshop, 2007, <<https://www.dns-oarc.net/files/workshop-2007/Dagon-Resolution-corruption.pdf>>.
- [darkreading-dns] Lemos, R., "Got Malware? Three Signs Revealed In DNS Traffic", InformationWeek Dark Reading, May 2013, <<http://www.darkreading.com/analytics/security-monitoring/got-malware-three-signs-revealed-in-dns-traffic/d/d-id/1139680>>.

[data-protection-directive]

European Parliament, "Directive 95/46/EC of the European Parliament and of the council on the protection of individuals with regard to the processing of personal data and on the free movement of such data", Official Journal L 281, pp. 0031 - 0050, November 1995, <<http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:EN:HTML>>.

[day-at-root]

Castro, S., Wessels, D., Fomenkov, M., and K. Claffy, "A Day at the Root of the Internet", ACM SIGCOMM Computer Communication Review, Vol. 38, Number 5, DOI 10.1145/1452335.1452341, October 2008, <<http://www.sigcomm.org/sites/default/files/ccr/papers/2008/October/1452335-1452341.pdf>>.

[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/>>.

[ditl]

CAIDA, "A Day in the Life of the Internet (DITL)", 2002, <<http://www.caida.org/projects/ditl/>>.

[dns-footprint]

Stoner, E., "DNS Footprint of Malware", OARC Workshop, October 2010, <<https://www.dns-oarc.net/files/workshop-201010/OARC-ers-20101012.pdf>>.

[dnshchanger]

Wikipedia, "DNSChanger", October 2013, <<https://en.wikipedia.org/w/index.php?title=DNSChanger&oldid=578749672>>.

[dnsmezzo]

Bortzmeyer, S., "DNSmezzo", 2009, <<http://www.dnsmezzo.net/>>.

[fangming-hori-sakurai]

Fangming, Z., Hori, Y., and K. Sakurai, "Analysis of Privacy Disclosure in DNS Query", 2007 International Conference on Multimedia and Ubiquitous Engineering (MUE 2007), Seoul, Korea, ISBN: 0-7695-2777-9, pp. 952-957, DOI 10.1109/MUE.2007.84, April 2007, <<http://dl.acm.org/citation.cfm?id=1262690.1262986>>.

- [federrath-fuchs-herrmann-piosecny]
Federrath, H., Fuchs, K., Herrmann, D., and C. Piosecny,
"Privacy-Preserving DNS: Analysis of Broadcast, Range
Queries and Mix-based Protection Methods", Computer
Security ESORICS 2011, Springer, page(s) 665-683,
ISBN 978-3-642-23821-5, 2011, <https://svs.informatik.uni-hamburg.de/publications/2011/2011-09-14_FFHP_PrivacyPreservingDNS_ESORICS2011.pdf>.
- [grangeia.snooping]
Grangeia, L., "DNS Cache Snooping or Snooping the Cache
for Fun and Profit", February 2004,
<http://www.msit2005.mut.ac.th/msit_media/1_2551/nete4630/materials/20080718130017Hc.pdf>.
- [herrmann-reidentification]
Herrmann, D., Gerber, C., Banse, C., and H. Federrath,
"Analyzing Characteristic Host Access Patterns for Re-
Identification of Web User Sessions",
DOI 10.1007/978-3-642-27937-9_10, 2012, <http://epub.uni-regensburg.de/21103/1/Paper_PUL_nordsec_published.pdf>.
- [I-D.ietf-dprive-bcp-op]
Dickinson, S., Overeinder, B., Rijswijk-Deij, R., and A.
Mankin, "Recommendations for DNS Privacy Service
Operators", draft-ietf-dprive-bcp-op-01 (work in
progress), December 2018.
- [I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed
and Secure Transport", draft-ietf-quic-transport-17 (work
in progress), December 2018.
- [I-D.ietf-tls-sni-encryption]
Huitema, C. and E. Rescorla, "Issues and Requirements for
SNI Encryption in TLS", draft-ietf-tls-sni-encryption-04
(work in progress), November 2018.
- [morecowbell]
Grothoff, C., Wachs, M., Ermert, M., and J. Appelbaum,
"NSA's MORECOWBELL: Knell for DNS", GNUnet e.V., January
2015, <<https://gnunet.org/morecowbell>>.
- [packetq]
Dot SE, "PacketQ, a simple tool to make SQL-queries
against PCAP-files", 2011,
<<https://github.com/dotse/packetq/wiki>>.

- [packetq-list] PacketQ, "PacketQ Mailing List", <<http://lists.iis.se/mailman/listinfo/packetq>>.
- [passive-dns] Weimer, F., "Passive DNS Replication", April 2005, <<http://www.enyo.de/fw/software/dnslogger/#2>>.
- [pitfalls-of-dns-encrption] Shulman, H., "Pretty Bad Privacy:Pitfalls of DNS Encryption", <<https://www.ietf.org/mail-archive/web/dns-privacy/current/pdfWqAIUmEl47.pdf>>.
- [prism] Wikipedia, "PRISM (surveillance program)", July 2015, <[https://en.wikipedia.org/w/index.php?title=PRISM_\(surveillance_program\)&oldid=673789455](https://en.wikipedia.org/w/index.php?title=PRISM_(surveillance_program)&oldid=673789455)>.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <<https://www.rfc-editor.org/info/rfc4033>>.
- [RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, DOI 10.17487/RFC5155, March 2008, <<https://www.rfc-editor.org/info/rfc5155>>.
- [RFC5936] Lewis, E. and A. Hoenes, Ed., "DNS Zone Transfer Protocol (AXFR)", RFC 5936, DOI 10.17487/RFC5936, June 2010, <<https://www.rfc-editor.org/info/rfc5936>>.
- [RFC6269] Ford, M., Ed., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, DOI 10.17487/RFC6269, June 2011, <<https://www.rfc-editor.org/info/rfc6269>>.
- [RFC7413] Cheng, Y., Chu, J., Radhakrishnan, S., and A. Jain, "TCP Fast Open", RFC 7413, DOI 10.17487/RFC7413, December 2014, <<https://www.rfc-editor.org/info/rfc7413>>.
- [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <<https://www.rfc-editor.org/info/rfc7525>>.

- [RFC7624] Barnes, R., Schneier, B., Jennings, C., Hardie, T., Trammell, B., Huitema, C., and D. Borkmann, "Confidentiality in the Face of Pervasive Surveillance: A Threat Model and Problem Statement", RFC 7624, DOI 10.17487/RFC7624, August 2015, <<https://www.rfc-editor.org/info/rfc7624>>.
- [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>>.
- [RFC7871] Contavalli, C., van der Gaast, W., Lawrence, D., and W. Kumari, "Client Subnet in DNS Queries", RFC 7871, DOI 10.17487/RFC7871, May 2016, <<https://www.rfc-editor.org/info/rfc7871>>.
- [RFC7873] Eastlake 3rd, D. and M. Andrews, "Domain Name System (DNS) Cookies", RFC 7873, DOI 10.17487/RFC7873, May 2016, <<https://www.rfc-editor.org/info/rfc7873>>.
- [RFC7929] Wouters, P., "DNS-Based Authentication of Named Entities (DANE) Bindings for OpenPGP", RFC 7929, DOI 10.17487/RFC7929, August 2016, <<https://www.rfc-editor.org/info/rfc7929>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [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>>.
- [RFC8499] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <<https://www.rfc-editor.org/info/rfc8499>>.
- [ripe-atlas-turkey] Aben, E., "A RIPE Atlas View of Internet Meddling in Turkey", March 2014, <<https://labs.ripe.net/Members/emileaben/a-ripe-atlas-view-of-internet-meddling-in-turkey>>.

[sidn-entrada]

Hesselman, C., Jansen, J., Wullink, M., Vink, K., and M. Simon, "A privacy framework for 'DNS big data' applications", November 2014, <https://www.sidnlabs.nl/uploads/tx_sidnpublications/SIDN_Labs_Privacyraamwerk_Position_Paper_V1.4_ENG.pdf>.

[thomas-ditl-tcp]

Thomas, M. and D. Wessels, "An Analysis of TCP Traffic in Root Server DITL Data", DNS-OARC 2014 Fall Workshop, October 2014, <<https://indico.dns-oarc.net/event/20/session/2/contribution/15/material/slides/1.pdf>>.

[tor-leak]

Tor, "DNS leaks in Tor", 2013, <<https://www.torproject.org/docs/faq.html.en#WarningsAboutSOCKSsandDNSInformationLeaks>>.

[yanbin-tsudik]

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

8.3. URIs

[1] <https://lists.dns-oarc.net/pipermail/dns-operations/2016-January/014141.html>

[2] <http://netres.ec/?b=11B99BD>

[3] https://www.researchgate.net/publication/320322146_DNS-DNS_DNS-based_De-NAT_Scheme

[4] <https://www.eugdpr.org/the-regulation.html>

Authors' Addresses

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

Email: bortzmeyer+ietf@nic.fr

Sara Dickinson
Sinodun IT
Magdalen Centre
Oxford Science Park
Oxford OX4 4GA
United Kingdom

Email: sara@sinodun.com

dprive
Internet-Draft
Intended status: Best Current Practice
Expires: January 17, 2019

S. Dickinson
Sinodun IT
B. Overeinder
NLnet Labs
R. van Rijswijk-Deij
SURFnet bv
A. Mankin
Salesforce
July 16, 2018

Recommendations for DNS Privacy Service Operators
draft-dickinson-dprive-bcp-op-01

Abstract

This document presents operational, policy and security considerations for DNS operators who choose to offer DNS Privacy services. With the recommendations, the operator can make deliberate decisions which services to provide, and how the decisions and alternatives impact the privacy of users.

This document also presents a framework to assist writers of DNS Privacy Policy and Practices Statements (analogous to DNS Security Extensions (DNSSEC) Policies and DNSSEC Practice Statements described in [RFC6841]).

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 January 17, 2019.

Copyright Notice

Copyright (c) 2018 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	3
2. Scope	5
3. Privacy related documents	5
4. Terminology	6
5. Recommendations for DNS privacy services	6
5.1. On the wire between client and server	7
5.1.1. Transport recommendations	7
5.1.2. Authentication of DNS privacy services	8
5.1.3. Protocol recommendations	9
5.1.4. Availability	10
5.1.5. Service options	11
5.1.6. Limitations of using a pure TLS proxy	11
5.2. Data at rest on the server	12
5.2.1. Data handling	12
5.2.2. Data minimization of network traffic	13
5.2.3. IP address pseudonymization and anonymization methods	14
5.2.4. Pseudonymization, anonymization or discarding of other correlation data	14
5.2.5. Cache snooping	15
5.3. Data sent onwards from the server	15
5.3.1. Protocol recommendations	15
5.3.2. Client query obfuscation	16
5.3.3. Data sharing	17
6. DNS privacy policy and practice statement	17
6.1. Recommended contents of a DPPPS	18
6.2. Current policy and privacy statements	19
6.2.1. Quad9	19
6.2.2. Cloudflare	19
6.2.3. Google	20
6.2.4. OpenDNS	20
6.2.5. Comparison	20

6.3. Enforcement/accountability	20
7. IANA considerations	21
8. Security considerations	21
9. Acknowledgements	21
10. Contributors	21
11. Changelog	21
12. References	22
12.1. Normative References	22
12.2. Informative References	23
12.3. URIs	25
Appendix A. Documents	26
A.1. Potential increases in DNS privacy	26
A.2. Potential decreases in DNS privacy	27
A.3. Related operational documents	27
Appendix B. IP address techniques	28
B.1. Google Analytics non-prefix filtering	29
B.2. dnswasher	29
B.3. Prefix-preserving map	29
B.4. Cryptographic Prefix-Preserving Pseudonymisation	30
B.5. Top-hash Subtree-replicated Anonymisation	30
B.6. ipcipher	30
B.7. Bloom filters	31
Authors' Addresses	31

1. Introduction

[NOTE: This document is submitted to the IETF for initial review and for feedback on the best forum for future versions of this document. Initial considerations for DoH [I-D.ietf-doh-dns-over-https] are included here in anticipation of that draft progressing to be an RFC but further analysis is required.]

The Domain Name System (DNS) is at the core of the Internet; almost every activity on the Internet starts with a DNS query (and often several). However the DNS was not originally designed with strong security or privacy mechanisms. A number of developments have taken place in recent years which aim to increase the privacy of the DNS system and these are now seeing some deployment. This latest evolution of the DNS presents new challenges to operators and this document attempts to provide an overview of considerations for privacy focussed DNS services.

In recent years there has also been an increase in the availability of "open resolvers" [I-D.ietf-dnsop-terminology-bis] which users may prefer to use instead of the default network resolver because they offer a specific feature (e.g. good reachability, encrypted transport, strong privacy policy, filtering (or lack of), etc.). These open resolvers have tended to be at the forefront of adoption

of privacy related enhancements but it is anticipated that operators of other resolver services will follow.

Whilst protocols that encrypt DNS messages on the wire provide protection against certain attacks, the resolver operator still has (in principle) full visibility of the query data and transport identifiers for each user. Therefore, a trust relationship exists. The ability of the operator to provide a transparent, well documented, and secure privacy service will likely serve as a major differentiating factor for privacy conscious users if they make an active selection of which resolver to use.

It should also be noted that the choice of a user to configure a single resolver (or a fixed set of resolvers) and an encrypted transport to use in all network environments has both advantages and disadvantages. For example the user has a clear expectation of which resolvers have visibility of their query data however this resolver/transport selection may provide an added mechanism to track them as they move across network environments. Commitments from operators to minimize such tracking are also likely to play a role in users selection of resolver.

More recently the global legislative landscape with regard to personal data collection, retention, and pseudonymization has seen significant activity with differing requirements active in different jurisdictions. For example the user of a service and the service itself may be in jurisdictions with conflicting legislation. It is an untested area that simply using a DNS resolution service constitutes consent from the user for the operator to process their query data. The impact of recent legislative changes on data pertaining to the users of both Internet Service Providers and DNS open resolvers is not fully understood at the time of writing.

This document has two main goals:

- o To provide operational and policy guidance related to DNS over encrypted transports and to outline recommendations for data handling for operators of DNS privacy services.
- o To introduce the DNS Privacy Policy and Practice Statement (DPPPS) and present a framework to assist writers of this document. A DPPPS is a document that an operator can publish outlining their operational practices and commitments with regard to privacy thereby providing a means for clients to evaluate the privacy properties of a given DNS privacy service. In particular, the framework identifies the elements that should be considered in formulating a DPPPS. This document does not, however, define a

particular Policy or Practice Statement, nor does it seek to provide legal advice or recommendations as to the contents.

Community insight [or judgment?] about operational practices can change quickly, and experience shows that a Best Current Practice (BCP) document about privacy and security is a point-in-time statement. Readers are advised to seek out any errata or updates that apply to this document.

2. Scope

"DNS Privacy Considerations" [I-D.bortzmeyer-dprive-rfc7626-bis] describes the general privacy issues and threats associated with the use of the DNS by Internet users and much of the threat analysis here is lifted from that document and from [RFC6873]. However this document is limited in scope to best practice considerations for the provision of DNS privacy services by servers (recursive resolvers) to clients (stub resolvers or forwarders). Privacy considerations specifically from the perspective of an end user, or those for operators of authoritative nameservers are out of scope.

This document includes (but is not limited to) considerations in the following areas (taken from [I-D.bortzmeyer-dprive-rfc7626-bis]):

1. Data "on the wire" between a client and a server
2. Data "at rest" on a server (e.g. in logs)
3. Data "sent onwards" from the server (either on the wire or shared with a third party)

Whilst the issues raised here are targeted at those operators who choose to offer a DNS privacy service, considerations for areas 2 and 3 could equally apply to operators who only offer DNS over unencrypted transports but who would like to align with privacy best practice.

3. Privacy related documents

There are various documents that describe protocol changes that have the potential to either increase or decrease the privacy of the DNS. Note this does not imply that some documents are good or bad, better or worse, just that (for example) some features may bring functional benefits at the price of a reduction in privacy and conversely some features increase privacy with an accompanying increase in complexity. A selection of the most relevant documents are listed in Appendix A for reference.

4. Terminology

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

Privacy terminology is as described in Section 3 of [RFC6973].

DNS terminology is as described in [I-D.ietf-dnsop-terminology-bis] with one modification: we use the definition of Privacy-enabling DNS server taken from [RFC8310]:

- o Privacy-enabling DNS server: A DNS server (most likely a full-service resolver) that implements DNS-over-TLS [RFC7858], and may optionally implement DNS-over-DTLS [RFC8094]. The server should also offer at least one of the credentials described in Section 8 and implement the (D)TLS profile described in Section 9.

TODO: Update the definition of Privacy-enabling DNS server in [I-D.ietf-dnsop-terminology-bis] to be complete and also include DoH, then reference that here.

- o DPPPS: DNS Privacy Policy and Practice Statement, see Section 6.
- o DNS privacy service: The service that is offered via a privacy-enabling DNS server and is documented either in an informal statement of policy and practice with regard to users privacy or a formal DPPPS.

5. Recommendations for DNS privacy services

We describe three classes of actions that operators of DNS privacy services can take:

- o Threat mitigation for well understood and documented privacy threats to the users of the service and in some cases to the operators of the service.
- o Optimization of privacy services from an operational or management perspective
- o Additional options that could further enhance the privacy and usability of the service

This document does not specify policy only best practice, however for DNS Privacy services to be considered compliant with these best practice guidelines they SHOULD implement (where appropriate) all:

- o Threat mitigations to be minimally compliant
- o Optimizations to be moderately compliant
- o Additional options to be maximally compliant

TODO: Some of the threats listed in the following sections are taken directly from Section 5 of RFC6973, some are just standalone descriptions, we need to go through all of them and see if we can use the RFC6973 threats where possible and make them consistent.

5.1. On the wire between client and server

In this section we consider both data on the wire and the service provided to the client.

5.1.1. Transport recommendations

Threats:

- o Surveillance: Passive surveillance of traffic on the wire
- o Intrusion: Active injection of spurious data or traffic

Mitigations:

A DNS privacy service can mitigate these threats by providing service over one or more of the following transports

- o DNS-over-TLS [RFC7858]
- o DoH [I-D.ietf-doh-dns-over-https]

Additional options:

- o A DNS privacy service can also be provided over DNS-over-DTLS [RFC8094], however note that this is an Experimental specification.

It is noted that DNS privacy service might be provided over IPSec, DNSCrypt or VPNs. However, use of these transports for DNS are not standardized and any discussion of best practice for providing such service is out of scope for this document.

5.1.2. Authentication of DNS privacy services

Threats:

- o Surveillance and Intrusion: Active attacks that can redirect traffic to rogue servers

Mitigations:

DNS privacy services should ensure clients can authenticate the server. Note that this, in effect, commits the DNS privacy service to a public identity users will trust.

When using DNS-over-TLS clients that select a 'Strict Privacy' usage profile [RFC8310] (to mitigate the threat of active attack on the client) require the ability to authenticate the DNS server. To enable this, DNS privacy services that offer DNS-over-TLS should provide credentials in the form of either X.509 certificates, SPKI pinsets or TLSA records.

When offering DoH [I-D.ietf-doh-dns-over-https], HTTPS requires authentication of the server as part of the protocol.

Optimizations:

DNS privacy services can also consider the following capabilities/options:

- o As recommended in [RFC8310] providing DANE TLSA records for the nameserver
 - * In particular, the service could provide TLSA records such that authenticating solely via the PKIX infrastructure can be avoided.
- o Implementing [I-D.ietf-tls-dnssec-chain-extension]
 - * This can decrease the latency of connection setup to the server and remove the need for the client to perform meta-queries to obtain and validate the DANE records.

5.1.2.1. Certificate management

Anecdotal evidence to date highlights the management of certificates as one of the more challenging aspects for operators of traditional DNS resolvers that choose to additionally provide a DNS privacy service as management of such credentials is new to those DNS operators.

It is noted that SPKI pinset management is described in [RFC7858] but that key pinning mechanisms in general have fallen out of favour operationally for various reasons.

Threats:

- o Invalid certificates, resulting in an unavailable service.
- o Mis-identification of a server by a client e.g. typos in URLs or authentication domain names

Mitigations:

It is recommended that operators:

- o Choose a short, memorable authentication name for their service
- o Automate the generation and publication of certificates
- o Monitor certificates to prevent accidental expiration of certificates

TODO: Could we provide references for certificate management best practice, for example Section 6.5 of RFC7525?

5.1.3. Protocol recommendations

5.1.3.1. DNS-over-TLS

Threats:

- o Known attacks on TLS (TODO: add a reference)
- o Traffic analysis (TODO: add a reference)
- o Potential for client tracking via transport identifiers
- o Blocking of well known ports (e.g. 853 for DNS-over-TLS)

Mitigations:

In the case of DNS-over-TLS, TLS profiles from Section 9 and the Countermeasures to DNS Traffic Analysis from section 11.1 of [RFC8310] provide strong mitigations. This includes but is not limited to:

- o Adhering to [RFC7525]

- o Implementing only (D)TLS 1.2 or later as specified in [RFC8310]
- o Implementing EDNS(0) Padding [RFC7830] using the guidelines in [I-D.ietf-dprive-padding-policy]
- o Clients should not be required to use TLS session resumption [RFC5077], Domain Name System (DNS) Cookies [RFC7873].
- o A DNS-over-TLS privacy service on both port 853 and 443. We note that this practice may require revision when DoH becomes more widely deployed, because of the potential use of the same ports for two incompatible types of service.

Optimizations:

- o Concurrent processing of pipelined queries, returning responses as soon as available, potentially out of order as specified in [RFC7766]. This is often called 'OOOR' - out-of-order responses. (Providing processing performance similar to HTTP multiplexing)
- o Management of TLS connections to optimize performance for clients using either
 - * [RFC7766] and EDNS(0) Keepalive [RFC7828] and/or
 - * DNS Stateful Operations [I-D.ietf-dnsop-session-signal]

Additional options that providers may consider:

- o Offer a .onion [RFC7686] service endpoint

5.1.3.2. DoH

TODO: Fill this in, a lot of overlap with DNS-over-TLS but we need to address DoH specific ones if possible.

Mitigations:

- o Clients should not be required to use HTTP Cookies [RFC6265].
- o Clients should not be required to include any headers beyond the absolute minimum to obtain service from a DoH server.

5.1.4. Availability

Threats:

- o A failed DNS privacy service could force the user to switch providers, fallback to cleartext or accept no DNS service for the outage.

Mitigations:

A DNS privacy service must be engineered for high availability. Particular care should be taken to protect DNS privacy services against denial-of-service attacks, as experience has shown that unavailability of DNS resolving because of attacks is a significant motivation for users to switch services.

TODO: Add reference to ongoing research on this topic.

5.1.5. Service options

Threats:

- o Unfairly disadvantaging users of the privacy service with respect to the services available. This could force the user to switch providers, fallback to cleartext or accept no DNS service for the outage.

Mitigations:

A DNS privacy service should deliver the same level of service offered on un-encrypted channels in terms of such options as filtering (or lack of), DNSSEC validation, etc.

5.1.6. Limitations of using a pure TLS proxy

Optimization:

Some operators may choose to implement DNS-over-TLS using a TLS proxy (e.g. nginx [1], haproxy [2] or stunnel [3]) in front of a DNS nameserver because of proven robustness and capacity when handling large numbers of client connections, load balancing capabilities and good tooling. Currently, however, because such proxies typically have no specific handling of DNS as a protocol over TLS or DTLS using them can restrict traffic management at the proxy layer and at the DNS server. For example, all traffic received by a nameserver behind such a proxy will appear to originate from the proxy and DNS techniques such as ACLs, RRL or DNS64 will be hard or impossible to implement in the nameserver.

Operators may choose to use a DNS aware proxy such as dnsmist.

5.2. Data at rest on the server

5.2.1. Data handling

Threats:

- o Surveillance
- o Stored data compromise
- o Correlation
- o Identification
- o Secondary use
- o Disclosure
- o Contravention of legal requirements not to process user data?

Mitigations:

The following are common activities for DNS service operators and in all cases should be minimized or completely avoided if possible for DNS privacy services. If data is retained it should be encrypted and either aggregated, pseudonymized or anonymized whenever possible. In general the principle of data minimization described in [RFC6973] should be applied.

- o Transient data (e.g. that is used for real time monitoring and threat analysis which might be held only memory) should be retained for the shortest possible period deemed operationally feasible.
- o The retention period of DNS traffic logs should be only those required to sustain operation of the service and, to the extent that such exists, meet regulatory requirements.
- o DNS privacy services should not track users except for the particular purpose of detecting and remedying technically malicious (e.g. DoS) or anomalous use of the service.
- o Data access should be minimized to only those personal who require access to perform operational duties.

5.2.2. Data minimization of network traffic

Data minimization refers to collecting, using, disclosing, and storing the minimal data necessary to perform a task, and this can be achieved by removing or obfuscating privacy-sensitive information in network traffic logs. This is typically personal data, or data that can be used to link a record to an individual, but may also include revealing other confidential information, for example on the structure of an internal corporate network.

The problem of effectively ensuring that DNS traffic logs contain no or minimal privacy-sensitive information is not one that currently has a generally agreed solution or any Standards to inform this discussion. This section presents an overview of current techniques to simply provide reference on the current status of this work.

Research into data minimization techniques (and particularly IP address pseudonymization/anonymization) was sparked in the late 1990s/early 2000s, partly driven by the desire to share significant corpuses of traffic captures for research purposes. Several techniques reflecting different requirements in this area and different performance/resource tradeoffs emerged over the course of the decade. Developments over the last decade have been both a blessing and a curse; the large increase in size between an IPv4 and an IPv6 address, for example, renders some techniques impractical, but also makes available a much larger amount of input entropy, the better to resist brute force re-identification attacks that have grown in practicality over the period.

Techniques employed may be broadly categorized as either anonymization or pseudonymization. The following discussion uses the definitions from [RFC6973] Section 3, with additional observations from van Dijkhuizen et al. [4]

- o Anonymization. To enable anonymity of an individual, there must exist a set of individuals that appear to have the same attribute(s) as the individual. To the attacker or the observer, these individuals must appear indistinguishable from each other.
- o Pseudonymization. The true identity is deterministically replaced with an alternate identity (a pseudonym). When the pseudonymization schema is known, the process can be reversed, so the original identity becomes known again.

In practice there is a fine line between the two; for example, how to categorize a deterministic algorithm for data minimization of IP addresses that produces a group of pseudonyms for a single given address.

5.2.3. IP address pseudonymization and anonymization methods

As [I-D.bortzmeyer-dprive-rfc7626-bis] makes clear, the big privacy risk in DNS is connecting DNS queries to an individual and the major vector for this in DNS traffic is the client IP address.

There is active discussion in the space of effective pseudonymization of IP addresses in DNS traffic logs, however there seems to be no single solution that is widely recognized as suitable for all or most use cases. There are also as yet no standards for this that are unencumbered by patents. This following table presents a high level comparison of various techniques employed or under development today and classifies them according to categorization of technique and other properties. The list of techniques includes the main techniques in current use, but does not claim to be comprehensive. Appendix B provides a more detailed survey of these techniques and definitions for the categories and properties listed below.

Figure showing comparison of IP address techniques (SVG) [5]

The choice of which method to use for a particular application will depend on the requirements of that application and consideration of the threat analysis of the particular situation.

For example, a common goal is that distributed packet captures must be in an existing data format such as PCAP [pcap] or C-DNS [I-D.ietf-dnsop-dns-capture-format] that can be used as input to existing analysis tools. In that case, use of a Format-preserving technique is essential. This, though, is not cost-free - several authors (e.g. Brenker & Arnes [6]) have observed that, as the entropy in a IPv4 address is limited, given a de-identified log from a target, if an attacker is capable of ensuring packets are captured by the target and the attacker can send forged traffic with arbitrary source and destination addresses to that target, any format-preserving pseudonymization is vulnerable to an attack along the lines of a cryptographic chosen plaintext attack.

5.2.4. Pseudonymization, anonymization or discarding of other correlation data

Threats:

- o IP TTL/Hoplimit can be used to fingerprint client OS
- o Tracking of TCP sessions
- o Tracking of TLS sessions and session resumption mechanisms

- o Resolvers *might* receive client identifiers e.g. MAC addresses in EDNS(0) options – some CPE devices are known to add them.

- o HTTP headers

Mitigations:

- o Data minimization or discarding of such correlation data

TODO: More analysis here.

5.2.5. Cache snooping

Threats:

- o Profiling of client queries by malicious third parties

Mitigations:

TODO: Describe techniques to defend against cache snooping

5.3. Data sent onwards from the server

In this section we consider both data sent on the wire in upstream queries and data shared with third parties.

5.3.1. Protocol recommendations

Threats:

- o Transmission of identifying data upstream.

Mitigations:

As specified in [RFC8310] for DNS-over-TLS but applicable to any DNS Privacy services the server should:

- o Implement QNAME minimization [RFC7816]
- o Honour a SOURCE PREFIX-LENGTH set to 0 in a query containing the EDNS(0) Client Subnet (ECS) option and not send an ECS option in upstream queries.

Optimizations:

- o The server should either
 - * not use the ECS option in upstream queries at all, or

- * offer alternative services, one that sends ECS and one that does not.

If operators do offer a service that sends the ECS options upstream they should use the shortest prefix that is operationally feasible (NOTE: the authors believe they will be able to add a reference for advice here soon) and ideally use a policy of whitelisting upstream servers to send ECS to in order to minimize data leakage. Operators should make clear in any policy statement what prefix length they actually send and the specific policy used.

Additional options:

- o Aggressive Use of DNSSEC-Validated Cache [RFC8198] to reduce the number of queries to authoritative servers to increase privacy.
- o Run a copy of the root zone on loopback [RFC7706] to avoid making queries to the root servers that might leak information.

5.3.2. Client query obfuscation

Additional options:

Since queries from recursive resolvers to authoritative servers are performed using cleartext (at the time of writing), resolver services need to consider the extent to which they may be directly leaking information about their client community via these upstream queries and what they can do to mitigate this further. Note, that even when all the relevant techniques described above are employed there may still be attacks possible, e.g. [Pitfalls-of-DNS-Encryption]. For example, a resolver with a very small community of users risks exposing data in this way and OUGHT obfuscate this traffic by mixing it with 'generated' traffic to make client characterization harder. The resolver could also employ aggressive pre-fetch techniques as a further measure to counter traffic analysis.

At the time of writing there are no standardized or widely recognized techniques to preform such obfuscation or bulk pre-fetches.

Another technique that particularly small operators may consider is forwarding local traffic to a larger resolver (with a privacy policy that aligns with their own practices) over an encrypted protocol so that the upstream queries are obfuscated among those of the large resolver.

5.3.3. Data sharing

Threats:

- o Surveillance
- o Stored data compromise
- o Correlation
- o Identification
- o Secondary use
- o Disclosure
- o Contravention of legal requirements not to process user data?

Mitigations:

Operators should not provide identifiable data to third-parties without explicit consent from clients (we take the stance here that simply using the resolution service itself does not constitute consent).

Even when consent is granted operators should employ data minimization techniques such as those described in Section 5.2.1 if data is shared with third-parties.

Operators should consider including specific guidelines for the collection of aggregated and/or anonymized data for research purposes, within or outside of their own organization.

TODO: More on data for research vs operations... how to still motivate operators to share anonymized data?

TODO: Guidelines for when consent is granted?

TODO: Applies to server data handling too.. could operators offer alternatives services one that implies consent for data processing, one that doesn't?

6. DNS privacy policy and practice statement

6.1. Recommended contents of a DPPPS

1 Policy

1.1 Recommendations. This section should explain, with reference to section Section 5 of this document which recommendations the DNS privacy service employs.

1.2 Data handling. This section should explain, with reference to section Section 5.2 of this document the policy for gathering and disseminating information collected by the DNS privacy service.

1.2.1 Specify clearly what data (including whether it is aggregated, pseudonymized or anonymized) is:

1.2.1.1 Collected and retained by the operator (and for how long)

1.2.1.2 Shared with partners

1.2.1.3 Shared, sold or rented to third-parties

1.2.2 Specify any exceptions to the above, for example technically malicious or anomalous behaviour

1.2.3 Declare any partners, third-party affiliations or sources of funding

1.2.4 Whether user DNS data is correlated or combined with any other personal information held by the operator

2 Practice. This section should explain the current operational practices of the service.

2.1 Specify any temporary or permanent deviations from the policy for operational reasons

2.2 With reference to section Section 5.1 provide specific details of which capabilities are provided on which address and ports

2.3 With reference to section Section 5.3 provide specific details of which capabilities are employed for upstream traffic from the server

2.4 Specify the authentication name to be used (if any) and if TLSA records are published (including options used in the TLSA records)

2.5 Specify the SPKI pinsets to be used (if any) and policy for rolling keys

2.6 Provide a contact email address for the service

6.2. Current policy and privacy statements

NOTE: An analysis of these statements will clearly only provide a snapshot at the time of writing. It is included in this version of the draft to provide a basis for the assessment of the contents of the DPPPS and is expected to be removed or substantially re-worked in a future version.

6.2.1. Quad9

UDP/TCP and TLS (port 853) service provided on two addresses:

- o 'Secure': 9.9.9.9, 149.112.112.112, 2620:fe::fe, 2620:fe::9
- o 'Unsecured': 9.9.9.10, 149.112.112.10, 2620:fe::10

Policy:

- o <<https://www.quad9.net/policy/>>
- o <<https://www.quad9.net/privacy/>>
- o <<https://www.quad9.net/faq/>>

6.2.2. Cloudflare

UDP/TCP and TLS (port 853) service provided on 1.1.1.1, 1.0.0.1, 2606:4700:4700::1111 and 2606:4700:4700::1001.

Policy:

- o <<https://developers.cloudflare.com/1.1.1.1/commitment-to-privacy/privacy-policy/privacy-policy/>>

DoH provided on: <<https://cloudflare-dns.com/dns-query>>

Policy:

- o <<https://developers.cloudflare.com/1.1.1.1/commitment-to-privacy/privacy-policy/firefox/>>

Tor endpoint: <<https://dns4torpnlfs2ifuz2s2yf3fc7rdmsbhm6rw75euj35pac6ap25zgqad.onion>>.

6.2.3. Google

UDP/TCP service provided on 8.8.8.8, 8.8.4.4, 2001:4860:4860::8888 and 2001:4860:4860::8844.

Policy: <<https://developers.google.com/speed/public-dns/privacy>>

6.2.4. OpenDNS

UDP/TCP service provided on 208.67.222.222 and 208.67.220.220 (no IPv6).

We could find no specific privacy policy for the DNS resolution, only a general one from Cisco that seems focussed on websites.

Policy: <<https://www.cisco.com/c/en/us/about/legal/privacy-full.html>>

6.2.5. Comparison

The following tables provides a high-level comparison of the policy and practice statements above and also some observations of practice measured at dnspriacy.org [7]. The data is not exhaustive and has not been reviewed or confirmed by the operators.

A question mark indicates no clear statement or data could be located on the issue. A dash indicates the category is not applicable to the service.

Table showing comparison of operators policies [8]

Table showing comparison of operators practices [9]

NOTE: Review and correction of any inaccuracies in the table would be much appreciated.

6.3. Enforcement/accountability

Transparency reports may help with building user trust that operators adhere to their policies and practices.

Independent monitoring should be performed where possible of:

- o ECS, QNAME minimization, EDNS(0) padding, etc.
- o Filtering
- o Uptime

7. IANA considerations

None

8. Security considerations

TODO: e.g. New issues for DoS defence, server admin policies

9. Acknowledgements

Many thanks to Amelia Andersdotter for a very thorough review of the first draft of this document. Thanks also to John Todd for discussions on this topic, and to Stephane Bortzmeyer for review.

Sara Dickinson thanks the Open Technology Fund for a grant to support the work on this document.

10. Contributors

The below individuals contributed significantly to the document:

John Dickinson
Sinodun Internet Technologies
Magdalen Centre
Oxford Science Park
Oxford OX4 4GA
United Kingdom

Jim Hague
Sinodun Internet Technologies
Magdalen Centre
Oxford Science Park
Oxford OX4 4GA
United Kingdom

11. Changelog

draft-dickinson-dprive-bcp-op-01

- o Update reference to RFC7626 to draft-bortzmeyer-rfc7626-bis
- o Fix a few typos

draft-dickinson-dprive-bcp-op-00

Name change to add dprive. Differences to draft-dickinson-bcp-op-00:

- o Reworked the Terminology, Introduction and Scope

- o Added Document section
 - o Reworked the Recommendations section to describe threat mitigations, optimizations and other options. Split the recommendations up into 3 subsections: on the wire, at rest and upstream
 - o Added much more information on data handling and IP address pseudonymization and anonymization
 - o Added more details and comparison of some existing policy/privacy policies
 - o Applied virtually all of Amelia Andersdotter's suggested changes.
- draft-dickinson-bcp-op-00
- o Initial commit

12. References

12.1. Normative References

- [I-D.ietf-dnsop-terminology-bis]
Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", draft-ietf-dnsop-terminology-bis-11 (work in progress), July 2018.
- [I-D.ietf-doh-dns-over-https]
Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", draft-ietf-doh-dns-over-https-12 (work in progress), June 2018.
- [I-D.ietf-dprive-padding-policy]
Mayrhofer, A., "Padding Policy for EDNS(0)", draft-ietf-dprive-padding-policy-05 (work in progress), April 2018.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5077] Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State", RFC 5077, DOI 10.17487/RFC5077, January 2008, <<https://www.rfc-editor.org/info/rfc5077>>.

- [RFC6265] Barth, A., "HTTP State Management Mechanism", RFC 6265, DOI 10.17487/RFC6265, April 2011, <<https://www.rfc-editor.org/info/rfc6265>>.
- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <<https://www.rfc-editor.org/info/rfc6973>>.
- [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <<https://www.rfc-editor.org/info/rfc7525>>.
- [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>>.
- [RFC7830] Mayrhofer, A., "The EDNS(0) Padding Option", RFC 7830, DOI 10.17487/RFC7830, May 2016, <<https://www.rfc-editor.org/info/rfc7830>>.
- [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>>.
- [RFC7873] Eastlake 3rd, D. and M. Andrews, "Domain Name System (DNS) Cookies", RFC 7873, DOI 10.17487/RFC7873, May 2016, <<https://www.rfc-editor.org/info/rfc7873>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8310] Dickinson, S., Gillmor, D., and T. Reddy, "Usage Profiles for DNS over TLS and DNS over DTLS", RFC 8310, DOI 10.17487/RFC8310, March 2018, <<https://www.rfc-editor.org/info/rfc8310>>.

12.2. Informative References

- [I-D.bortzmeyer-dprive-rfc7626-bis]
Bortzmeyer, S. and S. Dickinson, "DNS Privacy Considerations", draft-bortzmeyer-dprive-rfc7626-bis-00 (work in progress), July 2018.

- [I-D.ietf-dnsop-dns-capture-format]
Dickinson, J., Hague, J., Dickinson, S., Manderson, T.,
and J. Bond, "C-DNS: A DNS Packet Capture Format", draft-
ietf-dnsop-dns-capture-format-07 (work in progress), May
2018.
- [I-D.ietf-dnsop-dns-tcp-requirements]
Kristoff, J. and D. Wessels, "DNS Transport over TCP -
Operational Requirements", draft-ietf-dnsop-dns-tcp-
requirements-02 (work in progress), May 2018.
- [I-D.ietf-dnsop-session-signal]
Bellis, R., Cheshire, S., Dickinson, J., Dickinson, S.,
Lemon, T., and T. Pusateri, "DNS Stateful Operations",
draft-ietf-dnsop-session-signal-11 (work in progress),
July 2018.
- [I-D.ietf-tls-dnssec-chain-extension]
Shore, M., Barnes, R., Huque, S., and W. Toorop, "A DANE
Record and DNSSEC Authentication Chain Extension for TLS",
draft-ietf-tls-dnssec-chain-extension-07 (work in
progress), March 2018.
- [pcap] tcpdump.org, "PCAP", 2016, <<http://www.tcpdump.org/>>.
- [Pitfalls-of-DNS-Encryption]
Shulman, H., "Pretty Bad Privacy: Pitfalls of DNS
Encryption", 2014, <[https://www.ietf.org/mail-archive/web/
dns-privacy/current/pdfWqAIUmEl47.pdf](https://www.ietf.org/mail-archive/web/dns-privacy/current/pdfWqAIUmEl47.pdf)>.
- [RFC6235] Boschi, E. and B. Trammell, "IP Flow Anonymization
Support", RFC 6235, DOI 10.17487/RFC6235, May 2011,
<<https://www.rfc-editor.org/info/rfc6235>>.
- [RFC6841] Ljunggren, F., Eklund Lowinder, AM., and T. Okubo, "A
Framework for DNSSEC Policies and DNSSEC Practice
Statements", RFC 6841, DOI 10.17487/RFC6841, January 2013,
<<https://www.rfc-editor.org/info/rfc6841>>.
- [RFC6873] Salgueiro, G., Gurbani, V., and A. Roach, "Format for the
Session Initiation Protocol (SIP) Common Log Format
(CLF)", RFC 6873, DOI 10.17487/RFC6873, February 2013,
<<https://www.rfc-editor.org/info/rfc6873>>.
- [RFC7686] Appelbaum, J. and A. Muffett, "The ".onion" Special-Use
Domain Name", RFC 7686, DOI 10.17487/RFC7686, October
2015, <<https://www.rfc-editor.org/info/rfc7686>>.

- [RFC7706] Kumari, W. and P. Hoffman, "Decreasing Access Time to Root Servers by Running One on Loopback", RFC 7706, DOI 10.17487/RFC7706, November 2015, <<https://www.rfc-editor.org/info/rfc7706>>.
- [RFC7766] Dickinson, J., Dickinson, S., Bellis, R., Mankin, A., and D. Wessels, "DNS Transport over TCP - Implementation Requirements", RFC 7766, DOI 10.17487/RFC7766, March 2016, <<https://www.rfc-editor.org/info/rfc7766>>.
- [RFC7828] Wouters, P., Abley, J., Dickinson, S., and R. Bellis, "The edns-tcp-keepalive EDNS0 Option", RFC 7828, DOI 10.17487/RFC7828, April 2016, <<https://www.rfc-editor.org/info/rfc7828>>.
- [RFC7871] Contavalli, C., van der Gaast, W., Lawrence, D., and W. Kumari, "Client Subnet in DNS Queries", RFC 7871, DOI 10.17487/RFC7871, May 2016, <<https://www.rfc-editor.org/info/rfc7871>>.
- [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>>.
- [RFC8198] Fujiwara, K., Kato, A., and W. Kumari, "Aggressive Use of DNSSEC-Validated Cache", RFC 8198, DOI 10.17487/RFC8198, July 2017, <<https://www.rfc-editor.org/info/rfc8198>>.

12.3. URIs

- [1] <https://nginx.org/>
- [2] <https://www.haproxy.org/>
- [3] <https://kb.isc.org/article/AA-01386/0/DNS-over-TLS.html>
- [4] <https://doi.org/10.1145/3182660>
- [5] https://github.com/Sinodun/draft-dprive-bcp-op/blob/master/draft-01/ip_techniques_table.svg
- [6] <https://pdfs.semanticscholar.org/7b34/12c951cebe71cd2cddac5fda164fb2138a44.pdf>
- [7] <https://dnsprivacy.org/jenkins/job/dnsprivacy-monitoring/>

- [8] https://github.com/Sinodun/draft-dprive-bcp-op/blob/master/draft-01/policy_table.svg
- [9] https://github.com/Sinodun/draft-dprive-bcp-op/blob/master/draft-01/practice_table.svg
- [10] <https://support.google.com/analytics/answer/2763052?hl=en>
- [11] <https://www.conversionworks.co.uk/blog/2017/05/19/anonymize-ip-geo-impact-test/>
- [12] <https://github.com/edmonds/pdns/blob/master/pdns/dnswasher.cc>
- [13] <http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html>
- [14] <http://an.kaist.ac.kr/~sbmoon/paper/intl-journal/2004-cn-anon.pdf>
- [15] <https://www.cc.gatech.edu/computing/Telecomm/projects/cryptopan/>
- [16] http://mharvan.net/talks/noms-ip_anon.pdf
- [17] <https://medium.com/@bert.hubert/on-ip-address-encryption-security-analysis-with-respect-for-privacy-dabe1201b476>
- [18] <https://github.com/PowerDNS/ipcipher>
- [19] <https://github.com/veorq/ipcrypt>
- [20] <https://www.ietf.org/mail-archive/web/cfrg/current/msg09494.html>
- [21] <https://tncl8.geant.org/core/presentation/127>

Appendix A. Documents

This section provides an overview of some DNS privacy related documents, however, this is neither an exhaustive list nor a definitive statement on the characteristic of the document.

A.1. Potential increases in DNS privacy

These documents are limited in scope to communications between stub clients and recursive resolvers:

- o 'Specification for DNS over Transport Layer Security (TLS)' [RFC7858], referred to here as 'DNS-over-TLS'.

- o 'DNS over Datagram Transport Layer Security (DTLS)' [RFC8094], referred to here as 'DNS-over-DTLS'. Note that this document has the Category of Experimental.
- o 'DNS Queries over HTTPS (DoH)' [I-D.ietf-doh-dns-over-https] referred to here as DoH.
- o 'Usage Profiles for DNS over TLS and DNS over DTLS' [RFC8310]
- o 'The EDNS(0) Padding Option' [RFC7830] and 'Padding Policy for EDNS(0)' [I-D.ietf-dprive-padding-policy]

These documents apply to recursive to authoritative DNS but are relevant when considering the operation of a recursive server:

- o 'DNS Query Name minimization to Improve Privacy' [RFC7816] referred to here as 'QNAME minimization'

A.2. Potential decreases in DNS privacy

These documents relate to functionality that could provide increased tracking of user activity as a side effect:

- o 'Client Subnet in DNS Queries' [RFC7871]
- o 'Domain Name System (DNS) Cookies' [RFC7873])
- o 'Transport Layer Security (TLS) Session Resumption without Server-Side State' [RFC5077] referred to here as simply TLS session resumption.
- o 'A DNS Packet Capture Format' [I-D.ietf-dnsop-dns-capture-format]
- o Passive DNS [I-D.ietf-dnsop-terminology-bis]

Note that depending on the specifics of the implementation [I-D.ietf-doh-dns-over-https] may also provide increased tracking.

A.3. Related operational documents

- o 'DNS Transport over TCP - Implementation Requirements' [RFC7766]
- o 'Operational requirements for DNS-over-TCP' [I-D.ietf-dnsop-dns-tcp-requirements]
- o 'The edns-tcp-keepalive EDNS0 Option' [RFC7828]
- o 'DNS Stateful Operations' [I-D.ietf-dnsop-session-signal]

Appendix B. IP address techniques

Data minimization methods may be categorized by the processing used and the properties of their outputs. The following builds on the categorization employed in [RFC6235]:

- o Format-preserving. Normally when encrypting, the original data length and patterns in the data should be hidden from an attacker. Some applications of de-identification, such as network capture de-identification, require that the de-identified data is of the same form as the original data, to allow the data to be parsed in the same way as the original.
- o Prefix preservation. Values such as IP addresses and MAC addresses contain prefix information that can be valuable in analysis, e.g. manufacturer ID in MAC addresses, subnet in IP addresses. Prefix preservation ensures that prefixes are de-identified consistently; e.g. if two IP addresses are from the same subnet, a prefix preserving de-identification will ensure that their de-identified counterparts will also share a subnet. Prefix preservation may be fixed (i.e. based on a user selected prefix length identified in advance to be preserved) or general.
- o Replacement. A one-to-one replacement of a field to a new value of the same type, for example using a regular expression.
- o Filtering. Removing (and thus truncating) or replacing data in a field. Field data can be overwritten, often with zeros, either partially (grey marking) or completely (black marking).
- o Generalization. Data is replaced by more general data with reduced specificity. One example would be to replace all TCP/UDP port numbers with one of two fixed values indicating whether the original port was ephemeral (≥ 1024) or non-ephemeral (> 1024). Another example, precision degradation, reduces the accuracy of e.g. a numeric value or a timestamp.
- o Enumeration. With data from a well-ordered set, replace the first data item data using a random initial value and then allocate ordered values for subsequent data items. When used with timestamp data, this preserves ordering but loses precision and distance.
- o Reordering/shuffling. Preserving the original data, but rearranging its order, often in a random manner.
- o Random substitution. As replacement, but using randomly generated replacement values.

- o Cryptographic permutation. Using a permutation function, such as a hash function or cryptographic block cipher, to generate a replacement de-identified value.

B.1. Google Analytics non-prefix filtering

Since May 2010, Google Analytics has provided a facility [10] that allows website owners to request that all their users IP addresses are anonymized within Google Analytics processing. This very basic anonymization simply sets to zero the least significant 8 bits of IPv4 addresses, and the least significant 80 bits of IPv6 addresses. The level of anonymization this produces is perhaps questionable. There are some analysis results [11] which suggest that the impact of this on reducing the accuracy of determining the user's location from their IP address is less than might be hoped; the average discrepancy in identification of the user city for UK users is no more than 17%.

Anonymization: Format-preserving, Filtering (grey marking).

B.2. dnswasher

Since 2006, PowerDNS have included a de-identification tool dnswasher [12] with their PowerDNS product. This is a PCAP filter that performs a one-to-one mapping of end user IP addresses with an anonymized address. A table of user IP addresses and their de-identified counterparts is kept; the first IPv4 user addresses is translated to 0.0.0.1, the second to 0.0.0.2 and so on. The de-identified address therefore depends on the order that addresses arrive in the input, and running over a large amount of data the address translation tables can grow to a significant size.

Anonymization: Format-preserving, Enumeration.

B.3. Prefix-preserving map

Used in TCPdpriv [13], this algorithm stores a set of original and anonymised IP address pairs. When a new IP address arrives, it is compared with previous addresses to determine the longest prefix match. The new address is anonymized by using the same prefix, with the remainder of the address anonymized with a random value. The use of a random value means that TCPdpriv is not deterministic; different anonymized values will be generated on each run. The need to store previous addresses means that TCPdpriv has significant and unbounded memory requirements, and because of the need to allocated anonymized addresses sequentially cannot be used in parallel processing.

Anonymization: Format-preserving, prefix preservation (general).

B.4. Cryptographic Prefix-Preserving Pseudonymisation

Cryptographic prefix-preserving pseudonymisation was originally proposed as an improvement to the prefix-preserving map implemented in TCPdpriv, described in Xu et al. [14] and implemented in the Crypto-PAN tool [15]. Crypto-PAN is now frequently used as an acronym for the algorithm. Initially it was described for IPv4 addresses only; extension for IPv6 addresses was proposed in Harvan & Schoenwaelder [16] and implemented in snmpdump. This uses a cryptographic algorithm rather than a random value, and thus pseudonymity is determined uniquely by the encryption key, and is deterministic. It requires a separate AES encryption for each output bit, so has a non-trivial calculation overhead. This can be mitigated to some extent (for IPv4, at least) by pre-calculating results for some number of prefix bits.

Pseudonymization: Format-preserving, prefix preservation (general).

B.5. Top-hash Subtree-replicated Anonymisation

Proposed in Ramaswamy & Wolf, Top-hash Subtree-replicated Anonymisation (TSA) originated in response to the requirement for faster processing than Crypto-PAN. It used hashing for the most significant byte of an IPv4 address, and a pre-calculated binary tree structure for the remainder of the address. To save memory space, replication is used within the tree structure, reducing the size of the pre-calculated structures to a few Mb for IPv4 addresses. Address pseudonymization is done via hash and table lookup, and so requires minimal computation. However, due to the much increased address space for IPv6, TSA is not memory efficient for IPv6.

Pseudonymization: Format-preserving, prefix preservation (general).

B.6. ipcipher

A recently-released proposal from PowerDNS [17], ipcipher [18] is a simple pseudonymization technique for IPv4 and IPv6 addresses. IPv6 addresses are encrypted directly with AES-128 using a key (which may be derived from a passphrase). IPv4 addresses are similarly encrypted, but using a recently proposed encryption ipcrypt [19] suitable for 32bit block lengths. However, the author of ipcrypt has since indicated [20] that it has low security, and further analysis has revealed it is vulnerable to attack.

Pseudonymization: Format-preserving, cryptographic permutation.

B.7. Bloom filters

van Rijswijk-Deij et al. [21] have recently described work using Bloom filters to categorize query traffic and record the traffic as the state of multiple filters. The goal of this work is to allow operators to identify so-called Indicators of Compromise (IOCs) originating from specific subnets without storing information about, or be able to monitor the DNS queries of an individual user. By using a Bloom filter, it is possible to determine with a high probability if, for example, a particular query was made, but the set of queries made cannot be recovered from the filter. Similarly, by mixing queries from a sufficient number of users in a single filter, it becomes practically impossible to determine if a particular user performed a particular query. Large numbers of queries can be tracked in a memory-efficient way. As filter status is stored, this approach cannot be used to regenerate traffic, and so cannot be used with tools used to process live traffic.

Anonymized: Generalization.

Authors' Addresses

Sara Dickinson
Sinodun IT
Magdalen Centre
Oxford Science Park
Oxford OX4 4GA
United Kingdom

Email: sara@sinodun.com

Benno J. Overeinder
NLnet Labs
Science Park 400
Amsterdam 1098 XH
The Netherlands

Email: benno@nlnetLabs.nl

Roland M. van Rijswijk-Deij
SURFnet bv
PO Box 19035
Utrecht 3501 DA Utrecht
The Netherlands

Email: roland.vanrijswijk@surfnet.nl

Allison Mankin
Salesforce

Email: allison.mankin@gmail.com