Workgroup: ADD
Internet-Draft: draft-box-add-requirements-00
Published: September 4, 2020
Intended Status: Informational
Expires: March 8, 2021
Authors: C. Box T. Pauly C.A. Wood T. Reddy
BT Apple Cloudflare McAfee

Requirements for Adaptive DNS Discovery

Abstract

Adaptive DNS Discovery is chartered to define mechanisms that allow clients to discover and select encrypted DNS resolvers. This document describes several use cases for discovering DNS resolvers that support encrypted transports, and lists requirements that any proposed discovery and selection mechanisms should address.

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/ietf-wg-add/draft-add-requirements.

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 March 8, 2021.

Table of Contents

1. Introduction

Several protocols for protecting DNS traffic with encrypted transports have been defined, such as DNS-over-TLS (DoT) [RFC7858] and DNS-over-HTTPS (DoH) [RFC8484]. Encrypted DNS can provide many security and privacy benefits for network clients.

While it is possible for clients to hard-code encrypted DNS resolvers to use, dynamic discovery and provisioning of encrypted resolvers can expand the usefulness and applicability of encrypted DNS to many more use cases.

The Adaptive DNS Discovery (ADD) Working Group is chartered to define mechanisms that allow clients to automatically discover and select encrypted DNS resolvers in a wide variety of network environments. This document describes several use cases for discovering DNS resolvers that support encrypted transports, and lists requirements that any proposed discovery and selection mechanisms should address. They can do this either by providing a solution, or by explicitly stating why it is not in scope.

Use cases are described between Section 3 and Section 4.2. Each use case contains a narrative and a set of requirements that apply in that case. There are additional common requirements in Section 5. Each requirement is identified as "Ra.b" where a is the group number and b is the number within that group. Both a and b are integers starting with 1.

A summary of all requirements in listed in Section 6.

1.1. Requirements Language

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.

2. Terminology

This document makes use of the following terms.

Encrypted DNS: DNS-over-HTTPS [RFC8484], DNS-over-TLS [RFC7858], or any other encrypted DNS technology that the IETF may publish, such as DNS-over-QUIC [I-D.ietf-dprive-dnsoquic].

Associated resolver: A resolver operated by the same entity that provides the resolver the client started with. See Section 3.

Equivalent associated resolver: An associated resolver that provides DNS responses that are identical to the ones served by the original unencrypted resolver.

Alternative associated resolver: An associated resolver that serves different responses to some queries; see Section 3 for examples.

3. Discovery of associated resolvers

A client may begin with information about unencrypted resolvers from the attached networks (Section 3.1), and/or unencrypted resolvers known from configuration (Section 3.2). This information may be used to then discover one or more associated encrypted resolvers.

Associated resolvers are defined as resolvers operated by the same entity that provides the resolver the client started with. Such associated resolvers may come in two forms:

1. Equivalent - these provide DNS responses that are identical to the ones served by the unencrypted resolver.

2. Alternative - these serve different responses to some queries. For example one entity may offer a set of encrypted resolvers with different levels of filtering (none, just malware, or malware & adult content), or different proximity (local or central).

The client may wish to select an equivalent associated resolver, or select one of the alternatives.

Designs for resolver upgrade mechanisms can either add new parameters to existing provisioning mechanisms (for example, adding necessary information to use DoT or DoH to options in DHCP, RAs, or IKEv2) or else provide a way to communicate with a provisioned unencrypted DNS resolver and discover the associated encrypted DNS resolvers.

Requirement	Description
R1.1	There must be a mechanism for a client to learn the set of encrypted resolvers that are associated with an unencrypted resolver.
R1.2	Discovery must be possible even when the IP address of the encrypted resolver is only valid locally.

Table 1

3.1. Network-provisioned resolvers

DNS servers are often provisioned by a network as part of DHCP options [RFC2132], IPv6 Router Advertisement (RA) options [RFC8106], Point-to-Point Protocol (PPP) [RFC1877], 3GPP Protocol Configuration Options, or another mechanism. Historically this is usually one or more DNS resolver IP addresses, to be used for traditional unencrypted DNS. However it could also be a richer set of information.

Using an encrypted and authenticated resolver that is associated to the one provisioned by the network can provide several benefits that are not possible if only unencrypted DNS is used:

*Prevent other devices on the network from observing client DNS messages

*Verify that answers come from the selected DNS resolver

*Authenticate that the DNS resolver is the one provisioned by the network

Frequently, network-provisioned resolvers are forwarders running on a local router. The discovered encrypted resolvers in these cases may either be local forwarders themselves, or an associated resolver that is in the network (thus bypassing the router's DNS forwarder).

Requirement	Description
R2.1	Example requirement

Table 2

3.1.1. Unencrypted forwarder

If the resolver announced by the network is a classic unencrypted forwarder, it is frequently the case that such forwarders are difficult to upgrade to support encrypted operation. In such cases it is useful for the resolver provider to be able to declare which encrypted resolvers they provide, and for the client to be able to discover them. If the client wishes to, it can then use one of those resolvers and bypass the local forwarder.

Requirement	Description
R3.1	Example requirement

Table 3

3.1.2. Encrypted forwarder

If a subset of local resolvers supports encrypted DNS, the client may not initially be aware that its local resolver supports it. Discovering this may require communication with the local resolver, or an upstream resolver, over an unencrypted transport. Once discovered, the local encrypted forwarder may be selected by the client, gaining the benefits of encryption while retaining the benefits of a local caching forwarder with knowledge of the local topology.

Another benefit occurs with IoT devices. A common usage pattern for such devices is for it to "call home" to a service that resides on the public Internet, where that service is referenced through a domain name (A or AAAA record). As discussed in Manufacturer Usage Description Specification [RFC8520], because these devices tend to require access to very few sites, all other access should be considered suspect. However, if the query is not accessible for inspection, it becomes quite difficult for the infrastructure to suspect anything.

Requirement	Description
R4.1	Example requirement

Table 4

3.2. Client-selected resolvers

Client devices often allow the device administrator to select a specific DNS resolver to use on certain networks, or on all networks. Historically, this selection was specified only with an IP address.

Discovering which equivalent encrypted resolvers are offered by the same entity allows the client to "upgrade" connections to use encrypted DNS. This can provide several benefits:

- Prevent devices along the network path to the selected resolver from observing client DNS messages
- Verify that answers come from the selected DNS resolver
- Authenticate that the DNS resolver is the one selected by the client

In doing so it is critical that the new resolver is an equivalent resolver. Switching to a non-equivalent alternative resolver would break the expectation of the user who previously selected that resolver.

Requirement	Description
R5.1	Example requirement

Table 5

3.3. VPN resolvers

Virtual Private Networks (VPNs) also can provision DNS resolvers. In addition to being able to use DHCP or RAs, VPNs can provision DNS information in an explicit configuration message. For example, IKEv2 can provision DNS servers using Configuration Attributes [RFC7296].

VPNs can also configure Split DNS rules to limit the use of the configured resolvers to specific domain names [RFC8598].

Discovering an encrypted resolver that is provisioned by a VPN can provide the same benefits as doing so for a local network, but applied to the private network. When using Split DNS, it becomes possible to use one encrypted resolver for private domains, and another for other domains.

Requirement	Description
R6.1	Example requirement

Table 6

4. Discovery of limited domain resolvers

Similar to how VPN DNS configurations can use Split DNS for private names, other network environments can support resolution of names that are specific to the local environment. For example, an enterprise-managed Wi-Fi network might be able to access both the Internet and a private intranet. In such a scenario, the private domains managed by the enterprise might only be resolvable using a specific DNS resolver.

Discovering an encrypted resolver for a subset of names allows a client to perform Split DNS while maintaining the benefits of encrypted DNS. For example, a client could use a client-selected encrypted resolver for public domains, but use a different encrypted resolver for enterprise-private domains.

Such domain-specific resolver discovery mechanisms additionally need to provide some information about the applicability and capabilities of encrypted resolvers. This information can either be provisioned or can be discovered based on clients actively trying to access content.

Requirement	Description
R7.1	Example requirement

Table 7

4.1. Discover a mapping between a locally-hosted domain and a resolver

Narrative required.

Requirement	Description
R8.1	Example requirement

Table 8

4.1.1. Encrypted resolvers for local or home content

Accessing locally-hosted content can require the use of a specific resolver. For example, captive networks or networks with walled-garden content like media on airplane Wi-Fi networks can rely on using a resolver hosted on the local network.

In cases where a client is using an encrypted resolver provisioned by a network, and that encrypted resolver is able to resolve names of local content, this can fall into the use case described in Section 3.1. However, it might be necessary to discover a local encrypted resolver along with specific domains if:

* the network-provisioned encrypted resolver is not able to resolve local-only names, or

*the client has a more-preferred encrypted resolver for generic traffic, and would otherwise not be able to access local content

The first point can occur in a hybrid deployment, e.g. when the local resolver is unencrypted but a central one is encrypted. Clients choosing the encrypted resolver for most queries will need to be advised to refer to the local one for some names.

This case also include accessing content specific to a home network.

Requirement	Description
R9.1	Example requirement

Table 9

4.1.2. Locally-cached content

Narrative required.

Requirement	Description
R10.1	Example requirement

Table 10

4.1.3. Private enterprise names

As stated above, an enterprise-managed Wi-Fi network might be able to access both the Internet and a private intranet. The private domains managed by the enterprise might only be resolvable using a specific DNS resolver, hence use of that resolver is essential for such domains. However it does not necessarily mean that all queries for all domains have to flow through that resolver.

Only sending the necessary queries through the enterprise resolver, and not generic Internet queries, has the privacy benefit of only exposing traffic to the enterprise that fall within a limited set of domains.

Using encrypted DNS for private names also opens up the possibility of doing private name resolution outside of the content of a VPN or managed network. If the DNS resolver authenticates clients, it can offer its resolver for private names on a publicly accessible server, while still limiting the visibility of the DNS traffic.

Requirement	Description
R11.1	Example requirement

Table 11

4.2. Encrypted resolvers for content providers

Content Delivery Networks (CDNs), and content-providers more broadly, can also provide encrypted DNS resolvers that can be used by clients over the public Internet. These resolvers can either allow resolution of all public names (like normal recursive resolvers), or be designed to serve a subset of names managed by the content provider (like an authoritative resolver). Using these resolvers can allow the content provider to directly control how DNS answers are used for load balancing and address selection, which could improve performance of connections to the content provider.

Using a content-provider's encrypted resolver can also provide several privacy and security benefits:

* Prevent devices along the network path to the content-provider's resolver from observing client DNS messages

* Verify that answers come from the entity that manages the domains being resolved

* Reduce the number of entities able to monitor the specific names accessed by a client to only the client and the content provider, assuming that the content provider would already see the names upon a secure connection later being made based on the DNS answers (e.g., in the TLS SNI extension)

Requirement	Description
R12.1	Example requirement

Table 12

5. Privacy and security requirements

Encrypted (and authenticated) DNS improves the privacy and security of DNS queries and answers in the presence of malicious attackers. Such attackers are assumed to interfere with or otherwise impede DNS traffic and corresponding discovery mechanisms. They may be on-path or off-path between the client and entities with which the client communicates [RFC3552]. These attackers can inject, tamper, or otherwise interfere with traffic as needed. Given these capabilities, an attacker may have a variety of goals, including, though not limited to:

* Monitor and profile clients by observing unencrypted DNS traffic

* Modify unencrypted DNS traffic to filter or augment the user experience

* Block encrypted DNS

Clients cannot assume that their network does not have such an attacker unless given some means of authenticating or otherwise trusting the communication with their DNS resolver.

Given this type of attacker, resolver discovery mechanisms must be designed carefully to not worsen a client's security or privacy posture. In particular, attackers must not be able to:

- Redirect secure DNS traffic to themselves when they would not otherwise handle DNS traffic.

- Override or interfere with the resolver preferences of a user or administrator.

- Cause clients to use a discovered resolver which has no authenticated delegation from a client-known entity.

- Influence automatic discovery mechanisms such that a client uses one or more resolvers that are not otherwise involved with providing service to the client, such as: a network provider, a VPN server, a content provider being accessed, or a server that the client has manually configured.

Beyond these requirements, standards describing resolver discovery mechanisms must not place any requirements on clients to select particular resolvers over others.

When discovering DNS resolvers on a local network, clients have no mechanism to distinguish between cases where an active attacker with the above capabilities is interfering with discovery, and situations wherein the network has no encrypted resolver. Absent such a mechanism, an attacker can always succeed in these goals. Therefore, in such circumstances, viable solutions for local DNS resolver discovery should consider weaker attackers, such as those with only passive eavesdropping capabilities. It is unknown whether such relaxations represent a realistic attacker in practice. Thus, local discovery solutions designed around this threat model may have limited value.

5.1. On opportunistic encryption

Opportunistic encrypted DNS, when the client cannot authenticate the entity that provides encrypted DNS, does not meet the requirements laid out here for resolver discovery. While opportunistic encryption can provide some benefits, specifically in reducing the ability for other entities to observe traffic, it is not a viable solution against an on-path attacker.

Performing opportunistic encrypted DNS does not require specific discovery mechanisms. Section 4.1 of [RFC7858] already describes how to use DNS-over-TLS opportunistically.

5.2. Handling exceptions and failures

Even with encrypted DNS resolver discovery in place, clients must be prepared to handle certain scenarios where encrypted DNS cannot be used. In these scenarios, clients must consider if it is appropriate to fail open by sending the DNS queries without encryption, fail closed by not doing so, or presenting a choice to a user or administrator. The exact behavior is a local client policy decision.

Some networks that use Captive Portals will not allow any Internet connectivity until a client has interacted with the portal [I-D.ietf-capport-architecture]. If these networks do not use encrypted DNS for their own resolution, a client will need to perform unencrypted DNS queries in order to get out of captivity. Many operating systems have specific client code responsible for detecting and interacting with Captive Portals; these system components may be good candidates for failing open, since they do not generally represent user traffic.

Other networks may not allow any use of encrypted DNS, or any use of encrypted DNS to resolvers other than a network-provisioned resolver. Clients should not silently fail open in these cases, but if these networks are trusted by or administered by the user, the user may want to specifically follow the network's DNS policy instead of what the client would do on an unknown or untrusted network.

6. Requirements Summary

This sections lists the complete set of requirements described above, for ease of reference.

Requirement	Description
R1.1	There must be a mechanism for a client to learn the set of encrypted resolvers that are associated with a resolver that is known only by its IP address.
R1.2	Discovery must be possible even when the IP address is only valid locally.
R1.3	More to be added
R2.1	Example requirement
R3.1	Example requirement

Table 13

7. Security Considerations

All security considerations relevant to a particular use case are described under that section. Additional considerations common to all of them are described in Section 5.

8. IANA Considerations

This document has no IANA actions.

9. References

9.1. Normative References

[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>.

[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>.

9.2. Informative References

[I-D.ietf-capport-architecture] Larose, K., Dolson, D., and H. Liu, "Captive Portal Architecture", Work in Progress, Internet-Draft, draft-ietf-capport-architecture-09, August 8, 2020, <http://www.ietf.org/internet-drafts/draft-ietf-capport-architecture-09.txt>.

[I-D.ietf-dprive-dnsoquic] Huitema, C., Mankin, A., and S. Dickinson, "Specification of DNS over Dedicated QUIC Connections", Work in Progress, Internet-Draft, draft-ietf-dprive-dnsoquic-00, April 27, 2020, <http://www.ietf.org/internet-drafts/draft-ietf-dprive-dnsoquic-00.txt>.

[RFC1877] Cobb, S., "PPP Internet Protocol Control Protocol Extensions for Name Server Addresses", RFC 1877, DOI 10.17487/RFC1877, December 1995, <https://www.rfc-editor.org/info/rfc1877>.

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, DOI 10.17487/RFC2132, March 1997, <https://www.rfc-editor.org/info/rfc2132>.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI

10.17487/RFC3552, July 2003, <https://www.rfc-editor.org/info/rfc3552>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <https://www.rfc-editor.org/info/rfc7296>.

[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>.

[RFC8106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 8106, DOI 10.17487/RFC8106, March 2017, <https://www.rfc-editor.org/info/rfc8106>.

[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>.

[RFC8520] Lear, E., Droms, R., and D. Romascanu, "Manufacturer Usage Description Specification", RFC 8520, DOI 10.17487/RFC8520, March 2019, <https://www.rfc-editor.org/info/rfc8520>.

[RFC8598] Pauly, T. and P. Wouters, "Split DNS Configuration for the Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 8598, DOI 10.17487/RFC8598, May 2019, <https://www.rfc-editor.org/info/rfc8598>.

Acknowledgments

This document was started based on contributions during the ADD meeting of IETF108, on the list, and from draft-pauly-add-requirements.

More contributions are required! Please consider starting a github issue, submit a pull request, or simply raise the topic on the ADD list.

Authors' Addresses

Chris Box
BT
2000 Park Avenue
Bristol
United Kingdom

Email: chris.box@bt.com

Tommy Pauly
Apple
One Apple Park Way
Cupertino, California 95014,
United States of America

Email: tpauly@apple.com

Christopher A. Wood
Cloudflare
101 Townsend St
San Francisco,
United States of America

Email: caw@heapingbits.net

Tirumaleswar Reddy
McAfee
Embassy Golf Link Business Park
Bangalore
India

Email: TirumaleswarReddy_Konda@McAfee.com