

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 27 December 2020

E. Rescorla
Mozilla
J. Livingood
Comcast
25 June 2020

CNAME Discovery of Local DoH Resolvers
draft-rescorla-doh-cdisco-00

Abstract

This note describes a simple mechanism for determining whether an Internet Service Provider (ISP) network is operating a DNS over HTTPS [RFC8484] server on it for users connected to that network.

Discussion Venues

This note is to be removed before publishing as an RFC.

Source for this draft and an issue tracker can be found at <https://github.com/ekr/draft-rescorla-doh-cdisco> (<https://github.com/ekr/draft-rescorla-doh-cdisco>).

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 27 December 2020.

Copyright Notice

Copyright (c) 2020 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	2
2.	Conventions and Definitions	3
3.	DoH Resolver Discovery	3
3.1.	Why DNS?	4
3.2.	Why a CNAME?	5
4.	Security Considerations	6
5.	IANA Considerations	6
6.	References	6
6.1.	Normative References	6
6.2.	Informative References	7
	Acknowledgments	8
	Authors' Addresses	8

[1.](#) Introduction

Some applications perform their own name resolution rather than using the system resolver, typically using an encrypted protocol such as DoH [[RFC8484](#)]. These applications have the choice of using either the same recursive resolver configured into the system or of using a resolver chosen out of a preconfigured list of trusted resolvers in an application, such as in [[DOHTRR](#)].

If all of the trusted resolvers are publicly available, then there are a number of mechanisms for choosing between them, for instance randomly or based on performance.

[[I-D.arkko-abcd-distributed-resolver-selection](#)] describes a number of potential mechanisms. However, if the list of trusted resolvers includes Internet Service Providers (ISPs) and the client is on a network associated with such a provider, then it may be desirable to preferentially select the resolver associated with that provider. This provides the benefits both of using a DNS resolver with a known policy and using a resolver that has high quality local information

about the local network topology.

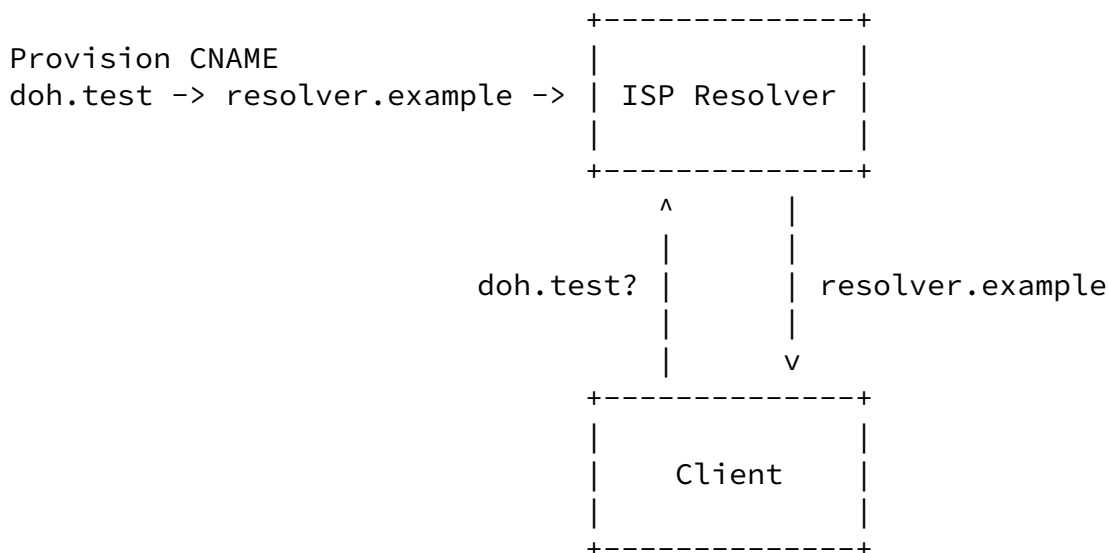
This document describes a mechanism to address this situation. This mechanism is being tested in the Firefox browser with Comcast's resolvers.

2. Conventions and Definitions

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.

3. DoH Resolver Discovery

The basic mechanism described in this document is straightforward and has been chosen for ease of implementation rather than architectural correctness.



A network provider can publish the fact that it has an associated DoH resolver on its network by configuring its own resolvers to serve a CNAME record at a well known domain name which cannot be otherwise registered. The current test deployment uses "doh.test" (see [[RFC2606](#)] for the definition of .test). This CNAME points to the domain name of the associated DoH resolver ("resolver.example" in the

diagram above).

[[OPEN ISSUE: doh.test is probably the wrong domain. We may pick something else later.]]

A client which wishes to test for the presence of a DoH resolver on the network takes the following steps:

1. Do any testing for whether DoH should be disabled, such as looking for canary domains [[I-D.grover-add-policy-detection](#)] or checking for local enterprise configuration.

2. Do a CNAME query for "doh.test" using either the system resolver or by talking directly to the recursive resolver IP address configured into the system.
3. If the query succeeds, then look up the CNAME record value in the list of preconfigured resolvers. If an exact match is found, then use the resolver address for the matching preconfigured resolver. Otherwise fall back to the ordinary DoH resolver selection logic.
4. If the query fails, then no associated resolver is present; fall back to the ordinary DoH resolver selection logic.

As noted above, this mechanism was designed for ease of implementation.

Comcast's resolvers and authoritative servers have been configured with some additional records to support the Firefox applications and potential future applications. The DNS behavior is as follows, where example.com is the domain used for naming provider services:

1. doh.test IN CNAME doh-discovery.example.com
2. doh-discovery.example.com must have at least one A and/or AAAA RR (address does not matter - can be 127.0.0.1)
3. doh-discovery.example.com IN URI https://doh.example.com/dns-query (the ISP DoH URI - not currently used by Firefox as the URI is preconfigured in the application)

The next few sections describe the reasoning for some of the design choices.

Considering that many applications do not act as a DNS client and instead use platform functions such as `getaddrinfo`, the domain of the associated resolver SHOULD also have an A record, so the call to `getaddrinfo` does not fail.

[3.1.](#) Why DNS?

There have been a number of discussions of using non-DNS mechanisms resolver information, for instance as in Section 4 of [\[I-D.pauly-add-resolver-discovery\]](#). While arguably more architecturally correct in terms of layering, they have a number of deployment drawbacks:

- * They require the client to have much tighter integration with the operating system in order to query the data. By contrast, with this mechanism, the client need only be able to do name resolution via the system resolver, which it generally already is able to do via standard APIs.
- * They require new types of configuration which ISPs may not already be set up to do. By contrast, configuring DNS records is generally well understood.
- * They rely on intermediate devices (e.g., NATs) being aware of the configuration information and passing it onto clients. These devices already do this with DNS information.

For these reasons, DNS seems to be the easiest solution to deploy quickly.

[3.2.](#) Why a CNAME?

Most other proposed designs (e.g., [\[I-D.pp-add-resinfo\]](#) and [\[I-D.pp-add-stub-upgrade\]](#), and [\[I-D.pauly-add-resolver-discovery\]](#)) use new RRtypes. While this may be the right answer eventually, it

is less convenient for immediate deployment, for several reasons:

1. It is somewhat more difficult (though not impossible) to look up new RRTypes on the client and provision them on the ISP resolver.
2. Some consumer-grade middleboxes (e.g., WiFi routers) may block unknown RRTypes. The data here is quite old and limited, but still not particularly promising.

The choice to use a CNAME does have one major drawback: it does not let us provide the URL template but only the name of the resolver. This is not a problem for our system in practice because Firefox will only connect to resolvers on a preconfigured list and thus will use the CNAME as a lookup key for that list. The Mozilla team is working to measure the rate of new RRTYPE interference and may revise this approach depending on the results of that.

[[OPEN ISSUE: We are working to measure the rate of new RRTYPE interference and may revise this approach depending on the results of that.]]

[4.](#) Security Considerations

Because the initial request for discovery is done over insecure DNS (Do53), a local attacker or malicious local resolver can substitute their own response. However, because this mechanism only selects from a list of preconfigured trusted resolvers, an attacker can only redirect you to a different resolver out of that list, which by definition is also trusted. Note: the URI field potentially has different security properties depending on how it is used. As noted above; Firefox does not use it.

If the server which is redirected to is not publicly available, this mechanism can be used as a DoS attack. Application clients should test the selected server before committing to it and otherwise fall back to their ordinary DoH selection logic.

Any local discovery mechanism has potential privacy impacts: suppose that a user uses their mobile device on ISP A, which redirects it to their own resolver, and ISP B which does not. In that case, the user's DNS queries will be spread over both ISP A's resolver and one of the public trusted resolvers, which could have an impact on the user's privacy. This has to be balanced against the improvement obtained by using a local resolver and the level of metadata leakage that currently occurs to the ISP, but can be mitigated through trusted recursive resolver policies.

[5.](#) IANA Considerations

This document has no IANA actions.

[6.](#) References

[6.1.](#) Normative References

[I-D.grover-add-policy-detection]

Grover, A. and P. Saint-Andre, "DNS Resolver-Based Policy Detection Domain", Work in Progress, Internet-Draft, [draft-grover-add-policy-detection-00](#), 8 July 2019, <<http://www.ietf.org/internet-drafts/draft-grover-add-policy-detection-00.txt>>.

[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>>.

[RFC2606] Eastlake 3rd, D. and A. Panitz, "Reserved Top Level DNS Names", [BCP 32](#), [RFC 2606](#), DOI 10.17487/RFC2606, June 1999, <<https://www.rfc-editor.org/info/rfc2606>>.

[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>>.

- [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>>.

6.2. Informative References

- [DOHTRR] Mozilla, ., "Trusted Recursive Resolver", n.d., <https://wiki.mozilla.org/Trusted_Recursive_Resolver>.
- [I-D.arkko-abcd-distributed-resolver-selection] Arkko, J., Thomson, M., and T. Hardie, "Selecting Resolvers from a Set of Distributed DNS Resolvers", Work in Progress, Internet-Draft, [draft-arkko-abcd-distributed-resolver-selection-01](#), 9 March 2020, <<http://www.ietf.org/internet-drafts/draft-arkko-abcd-distributed-resolver-selection-01.txt>>.
- [I-D.pauly-add-resolver-discovery] Pauly, T., Kinnear, E., Wood, C., McManus, P., and T. Jensen, "Adaptive DNS Resolver Discovery", Work in Progress, Internet-Draft, [draft-pauly-add-resolver-discovery-00](#), 20 May 2020, <<http://www.ietf.org/internet-drafts/draft-pauly-add-resolver-discovery-00.txt>>.
- [I-D.pp-add-resinfo] Sood, P. and P. Hoffman, "DNS Resolver Information Self-publication", Work in Progress, Internet-Draft, [draft-pp-add-resinfo-01](#), 14 May 2020, <<http://www.ietf.org/internet-drafts/draft-pp-add-resinfo-01.txt>>.
- [I-D.pp-add-stub-upgrade] Sood, P. and P. Hoffman, "Upgrading Communication from Stub Resolvers to DoT or DoH", Work in Progress, Internet-Draft, [draft-pp-add-stub-upgrade-01](#), 14 May 2020, <<http://www.ietf.org/internet-drafts/draft-pp-add-stub-upgrade-01.txt>>.

TODO acknowledge.

Authors' Addresses

Eric Rescorla
Mozilla

Email: ekr@rtfm.com

Jason Livingood
Comcast

Email: jason_livingood@comcast.com