Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 6, 2014

R. Arends
Nominet
J. Abley
J. Damas
Dyn, Inc.

March 5, 2014

DNS Privacy with a Hint of Onion
draft-jabley-dnsop-dns-onion-00

Abstract

The Domain Name System (DNS) has no inherent capability to protect the privacy of end users. The data associated with DNS queries and responses can be observed by intermediate systems, and such observations could provide a source of metadata relating to end user behaviour.

This document describes an approach which separates the data in DNS queries and responses from the identity of the DNS resolver used by DNS clients.

This approach does not address privacy concerns between a stub resolver and a recursive resolver.

This approach imposes no requirement for modification of authority servers, and does not depend upon widespread deployment of DNSSEC signing or validation.

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 September 6, 2014.

Copyright Notice

Copyright (c) 2014 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

The Domain Name System (DNS), as described in [RFC1034] and [RFC1035], has no inherent capability to protect the privacy of end users. Privacy concerns are described in [I-D.bortzmeyer-dnsop-dns-privacy] and [I-D.koch-perpass-dns-confidentiality].

This document describes an approach which separates the data in DNS queries and responses from the identity of the DNS resolver used by DNS clients.

This approach does not address privacy between a stub resolver and a recursive resolver.

This approach imposes no requirement for modification of authority servers, and does not depend upon widespread deployment of DNSSEC signing or validation.

The approach described here is derived from (and is similar or identical in many respects to) the Tor project [Tor]. The motivation to write up a DNS-specific, Tor-like solution is to explore opportunities to optimise the solution space specifically for the DNS, e.g. for very short-lived transactions.

2. Notes to Readers

This is an incomplete proposal. It has been distributed in its current form for the purposes of discussion, such that the high-level approach can be considered amongst other options in the general consideration of DNS privacy.

The authors have called out particular gaps in this document. The authors are confident that there are many other gaps that have not been mentioned. The absence of a description of a gap in this document does not imply there is no gap. Contents may have settled in transit. Your statutory rights are not affected.

The origins of this document lie in a beer-soaked afternoon conversation in the lobby bar of the Hilton Metropole, London, UK. Should this document play any future part in preserving human life or dignity, the authors recommend the installation of a small but elegant brass plaque, the text embossed upon which should naturally be encrypted.

3. Nomenclature

The following terms used in this document are intended in the sense described below, in the interests of avoiding ambiguity. The definitions presented here are abridged and tilted towards the subject matter of this document. For more exhaustive treatment please consult [RFC1034] and [RFC1035].

Authority Server A DNS component that provides authoritative responses on behalf of a Zone Manager, typically in response to queries received from Recursive Resolvers (q.v.); also known as "authoritative server" and "authoritative-only server".

Recursive Resolver A DNS component that finds answers for queries on behalf of a Stub Resolver (q.v.). A Recursive resolver draws upon data stored in a local cache and fills in where necessary using an iterative process of sending relevant queries to Authority Servers. A Recursive Resolver may be located on the same host as its dependant Stub Resolver, or it may be located on a different host and be used remotely across a network by multiple Stub Resolvers.

Stub Resolver A DNS component, present on a host used locally by an end user, that sends DNS queries to and receives responses from a Recursive Resolver (q.v.)

Zone Manager The party responsible for the contents of a DNS zone, and consequently (directly or indirectly) for the provisioning of the Authority Servers (q.v.) for that zone.

The following terms are specific to this proposal, and are used in this document accordingly. These are not terms commonly used within the taxonomy of DNS. See Section 4 for more details.

Entry Resolver A component of a Recursive Resolver service which accepts queries from a Stub Resolver, encrypts the query towards one or more Relay Resolvers and an Exit Resolver, and forwards towards the first Relay Resolver.

Relay Resolver A component of a Recursive Resolver service that accepts an encrypted query from an Entry Resolver, decrypts it and forwards it to the next Relay Resolver.

Exit Resolver The last Relay Resolver in a chain of Relay Resolvers.

4. General Approach

For the purposes of this document, consider that the network path between a Stub Resolver and a Recursive Resolver is entirely trustworthy. The Recursive Resolver might run on the same host as the Stub Resolver, for example, or might lie within the same trust perimeter as the Stub Resolver in an enterprise network.

A query received by an Entry Resolver is assigned a chain of Relay Resolvers, the number and choice of which are decided according to local policy. The Entry Resolver must have available a public key and knowledge of and capability of using the appropriate corresponding encryption algorithm for each selected Relay Resolver along the chain.

```
  ,------.              ,----------------.
  | Stub | ----===----> | Entry Resolver |
  `------'      ^       `----------------'
                |
                `---- [ Standard-format DNS request ]
```

An Entry Resolver generates a symmetric session key and encrypts it towards the Exit Resolver.

The Entry Resolver encrypts the query once for each Relay Resolver on the selected chain, in order.

The Entry Resolver constructs a package that consists of the encrypted session key and the multiply-encrypted (onion-wrapped) query and forwards it to the first Relay Resolver.

```
  ,----------------.              ,------------------.
  | Entry Resolver | ----===----> | Relay Resolver 1 |
  `----------------'      ^       `------------------'
                          |
                          `---- [ encrypted session key,
                                  onion-wrapped query ]
```

Each Relay Resolver in the chain decrypts the query and identifies from the result the next Relay Resolver in the chain. The source of the query from the Relay Resolver's perspective (the Entry Resolver, or the previous Relay Resolver) is encrypted towards the Relay Resolver itself and included in the package with the peeled query. The resulting package is forwarded to the next Relay Resolver in the chain.

```
   ,------------------.              ,------------------.
   | Relay Resolver 1 | ----===----> | Relay Resolver 2 |
   `------------------'      ^       `------------------'
                             |
                             `---- [ encrypted session key,
                                     onion-wrapped query (peeled once),
                                     onion-wrapped path (wrapped once) ]
```

A Relay Resolver that receives a package in which the query component is encrypted only towards itself is the Exit Resolver for this chain. The Relay Resolver decrypts the query and follows the normal DNS Resolver process to obtain a response.

```
     ,---------------.              ,-------------------------.
     | Exit Resolver | ----===----> | Various Authorit Servers |
     `---------------'      ^       `-------------------------'
              |             |
       ( local cache )      `---- [ standard-format DNS requests ]
```

The Exit Resolver obtains the session key from the original package and encrypts the response using it. The Exit Resolver then sends the package back down the chain, each Relay Resolver in turn identifying the next hop for the package using the directions it encrypted on the forward path.

```
     ,----------------------.             ,---------------.
     | Relay Resolver [i-1] | <---===---- | Exit Resolver |
     `----------------------'      ^      `---------------'
                                   |
                                   `--- [ encrypted response,
                                           onion-wrapped path ]
```

Once the response is received by the Entry resolver, it is decrypted with the session key and a response is dispatched to the original stub resolver.

```
     ,---------------.             ,----------------.
     | Stub Resolver | <---===---- | Entry Resolver |
     `---------------'      ^      `----------------'
                            |
                            `--- [ standard-format DNS response ]
```

5. Operational Considerations

An Entry Resolver might be primed with information about a large number of candidate Relay Resolvers, together with local policy relating to the minimum chain length required for particular (or, e.g., any) outbound queries. An Entry Resolver might build random chains from the available pool of Entry Resolvers and select between them when dealing with particular queries.

Care should be taken when re-using session keys for particular Exit Resolvers, since repeated use of the same session keys might be used to identify that different queries originate from the same user. A sufficiently large pool of candidate chains might provide an opportunity for session key regeneration in parallel to query processing.

An Entry Resolver might be configured to send padding queries down particular chains (e.g. CHAOS-class queries that can be resolved cheaply on an Exit Resolver) in order to reduce the opportunity to compare query frequency between different Resolver Relays and make inferences about chain construction by particular Entry Resolvers.

All Relay Resolvers ought to be usable as Exit Resolvers, and hence every Relay Resolver has an opportunity to build and maintain a DNS cache in the manner of a conventional DNS Resolver. The cache of course will only be used in the event that a particular Relay Resolver is acting as an Exit Resolver for a particular chain.

It should be expected that particular Relay Resolvers will become unavailable from time to time, e.g. due to scheduled maintenance or unexpected device failure. Entry Resolvers should time out and retry in the event that a chain is broken, and should take observed failure into account when building candidate chains for use for queries yet to be sent.

There is no requirement for the communication between Entry Resolvers and Relay Resolvers, or between Relay Resolvers, to use the DNS protocol. We might imagine that communication being made using modern APIs and dynamically-provisioned pools of TCP sessions, for example. The only requirement for the standard DNS protocol is between the Stub Resolver and the Entry Resolver, and between the Exit Resolver and Authority Servers.

6. Security Considerations

This document describes an approach for improving the privacy of the DNS, reducing opportunities to map an end user identity to data present in the DNS queries triggered by end user behaviour.

This document does not include an assessment of the impact of the proposed approach on the use of the DNS to launch denial of service (or other) attacks. Such analysis seems prudent to include in future revisions of this document, should there be interest in proceeding with it.

The ability of a chain of Relay Resolvers to provide privacy for an Entry Resolver depends on choosing a chain that crosses privacy domains (e.g. organisational or geopolitical boundaries). This document is missing guidance on how this might be done reliably.

7. IANA Considerations

This document makes no request of the IANA.

8. Acknowledgements

Many aspects of the approach described in this document are similar or identical to the approach taken in the design and implemnetation of The Onion Router [Tor], a project which has produced software that is widely-used to protect end-user privacy.

Also, your name here, etc.

9. References

9.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

9.2. Informative References

[I-D.bortzmeyer-dnsop-dns-privacy]
Bortzmeyer, S., "DNS privacy problem statement", draft-bortzmeyer-dnsop-dns-privacy-01 (work in progress), December 2013.

[I-D.koch-perpass-dns-confidentiality]
Koch, P., "Confidentiality Aspects of DNS Data, Publication, and Resolution", draft-koch-perpass-dns-confidentiality-00 (work in progress), November 2013.

[Tor] Dingledine, R. and N. Mathewson, "Tor Protocol Specification".

Appendix A. Editorial Notes

This section (and sub-sections) to be removed prior to publication.

A.1. Change History

00 Initial idea, circulated for the purposes of entertainment.

Authors' Addresses

Roy Arends
Nominet
Sandford Gate
Sandy Lane West
Oxford OX4 6LB
UK

Email: roy@nominet.org.uk

Joe Abley
Dyn, Inc.
470 Moore Street
London, ON N6C 2C2
Canada

Phone: +1 519 670 9327
Email: jabley@dyn.com

Joao Luis Silva Damas
Dyn, Inc.
Avenida de la Albufera 14
San Sebastian de los Reyes, Madrid 28701
Spain

Email: jdamas@dyn.com