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.

Table of Contents

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 queries.

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, targetted to for BCP, that n would offers 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:

- 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.

- 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

- 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)

- 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.

- The authoritative server for ODNS queries decrypts the session key, which it uses to decrypt the domain in the query.

- 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.

- 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) that the space alloted 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