Discovery of Designated Resolvers
draft-ietf-add-ddr-08

Abstract

This document defines Discovery of Designated Resolvers (DDR), a mechanism for DNS clients to use DNS records to discover a resolver’s encrypted DNS configuration. An encrypted resolver discovered in this manner is referred to as a "Designated Resolver". This mechanism can be used to move from unencrypted DNS to encrypted DNS when only the IP address of a resolver is known. This mechanism is designed to be limited to cases where unencrypted resolvers and their designated resolvers are operated by the same entity or cooperating entities. It can also be used to discover support for encrypted DNS protocols when the name of an encrypted resolver is known.

Discussion Venues

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

Discussion of this document takes place on the Adaptive DNS Discovery Working Group mailing list (add@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/add/.

Source for this draft and an issue tracker can be found at https://github.com/ietf-wg-add/draft-ietf-add-ddr.

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 6 January 2023.

Copyright Notice

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction ............................................. 3
   1.1. Specification of Requirements ........................ 3
2. Terminology .............................................. 3
3. DNS Service Binding Records .............................. 4
4. Discovery Using Resolver IP Addresses .................... 5
   4.1. Use of Designated Resolvers .......................... 6
   4.1.1. Use of Designated Resolvers across network changes .. 7
   4.2. Verified Discovery .................................. 7
   4.3. Opportunistic Discovery .............................. 8
5. Discovery Using Resolver Names ........................... 9
6. Deployment Considerations .............................. 10
   6.1. Caching Forwarders .................................. 10
   6.2. Certificate Management .............................. 10
   6.3. Server Name Handling ................................ 10
   6.4. Handling non-DDR queries for resolver.arpa .......... 11
   6.5. Interaction with Network-Designated Resolvers ....... 11
7. Security Considerations ................................. 11
8. IANA Considerations ..................................... 12
   8.1. Special Use Domain Name "resolver.arpa" .............. 12
9. References .............................................. 13
   9.1. Normative References ................................ 13
   9.2. Informative References .............................. 14
Appendix A. Rationale for using SVCB records ............... 16
Authors’ Addresses ......................................... 17
1. Introduction

When DNS clients wish to use encrypted DNS protocols such as DNS-over-TLS (DoT) [RFC7858], DNS-over-QUIC (DoQ) [RFC9250], or DNS-over-HTTPS (DoH) [RFC8484], they require additional information beyond the IP address of the DNS server, such as the resolver’s hostname, non-standard ports, or URI templates. However, common configuration mechanisms only provide the resolver’s IP address during configuration. Such mechanisms include network provisioning protocols like DHCP [RFC2132] [RFC8415] and IPv6 Router Advertisement (RA) options [RFC8106], as well as manual configuration.

This document defines two mechanisms for clients to discover designated resolvers using DNS server Service Binding (SVCB, [I-D.ietf-dnsop-svcb-https]) records:

1. When only an IP address of an Unencrypted Resolver is known, the client queries a special use domain name (SUDN) [RFC6761] to discover DNS SVCB records associated with one or more Encrypted Resolvers the Unencrypted Resolver has designated for use when support for DNS encryption is requested (Section 4).

2. When the hostname of an Encrypted Resolver is known, the client requests details by sending a query for a DNS SVCB record. This can be used to discover alternate encrypted DNS protocols supported by a known server, or to provide details if a resolver name is provisioned by a network (Section 5).

Both of these approaches allow clients to confirm that a discovered Encrypted Resolver is designated by the originally provisioned resolver. "Designated" in this context means that the resolvers are operated by the same entity or cooperating entities; for example, the resolvers are accessible on the same IP address, or there is a certificate that claims ownership over the IP address for the original designating resolver.

1.1. Specification of Requirements

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 defines the following terms:
DDR: Discovery of Designated Resolvers. Refers to the mechanisms defined in this document.

Designated Resolver: A resolver, presumably an Encrypted Resolver, designated by another resolver for use in its own place. This designation can be verified with TLS certificates.

Encrypted Resolver: A DNS resolver using any encrypted DNS transport. This includes current mechanisms such as DoH, DoT, and DoQ, as well as future mechanisms.

Unencrypted Resolver: A DNS resolver using TCP or UDP port 53 without encryption.

3. DNS Service Binding Records

DNS resolvers can advertise one or more Designated Resolvers that may offer support over encrypted channels and are controlled by the same entity.

When a client discovers Designated Resolvers, it learns information such as the supported protocols and ports. This information is provided in ServiceMode Service Binding (SVCB) records for DNS Servers, although AliasMode SVCB records can be used to direct clients to the needed ServiceMode SVCB record per [I-D.ietf-dnsop-svcb-https]. The formatting of these records, including the DNS-unique parameters such as "dohpath", are defined by [I-D.ietf-add-svcb-dns].

The following is an example of an SVCB record describing a DoH server discovered by querying for _dns.example.net:

```dns
_dns.example.net. 7200 IN SVCB 1 example.net. (alpn=h2 dohpath=/dns-query{?dns})
```

The following is an example of an SVCB record describing a DoT server discovered by querying for _dns.example.net:

```dns
_dns.example.net. 7200 IN SVCB 1 dot.example.net (alpn=dot port=8530)
```

The following is an example of an SVCB record describing a DoQ server discovered by querying for _dns.example.net:

```dns
_dns.example.net. 7200 IN SVCB 1 doq.example.net (alpn=doq port=8530)
```
If multiple Designated Resolvers are available, using one or more encrypted DNS protocols, the resolver deployment can indicate a preference using the priority fields in each SVCB record [I-D.ietf-dnsop-svcb-https].

If the client encounters a mandatory parameter in an SVCB record it does not understand, it MUST NOT use that record to discover a Designated Resolver. The client can still use others records in the same response if the client can understand all of their mandatory parameters. This allows future encrypted deployments to simultaneously support protocols even if a given client is not aware of all those protocols. For example, if the Unencrypted Resolver returns three SVCB records, one for DoH, one for DoT, and one for a yet-to-exist protocol, a client which only supports DoH and DoT should be able to use those records while safely ignoring the third record.

To avoid name lookup deadlock, Designated Resolvers SHOULD follow the guidance in Section 10 of [RFC8484] regarding the avoidance of DNS-based references that block the completion of the TLS handshake.

This document focuses on discovering DoH, DoT, and DoQ Designated Resolvers. Other protocols can also use the format defined by [I-D.ietf-add-svcb-dns]. However, if any such protocol does not involve some form of certificate validation, new validation mechanisms will need to be defined to support validating designation as defined in Section 4.2.

4. Discovery Using Resolver IP Addresses

When a DNS client is configured with an Unencrypted Resolver IP address, it SHOULD query the resolver for SVCB records for the name "resolver.arpa" before making other queries. Specifically, the client issues a query for _dns.resolver.arpa with the SVCB resource record type (64) [I-D.ietf-dnsop-svcb-https].

Because this query is for an SUDN, which no entity can claim ownership over, the ServiceMode SVCB response MUST NOT use the "." value for the TargetName. Instead, the domain name used for DoT/DoQ or used to construct the DoH template MUST be provided.

The following is an example of an SVCB record describing a DoH server discovered by querying for _dns.resolver.arpa:

```plaintext
_dns.resolver.arpa. 7200 IN SVCB 1 doh.example.net (alpn=h2 dohpath=/dns-query{?dns})
```

Pauly, et al. Expires 6 January 2023
The following is an example of an SVCB record describing a DoT server discovered by querying for _dns.resolver.arpa:

_dns.resolver.arpa.  7200  IN  SVCB 1 dot.example.net (alpn=dot port=8530)

The following is an example of an SVCB record describing a DoQ server discovered by querying for _dns.resolver.arpa:

_dns.resolver.arpa.  7200  IN  SVCB 1 doq.example.net (alpn=doq port=8530)

If the recursive resolver that receives this query has one or more Designated Resolvers, it will return the corresponding SVCB records. When responding to these special queries for "resolver.arpa", the recursive resolver SHOULD include the A and AAAA records for the name of the Designated Resolver in the Additional Answers section. This will save the DNS client an additional round trip to retrieve the address of the designated resolver; see Section 5 of [I-D.ietf-dnsop-svcb-https].

Designated Resolvers SHOULD be accessible using the IP address families that are supported by their associated Unencrypted Resolvers. If an Unencrypted Resolver is accessible using an IPv4 address, it ought to provide an A record for an IPv4 address of the Designated Resolver; similarly, if it is accessible using an IPv6 address, it ought to provide a AAAA record for an IPv6 address of the Designated Resolver. The Designated Resolver can support more address families than the Unencrypted Resolver, but it ought not to support fewer. If this is not done, clients that only have connectivity over one address family might not be able to access the Designated Resolver.

If the recursive resolver that receives this query has no Designated Resolvers, it SHOULD return NODATA for queries to the "resolver.arpa" SUDN.

4.1. Use of Designated Resolvers

When a client discovers Designated Resolvers from an Unencrypted Resolver IP address, it can choose to use these Designated Resolvers either automatically, or based on some other policy, heuristic, or user choice.

This document defines two preferred methods to automatically use Designated Resolvers:

Pauly, et al. Expires 6 January 2023
* Verified Discovery (Section 4.2), for when a TLS certificate can be used to validate the resolver’s identity.

* Opportunistic Discovery (Section 4.3), for when a resolver’s IP address is a private or local address.

A client MAY additionally use a discovered Designated Resolver without either of these methods, based on implementation-specific policy or user input. Details of such policy are out of scope of this document. Clients MUST NOT automatically use a Designated Resolver without some sort of validation, such as the two methods defined in this document or a future mechanism.

A client MUST NOT re-use a designation discovered using the IP address of one Unencrypted Resolver in place of any other Unencrypted Resolver. Instead, the client SHOULD repeat the discovery process to discover the Designated Resolver of the other Unencrypted Resolver. In other words, designations are per-resolver and MUST NOT be used to configure the client’s universal DNS behavior. This ensures in all cases that queries are being sent to a party designated by the resolver originally being used.

4.1.1. Use of Designated Resolvers across network changes

If a client is configured with the same Unencrypted Resolver IP address on multiple different networks, a Designated Resolver that has been discovered on one network SHOULD NOT be reused on any of the other networks without repeating the discovery process for each network.

However, if a given Unencrypted Resolver designates a Designated Resolver that does not use a private or local IP address and can be verified using the mechanism described in Section 4.2, it MAY be used on different network connections so long as the subsequent connections over other networks can also be successfully verified using the mechanism described in Section 4.2. This is a tradeoff between performance (by having no delay in establishing an encrypted DNS connection on the new network) and functionality (if the Unencrypted Resolver intends to designate different Designated Resolvers based on the network from which clients connect).

4.2. Verified Discovery

Verified Discovery is a mechanism that allows automatic use of a Designated Resolver that supports DNS encryption that performs a TLS handshake.
In order to be considered a verified Designated Resolver, the TLS certificate presented by the Designated Resolver needs to pass the following checks made by the client:

1. The client MUST verify the chain of certificates up to a trust anchor as described in Section 6 of [RFC5280]. This SHOULD use the default system or application trust anchors.

2. The client MUST verify that the certificate contains the IP address of the designating Unencrypted Resolver in a subjectAltName extension.

If these checks pass, the client SHOULD use the discovered Designated Resolver for any cases in which it would have otherwise used the Unencrypted Resolver.

If these checks fail, the client MUST NOT automatically use the discovered Designated Resolver. Additionally, the client SHOULD suppress any further queries for Designated Resolvers using this Unencrypted Resolver for the length of time indicated by the SVCB record’s Time to Live (TTL).

If the Designated Resolver and the Unencrypted Resolver share an IP address, clients MAY choose to opportunistically use the Designated Resolver even without this certificate check (Section 4.3).

If resolving the name of a Designated Resolver from an SVCB record yields an IP address that was not presented in the Additional Answers section or ipv4hint or ipv6hint fields of the original SVCB query, the connection made to that IP address MUST pass the same TLS certificate checks before being allowed to replace a previously known and validated IP address for the same Designated Resolver name.

4.3. Opportunistic Discovery

There are situations where Verified Discovery of encrypted DNS configuration over unencrypted DNS is not possible. This includes Unencrypted Resolvers on private IP addresses [RFC1918], Unique Local Addresses (ULAs) [RFC4193], and Link Local Addresses [RFC3927] [RFC4291], whose identity cannot be confirmed using TLS certificates under most conditions.

Opportunistic Privacy is defined for DoT in Section 4.1 of [RFC7858] as a mode in which clients do not validate the name of the resolver presented in the certificate. Opportunistic Privacy similarly applies to DoQ [RFC9250]. A client MAY use information from the SVCB record for "resolver.arpa" with this "opportunistic" approach (not validating the names presented in the SubjectAlternativeName field of
the certificate) as long as the IP address of the Encrypted Resolver does not differ from the IP address of the Unencrypted Resolver. Clients SHOULD use this mode only for resolvers using private or local IP addresses. This approach can be used for any encrypted DNS protocol that uses TLS.

5. Discovery Using Resolver Names

A DNS client that already knows the name of an Encrypted Resolver can use DDR to discover details about all supported encrypted DNS protocols. This situation can arise if a client has been configured to use a given Encrypted Resolver, or if a network provisioning protocol (such as DHCP or IPv6 Router Advertisements) provides a name for an Encrypted Resolver alongside the resolver IP address, such as by using Discovery of Network Resolvers (DNR) [I-D.ietf-add-dnr].

For these cases, the client simply sends a DNS SVCB query using the known name of the resolver. This query can be issued to the named Encrypted Resolver itself or to any other resolver. Unlike the case of bootstrapping from an Unencrypted Resolver (Section 4), these records SHOULD be available in the public DNS.

For example, if the client already knows about a DoT server resolver.example.com, it can issue an SVCB query for _dns.resolver.example.com to discover if there are other encrypted DNS protocols available. In the following example, the SVCB answers indicate that resolver.example.com supports both DoH and DoT, and that the DoH server indicates a higher priority than the DoT server.

```
_dns.resolver.example.com. 7200 IN SVCB 1 resolver.example.com. (alpn=h2 dohpath=/dns-query{?dns} )
_dns.resolver.example.com. 7200 IN SVCB 1 resolver.example.com. (alpn=dot )
```

Clients MUST validate that for any Encrypted Resolver discovered using a known resolver name, the TLS certificate of the resolver contains the known name in a subjectAltName extension. In the example above, this means that both servers need to have certificates that cover the name resolver.example.com. Often, the various supported encrypted DNS protocols will be specified such that the SVCB TargetName matches the known name, as is true in the example above. However, even when the TargetName is different (for example, if the DoH server had a TargetName of doh.example.com), the clients still check for the original known resolver name in the certificate.

Note that this resolver validation is not related to the DNS resolver that provided the SVCB answer.
As another example, being able to discover a Designated Resolver for a known Encrypted Resolver is useful when a client has a DoT configuration for foo.resolver.example.com but is on a network that blocks DoT traffic. The client can still send a query to any other accessible resolver (either the local network resolver or an accessible DoH server) to discover if there is a designated DoH server for foo.resolver.example.com.

6. Deployment Considerations

Resolver deployments that support DDR are advised to consider the following points.

6.1. Caching Forwarders

A DNS forwarder SHOULD NOT forward queries for "resolver.arpa" upstream. This prevents a client from receiving an SVCB record that will fail to authenticate because the forwarder’s IP address is not in the upstream resolver’s Designated Resolver’s TLS certificate SAN field. A DNS forwarder which already acts as a completely blind forwarder MAY choose to forward these queries when the operator expects that this does not apply, either because the operator knows that the upstream resolver does have the forwarder’s IP address in its TLS certificate’s SAN field or that the operator expects clients of the unencrypted resolver to use the SVCB information opportunistically.

Operators who choose to forward queries for "resolver.arpa" upstream should note that client behavior is never guaranteed and use of DDR by a resolver does not communicate a requirement for clients to use the SVCB record when it cannot be verified.

6.2. Certificate Management

Resolver owners that support Verified Discovery will need to list valid referring IP addresses in their TLS certificates. This may pose challenges for resolvers with a large number of referring IP addresses.

6.3. Server Name Handling

Clients MUST NOT use "resolver.arpa" as the server name either in the TLS Server Name Indication (SNI) ([RFC8446]) for DoT, DoQ, or DoH connections, or in the URI host for DoH requests.

When performing discovery using resolver IP addresses, clients MUST use the IP address as the URI host for DoH requests.
Note that since IP addresses are not supported by default in the TLS SNI, resolvers that support discovery using IP addresses will need to be configured to present the appropriate TLS certificate when no SNI is present for DoT, DoQ, and DoH.

6.4. Handling non-DDR queries for resolver.arpa

DNS resolvers that support DDR by responding to queries for _dns.resolver.arpa SHOULD treat resolver.arpa as a locally served zone per [RFC6303]. In practice, this means that resolvers SHOULD respond to queries of any type other than SVCB for _dns.resolver.arpa with NODATA and queries of any type for any domain name under resolver.arpa with NODATA.

6.5. Interaction with Network-Designated Resolvers

Discovery of network-designated resolvers (DNR, [I-D.ietf-add-dnr]) allows a network to provide designation of resolvers directly through DHCP [RFC2132] [RFC8415] and IPv6 Router Advertisement (RA) [RFC4861] options. When such indications are present, clients can suppress queries for "resolver.arpa" to the unencrypted DNS server indicated by the network over DHCP or RAs, and the DNR indications SHOULD take precedence over those discovered using "resolver.arpa" for the same resolver if there is a conflict.

The designated resolver information in DNR might not contain a full set of SvcParams needed to connect to an encrypted resolver. In such a case, the client can use an SVCB query using a resolver name, as described in Section 5, to the authentication-domain-name (ADN).

7. Security Considerations

Since clients can receive DNS SVCB answers over unencrypted DNS, on-path attackers can prevent successful discovery by dropping SVCB queries or answers, and thus prevent clients from switching to use encrypted DNS. Clients should be aware that it might not be possible to distinguish between resolvers that do not have any Designated Resolver and such an active attack. To limit the impact of discovery queries being dropped either maliciously or unintentionally, clients can re-send their SVCB queries periodically.

Section 8.2 of [I-D.ietf-add-svcb-dns] describes a second downgrade attack where an attacker can block connections to the encrypted DNS server, and recommends that clients prevent it by switching to SVCB-reliant behavior once SVCB resolution does succeed. For DDR, this means that once a client discovers a compatible Designated Resolver, it SHOULD NOT use unencrypted DNS until the SVCB record expires, unless verification of the resolver fails.
DoH resolvers that allow discovery using DNS SVCB answers over unencrypted DNS MUST NOT provide differentiated behavior based on the HTTP path alone, since an attacker could modify the "dohpath" parameter. For example, if a DoH resolver provides a filtering service for one URI path, and a non-filtered service for another URI path, an attacker could select which of these services is used by modifying the "dohpath" parameter. These attacks can be mitigated by providing separate resolver IP addresses or hostnames.

While the IP address of the Unencrypted Resolver is often provisioned over insecure mechanisms, it can also be provisioned securely, such as via manual configuration, a VPN, or on a network with protections like RA-Guard [RFC6105]. An attacker might try to direct Encrypted DNS traffic to itself by causing the client to think that a discovered Designated Resolver uses a different IP address from the Unencrypted Resolver. Such a Designated Resolver might have a valid certificate, but be operated by an attacker that is trying to observe or modify user queries without the knowledge of the client or network.

If the IP address of a Designated Resolver differs from that of an Unencrypted Resolver, clients applying Verified Discovery (Section 4.2) MUST validate that the IP address of the Unencrypted Resolver is covered by the SubjectAlternativeName of the Designated Resolver’s TLS certificate.

Clients using Opportunistic Discovery (Section 4.3) MUST be limited to cases where the Unencrypted Resolver and Designated Resolver have the same IP address.

The constraints on the use of Designated Resolvers specified here apply specifically to the automatic discovery mechanisms defined in this document, which are referred to as Verified Discovery and Opportunistic Discovery. Clients MAY use some other mechanism to verify and use Designated Resolvers discovered using the DNS SVCB record. However, use of such an alternate mechanism needs to take into account the attack scenarios detailed here.

8. IANA Considerations

8.1. Special Use Domain Name "resolver.arpa"

This document calls for the addition of "resolver.arpa" to the Special-Use Domain Names (SUDN) registry established by [RFC6761]. This will allow resolvers to respond to queries directed at themselves rather than a specific domain name. While this document uses "resolver.arpa" to return SVCB records indicating designated encrypted capability, the name is generic enough to allow future
reuse for other purposes where the resolver wishes to provide information about itself to the client.

The "resolver.arpa" SUDN is similar to "ipv4only.arpa" in that the querying client is not interested in an answer from the authoritative "arpa" name servers. The intent of the SUDN is to allow clients to communicate with the Unencrypted Resolver much like "ipv4only.arpa" allows for client-to-middlebox communication. For more context, see the rationale behind "ipv4only.arpa" in [RFC8880].

IANA is requested to add an entry in "Transport-Independent Locally-Served DNS Zones" registry for ‘resolver.arpa.’ with the description "DNS Resolver Special-Use Domain", listing this document as the reference.

9. References

9.1. Normative References

[I-D.ietf-add-svcb-dns]

[I-D.ietf-dnsop-svcb-https]


9.2. Informative References

Jensen, "DHCP and Router Advertisement Options for the
Discovery of Network-designated Resolvers (DNR)",
Work in Progress, Internet-Draft, draft-ietf-add-dnr-09, 24 June
2022, <https://datatracker.ietf.org/doc/html/draft-ietf-
add-dnr-09>.

Encrypted Client Hello", Work in Progress, Internet-Draft,
draft-ietf-tls-esni-14, 13 February 2022,
<https://datatracker.ietf.org/doc/html/draft-ietf-tls-
esni-14>.

Hints for HTTP", Work in Progress, Internet-Draft,
draft-schinazi-httpbis-doh-preference-hints-02, 13 July 2020,
<https://datatracker.ietf.org/doc/html/draft-schinazi-
httpbis-doh-preference-hints-02>.

Extensions", RFC 2132, DOI 10.17487/RFC2132, March 1997,

[RFC4861]  Narten, T., Nordmark, E., Simpson, W., and H. Soliman,
"Neighbor Discovery for IP version 6 (IPv6)", RFC 4861,
DOI 10.17487/RFC4861, September 2007,

Ed., "Design Choices When Expanding the DNS", RFC 5507,
DOI 10.17487/RFC5507, April 2009,

[RFC6105]  Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J.
Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105,
DOI 10.17487/RFC6105, February 2011,

[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,
Appendix A. Rationale for using SVCB records

This mechanism uses SVCB/HTTPS resource records [I-D.ietf-dnsop-svcb-https] to communicate that a given domain designates a particular Designated Resolver for clients to use in place of an Unencrypted Resolver (using a SUDN) or another Encrypted Resolver (using its domain name).

There are various other proposals for how to provide similar functionality. There are several reasons that this mechanism has chosen SVCB records:

* Discovering encrypted resolver using DNS records keeps client logic for DNS self-contained and allows a DNS resolver operator to define which resolver names and IP addresses are related to one another.

* Using DNS records also does not rely on bootstrapping with higher-level application operations (such as [I-D.schinazi-httpbis-doh-preference-hints]).

* SVCB records are extensible and allow definition of parameter keys. This makes them a superior mechanism for extensibility as compared to approaches such as overloading TXT records. The same keys can be used for discovering Designated Resolvers of different transport types as well as those advertised by Unencrypted Resolvers or another Encrypted Resolver.

* Clients and servers that are interested in privacy of names will already need to support SVCB records in order to use Encrypted TLS Client Hello [I-D.ietf-tls-esni]. Without encrypting names in TLS, the value of encrypting DNS is reduced, so pairing the solutions provides the largest benefit.
* Clients that support SVCB will generally send out three queries when accessing web content on a dual-stack network: A, AAAA, and HTTPS queries. Discovering a Designated Resolver as part of one of these queries, without having to add yet another query, minimizes the total number of queries clients send. While [RFC5507] recommends adding new RRTypes for new functionality, SVCB provides an extension mechanism that simplifies client behavior.

Authors’ Addresses

Tommy Pauly
Apple Inc.
One Apple Park Way
Cupertino, California 95014,
United States of America
Email: tpauly@apple.com

Eric Kinnear
Apple Inc.
One Apple Park Way
Cupertino, California 95014,
United States of America
Email: ekinnear@apple.com

Christopher A. Wood
Cloudflare
101 Townsend St
San Francisco,
United States of America
Email: caw@heapingbits.net

Patrick McManus
Fastly
Email: mcmanus@ducksong.com

Tommy Jensen
Microsoft
Email: tojens@microsoft.com
DHCP and Router Advertisement Options for the Discovery of Network-designated Resolvers (DNR)
draft-ietf-add-dnr-11

Abstract

The document specifies new DHCP and IPv6 Router Advertisement options to discover encrypted DNS resolvers (e.g., DNS-over-HTTPS, DNS-over-TLS, DNS-over-QUIC). Particularly, it allows a host to learn an authentication domain name together with a list of IP addresses and a set of service parameters to reach such encrypted DNS resolvers.

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 12 January 2023.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction ................................................. 3
2. Terminology .................................................. 3
3. Overview ....................................................... 4
  3.1. Configuration Data for Encrypted DNS .................... 4
    3.1.1. ADN as the Reference Identifier for DNS Authentication  . 4
    3.1.2. Avoiding Dependency on External Resolvers ............ 4
    3.1.3. Single vs. Multiple IP Addresses ................... 5
    3.1.4. Why Not Separate Options for ADN and IP Addresses?  . 5
    3.1.5. Service Parameters .................................. 5
    3.1.6. ADN Only Mode ....................................... 6
    3.1.7. Encrypted DNS Options Ordering .................... 6
    3.1.8. DNR Validation Checks ............................... 6
    3.1.9. Recommended DNR Information ....................... 7
  3.2. Handling Configuration Data Conflicts .................... 7
3.3. Connection Establishment .................................. 8
3.4. Multihoming Is Out Of Scope .............................. 8
4. DHCPv6 Encrypted DNS Option .................................. 8
  4.1. Option Format ........................................... 8
5. DHCPv4 Encrypted DNS Option .................................. 11
  5.1. Option Format ........................................... 11
5.2. DHCPv4 Client Behavior ................................... 13
6. IPv6 RA Encrypted DNS Option ................................ 14
  6.1. Option Format ........................................... 14
6.2. IPv6 Host Behavior ....................................... 16
7. Security Considerations ...................................... 16
  7.1. Spoofing Attacks ........................................ 16
  7.2. Deletion Attacks ....................................... 17
  7.3. Passive Attacks ....................................... 18
8. IANA Considerations .......................................... 18
  8.1. DHCPv6 Option .......................................... 18
  8.2. DHCPv4 Option .......................................... 19
  8.3. Neighbor Discovery Option ............................. 19
9. Acknowledgements ............................................ 19
10. Contributing Authors ....................................... 20

Table of Contents
1. Introduction

This document focuses on the support of encrypted DNS such as DNS-over-HTTPS (DoH) [RFC8484], DNS-over-TLS (DoT) [RFC7858], or DNS-over-QUIC (DoQ) [RFC9250] in local networks.

In particular, the document specifies how a local encrypted DNS resolver can be discovered by connected hosts by means of DHCPv4 [RFC2132], DHCPv6 [RFC8415], and IPv6 Router Advertisement (RA) [RFC4861] options. These options are designed to convey the following information: the DNS Authentication Domain Name (ADN), a list of IP addresses, and a set of service parameters. This procedure is called Discovery of Network-designated Resolvers (DNR).

The options defined in this document can be deployed in a variety of deployments (e.g., local networks with Customer Premises Equipment (CPEs) that may or may not be managed by an Internet Service Provider (ISP), or local networks with or without DNS forwarders). It is out of the scope of this document to provide an inventory of such deployments.

2. 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.

This document makes use of the terms defined in [RFC8499]. The following additional terms are used:

Do53: refers to unencrypted DNS.

DNR: refers to the Discovery of Network-designated Resolvers procedure.

Encrypted DNS: refers to a scheme where DNS exchanges are transported over an encrypted channel. Examples of encrypted DNS are DoT, DoH, or DoQ.

Encrypted DNS resolver: refers to a DNS resolver that supports any encrypted DNS scheme.
Encrypted DNS options: refers to the options defined in Sections 4, 5, and 6.

DHCP: refers to both DHCPv4 and DHCPv6.

3. Overview

This document describes how a DNS client can discover local encrypted DNS resolvers using DHCP (Sections 4 and 5) and Neighbor Discovery protocol (Section 6): Encrypted DNS options.

These options configure an authentication domain name, a list of IP addresses, and a set of service parameters of the encrypted DNS resolver. More information about the design of these options is provided in the following subsections.

3.1. Configuration Data for Encrypted DNS

3.1.1. ADN as the Reference Identifier for DNS Authentication

In order to allow for PKIX-based authentication between a DNS client and an encrypted DNS resolver, the Encrypted DNS options are designed to include an authentication domain name. This ADN is presented as a reference identifier for DNS authentication purposes. This design accommodates the current best practices for issuing certificates as per Section 1.7.2 of [RFC6125]:

> Some certification authorities issue server certificates based on IP addresses, but preliminary evidence indicates that such certificates are a very small percentage (less than 1%) of issued certificates.

3.1.2. Avoiding Dependency on External Resolvers

To avoid adding a dependency on another server to resolve the ADN, the Encrypted DNS options return the IP address(es) to locate the encrypted DNS resolver. These encrypted DNS resolvers may be hosted on the same or distinct IP addresses. Such a decision is deployment specific.

In order to optimize the size of discovery messages when all DNS resolvers terminate on the same IP address, early versions of this document considered relying upon the discovery mechanisms specified in [RFC2132][RFC3646][RFC8106] to retrieve a list of IP addresses to reach their DNS resolvers. Nevertheless, this approach requires a client that supports more than one encrypted DNS protocol (e.g., DoH and DoT) to probe that list of IP addresses. To avoid such a
probing, the options defined in Sections 4, 5, and 6 associate an encrypted DNS protocol with an IP address. No probing is required in such a design.

3.1.3. Single vs. Multiple IP Addresses

A list of IP addresses to reach an encrypted DNS resolver may be returned in an Encrypted DNS option to accommodate current deployments relying upon primary and backup resolvers. Also, DNR can be used in contexts where other DNS redundancy schemes (e.g., anycast as in BCP 126 [RFC4786]) are used.

Whether one or more IP addresses are returned in an Encrypted DNS option is deployment specific. For example, a router embedding a recursive server or a forwarder has to include one single IP address pointing to one of its LAN-facing interfaces. Typically, this IP address can be a private IPv4 address, a link-local address, a Unique Local IPv6 unicast Address (ULA), or a Global Unicast Address (GUA).

If multiple IP addresses are to be returned in an Encrypted DNS option, these addresses are ordered in the preference for use by the client.

3.1.4. Why Not Separate Options for ADN and IP Addresses?

A single option is used to convey both the ADN and IP addresses because otherwise means to correlate an IP address conveyed in an option with an ADN conveyed in another option will be required if, for example, more than one ADN is supported by the network.

3.1.5. Service Parameters

Because distinct encrypted DNS protocols (e.g., DoT, DoH, and DoQ) may be provisioned by a network and that some of these protocols may make use of customized port numbers instead of default ones, the Encrypted DNS options are designed to return a set of service parameters. These parameters are encoded following the same rules for encoding SvcParams in Section 2.1 of [I-D.ietf-dnsop-svcb-https]. This encoding approach may increase the size of the options but it has the merit relying upon an existing IANA registry and, thus, accommodating new encrypted DNS protocols and service parameters that may be defined in the future.

The following service parameters MUST be supported by a DNR implementation:

alpn: Used to indicate the set of supported protocols (Section 7.1 of [I-D.ietf-dnsop-svcb-https]).
port: Used to indicate the target port number for the encrypted DNS connection (Section 7.2 of [I-D.ietf-dnsop-svcb-https]).

In addition, the following service parameters are RECOMMENDED to be supported by a DNR implementation:

- ech: Used to enable Encrypted ClientHello (ECH) (Section 7.3 of [I-D.ietf-dnsop-svcb-https]).
- dohpath: Used to supply a relative DoH URI Template (Section 5.1 of [I-D.ietf-add-svcb-dns]).

3.1.6. ADN Only Mode

The provisioning mode in which an ADN, a list of IP addresses, and a set of service parameters of the encrypted DNS resolver are supplied to a host SHOULD be used because the Encrypted DNS options are self-contained and do not require any additional DNS queries. The reader may refer to [RFC7969] for an overview of advanced capabilities that are supported by DHCP servers to populate configuration data (e.g., issue DNS queries).

In contexts where putting additional complexity on requesting hosts is acceptable, returning an ADN only can be considered. The supplied ADN will be passed to a local resolution library (a DNS client, typically) which will then issues Service Binding (SVCB) queries [I-D.ietf-add-svcb-dns]. These SVCB queries can be sent to the discovered encrypted DNS resolver itself or to the network-designated Do53 resolver. Note that this mode may be subject to active attacks, which can be mitigated by DNSSEC.

How an ADN is passed to a local resolution library is implementation specific.

3.1.7. Encrypted DNS Options Ordering

The DHCP options defined in Sections 4 and 5 follow the option ordering guidelines in Section 17 of [RFC7227].

Likewise, the RA option (Section 6) adheres to the recommendations in Section 9 of [RFC4861].

3.1.8. DNR Validation Checks

On receipt of an Encrypted DNS option, the client makes the following validation checks:
* The ADN is encoded as per Section 10 of [RFC8415].

* If additional data is supplied:
  
  - the service parameters are encoded following the rules specified in Section 2.1 of [I-D.ietf-dnsop-svcb-https].

  - the option includes at least one valid IP address and the "alpn" service parameter.

  - the service parameters do not include "ipv4hint" or "ipv6hint" service parameters.

If any of the checks fail, the receiver discards the received Encrypted DNS option.

3.1.9. Recommended DNR Information

Other mechanisms may be considered in other contexts (e.g., secure discovery) for the provisioning of encrypted DNS resolvers. It is RECOMMENDED that at least the following DNR information is made available to a requesting host:

* A service priority whenever the discovery mechanism does not rely on implicit ordering if multiple instances of the encrypted DNS are used.

* An authentication domain name.

* A list of IP addresses to locate the encrypted DNS resolver.

* A set of service parameters.

3.2. Handling Configuration Data Conflicts

If the encrypted DNS is discovered by a host using both RA and DHCP, the rules discussed in Section 5.3.1 of [RFC8106] MUST be followed.

DHCP/RA options to discover encrypted DNS resolvers (including, DoH URI Templates) takes precedence over Discovery of Designated Resolvers (DDR) [I-D.ietf-add-ddr] since DDR uses Do53 to an external DNS resolver, which is susceptible to both internal and external attacks whereas DHCP/RA is typically protected using the mechanisms discussed in Section 7.1.
If a client learns both Do53 and encrypted DNS resolvers from the same network, and absent explicit configuration otherwise, it is RECOMMENDED that the client uses the encrypted DNS resolvers for that network.

3.3. Connection Establishment

If the local DNS client supports one of the discovered Encrypted DNS protocols identified by Application Layer Protocol Negotiation (ALPN) protocol identifiers, the DNS client establishes an encrypted DNS session following the order of the discovered resolvers.

The DNS client verifies the connection based on PKIX validation [RFC5280] of the DNS resolver certificate and uses the validation techniques as described in [RFC6125] to compare the authentication domain name conveyed in the Encrypted DNS options to the certificate provided (see Section 8.1 of [RFC8310] for more details). The DNS client uses Web PKI trust anchors by default unless configured otherwise to use explicit trust anchors. ALPN-related considerations can be found in Section 6.1 of [I-D.ietf-dnsop-svcb-https].

3.4. Multihoming Is Out Of Scope

Devices may be connected to multiple networks; each providing their own DNS configuration using the discovery mechanisms specified in this document. Nevertheless, it is out of the scope of this specification to discuss DNS selection of multi-interface devices. The reader may refer to [RFC6731] for a discussion of issues and an example of DNS resolver selection for multi-interfaced devices.

4. DHCPv6 Encrypted DNS Option

4.1. Option Format

The format of the DHCPv6 Encrypted DNS option is shown in Figure 1.
Figure 1: DHCPv6 Encrypted DNS Option

The fields of the option shown in Figure 1 are as follows:

Option-code: OPTION_V6_DNR (TBA1, see Section 8.1)

Option-length: Length of the enclosed data in octets. The option length is ('ADN Length' + 4) when only an ADN is included in the option.

Service Priority: The priority of this OPTION_V6_DNR instance compared to other instances. This 16-bit unsigned integer is interpreted following the rules specified in Section 2.4.1 of [I-D.ietf-dnsop-svcb-https].

ADN Length: Length of the authentication-domain-name field in octets.

authentication-domain-name (variable length): A fully qualified domain name of the encrypted DNS resolver. This field is formatted as specified in Section 10 of [RFC8415].

An example of the authentication-domain-name encoding is shown in Figure 2. This example conveys the FQDN "doh1.example.com.", and the resulting Option-length field is 18.
Figure 2: An Example of the DNS authentication-domain-name Encoding

Addr Length: Length of enclosed IPv6 addresses in octets. When present, it MUST be a multiple of 16.

ipv6-address(es) (variable length): Indicates one or more IPv6 addresses to reach the encrypted DNS resolver. An address can be link-local, ULA, or GUA. The format of this field is shown in Figure 3.

```
+-----------------------------+
| ipv6-address                |
+-----------------------------+
|                              |
+-----------------------------+
|                              |
+-----------------------------+  ...
```

Figure 3: Format of the IPv6 Addresses Field

Service Parameters (SvcParams) (variable length): Specifies a set of service parameters that are encoded following the rules in Section 2.1 of [I-D.ietf-dnsop-svcb-https]. Service parameters may include, for example, a list of ALPN protocol identifiers or alternate port numbers. This field MUST include at least "alpn" SvcParam (Section 4.1 of [I-D.ietf-add-svcb-dns]). The service parameters MUST NOT include "ipv4hint" or "ipv6hint" SvcParams as they are superseded by the included IP addresses.

If no port service parameter is included, this indicates that default port numbers should be used. As a reminder, the default port number is 853 for DoT, 443 for DoH, and 853 for DoQ.

The length of this field is ('Option-length' – 6 – 'ADN Length' – 'Addr Length').

4.2. DHCPv6 Client Behavior

To discover an encrypted DNS resolver, the DHCPv6 client MUST include OPTION_V6_DNR in an Option Request Option (ORO), as in Sections 18.2.1, 18.2.2, 18.2.4, 18.2.5, 18.2.6, and 21.7 of [RFC8415].
The DHCPv6 client MUST be prepared to receive multiple instances of the OPTION_V6_DNR option; each option is to be treated as a separate encrypted DNS resolver. These instances SHOULD be processed following their service priority (i.e., smaller service priority indicates a higher preference).

The DHCPv6 client MUST silently discard any OPTION_V6_DNR that fails to pass the validation steps defined in Section 3.1.8.

The DHCPv6 client MUST silently discard multicast and host loopback addresses conveyed in OPTION_V6_DNR.

5. DHCPv4 Encrypted DNS Option

5.1. Option Format

The format of the DHCPv4 Encrypted DNS option is illustrated in Figure 4.

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| OPTION_V4_DNR |     Length    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
`--- DNR Instance Data #1 "
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          ...                    . |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
`--- DNR Instance Data #n "
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: DHCPv4 Encrypted DNS Option

The fields of the option shown in Figure 4 are as follows:

Code: OPTION_V4_DNR (TBA2, see Section 8.2).

Length: Indicates the length of the enclosed data in octets.

DNR Instance Data: Includes the configuration data of an encrypted DNS resolver. The format of this field is shown in Figure 5.
When several encrypted DNS resolvers are to be included, the "DNR Instance Data" field is repeated.

The fields shown in Figure 5 are as follows:

DNR Instance Data Length: Length of all following data in octets. This field is set to ('ADN Length' + 3) when only an ADN is provided for a DNR instance.

Service Priority: The priority of this instance compared to other DNR instances. This 16-bit unsigned integer is interpreted following the rules specified in Section 2.4.1 of [I-D.ietf-dnsop-svcb-https].

ADN Length: Length of the authentication-domain-name in octets.

authentication-domain-name (variable length): The authentication domain name of the encrypted DNS resolver. This field is formatted as specified in Section 10 of [RFC8415]. An example is provided in Figure 2.

Addr Length: Length of included IPv4 addresses in octets. When present, it MUST be a multiple of 4.

IPv4 Address(es) (variable length): Indicates one or more IPv4
addresses to reach the encrypted DNS resolver. Both private and public IPv4 addresses can be included in this field. The format of this field is shown in Figure 6. This format assumes that an IPv4 address is encoded as a1.a2.a3.a4.

```
  0 8 16 24 32 40 48
+-----+-----+-----+-----+-----+-----+--
| a1 | a2 | a3 | a4 | a1 | a2 | ...
+-----+-----+-----+-----+-----+-----+--
```

**Figure 6: Format of the IPv4 Addresses Field**

Service Parameters (SvcParams) (variable length): Specifies a set of service parameters that are encoded following the rules in Section 2.1 of [I-D.ietf-dnsop-svcb-https]. Service parameters may include, for example, a list of ALPN protocol identifiers or alternate port numbers. This field MUST include at least "alpn" SvcParam (Section 4.1 of [I-D.ietf-add-svcb-dns]). The service parameters MUST NOT include "ipv4hint" or "ipv6hint" SvcParams as they are superseded by the included IP addresses.

If no port service parameter is included, this indicates that default port numbers should be used.

The length of this field is ('DNR Instance Data Length' - 4 - 'ADN Length' - 'Addr Length').

OPTION_V4_DNR is a concatenation-requiring option. As such, the mechanism specified in [RFC3396] MUST be used if OPTION_V4_DNR exceeds the maximum DHCPv4 option size of 255 octets.

### 5.2. DHCPv4 Client Behavior

To discover an encrypted DNS resolver, the DHCPv4 client requests the encrypted DNS resolver by including OPTION_V4_DNR in a Parameter Request List option [RFC2132].

The DHCPv4 client MUST be prepared to receive multiple DNR instance data in the OPTION_V4_DNR option; each instance is to be treated as a separate encrypted DNS resolver. These instances SHOULD be processed following their service priority (i.e., smaller service priority indicates a higher preference).

The DHCPv4 client MUST silently discard any OPTION_V4_DNR that fails to pass the validation steps defined in Section 3.1.8.
The DHCPv4 client MUST silently discard multicast and host loopback addresses conveyed in OPTION_V4_DNR.

6. IPv6 RA Encrypted DNS Option

6.1. Option Format

This section defines a new Neighbor Discovery option [RFC4861]: IPv6 RA Encrypted DNS option. This option is useful in contexts similar to those discussed in Section 1.1 of [RFC8106].

The format of the IPv6 RA Encrypted DNS option is illustrated in Figure 7.

```
+---------------+-------------------+-------------------+
|    TBA3      |     Length     |        Service Priority       |
+---------------+-------------------+-------------------+
|     Lifetime  |         ADN Length |                               |
+---------------+-------------------+-------------------+
| authentication-domain-name | Addr Length |
+-------------------+-------------------+
| ipv6-address(es) |                  |
+-------------------+-------------------+
|                  | Service Parameters (SvcParams) |
+-------------------+-------------------+
```

Figure 7: RA Encrypted DNS Option

The fields of the option shown in Figure 7 are as follows:

Type: 8-bit identifier of the Encrypted DNS option as assigned by IANA (TBA3, see Section 8.3).

Length: 8-bit unsigned integer. The length of the option (including the Type and Length fields) is in units of 8 octets.

Service Priority: 16-bit unsigned integer. The priority of this
Encrypted DNS option instance compared to other instances. This field is interpreted following the rules specified in Section 2.4.1 of [I-D.ietf-dnsop-svcb-https].

Lifetime: 32-bit unsigned integer. The maximum time in seconds (relative to the time the packet is received) over which the discovered Authentication Domain Name is valid.

The value of Lifetime SHOULD by default be at least 3 * MaxRtrAdvInterval, where MaxRtrAdvInterval is the maximum RA interval as defined in [RFC4861].

A value of all one bits (0xffffffff) represents infinity.

A value of zero means that this Authentication Domain Name MUST no longer be used.

ADN Length: 16-bit unsigned integer. This field indicates the length of the authentication-domain-name field in octets.

authentication-domain-name (variable length): The authentication domain name of the encrypted DNS resolver. This field is formatted as specified in Section 10 of [RFC8415].

Addr Length: 16-bit unsigned integer. This field indicates the length of enclosed IPv6 addresses in octets. When present, it MUST be a multiple of 16.

ipv6-address(es) (variable length): One or more IPv6 addresses of the encrypted DNS resolver. An address can be link-local, ULA, or GUA.

All of the addresses share the same Lifetime value. Similar to [RFC8106], if it is desirable to have different Lifetime values per IP address, multiple Encrypted DNS options may be used.

The format of this field is shown in Figure 3.

SvcParams Length: 16-bit unsigned integer. This field indicates the length of the Service Parameters field in octets.

Service Parameters (SvcParams) (variable length): Specifies a set of
service parameters that are encoded following the rules in
Section 2.1 of [I-D.ietf-dnsop-svcb-https]. Service parameters
may include, for example, a list of ALPN protocol identifiers or
alternate port numbers. This field MUST include at least "alpn"
SvcParam (Section 4.1 of [I-D.ietf-add-svcb-dns]). The service
parameters MUST NOT include "ipv4hint" or "ipv6hint" SvcParams as
they are superseded by the included IP addresses.

If no port service parameter is included, this indicates that
default port numbers should be used.

The option MUST be padded with zeros so that the full enclosed data
is a multiple of 8 octets (Section 4.6 of [RFC4861]).

6.2. IPv6 Host Behavior

The procedure for DNS configuration is the same as it is with any
other Neighbor Discovery option [RFC4861]. In addition, the host
follows the same procedure as the one described in Section 5.3.1 of
[RFC8106] for processing received Encrypted DNS options with the
formatting requirements in Section 6.1 and validation checks in
Section 3.1.8 substituted for the length and fields validation.

The host MUST be prepared to receive multiple Encrypted DNS options
in RAs. These instances SHOULD be processed following their service
priority (i.e., smaller service priority indicates a higher
preference).

The host MUST silently discard multicast and host loopback addresses
conveyed in the Encrypted DNS options.

7. Security Considerations

7.1. Spoofing Attacks

DHCP/RA messages are not encrypted or protected against modification
within the LAN. Unless mitigated (described below), the content of
DHCP and RA messages can be spoofed or modified by active attackers,
such as compromised devices within the local network. An active
attacker (Section 3.3 of [RFC3552]) can spoof the DHCP/RA response to
provide the attacker’s encrypted DNS resolver. Note that such an
attacker can launch other attacks as discussed in Section 22 of
[RFC8415]. The attacker can get a domain name with a domain-
validated public certificate from a CA and host an encrypted DNS
resolver.
Attacks of spoofed or modified DHCP responses and RA messages by attackers within the local network may be mitigated by making use of the following mechanisms:

* DHCPv6-Shield [RFC7610]: the router (e.g., a border router, a CPE) discards DHCP response messages received from any local endpoint.

* RA-Guard [RFC7113]: the router discards RAs messages received from any local endpoint.

* Source Address Validation Improvement (SAVI) solution for DHCP [RFC7513]: the router filters packets with forged source IP addresses.

The above mechanisms would ensure that the endpoint receives the correct configuration information of the encrypted DNS resolvers selected by the DHCP server (or RA sender), but cannot provide any information about the DHCP server or the entity hosting the DHCP server (or RA sender).

Encrypted DNS sessions with rogue resolvers that spoof the IP address of a DNS resolver will fail because the DNS client will fail to authenticate that rogue resolver based upon PKIX authentication [RFC6125], particularly the authentication domain name in the Encrypted DNS Option. DNS clients that ignore authentication failures and accept spoofed certificates will be subject to attacks (e.g., redirect to malicious resolvers, intercept sensitive data).

By default, Encrypted DNS connections received from outside the local network MUST be discarded by the encrypted DNS forwarder in a CPE. This behavior adheres to REQ#8 in [RFC6092]; it MUST apply for both IPv4 and IPv6. This recommendation is meant to isolate local network DNS resolver services from the public Internet and prevent external attacks against the local Encrypted DNS resolver.

7.2. Deletion Attacks

If the DHCP responses or RAs are dropped by the attacker, the client can fallback to use a preconfigured encrypted DNS resolver. However, the use of policies to select resolvers is out of the scope of this document.

Note that deletion attack is not specific to DHCP/RA.
7.3. Passive Attacks

A passive attacker (Section 3.2 of [RFC3552]) can identify a host is using DHCP/RA to discover an encrypted DNS resolver and can infer that host is capable of using DoH/DoT/DoQ to encrypt DNS messages. However, a passive attacker cannot spoof or modify DHCP/RA messages.


Wireless LAN (WLAN) as frequently deployed in local networks (e.g., home networks) is vulnerable to various attacks (e.g., [Evil-Twin], [Krack], [Dragonblood]). Because of these attacks, only cryptographically authenticated communications are trusted on WLANs. This means that an information (e.g., NTP server, DNS resolver, domain search list) provided by such networks via DHCP, DHCPv6, or RA are untrusted because DHCP and RA messages are not authenticated.

If the pre-shared key (PSK) is the same for all clients that connect to the same WLAN (e.g., WPA-PSK), the shared key will be available to all nodes, including attackers. As such, it is possible to mount an active on-path attack. On-path attacks are possible within local networks because such a WLAN authentication lacks peer entity authentication.

This leads to the need for provisioning unique credentials for different clients. Endpoints can be provisioned with unique credentials (username and password, typically) provided by the local network administrator to mutually authenticate to the local WLAN Access Point (e.g., 802.1x Wireless User Authentication on OpenWRT [dot1x], EAP-pwd [RFC8146]). Not all endpoint devices (e.g., IoT devices) support 802.1x supplicant and need an alternate mechanism to connect to the local network. To address this limitation, unique pre-shared keys can be created for each such devices and WPA-PSK is used (e.g., [IPSK]).

8. IANA Considerations

8.1. DHCPv6 Option

IANA is requested to assign the following new DHCPv6 Option Code in the registry maintained in [DHCPV6].
8.2. DHCPv4 Option

IANA is requested to assign the following new DHCP Option Code in the registry maintained in [BOOTP].

<table>
<thead>
<tr>
<th>Tag</th>
<th>Name</th>
<th>Data Length</th>
<th>Meaning</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBA2</td>
<td>OPTION_V4_DNR</td>
<td>N</td>
<td>Encrypted DNS Server</td>
<td>[ThisDocument]</td>
</tr>
</tbody>
</table>

Table 2: DHCPv4 Encrypted DNS Option

8.3. Neighbor Discovery Option

IANA is requested to assign the following new IPv6 Neighbor Discovery Option type in the "IPv6 Neighbor Discovery Option Formats" sub-registry under the "Internet Control Message Protocol version 6 (ICMPv6) Parameters" registry maintained in [ND].

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBA3</td>
<td>Encrypted DNS Option</td>
<td>[ThisDocument]</td>
</tr>
</tbody>
</table>

Table 3: Neighbor Discovery Encrypted DNS Option

9. Acknowledgements

Many thanks to Christian Jacquenet and Michael Richardson for the review.

Thanks to Stephen Farrell, Martin Thomson, Vittorio Bertola, Stephane Bortzmeyer, Ben Schwartz, Iain Sharp, and Chris Box for the comments.
Thanks to Mark Nottingham for the feedback on HTTP redirection that was discussed in previous versions of this specification.

The use of DHCP to retrieve an authentication domain name was discussed in Section 7.3.1 of [RFC8310] and [I-D.pusateri-dhc-dns-driu].

Thanks to Bernie Volz for the review of the DHCP part.

Thanks to Andrew Campling for the Shepherd review and Eric Vyncke for the AD review.

Thanks to Rich Salz for the secdir review, Joe Clarke for the ops-dir review, and Robert Sparks for the artart review.

10. Contributing Authors

Nicolai Leymann
Deutsche Telekom
Germany
Email: n.leymann@telekom.de

Zhiwei Yan
CNNIC
No.4 South 4th Street, Zhongguancun
Beijing
100190
China
Email: yan@cnnic.cn

11. References

11.1. Normative References

[I-D.ietf-add-svcb-dns]
Schwartz, B., "Service Binding Mapping for DNS Servers",
Work in Progress, Internet-Draft, draft-ietf-add-svcb-dns-06, 5 July 2022,
[I-D.ietf-dnsop-svcb-https]


11.2. Informative References


Authors’ Addresses

Mohamed Boucadair (editor)
Orange
35000 Rennes
France
Email: mohamed.boucadair@orange.com

DHCP and Router Advertisement Options for the Discovery of Network-designated Resolvers (DNR)
draft-ietf-add-dnr-12

Abstract

The document specifies new DHCP and IPv6 Router Advertisement options to discover encrypted DNS resolvers (e.g., DNS-over-HTTPS, DNS-over-TLS, DNS-over-QUIC). Particularly, it allows a host to learn an authentication domain name together with a list of IP addresses and a set of service parameters to reach such encrypted DNS resolvers.

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 25 January 2023.

Copyright Notice

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction ........................................... 3
2. Terminology ........................................... 3
3. Overview ............................................. 4
   3.1. Configuration Data for Encrypted DNS ................ 4
      3.1.1. ADN as the Reference Identifier for DNS Authentication ........................................ 4
      3.1.2. Avoiding Dependency on External Resolvers .......... 4
      3.1.3. Single vs. Multiple IP Addresses ................ 5
      3.1.4. Why Not Separate Options for ADN and IP Addresses? .... 5
      3.1.5. Service Parameters ................................ 5
      3.1.6. ADN Only Mode ................................... 6
      3.1.7. Encrypted DNS Options Ordering .................. 6
      3.1.8. DNR Validation Checks ............................ 7
      3.1.9. DNR Information Using Other Provisioning Mechanisms ......................... 7
      3.2. Handling Configuration Data Conflicts ................. 8
      3.3. Validating Discovered Resolvers ........................ 8
      3.4. Multihoming Considerations ........................... 9
4. DHCPv6 Encrypted DNS Option ............................ 9
   4.1. Option Format ...................................... 9
   4.2. DHCPv6 Client Behavior ............................. 11
5. DHCPv4 Encrypted DNS Option ............................ 11
   5.1. Option Format ...................................... 11
   5.2. DHCPv4 Client Behavior ............................. 14
6. IPv6 RA Encrypted DNS Option ............................. 14
   6.1. Option Format ...................................... 14
   6.2. IPv6 Host Behavior .................................. 17
7. Security Considerations .................................. 17
   7.1. Spoofing Attacks .................................... 17
   7.2. Deletion Attacks .................................... 18
   7.3. Passive Attacks .................................... 18
8. Privacy Considerations ................................... 19
9. IANA Considerations ..................................... 19
   9.1. DHCPv6 Option ..................................... 19
   9.2. DHCPv4 Option ..................................... 20
   9.3. Neighbor Discovery Option ........................... 20
1. Introduction

This document focuses on the discovery of encrypted DNS such as DNS-over-HTTPS (DoH) [RFC8484], DNS-over-TLS (DoT) [RFC7858], or DNS-over-QUIC (DoQ) [RFC9250] in local networks.

In particular, the document specifies how a local encrypted DNS resolver can be discovered by connected hosts by means of DHCPv4 [RFC2132], DHCPv6 [RFC8415], and IPv6 Router Advertisement (RA) [RFC4861] options. These options are designed to convey the following information: the DNS Authentication Domain Name (ADN), a list of IP addresses, and a set of service parameters. This procedure is called Discovery of Network-designated Resolvers (DNR).

The options defined in this document can be deployed in a variety of deployments (e.g., local networks with Customer Premises Equipment (CPEs) that may or may not be managed by an Internet Service Provider (ISP), or local networks with or without DNS forwarders). It is out of the scope of this document to provide an inventory of such deployments.

Resolver selection considerations are out of scope. Likewise, policies (including any interactions with users) are out of scope.

2. 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.

This document makes use of the terms defined in [RFC8499]. The following additional terms are used:

Do53: refers to unencrypted DNS.

DNR: refers to the Discovery of Network-designated Resolvers procedure.

Encrypted DNS: refers to a scheme where DNS exchanges are
transported over an encrypted channel. Examples of encrypted DNS are DoT, DoH, or DoQ.

Encrypted DNS resolver: refers to a DNS resolver that supports any encrypted DNS scheme.

Encrypted DNS options: refers to the options defined in Sections 4, 5, and 6.

DHCP: refers to both DHCPv4 and DHCPv6.

3. Overview

This document describes how a DNS client can discover local encrypted DNS resolvers using DHCP (Sections 4 and 5) and Neighbor Discovery protocol (Section 6): Encrypted DNS options.

These options configure an authentication domain name, a list of IP addresses, and a set of service parameters of the encrypted DNS resolver. More information about the design of these options is provided in the following subsections.

3.1. Configuration Data for Encrypted DNS

3.1.1. ADN as the Reference Identifier for DNS Authentication

In order to allow for PKIX-based authentication between a DNS client and an encrypted DNS resolver, the Encrypted DNS options are designed to always include an authentication domain name. This ADN is presented as a reference identifier for DNS authentication purposes. This design accommodates the current best practices for issuing certificates as per Section 1.7.2 of [RFC6125]:

Some certification authorities issue server certificates based on IP addresses, but preliminary evidence indicates that such certificates are a very small percentage (less than 1%) of issued certificates.

3.1.2. Avoiding Dependency on External Resolvers

To avoid adding a dependency on another server to resolve the ADN, the Encrypted DNS options return the IP address(es) to locate the encrypted DNS resolver. These encrypted DNS resolvers may be hosted on the same or distinct IP addresses. Such a decision is deployment specific.
In order to optimize the size of discovery messages when all DNS resolvers terminate on the same IP address, early versions of this document considered relying upon the discovery mechanisms specified in [RFC2132][RFC3646][RFC8106] to retrieve a list of IP addresses to reach their DNS resolvers. Nevertheless, this approach requires a client that supports more than one encrypted DNS protocol (e.g., DoH and DoT) to probe that list of IP addresses. To avoid such a probing, the options defined in Sections 4, 5, and 6 associate an encrypted DNS protocol with an IP address. No probing is required in such a design.

3.1.3. Single vs. Multiple IP Addresses

A list of IP addresses to reach an encrypted DNS resolver may be returned in an Encrypted DNS option to accommodate current deployments relying upon primary and backup resolvers. Also, DNR can be used in contexts where other DNS redundancy schemes (e.g., anycast as in BCP 126 [RFC4786]) are used.

Whether one or more IP addresses are returned in an Encrypted DNS option is deployment specific. For example, a router embedding a recursive server or a forwarder has to include one single IP address pointing to one of its LAN-facing interfaces. Typically, this IP address can be a private IPv4 address, a link-local address, a Unique Local IPv6 unicast Address (ULA), or a Global Unicast Address (GUA).

If multiple IP addresses are to be returned in an Encrypted DNS option, these addresses are ordered in the preference for use by the client.

3.1.4. Why Not Separate Options for ADN and IP Addresses?

A single option is used to convey both the ADN and IP addresses. Otherwise a means to correlate an IP address conveyed in an option with an ADN conveyed in another option will be required if, for example, more than one ADN is supported by the network.

3.1.5. Service Parameters

Because distinct encrypted DNS protocols (e.g., DoT, DoH, and DoQ) may be provisioned by a network and that some of these protocols may make use of customized port numbers instead of default ones, the Encrypted DNS options are designed to return a set of service parameters. These parameters are encoded following the same rules for encoding SvcParams in Section 2.1 of [I-D.ietf-dnsop-svcb-https]. This encoding approach may increase the size of the options but it has the merit of relying upon an existing IANA registry and, thus, accommodating new encrypted DNS protocols and service parameters that
may be defined in the future.

The following service parameters MUST be supported by a DNR implementation:

alpn: Used to indicate the set of supported protocols (Section 7.1 of [I-D.ietf-dnsop-svcb-https]).

port: Used to indicate the target port number for the encrypted DNS connection (Section 7.2 of [I-D.ietf-dnsop-svcb-https]).

In addition, the following service parameters are RECOMMENDED to be supported by a DNR implementation:

ech: Used to enable Encrypted ClientHello (ECH) (Section 7.3 of [I-D.ietf-dnsop-svcb-https]).

dohpath: Used to supply a relative DoH URI Template (Section 5.1 of [I-D.ietf-add-svcb-dns]).

3.1.6. ADN Only Mode

The provisioning mode in which an ADN, a list of IP addresses, and a set of service parameters of the encrypted DNS resolver are supplied to a host SHOULD be used because the Encrypted DNS options are self-contained and do not require any additional DNS queries. The reader may refer to [RFC7969] for an overview of advanced capabilities that are supported by DHCP servers to populate configuration data (e.g., issue DNS queries).

In contexts where putting additional complexity on requesting hosts is acceptable, returning an ADN only can be considered. The supplied ADN will be passed to a local resolution library (a DNS client, typically) which will then issue Service Binding (SVCB) queries [I-D.ietf-add-svcb-dns]. These SVCB queries can be sent to the discovered encrypted DNS resolver itself or to the network-designated Do53 resolver. Note that this mode may be subject to active attacks, which can be mitigated by DNSSEC.

| How an ADN is passed to a local resolution library is implementation specific.

3.1.7. Encrypted DNS Options Ordering

The DHCP options defined in Sections 4 and 5 follow the option ordering guidelines in Section 17 of [RFC7227].
Likewise, the RA option (Section 6) adheres to the recommendations in Section 9 of [RFC4861].

3.1.8. DNR Validation Checks

On receipt of an Encrypted DNS option, the DHCP client (or IPv6 host) makes the following validation checks:

* The ADN is present and encoded as per Section 10 of [RFC8415].
* If additional data is supplied:
  - the service parameters are encoded following the rules specified in Section 2.1 of [I-D.ietf-dnsop-svcb-https].
  - the option includes at least one valid IP address and the "alpn" service parameter.
  - the service parameters do not include "ipv4hint" or "ipv6hint" service parameters.

If any of the checks fail, the receiver discards the received Encrypted DNS option.

3.1.9. DNR Information Using Other Provisioning Mechanisms

The provisioning mechanisms specified in this document may not be available in specific networks (e.g., some cellular networks exclusively use Protocol Configuration Options (PCOs) [TS.24008]) or may not be suitable in some contexts (e.g., need for a secure discovery). Other mechanisms may be considered in these contexts for the provisioning of encrypted DNS resolvers. It is RECOMMENDED that at least the following DNR information is made available to a requesting host:

* A service priority whenever the discovery mechanism does not rely on implicit ordering if multiple instances of the encrypted DNS are used.
* An authentication domain name. This parameter is mandatory.
* A list of IP addresses to locate the encrypted DNS resolver.
* A set of service parameters.
3.2. Handling Configuration Data Conflicts

If the encrypted DNS is discovered by a host using both RA and DHCP, the rules discussed in Section 5.3.1 of [RFC8106] MUST be followed.

DHCP/RA options to discover encrypted DNS resolvers (including, DoH URI Templates) takes precedence over Discovery of Designated Resolvers (DDR) [I-D.ietf-add-ddr] since DDR uses Do53 to an external DNS resolver, which is susceptible to both internal and external attacks whereas DHCP/RA is typically protected using the mechanisms discussed in Section 7.1.

If a client learns both Do53 and encrypted DNS resolvers from the same network, and absent explicit configuration otherwise, it is RECOMMENDED that the client uses the encrypted DNS resolvers for that network.

3.3. Validating Discovered Resolvers

This section describes a set of validation checks to confirm that an encrypted DNS resolver matches what is provided using DNR (e.g., DHCP or RA). Such validation checks do not intend to validate the security of the DNR provisioning mechanisms or the user’s trust relationship to the network.

If the local DNS client supports one of the discovered encrypted DNS protocols identified by Application Layer Protocol Negotiation (ALPN) protocol identifiers, the DNS client establishes an encrypted DNS session following the service priority of the discovered encrypted resolvers.

The DNS client verifies the connection based on PKIX validation [RFC5280] of the DNS resolver certificate and uses the validation techniques as described in [RFC6125] to compare the authentication domain name conveyed in the Encrypted DNS options to the certificate provided (see Section 8.1 of [RFC8310] for more details). The DNS client uses the default system or application PKI trust anchors unless configured otherwise to use explicit trust anchors. ALPN-related considerations can be found in Section 6.1 of [I-D.ietf-dnsop-svcb-https]. Operation considerations to check the revocation status of the certificate of an encrypted DNS resolver are discussed in Section 10 of [RFC8484].
3.4. Multihoming Considerations

Devices may be connected to multiple networks; each providing their own DNS configuration using the discovery mechanisms specified in this document. Nevertheless, it is out of the scope of this specification to discuss DNS selection of multi-interface devices. The reader may refer to [RFC6731] for a discussion of issues and an example of DNS resolver selection for multi-interfaced devices. Also, the reader may refer to [I-D.ietf-add-split-horizon-authority] for a discussion on how DNR and Provisioning Domains (PvDs) Key "dnsZones" (Section 4.3 of [RFC8801]) can be used in Split DNS environments (Section 6 of [RFC8499]).

4. DHCPv6 Encrypted DNS Option

4.1. Option Format

The format of the DHCPv6 Encrypted DNS option is shown in Figure 1.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       OPTION_V6_DNR           |         Option-length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Service Priority        |         ADN Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
-                        authentication-domain-name                  -
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Addr Length           |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
-                        ipv6-address(es)                       -
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
-                 Service Parameters (SvcParams)                -
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Figure 1: DHCPv6 Encrypted DNS Option
```

The fields of the option shown in Figure 1 are as follows:

Option-code: OPTION_V6_DNR (TBA1, see Section 9.1)

Option-length: Length of the enclosed data in octets. The option length is ('ADN Length' + 4) when only an ADN is included in the option.

Service Priority: The priority of this OPTION_V6_DNR instance
compared to other instances. This 16-bit unsigned integer is interpreted following the rules specified in Section 2.4.1 of [I-D.ietf-dnsop-svcb-https].

ADN Length: Length of the authentication-domain-name field in octets.

authentication-domain-name (variable length): A fully qualified domain name of the encrypted DNS resolver. This field is formatted as specified in Section 10 of [RFC8415].

An example of the authentication-domain-name encoding is shown in Figure 2. This example conveys the FQDN "doh1.example.com.", and the resulting Option-length field is 18.

```
+------+------+------+------+------+------+------+------+------+
| 0x04 |   d  |   o  |   h  |  1   | 0x07 |   e  |   x  |   a  |
+------+------+------+------+------+------+------+------+------+
|   m  |   p  |   l  |   e  | 0x03 |   c  |   o  |   m  | 0x00 |
+------+------+------+------+------+------+------+------+------+
```

Figure 2: An Example of the DNS authentication-domain-name Encoding

Addr Length: Length of enclosed IPv6 addresses in octets. When present, it MUST be a multiple of 16.

ipv6-address(es) (variable length): Indicates one or more IPv6 addresses to reach the encrypted DNS resolver. An address can be link-local, ULA, or GUA. The format of this field is shown in Figure 3.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-------------------------------------------------------------------+
| ipv6-address                                                        |
+-------------------------------------------------------------------+
| ...                                                                |
+-------------------------------------------------------------------+
```

Figure 3: Format of the IPv6 Addresses Field

Service Parameters (SvcParams) (variable length): Specifies a set of service parameters that are encoded following the rules in Section 2.1 of [I-D.ietf-dnsop-svcb-https]. Service parameters may include, for example, a list of ALPN protocol identifiers or
alternate port numbers. This field MUST include at least "alpn" SvcParam (Section 4.1 of [I-D.ietf-add-svcb-dns]). The service parameters MUST NOT include "ipv4hint" or "ipv6hint" SvcParams as they are superseded by the included IP addresses.

If no port service parameter is included, this indicates that default port numbers should be used. As a reminder, the default port number is 853 for DoT, 443 for DoH, and 853 for DoQ.

The length of this field is ('Option-length' - 6 - 'ADN Length' - 'Addr Length').

Note that "Addr Length", "ipv6-address(es)", and "Service Parameters (SvcParams)" fields are not present if the ADN-only mode is used (Section 3.1.6).

4.2. DHCPv6 Client Behavior

To discover an encrypted DNS resolver, the DHCPv6 client MUST include OPTION_V6_DNR in an Option Request Option (ORO), as in Sections 18.2.1, 18.2.2, 18.2.4, 18.2.5, 18.2.6, and 21.7 of [RFC8415].

The DHCPv6 client MUST be prepared to receive multiple instances of the OPTION_V6_DNR option; each option is to be treated as a separate encrypted DNS resolver. These instances SHOULD be processed following their service priority (i.e., smaller service priority indicates a higher preference).

The DHCPv6 client MUST silently discard any OPTION_V6_DNR that fails to pass the validation steps defined in Section 3.1.8.

The DHCPv6 client MUST silently discard multicast and host loopback addresses conveyed in OPTION_V6_DNR.

5. DHCPv4 Encrypted DNS Option

5.1. Option Format

The format of the DHCPv4 Encrypted DNS option is illustrated in Figure 4.
Figure 4: DHCPv4 Encrypted DNS Option

The fields of the option shown in Figure 4 are as follows:

Code: OPTION_V4_DNR (TBA2, see Section 9.2).

Length: Indicates the length of the enclosed data in octets.

DNR Instance Data: Includes the configuration data of an encrypted DNS resolver. The format of this field is shown in Figure 5.

Figure 5: DNR Instance Data Format

When several encrypted DNS resolvers are to be included, the "DNR Instance Data" field is repeated.

The fields shown in Figure 5 are as follows:
DNR Instance Data Length: Length of all following data in octets. This field is set to (`ADN Length' + 3) when only an ADN is provided for a DNR instance.

Service Priority: The priority of this instance compared to other DNR instances. This 16-bit unsigned integer is interpreted following the rules specified in Section 2.4.1 of [I-D.ietf-dnsop-svcb-https].

ADN Length: Length of the authentication-domain-name in octets.

authentication-domain-name (variable length): The authentication domain name of the encrypted DNS resolver. This field is formatted as specified in Section 10 of [RFC8415]. An example is provided in Figure 2.

Addr Length: Length of included IPv4 addresses in octets. When present, it MUST be a multiple of 4.

IPv4 Address(es) (variable length): Indicates one or more IPv4 addresses to reach the encrypted DNS resolver. Both private and public IPv4 addresses can be included in this field. The format of this field is shown in Figure 6. This format assumes that an IPv4 address is encoded as a1.a2.a3.a4.

```
|  a1 |  a2 |  a3 |  a4 |  a1 |  a2 | ... |
```

Figure 6: Format of the IPv4 Addresses Field

Service Parameters (SvcParams) (variable length): Specifies a set of service parameters that are encoded following the rules in Section 2.1 of [I-D.ietf-dnsop-svcb-https]. Service parameters may include, for example, a list of ALPN protocol identifiers or alternate port numbers. This field MUST include at least "alpn" SvcParam (Section 4.1 of [I-D.ietf-add-svcb-dns]). The service parameters MUST NOT include "ipv4hint" or "ipv6hint" SvcParams as they are superseded by the included IP addresses.

If no port service parameter is included, this indicates that default port numbers should be used.

The length of this field is (`DNR Instance Data Length' - 4 - `ADN Length' - `Addr Length').
Note that "Addr Length", "IPv4 Address(es)", and "Service Parameters (SvcParams)" fields are not present if the ADN-only mode is used (Section 3.1.6).

OPTION_V4_DNR is a concatenation-requiring option. As such, the mechanism specified in [RFC3396] MUST be used if OPTION_V4_DNR exceeds the maximum DHCPv4 option size of 255 octets.

5.2. DHCPv4 Client Behavior

To discover an encrypted DNS resolver, the DHCPv4 client requests the encrypted DNS resolver by including OPTION_V4_DNR in a Parameter Request List option [RFC2132].

The DHCPv4 client MUST be prepared to receive multiple DNR instance data in the OPTION_V4_DNR option; each instance is to be treated as a separate encrypted DNS resolver. These instances SHOULD be processed following their service priority (i.e., smaller service priority indicates a higher preference).

The DHCPv4 client MUST silently discard any OPTION_V4_DNR that fails to pass the validation steps defined in Section 3.1.8.

The DHCPv4 client MUST silently discard multicast and host loopback addresses conveyed in OPTION_V4_DNR.

6. IPv6 RA Encrypted DNS Option

6.1. Option Format

This section defines a new Neighbor Discovery option [RFC4861]: IPv6 RA Encrypted DNS option. This option is useful in contexts similar to those discussed in Section 1.1 of [RFC8106].

The format of the IPv6 RA Encrypted DNS option is illustrated in Figure 7.
The fields of the option shown in Figure 7 are as follows:

Type: 8-bit identifier of the Encrypted DNS option as assigned by IANA (TBA3, see Section 9.3).

Length: 8-bit unsigned integer. The length of the option (including the Type and Length fields) is in units of 8 octets.

Service Priority: 16-bit unsigned integer. The priority of this Encrypted DNS option instance compared to other instances. This field is interpreted following the rules specified in Section 2.4.1 of [I-D.ietf-dnsop-svcb-https].

Lifetime: 32-bit unsigned integer. The maximum time in seconds (relative to the time the packet is received) over which the discovered Authentication Domain Name is valid.

The value of Lifetime SHOULD by default be at least 3 * MaxRtrAdvInterval, where MaxRtrAdvInterval is the maximum RA interval as defined in [RFC4861].

A value of all one bits (0xffffffff) represents infinity.

A value of zero means that this Authentication Domain Name MUST no longer be used.
ADN Length: 16-bit unsigned integer. This field indicates the length of the authentication-domain-name field in octets.

authentication-domain-name (variable length): The authentication domain name of the encrypted DNS resolver. This field is formatted as specified in Section 10 of [RFC8415].

Addr Length: 16-bit unsigned integer. This field indicates the length of enclosed IPv6 addresses in octets. When present, it MUST be a multiple of 16.

ipv6-address(es) (variable length): One or more IPv6 addresses of the encrypted DNS resolver. An address can be link-local, ULA, or GUA.

All of the addresses share the same Lifetime value. Similar to [RFC8106], if it is desirable to have different Lifetime values per IP address, multiple Encrypted DNS options may be used.

The format of this field is shown in Figure 3.

SvcParams Length: 16-bit unsigned integer. This field indicates the length of the Service Parameters field in octets.

Service Parameters (SvcParams) (variable length): Specifies a set of service parameters that are encoded following the rules in Section 2.1 of [I-D.ietf-dnsop-svcb-https]. Service parameters may include, for example, a list of ALPN protocol identifiers or alternate port numbers. This field MUST include at least "alpn" SvcParam (Section 4.1 of [I-D.ietf-add-svcb-dns]). The service parameters MUST NOT include "ipv4hint" or "ipv6hint" SvcParams as they are superseded by the included IP addresses.

If no port service parameter is included, this indicates that default port numbers should be used.

Note that "Addr Length", "ipv6-address(es)", and "Service Parameters (SvcParams)" fields are not present if the ADN-only mode is used (Section 3.1.6).

The option MUST be padded with zeros so that the full enclosed data is a multiple of 8 octets (Section 4.6 of [RFC4861]).
6.2. IPv6 Host Behavior

The procedure for DNS configuration is the same as it is with any other Neighbor Discovery option [RFC4861]. In addition, the host follows the same procedure as the one described in Section 5.3.1 of [RFC8106] for processing received Encrypted DNS options with the formatting requirements in Section 6.1 and validation checks in Section 3.1.8 substituted for the length and fields validation.

The host MUST be prepared to receive multiple Encrypted DNS options in RAs. These instances SHOULD be processed following their service priority (i.e., smaller service priority indicates a higher preference).

The host MUST silently discard multicast and host loopback addresses conveyed in the Encrypted DNS options.

7. Security Considerations

7.1. Spoofing Attacks

DHCP/RA messages are not encrypted or protected against modification within the LAN. Unless mitigated (described below), the content of DHCP and RA messages can be spoofed or modified by active attackers, such as compromised devices within the local network. An active attacker (Section 3.3 of [RFC3552]) can spoof the DHCP/RA response to provide the attacker’s encrypted DNS resolver. Note that such an attacker can launch other attacks as discussed in Section 22 of [RFC8415]. The attacker can get a domain name with a domain-validated public certificate from a CA and host an encrypted DNS resolver.

Attacks of spoofed or modified DHCP responses and RA messages by attackers within the local network may be mitigated by making use of the following mechanisms:

* DHCPv6-Shield [RFC7610]: the network access node (e.g., a border router, a CPE, an Access Point (AP)) discards DHCP response messages received from any local endpoint.

* RA-Guard [RFC7113]: the network access node discards RAs messages received from any local endpoint.

* Source Address Validation Improvement (SAVI) solution for DHCP [RFC7513]: the network access node filters packets with forged source IP addresses.
The above mechanisms would ensure that the endpoint receives the correct configuration information of the encrypted DNS resolvers selected by the DHCP server (or RA sender), but cannot provide any information about the DHCP server or the entity hosting the DHCP server (or RA sender).

Encrypted DNS sessions with rogue resolvers that spoof the IP address of a DNS resolver will fail because the DNS client will fail to authenticate that rogue resolver based upon PKIX authentication [RFC6125], particularly the authentication domain name in the Encrypted DNS Option. DNS clients that ignore authentication failures and accept spoofed certificates will be subject to attacks (e.g., redirect to malicious resolvers, intercept sensitive data).

7.2. Deletion Attacks

If the DHCP responses or RAs are dropped by the attacker, the client can fall back to use a preconfigured encrypted DNS resolver. However, the use of policies to select resolvers is out of the scope of this document.

Note that deletion attack is not specific to DHCP/RA.

7.3. Passive Attacks

A passive attacker (Section 3.2 of [RFC3552]) can identify a host is using DHCP/RA to discover an encrypted DNS resolver and can infer that host is capable of using DoH/DoT/DoQ to encrypt DNS messages. However, a passive attacker cannot spoof or modify DHCP/RA messages.


Wireless LAN (WLAN) as frequently deployed in local networks (e.g., home networks) is vulnerable to various attacks (e.g., [Evil-Twin], [Krack], [Dragonblood]). Because of these attacks, only cryptographically authenticated communications are trusted on WLANs. This means that any information (e.g., NTP server, DNS resolver, domain search list) provided by such networks via DHCP, DHCPv6, or RA is untrusted because DHCP and RA messages are not authenticated.

If the pre-shared key (PSK) is the same for all clients that connect to the same WLAN (e.g., WPA-PSK), the shared key will be available to all nodes, including attackers. As such, it is possible to mount an active on-path attack. On-path attacks are possible within local networks because such a WLAN authentication lacks peer entity authentication.
This leads to the need for provisioning unique credentials for different clients. Endpoints can be provisioned with unique credentials (username and password, typically) provided by the local network administrator to mutually authenticate to the local WLAN AP (e.g., 802.1x Wireless User Authentication on OpenWRT [dot1x], EAP-pwd [RFC8146]). Not all endpoint devices (e.g., IoT devices) support 802.1x supplicant and need an alternate mechanism to connect to the local network. To address this limitation, unique pre-shared keys can be created for each such devices and WPA-PSK is used (e.g., [IPSK]).

8. Privacy Considerations

Privacy considerations that are specific to DNR provisioning mechanisms are discussed in Section 23 of [RFC8415] or [RFC7824]. Anonymity profiles for DHCP clients are discussed in [RFC7844]. The mechanism defined in this document can be used to infer that a DHCP client or IPv6 host support encrypted DNS options, but does not explicitly reveal whether local DNS clients are able to consume these options or infer their encryption capabilities. Other than that, this document does not expose more privacy information compared to Do53 discovery options.

As discussed in [RFC9076], the use of encrypted DNS does not reduce the data available in the DNS resolver. For example, the reader may refer to Section 8 of [RFC8484] or Section 7 of [RFC9250] for a discussion on specific privacy considerations to encrypted DNS.

9. IANA Considerations

9.1. DHCPv6 Option

IANA is requested to assign the following new DHCPv6 Option Code in the registry maintained in [DHCPV6].

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Client ORO</th>
<th>Singleton Option</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBA1</td>
<td>OPTION_V6_DNR</td>
<td>Yes</td>
<td>No</td>
<td>[ThisDocument]</td>
</tr>
</tbody>
</table>

Table 1: DHCPv6 Encrypted DNS Option
9.2. DHCPv4 Option

IANA is requested to assign the following new DHCP Option Code in the registry maintained in [BOOTP].

<table>
<thead>
<tr>
<th>Tag</th>
<th>Name</th>
<th>Data Length</th>
<th>Meaning</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBA2</td>
<td>OPTION_V4_DNR</td>
<td>N</td>
<td>Encrypted DNS Server</td>
<td>[ThisDocument]</td>
</tr>
</tbody>
</table>

Table 2: DHCPv4 Encrypted DNS Option

9.3. Neighbor Discovery Option

IANA is requested to assign the following new IPv6 Neighbor Discovery Option type in the "IPv6 Neighbor Discovery Option Formats" sub-registry under the "Internet Control Message Protocol version 6 (ICMPv6) Parameters" registry maintained in [ND].

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBA3</td>
<td>Encrypted DNS Option</td>
<td>[ThisDocument]</td>
</tr>
</tbody>
</table>

Table 3: Neighbor Discovery Encrypted DNS Option

10. Acknowledgements

Many thanks to Christian Jacquenet and Michael Richardson for the review.

Thanks to Stephen Farrell, Martin Thomson, Vittorio Bertola, Stephane Bortzmeyer, Ben Schwartz, Iain Sharp, and Chris Box for the comments.

Thanks to Mark Nottingham for the feedback on HTTP redirection that was discussed in previous versions of this specification.

The use of DHCP to retrieve an authentication domain name was discussed in Section 7.3.1 of [RFC8310] and [I-D.pusateri-dhc-dns-driu].

Thanks to Bernie Volz for the review of the DHCP part.
Thanks to Andrew Campling for the Shepherd review and Eric Vyncke for the AD review.

Thanks to Rich Salz for the secdir review, Joe Clarke for the ops-dir review, and Robert Sparks for the artart review.

Thanks to Lars Eggert, Roman Danyliw, Erik Kline, Martin Duke, Robert Wilton, and Paul Wouters for the IESG review.

11. Contributing Authors

Nicolai Leymann
Deutsche Telekom
Germany
Email: n.leymann@telekom.de

Zhiwei Yan
CNNIC
No.4 South 4th Street, Zhongguancun
Beijing
100190
China
Email: yan@cnnic.cn

12. References

12.1. Normative References

[I-D.ietf-add-svcb-dns]

[I-D.ietf-dnsop-svcb-https]
12.2. Informative References


[Dragonblood]

[Evil-Twin]

[I-D.ietf-add-ddr]

[I-D.ietf-add-split-horizon-authority]

[I-D.pusateri-dhc-dns-driu]

[IPSK]

[Krack]

[ND]
"IPv6 Neighbor Discovery Option Formats", <https://www.iana.org/assignments/icmpv6-parameters/icmpv6-parameters.xhtml#icmpv6-parameters-5>.

[RFC3552]


Authors’ Addresses

Mohamed Boucadair (editor)
Orange
35000 Rennes
France
Email: mohamed.boucadair@orange.com

Tirumaleswar Reddy (editor)
Akamai
Embassy Golf Link Business Park
Bangalore 560071
Karnataka
India
Email: kondtir@gmail.com

Dan Wing
Citrix Systems, Inc.
United States of America
Email: dwing-ietf@fuggles.com

Neil Cook
Open-Xchange
United Kingdom
Email: neil.cook@noware.co.uk

Tommy Jensen
Microsoft
United States of America
Email: tojens@microsoft.com
Abstract

When split-horizon DNS is deployed by a network, certain domains can be resolved authoritatively by the network-provided DNS resolver. DNS clients that don't always use this resolver might wish to do so for these domains. This specification describes how clients can confirm the local resolver’s authority over these domains.

Discussion Venues

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

Discussion of this document takes place on the Adaptive DNS Discovery Working Group mailing list (add@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/add/.

Source for this draft and an issue tracker can be found at https://github.com/ietf-wg-add/draft-ietf-add-split-horizon-authority.

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 26 December 2022.

Copyright Notice

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction ........................................... 3
2. Terminology ........................................... 4
3. Scope .................................................. 4
4. Local Domain Hint Mechanisms ............................ 5
4.1. DHCP Options ........................................ 5
4.2. Host Configuration ................................... 6
4.3. Provisioning Domains dnsZones ......................... 6
4.4. Split DNS Configuration for IKEv2 ..................... 6
5. Establishing Local DNS Authority ......................... 7
6. Validating Authority over Local Domain Hints .............. 7
6.1. Using a Pre-configured External Resolver .............. 7
6.2. Using DNSSEC ........................................ 8
7. Examples of Split-Horizon DNS Configuration .............. 8
7.1. Split-Horizon Entire Zone ............................ 8
7.1.1. Verification using an external resolver ........... 10
7.1.2. Verification using DNSSEC ........................ 11
7.2. Internal-only Subdomains ............................. 12
8. Validation with IKEv2 .................................. 13
9. Security Considerations ................................ 13
10. IANA Considerations .................................. 13
11. Acknowledgements ..................................... 13
12. References ........................................... 14
12.1. Normative References ............................... 14
12.2. Informative References ................................ 14
Authors’ Addresses ....................................... 16
1. Introduction

To resolve a DNS query, there are three essential behaviors that an implementation can apply: (1) answer from a local database, (2) query the relevant authorities and their parents, or (3) ask a server to query those authorities and return the final answer. Implementations that use these behaviors are called "authoritative nameservers", "full resolvers", and "forwarders" (or "stub resolvers"). However, an implementation can also implement a mixture of these behaviors, depending on a local policy, for each query. We term such an implementation a "hybrid resolver".

Most DNS resolvers are hybrids of some kind. For example, stub resolvers frequently support a local "hosts file" that preempts query forwarding, and most DNS forwarders and full resolvers can also serve responses from a local zone file. Other standardized hybrid resolution behaviors include Local Root [RFC8806], mDNS [RFC6762], and NXDOMAIN synthesis for .onion [RFC7686].

In many network environments, the network offers clients a DNS server (e.g. DHCP OFFER, IPv6 Router Advertisement). Although this server is formally specified as a recursive resolver (e.g. Section 5.1 of [RFC6106]), some networks provide a hybrid resolver instead. If this resolver acts as an authoritative server for some names, we say that the network has "split-horizon DNS", because those names resolve in this way only from inside the network.

Network clients that use pure stub resolution, sending all queries to the network-provided resolver, will always receive the split-horizon results. Conversely, clients that send all queries to a different resolver or implement pure full resolution locally will never receive them. Clients that strictly implement either of these resolution behaviors are out of scope for this specification. Instead, this specification enables hybrid clients to access split-horizon results from a network-provided hybrid resolver, while using a different resolution method for some or all other names.

There are several existing mechanisms for a network to provide clients with "local domain hints", listing domain names that have special treatment in this network (Section 4). However, none of the local domain hint mechanisms enable clients to determine whether this special treatment is authorized by the domain owner. Instead, these specifications require clients to make their own determinations about whether to trust and rely on these hints.
This specification describes a protocol between domains, networks, and clients that allows the network to establish its authority over a domain to a client (Section 5). Clients can use this protocol to confirm that a local domain hint was authorized by the domain (Section 6), which might influence its processing of that hint.

This specification relies on securely identified local DNS servers and globally valid NS records. Use of this specification is therefore limited to servers that support authenticated encryption and split-horizon DNS names that are properly rooted in the global DNS.

2. 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.

This document makes use of the terms defined in [RFC8499], e.g. "Global DNS". The following additional terms are used throughout the document:

Encrypted DNS  A DNS protocol that provides an encrypted channel between a DNS client and server (e.g., DoT, DoH, or DoQ).

Split-Horizon DNS  The DNS service provided by a resolver that also acts as an authoritative server for some names, providing resolution results that are meaningfully different from those in the Global DNS. (See "Split DNS" in Section 6 of [RFC8499].)

Validated Split-Horizon A split horizon configuration for some name is considered "validated" if the client has confirmed that a parent of that name has authorized this resolver to serve its own responses for that name. Such authorization generally extends to the entire subtree of names below the authorization point.

3. Scope

The protocol in this document is designed to support the ability of a domain owner to create or authorize a split-horizon view of their domain. The protocol does not support split-horizon views created by any other entity. Thus, DNS filtering is not enabled by this protocol.
The protocol is applicable to any type of network offering split-horizon DNS configuration. The endpoint does not need any prior configuration to confirm that a local domain hint was indeed authorized by the domain.

4. Local Domain Hint Mechanisms

There are various mechanisms by which a network client might learn "local domain hints", which indicate a special treatment for particular domain names upon joining a network. This section provides a review of some common and standardized mechanisms for receiving domain hints.

4.1. DHCP Options

There are several DHCP options that convey local domain hints of different kinds. The most directly relevant is RDNSS Selection [RFC6731], which provides "a list of domains ... about which the RDNSS has special knowledge", along with a "High", "Medium", or "Low" preference for each name. The specification notes the difficulty of relying on these hints without validation:

| Trustworthiness of an interface and configuration information received over the interface is implementation and/or node deployment dependent, and the details of determining that trust are beyond the scope of this specification. |

Other local domain hints in DHCP include the "Domain Name" [RFC2132], "Access Network Domain Name" [RFC5986], "Client FQDN" [RFC4702][RFC4704], and "Name Service Search" [RFC2937] options. This specification may help clients to interpret these hints. For example, a rogue DHCP server could use the "Client FQDN" option to assign a client the name "www.example.com" in order to prevent the client from reaching the true "www.example.com". A client could use this specification to check the network’s authority over this name, and adjust its behavior to avoid this attack if authority is not established.

The Domain Search option [RFC3397][RFC3646], which offers clients a way to expand short names into Fully Qualified Domain Names, is not a "local domain hint" by this definition, because it does not modify the processing of any specific domain. (The specification notes that this option can be a "fruitful avenue of attack for a rogue DHCP server", and provides a number of cautions against accepting it unconditionally.)
4.2. Host Configuration

A host can be configured with DNS information when it joins a network, including when it brings up VPN (which is also considered joining a(n additional) network, detailed in Section 8). Existing implementations determine the host has joined a certain network via SSID, IP subnet assigned, DNS server IP address or name, and other similar mechanisms. For example, one existing implementation determines the host has joined an internal network because the DHCP-assigned IP address belongs to the company’s IP range (as assigned by the regional IP addressing authority) and the DHCP-advertised DNS IP address is one used by IT at that network. Other mechanisms exist in other products but are not interesting to this specification; rather what is interesting is this step to determine "we have joined the internal corporate network" occurred and the DNS server is configured as authoritative for certain DNS zones (e.g., *.example.com).

Because a rogue network can simulate all or most of the above characteristics, this specification details how to validate these claims in Section 6.

4.3. Provisioning Domains dnsZones

Provisioning Domains (PvDs) are defined in [RFC7556] as sets of network configuration information that clients can use to access networks, including rules for DNS resolution and proxy configuration. The PvD Key "dnsZones" is defined in [RFC8801] as a list of "DNS zones searchable and accessible" in this provisioning domain. Attempting to resolve these names via another resolver might fail or return results that are not correct for this network.

4.4. Split DNS Configuration for IKEv2

In IKEv2 VPNs, the INTERNAL_DNS_DOMAIN configuration attribute can be used to indicate that a domain is "internal" to the VPN [RFC8598]. To prevent abuse, the specification notes various possible restrictions on the use of this attribute:

- If a client is configured by local policy to only accept a limited set of INTERNAL_DNS_DOMAIN values, the client MUST ignore any other INTERNAL_DNS_DOMAIN values. ([RFC8598], Section 5)

- IKE clients MAY want to require whitelisted domains for Top-Level Domains (TLDs) and Second-Level Domains (SLDs) to further prevent malicious DNS redirections for well-known domains. ([RFC8598], Section 9)
Within these guidelines, a client could adopt a local policy of accepting INTERNAL_DNS_DOMAIN values only when it can validate the local DNS server’s authority over those names as described in this specification.

5. Establishing Local DNS Authority

To establish its authority over some DNS zone, a participating network MUST offer one or more encrypted resolvers via DNR [I-D.ietf-add-dnr] or an equivalent mechanism (see Section 8). At least one of these resolvers’ Authentication Domain Names (ADNs) MUST appear in an NS record for that zone. This arrangement establishes this resolver’s authority over the zone.

6. Validating Authority over Local Domain Hints

To validate the network’s authority over a domain name, participating clients MUST resolve the NS record for that name. If the resolution result is NODATA, the client MUST remove the last label and repeat the query until a response other than NODATA is received.

Once the NS record has been resolved, the client MUST check if each local encrypted resolver’s Authentication Domain Name appears in the NS record. The client SHALL regard each such resolver as authoritative for the zone of this NS record.

Each validation of authority applies only to the specific resolvers whose names appear in the NS RRSet. If a network offers multiple encrypted resolvers, each DNS entry may be authorized for a distinct subset of the network-provided resolvers.

A zone is termed a "Validated Split-Horizon zone" after successful validation using a "tamperproof" NS resolution method, i.e. a method that is not subject to interference by the local network operator. Two possible tamperproof resolution methods are presented below.

6.1. Using a Pre-configured External Resolver

This method applies only if the client is already configured with a default resolution strategy that sends queries to a resolver outside of the network over a secure transport. That resolution strategy is considered "tamperproof" because any actor who could modify the NS response could already modify all of the user’s other DNS responses.

To ensure that this assumption holds, clients MUST NOT relax the acceptance rules they would otherwise apply when using this resolver. For example, if the client would check the AD bit or validate RRSIGs locally when using this resolver, it must also do so when resolving...
NS records for this purpose. Alternatively, a client might perform DNSSEC validation for the NS query used for this purpose even if it has disabled DNSSEC validation for other DNS queries.

6.2. Using DNSSEC

The client resolves the NS record using any resolution method of its choice (e.g. querying one of the network-provided resolvers, performing iterative resolution locally), and performs full DNSSEC validation locally [RFC6698]. The result is processed based on its DNSSEC validation state ([RFC4035], Section 4.3):

*Secure*: The response is used for validation.

*Bogus* or *Indeterminate*: The response is rejected and validation is considered to have failed.

*Insecure*: The client SHOULD retry the validation process using a different method, such as the one in Section 6.1, to ensure compatibility with unsigned names.

7. Examples of Split-Horizon DNS Configuration

Two examples are shown below. The first example shows a company with an internal-only DNS server that claims the entire zone for that company (e.g., *.example.com). In the second example, the internal servers resolves only a subdomain of the company’s zone (e.g., *.internal.example.com).

7.1. Split-Horizon Entire Zone

Consider an organization that operates "example.com", and runs a different version of its global domain on its internal network. Today, on the Internet it publishes two NS records, "ns1.example.com" and "ns2.example.com".

First, the host and network both need to support one of the discovery mechanisms described in Section 4. Figure 1 shows discovery using DNR and PvD.

Validation is then performed using either an external resolver (Section 7.1.1) or DNSSEC (Section 7.1.2).

*Steps 1-2*: The client determines the network’s DNS server (ns1.example.com) and Provisioning Domain (pvd.example.com) using DNR [I-D.ietf-add-dnr] and PvD [RFC8801], using one of DNR Router Solicitation, DHCPv4, or DHCPv6.
*Step 3-5*: The client connects to ns1.example.com using an encrypted transport as indicated in DNR [I-D.ietf-add-dnr], authenticating the server to its name using TLS ([RFC8310], Section 8), and sends it a query for the address of pvd.example.com.

*Steps 6-7*: The client connects to the PvD server, validates its certificate, and retrieves the provisioning domain JSON information indicated by the associated PvD. The PvD contains:

```json
{
    "identifier": "pvd.example.com",
    "expires": "2020-05-23T06:00:00Z",
    "prefixes": ["2001:db8:1::/48", "2001:db8:4::/48"],
    "dnsZones": ["example.com"]
}
```

The JSON keys "identifier", "expires", and "prefixes" are defined in [RFC8801].
7.1.1.  Verification using an external resolver

The figure below shows the steps performed to verify the local claims of DNS authority using an external resolver.
Steps 1-2*: The client uses an encrypted DNS connection to an external resolver (e.g., 1.1.1.1) to issue NS queries for the domains in dnsZones. The NS lookup for "example.com" will return "ns1.example.com" and "ns2.example.com".

Step 3*: The network-provided DNS servers are listed in the NS record for example.com, which was retrieved from an external resolver over a secure transport, so these ADNs are authorized. When the client connects using an encrypted transport as indicated in DNR [I-D.ietf-add-dnr], it will authenticate the server to its name using TLS ([RFC8310], Section 8), and send queries to resolve any names that fall within the dnsZones from PvD.

![Diagram](image)

Figure 2: Verifying claims using an external resolver

7.1.2. Verification using DNSSEC

The figure below shows the steps performed to verify the local claims of DNS authority using DNSSEC.
*Steps 1-2*: The DNSSEC-validating client queries the network encrypted resolver to issue NS queries for the domains in dnsZones. The NS lookup for "example.com" will return a signed response containing "ns1.example.com" and "ns2.example.com". The client then performs full DNSSEC validation locally.

*Step 3*: The DNSSEC validation is successful and the network-provided DNS servers are listed in the signed NS record for example.com, so these ADNs are authorized. When the client connects using an encrypted transport as indicated in DNR [I-D.ietf-add-dnr], it will authenticate the server to its name using TLS ([RFC8310], Section 8), and send queries to resolve any names that fall within the dnsZones from PvD.

---

**Figure 3: Verifying claims using DNSSEC**

7.2. Internal-only Subdomains

In many split-horizon deployments, all non-public domain names are placed in a separate child zone (e.g., internal.example.com). In this configuration, the message flow is similar to Section 7.1, except that queries for hosts not within the subdomain (e.g., www.example.com) are sent to the external resolver rather than resolver for internal.example.com.
As in Section 7.1, the internal DNS server will need a certificate signed by a CA trusted by the client.

8. Validation with IKEv2

When the VPN tunnel is IPsec, the encrypted DNS resolver hosted by the VPN service provider can be securely discovered by the endpoint using the ENCDNS_IP*_* IKEv2 Configuration Payload Attribute Types defined in [I-D.ietf-ipsecme-add-ike].

Other VPN tunnel types have similar configuration capabilities, not detailed here.

9. Security Considerations

This specification does not alter DNSSEC validation behaviour. To ensure compatibility with validating clients, network operators MUST ensure that names under the split-horizon are correctly signed or place them in an unsigned zone.

If an internal zone name (e.g., internal.example.com) is used with this specification and a public certificate is obtained for validation, that internal zone name will exist in Certificate Transparency logs [RFC9162]. In order to not leak the internal domains to an external resolver, the internal domains can be kept in a child zone of the local domain hints advertised by the network. For example, if the Pvd "dnsZones" entry is "internal.example.com" and the network-provided DNS resolver is "ns1.internal.example.com", the network operator can structure the internal domain names as "privatel.internal.example.com", "private2.internal.example.com", etc. The network-designated resolver will be used to resolve the subdomains of the local domain hint ".internal.example.com". Further, adversaries that monitor a network such as through passive monitoring or active probing of protocols, such as DHCP will only learn the local domain hints but not learn the labels below internal.example.com. However, security by obscurity may not maintain or increase the security of the internal domain names, as they may be leaked in various other ways (e.g., browser reload).

10. IANA Considerations

This document has no IANA actions.

11. Acknowledgements

Thanks to Mohamed Boucadair, Jim Reid, Tommy Pauly, Paul Vixie, Paul Wouters, Michael Richardson and Vinny Parla for the discussion and comments.
12. References

12.1. Normative References


12.2. Informative References


Establishing Local DNS Authority


Authors’ Addresses

Tirumaleswar Reddy
Akamai
Embassy Golf Link Business Park
Bangalore 560071
Karnataka
India
Email: kondtir@gmail.com

Abstract

The SVCB DNS resource record type expresses a bound collection of endpoint metadata, for use when establishing a connection to a named service. DNS itself can be such a service, when the server is identified by a domain name. This document provides the SVCB mapping for named DNS servers, allowing them to indicate support for encrypted transport protocols.

Discussion Venues

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

Discussion of this document takes place on the ADD Working Group mailing list (add@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/add/.

Source for this draft and an issue tracker can be found at https://github.com/bemasc/svcb-dns.

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 6 January 2023.
1. Introduction

The SVCB resource record type [SVCB] provides clients with information about how to reach alternative endpoints for a service, which may have improved performance or privacy properties. The service is identified by a "scheme" indicating the service type, a hostname, and optionally other information such as a port number. A DNS server is often identified only by its IP address (e.g., in DHCP), but in some contexts it can also be identified by a hostname (e.g., "NS" records, manual resolver configuration) and sometimes also a non-default port number.

Use of the SVCB resource record type requires a mapping document for each service type (Section 2.4.3 of [SVCB]), indicating how a client for that service can interpret the contents of the SVCB SvcParams. This document provides the mapping for the "dns" service type, allowing DNS servers to offer alternative endpoints and transports, including encrypted transports like DNS over TLS (DoT) [RFC7858], DNS over HTTPS (DoH) [RFC8484], and DNS over QUIC (DoQ) [RFC9250].

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. Identities and Names

SVCB record names (i.e., QNAMEs) for DNS services are formed using Port-Prefix Naming (Section 2.3 of [SVCB]), with a scheme of "dns". For example, SVCB records for a DNS service identified as dns1.example.com would be queried at _dns.dns1.example.com.

In some use cases, the name used for retrieving these DNS records is different from the server identity used to authenticate the secure transport. To distinguish between these, this document uses the following terms:

* Binding authority - The service name (Section 1.4 of [SVCB]) and optional port number used as input to Port-Prefix Naming.

* Authentication name - The name used for secure transport authentication. This MUST be a DNS hostname or a literal IP address. Unless otherwise specified, this is the service name from the binding authority.
3.1. Special case: non-default ports

Normally, a DNS service is identified by an IP address or a domain name. When connecting to the service using unencrypted DNS over UDP or TCP, clients use the default port number for DNS (53). However, in rare cases, a DNS service might be identified by both a name and a port number. For example, the "dns:" URI scheme [DNSURI] optionally includes an authority, comprised of a host and a port number (with a default of 53). DNS URIs normally omit the authority, or specify an IP address, but a hostname and non-default port number are allowed.

When the binding authority specifies a non-default port number, Port-Prefix Naming places the port number in an additional a prefix on the name. For example, if the binding authority is "dns1.example.com:9953", the client would query for SVCB records at _9953._dns.dns1.example.com. If two DNS services operating on different port numbers provide different behaviors, this arrangement allows them to preserve the distinction when specifying alternative endpoints.

4. Applicable existing SvcParamKeys

4.1. alpn

This key indicates the set of supported protocols (Section 6.1 of [SVCB]). There is no default protocol, so the "no-default-alpn" key does not apply, and the "alpn" key MUST be present.

If the protocol set contains any HTTP versions (e.g., "h2", "h3"), then the record indicates support for DoH, and the "dohpath" key MUST be present (Section 5.1). All keys specified for use with the HTTPS record are also permissible, and apply to the resulting HTTP connection.

If the protocol set contains protocols with different default ports, and no port key is specified, then protocols are contacted separately on their default ports. Note that in this configuration, ALPN negotiation does not defend against cross-protocol downgrade attacks.

4.2. port

This key is used to indicate the target port for connection (Section 6.2 of [SVCB]). If omitted, the client SHALL use the default port number for each transport protocol (853 for DoT and DoQ, 443 for DoH).
This key is automatically mandatory for this binding. This means that a client that does not respect the "port" key MUST ignore any SVCB record that contains this key. (See Section 7 of [SVCB] for the definition of "automatically mandatory").

Support for the "port" key can be unsafe if the client has implicit elevated access to some network service (e.g., a local service that is inaccessible to remote parties) and that service uses a TCP-based protocol other than TLS. A hostile DNS server might be able to manipulate this service by causing the client to send a specially crafted TLS SNI or session ticket that can be misparsed as a command or exploit. To avoid such attacks, clients SHOULD NOT support the "port" key unless one of the following conditions applies:

* The client is being used with a DNS server that it trusts not attempt this attack.

* The client is being used in a context where implicit elevated access cannot apply.

* The client restricts the set of allowed TCP port values to exclude any ports where a confusion attack is likely to be possible (e.g., the "bad ports" list from the "Port blocking" section of [FETCH]).

4.3. Other applicable SvcParamKeys

These SvcParamKeys from [SVCB] apply to the "dns" scheme without modification:

* mandatory
* ech
* ipv4hint
* ipv6hint

Future SvcParamKeys might also be applicable.

5. New SvcParamKeys
5.1. dohpath

"dohpath" is a single-valued SvcParamKey whose value (both in presentation and wire format) MUST be a URI Template in relative form ([RFC6570], Section 1.1) encoded in UTF-8 [RFC3629]. If the "alpn" SvcParam indicates support for HTTP, "dohpath" MUST be present. The URI Template MUST contain a "dns" variable, and MUST be chosen such that the result after DoH template expansion (Section 6 of [RFC8484]) is always a valid and functional ":path" value ([RFC9113], Section 8.3.1).

When using this SVCB record, the client MUST send any DoH requests to the HTTP origin identified by the "https" scheme, the authentication name, and the port from the "port" SvcParam (if present). HTTP requests MUST be directed to the resource resulting from DoH template expansion of the "dohpath" value.

Clients SHOULD NOT query for any "HTTPS" RRs when using "dohpath". Instead, the SvcParams and address records associated with this SVCB record SHOULD be used for the HTTPS connection, with the same semantics as an HTTPS RR. However, for consistency, service operators SHOULD publish an equivalent HTTPS RR, especially if clients might learn about this DoH service through a different channel.

6. Limitations

This document is concerned exclusively with the DNS transport, and does not affect or inform the construction or interpretation of DNS messages. For example, nothing in this document indicates whether the service is intended for use as a recursive or authoritative DNS server. Clients need to know the intended use of services based on their context.

Not all features of this specification will be applicable or effective in all contexts:

* If the authentication name is received over an insecure channel (e.g., a glue NS record), this specification cannot prevent the client from connecting to an attacker.

* Different transports might prove to be popular for different purposes (e.g., stub resolution vs. iterative resolution). Implementors are not obligated to implement all the defined transports, although doing so is beneficial for compatibility.
Where resolution speed is a high priority, the SVCB TargetName SHOULD follow the convention described in Section 11.2 of [SVCB], and the use of AliasMode records (Section 2.4.2 of [SVCB]) is NOT RECOMMENDED.

7.  Examples

* A resolver known as simple.example that supports DNS over TLS on port 853 (implicitly, as this is its default port):

  _dns.simple.example. 7200 IN SVCB 1 simple.example. alpn=dot

* A DoH-only resolver at https://doh.example/dns-query{?dns}. (DNS over TLS is not supported.):

  _dns.doh.example. 7200 IN SVCB 1 doh.example. (
    alpn=h2 dohpath=/dns-query{?dns} )

* A resolver known as resolver.example that supports:

  - DoT on resolver.example ports 853 (implicit in record 1) and 8530 (explicit in record 2), with "resolver.example" as the Authentication Domain Name,

  - DoQ on resolver.example port 853 (record 1),

  - DoH at https://resolver.example/dns-query{?dns} (record 1), and

  - an experimental protocol on fooexp.resolver.example:5353 (record 3):

    _dns.resolver.example. 7200 IN SVCB 1 resolver.example. (alpn=dot,doq,h2,h3 dohpath=/dns-query{?dns} )
    _dns.resolver.example. 7200 IN SVCB 2 resolver.example. (alpn=dot port=8530)
    _dns.resolver.example. 7200 IN SVCB 3 fooexp (port=5353 alpn=foo foo-info=... )

* A nameserver named ns.example. whose service configuration is published on a different domain:

  _dns.ns.example. 7200 IN SVCB 0 _dns.ns.nic.example.

8.  Security Considerations
8.1. Adversary on the query path

This section considers an adversary who can add or remove responses to the SVCB query.

During secure transport establishment, clients MUST authenticate the server to its authentication name, which is not influenced by the SVCB record contents. Accordingly, this draft does not mandate the use of DNSSEC. This draft also does not specify how clients authenticate the name (e.g., selection of roots of trust), which might vary according to the context.

8.1.1. Downgrade attacks

This attacker cannot impersonate the secure endpoint, but it can forge a response indicating that the requested SVCB records do not exist. For a SVCB-reliant client ([SVCB], Section 3) this only results in a denial of service. However, SVCB-optional clients will generally fall back to insecure DNS in this case, exposing all DNS traffic to attacks.

8.1.2. Redirection attacks

SVCB-reliant clients always enforce the authentication domain name, but they are still subject to attacks using the transport, port number, and "dohpath" value, which are controlled by this adversary. By changing these values in the SVCB answers, the adversary can direct DNS queries for $HOSTNAME to any port on $HOSTNAME, and any path on "https://$HOSTNAME". If the DNS client uses shared TLS or HTTP state, the client could be correctly authenticated (e.g., using a TLS client certificate or HTTP cookie).

This behavior creates a number of possible attacks for certain server configurations. For example, if https://$HOSTNAME/upload accepts any POST request as a public file upload, the adversary could forge a SVCB record containing dohpath=/upload{?dns}. This would cause the client to upload and publish every query, resulting in unexpected storage costs for the server and privacy loss for the client. Similarly, if two DoH endpoints are available on the same origin, and the service has designated one of them for use with this specification, this adversary can cause clients to use the other endpoint instead.

To mitigate redirection attacks, a client of this SVCB mapping MUST NOT identify or authenticate itself when performing DNS queries, except to servers that it specifically knows are not vulnerable to such attacks. If an endpoint sends an invalid response to a DNS query, the client SHOULD NOT send more queries to that endpoint.
Multiple DNS services MUST NOT share a hostname identifier (Section 3) unless they are so similar that it is safe to allow an attacker to choose which one is used.

8.2. Adversary on the transport path

This section considers an adversary who can modify network traffic between the client and the alternative service (identified by the TargetName).

For a SVCB-reliant client, this adversary can only cause a denial of service. However, because DNS is unencrypted by default, this adversary can execute a downgrade attack against SVCB-optional clients. Accordingly, when use of this specification is optional, clients SHOULD switch to SVCB-reliant behavior if SVCB resolution succeeds. Specifications making using of this mapping MAY adjust this fallback behavior to suit their requirements.

9. IANA Considerations

Per [SVCB] IANA is directed to add the following entry to the SVCB Service Parameters registry.

<table>
<thead>
<tr>
<th>Number</th>
<th>Name</th>
<th>Meaning</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>dohpath</td>
<td>DNS over HTTPS path template</td>
<td>(This document)</td>
</tr>
</tbody>
</table>

Table 1

Per [Attrleaf], IANA is directed to add the following entry to the DNS Underscore Global Scoped Entry Registry:

<table>
<thead>
<tr>
<th>RR TYPE</th>
<th>_NODE NAME</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVCB</td>
<td>_dns</td>
<td>(This document)</td>
</tr>
</tbody>
</table>

Table 2

10. References

10.1. Normative References
10.2. Informative References


Appendix A. Mapping Summary

This table serves as a non-normative summary of the DNS mapping for SVCB.

<table>
<thead>
<tr>
<th><em>Mapped scheme</em></th>
<th>&quot;dns&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>RR type</em></td>
<td>SVCB (64)</td>
</tr>
<tr>
<td><em>Name prefix</em></td>
<td>_dns for port 53, else _$PORT._dns</td>
</tr>
<tr>
<td><em>Required keys</em></td>
<td>alpn</td>
</tr>
<tr>
<td><em>Automatically Mandatory Keys</em></td>
<td>port</td>
</tr>
<tr>
<td><em>Special behaviors</em></td>
<td>Supports all HTTPS RR SvcParamKeys</td>
</tr>
<tr>
<td></td>
<td>Overrides the HTTPS RR for DoH</td>
</tr>
<tr>
<td></td>
<td>Default port is per-transport</td>
</tr>
<tr>
<td></td>
<td>No encrypted -&gt; cleartext fallback</td>
</tr>
</tbody>
</table>

Table 3

Acknowledgments

Thanks to the many reviewers and contributors, including Andrew Campling, Peter van Dijk, Paul Hoffman, Daniel Migault, Matt Norhoff, Eric Rescorla, Andreas Schulze, and Eric Vyncke.

Author's Address
Benjamin Schwartz
Google LLC
Email: bemasc@google.com
Unilateral Opportunistic Deployment of Encrypted Recursive-to-Authoritative DNS
draft-ietf-dprive-unilateral-probing-01

Abstract

This document sets out steps that DNS servers (recursive resolvers and authoritative servers) can take unilaterally (without any coordination with other peers) to defend DNS query privacy against a passive network monitor. The steps in this document can be defeated by an active attacker, but should be simpler and less risky to deploy than more powerful defenses.

The goal of this document is to simplify and speed deployment of opportunistic encrypted transport in the recursive-to-authoritative hop of the DNS ecosystem. With wider easy deployment of the underlying transport on an opportunistic basis, we hope to facilitate the future specification of stronger cryptographic protections against more powerful attacks.

About This Document

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

The latest revision of this draft can be found at https://dkg.gitlab.io/dprive-unilateral-probing/. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-dprive-unilateral-probing/.

Discussion of this document takes place on the DPRIVE Working Group mailing list (mailto:dns-privacy@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/dns-privacy/.

Source for this draft and an issue tracker can be found at https://gitlab.com/dkg/dprive-unilateral-probing.
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 12 January 2023.

Copyright Notice

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction ................................................. 3
   1.1. Requirements Language .................................. 4
   1.2. Terminology ............................................. 4
2. Priorities ................................................... 4
   2.1. Minimizing Negative Impacts ............................ 5
   2.2. Protocol Choices ....................................... 5
3. Guidance for Authoritative Servers .......................... 5
   3.1. Pooled Authoritative Servers Behind a Single IP
        Address .................................................. 6
   3.2. Authentication .......................................... 6
   3.3. Server Name Indication .................................. 7
   3.4. Resource Exhaustion ..................................... 7
       3.4.1. Pad Responses to Mitigate Traffic Analysis ...... 8
4. Guidance for Recursive Resolvers ............................ 8
   4.1. High-level Overview .................................... 8
1. Introduction

This document aims to provide guidance to implementers who want to simply enable protection against passive network observers.
In particular, it focuses on mechanisms that can be adopted unilaterally by recursive resolvers and authoritative servers, without any explicit coordination with the other parties. This guidance provides opportunistic security (see [RFC7435]) -- encrypting things that would otherwise be in the clear, without interfering with or weakening stronger forms of security.

The document also briefly introduces (but does not try to specify) how a future protocol might permit defense against an active attacker in Appendix A.

The protocol described here offers three concrete advantages to the Internet ecosystem:

* Protection from passive attackers of DNS queries in transit between recursive and authoritative servers.

* A roadmap for gaining real-world experience at scale with encrypted protections of this traffic.

* A bridge to some possible future protection against a more powerful attacker.

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.

1.2. Terminology

Unilateral: capable of opportunistic probing deployment without external coordination with any of the other parties

Do53: traditional cleartext DNS over port 53 ([RFC1035])

DoQ: DNS-over-QUIC ([I-D.ietf-dprive-dnsoquic])

DoT: DNS-over-TLS ([RFC7858])

DoH: DNS-over-HTTPS ([RFC8484])

Encrypted transports: DoQ, DoT, and DoH collectively

2. Priorities
2.1. Minimizing Negative Impacts

This document aims to minimize potentially negative impacts caused by the probing of encrypted transports -- for the systems that adopt these guidelines, for the parties that they communicate with, and for uninvolved third parties. The negative impacts that we specifically try to minimize are:

* excessive bandwidth use
* excessive use of computational resources (CPU and memory in particular)
* the potential for amplification attacks (where DNS resolution infrastructure is wielded as part of a DoS attack)

2.2. Protocol Choices

Although this document focuses specifically on strategies used by DNS servers, it does not go into detail on the specific protocols used because those protocols, in particular DoT and DoQ, are described in other documents.

This document does not pursue the use of DoH in this context, because a DoH client needs to know the path part of a DoH endpoint URL, and there are currently no mechanisms for a DNS resolver to predict the path on its own, in an opportunistic or unilateral fashion, without incurring in excessive use of resources. For instance, a recursive resolver in theory could guess the full path to a queried IP address by trying all the URL paths that the client has in records and see if one of those works, but even though it can be expected that this would work 99% of the time with fewer than 100 probes, this technique would likely incur in excessive resource consumption potentially leading to vulnerabilities and amplification attacks. The authors of this document particularly welcome ideas and contributions from the community that lead to a suitable mechanism for unilaterally probing for DoH-capable authoritative servers, for later consideration in this or other documents.

3. Guidance for Authoritative Servers

An authoritative server SHOULD implement and deploy DNS-over-TLS (DoT) on TCP port 853.

An authoritative server SHOULD implement and deploy DNS-over-QUIC (DoQ) on UDP port 853.
An authoritative server implementing the protocol described in this document MUST implement at least one of DoT or DoQ on port 853.

3.1. Pooled Authoritative Servers Behind a Single IP Address

Some authoritative DNS servers are structured as a pool of authoritatives standing behind a load-balancer that runs on a single IP address, forwarding queries to members of the pool.

In such a deployment, individual members of the pool typically get updated independently from each other.

A recursive resolver following the guidance in Section 4 that interacts with such a pool likely does not know that it is a pool. If some members of the pool follow this guidance while others do not, the recursive client might see the pool as a single authoritative server that sometimes offers and sometimes refuses encrypted transport.

To avoid incurring additional minor timeouts for such a recursive resolver, the pool operator SHOULD either:

* ensure that all members of the pool enable the same encrypted transport(s) within the span of a few seconds, or

* ensure that the load balancer maps client requests to pool members based on client IP addresses.

Similar concerns apply to authoritative servers responding from an anycast IP address. As long as the pool of servers is in a heterogeneous state, any flapping route that switches a given client IP address to a different responder risks incurring an additional timeout. Frequent changes of routing for anycast listening IP addresses are also likely to cause problems for TLS, TCP, or QUIC connection state as well, so stable routes are important to ensure that the service remains available and responsive.

3.2. Authentication

For unilateral deployment, an authoritative server does not need to offer any particular form of authentication.

The simplest deployment would simply provide a self-issued, regularly-updated X.509 certificate. This mechanism is supported by many TLS and QUIC clients, and will be acceptable for any opportunistic connection.
3.3. Server Name Indication

An authoritative DNS server that wants to handle unilateral queries MAY rely on Server Name Indication (SNI) to select alternate server credentials. However, such a server MUST NOT serve resource records that differ based on SNI (or on the lack of SNI) provided by the client, as a probing recursive resolver that offers SNI might or might not have used the right server name to get the records it’s looking for.

3.4. Resource Exhaustion

A well-behaved recursive resolver may keep an encrypted connection open to an authoritative server, to amortize the costs of connection setup for both parties.

However, some authoritative servers may have insufficient resources available to keep many connections open concurrently.

To keep resources under control, authoritative servers should proactively manage their encrypted connections. Section 6.5 of [I-D.ietf-dprive-dnsoquic] ("Connection Handling") offers useful guidance for servers managing DoQ connections. Section 3.4 of [RFC7858] offers useful guidance for servers managing DoT connections.

An authoritative server facing unforeseen resource exhaustion SHOULD cleanly close open connections from recursive resolvers based on the authoritative’s preferred prioritization.

In the case of unanticipated resource exhaustion, a reasonable prioritization scheme would be to close connections in this order, until resources are back in control:

* connections with no outstanding queries, ordered by idle time (longest idle time gets closed first)

* connections with outstanding queries, ordered by age of outstanding query (oldest outstanding query gets closed first)

When resources are especially tight, the authoritative server may also decline to accept new connections over encrypted transport.
3.4.1. Pad Responses to Mitigate Traffic Analysis

To increase the anonymity set for each response, the authoritative server SHOULD use a sensible padding mechanism for all responses it sends. For example, an implementation might use EDNS(0) padding [RFC7830] within an encrypted transport, or a DoQ client might make use of the PADDING frames found in Section 19.1 of [QUIC]). How much to pad is out of scope of this document, but a reasonable suggestion can be found in [RFC8467].

4. Guidance for Recursive Resolvers

This section outlines a probing policy suitable for unilateral adoption by any recursive resolver. Following this policy should not result in failed resolutions or significant delay.

4.1. High-level Overview

In addition to querying on Do53, the recursive resolver will try either or both of DoT and DoQ concurrently. The recursive resolver remembers what opportunistic encrypted transport protocols have worked recently based on a (clientIP, serverIP, protocol) tuple.

If a query needs to go to a given authoritative server, and the recursive resolver remembers a recent successful encrypted transport to that server, then it doesn’t send the query over Do53 at all. Rather, it only sends the query using the recently-good encrypted transport protocol.

If the encrypted transport protocol fails, the recursive resolver falls back to Do53 for that tuple. When any encrypted transport fails, the recursive resolver remembers that failure for a reasonable amount of time to avoid flooding a non-compatible server with requests that it cannot accept.

See the subsections below for a more detailed description of this protocol.

4.2. Overall Recursive Resolver Settings

A recursive resolver implementing this document needs to set system-wide values for some default parameters. These parameters may be set independently for each supported encrypted transport, though a simple implementation may keep the parameters constant across encrypted transports.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Suggested Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>persistence</td>
<td>How long should the recursive resolver remember successful encrypted transport connections?</td>
<td>3 days (259200 seconds)</td>
</tr>
<tr>
<td>damping</td>
<td>How long should the recursive resolver remember unsuccessful encrypted transport connections?</td>
<td>1 day (86400 seconds)</td>
</tr>
<tr>
<td>timeout</td>
<td>How long should the recursive resolver wait for an initiated encrypted connection to complete?</td>
<td>4 seconds</td>
</tr>
</tbody>
</table>

Table 1: Recursive resolver system parameters per encrypted transport

This document uses the notation E-foo to refer to the foo parameter for the encrypted transport E.

For example DoT-persistence would indicate the length of time that the recursive resolver will remember that an authoritative server had a successful connection over DoT.

This document also assumes that the resolver maintains a list of outstanding cleartext queries destined for the authoritative server’s IP address X. This list is referred to as Do53-queries[X]. This document does not attempt to describe the specific operation of sending and receiving cleartext DNS queries (Do53) for a recursive resolver. Instead it describes a "bolt-on" mechanism that extends the recursive resolver’s operation on a few simple hooks into the recursive resolver’s existing handling of Do53.

Implementers or deployers of DNS recursive resolvers that follow the strategies in this document are encouraged to report their preferred values of these parameters.

4.3. Recursive Resolver Requirements

To follow this guidance, a recursive resolver MUST implement at least one of either DoT or DoQ in its capacity as a client of authoritative nameservers.
A recursive resolver SHOULD implement the client side of DNS-over-TLS (DoT). A recursive resolver SHOULD implement the client side of DNS-over-QUIC (DoQ).

DoT queries from the recursive resolver MUST target TCP port 853, with an ALPN of "dot". DoQ queries from the recursive resolver MUST target UDP port 853, with an ALPN of "doq". ALPN is described in [RFC7301].

While this document focuses on the recursive-to-authoritative hop, a recursive resolver implementing these strategies SHOULD also accept queries from its clients over some encrypted transport (current common transports are DoH or DoT).

4.4. Authoritative Server Encrypted Transport Connection State

The recursive resolver SHOULD keep a record of the state for each authoritative server it contacts, indexed by the IP address of the authoritative server and the encrypted transports supported by the recursive resolver.

Each record should contain the following fields for each supported encrypted transport, each of which would initially be null:
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Retain Across Reset</th>
</tr>
</thead>
<tbody>
<tr>
<td>session</td>
<td>The associated state of any existing, established session (the structure of this value is dependent on the encrypted transport implementation). If session is not null, it may be in one of two states: pending or established</td>
<td>no</td>
</tr>
<tr>
<td>initiated</td>
<td>Timestamp of most recent connection attempt</td>
<td>yes</td>
</tr>
<tr>
<td>completed</td>
<td>Timestamp of most recent completed handshake</td>
<td>yes</td>
</tr>
<tr>
<td>status</td>
<td>Enumerated value of success or fail or timeout, associated with the completed handshake</td>
<td>yes</td>
</tr>
<tr>
<td>last-response</td>
<td>A timestamp of the most recent response received on the connection</td>
<td>yes</td>
</tr>
<tr>
<td>resumptions</td>
<td>A stack of resumption tickets (and associated parameters) that could be used to resume a prior successful connection</td>
<td>yes</td>
</tr>
<tr>
<td>queries</td>
<td>A queue of queries intended for this authoritative server, each of which has additional status early, unsent, or sent</td>
<td>no</td>
</tr>
<tr>
<td>last-activity</td>
<td>A timestamp of the most recent activity on the connection</td>
<td>no</td>
</tr>
</tbody>
</table>

Table 2: Recursive resolver state per authoritative IP, per encrypted transport

Note that the session fields in aggregate constitute a pool of open connections to different servers.
With the exception of the session, queries, and last-activity fields, this cache information should be kept across restart of the server unless explicitly cleared by administrative action.

This document uses the notation E-foo[X] to indicate the value of field foo for encrypted transport E to IP address X.

For example, DoT-initiated[192.0.2.4] represents the timestamp when the most recent DoT connection packet was sent to IP address 192.0.2.4.

4.4.1. Separate State for Each of the Recursive Resolver’s Own IP Addresses

Note that the recursive resolver should record this per-authoritative-IP state for each IP address it uses as it sends its queries. For example, if a recursive resolver can send a packet to authoritative servers from IP addresses 192.0.2.100 and 192.0.2.200, it should keep two distinct sets of per-authoritative-IP state, one for each source address it uses. Keeping these state tables distinct for each source address makes it possible for a pooled authoritative server behind a load balancer to do a partial rollout while minimizing accidental timeouts (see Section 3.1).

4.5. Maintaining Authoritative State by IP Address

In designing a probing strategy, the recursive resolver could record its knowledge about any given authoritative server with different strategies, including at least:

* the authoritative server’s IP address,

* the authoritative server’s name (the NS record used), or

* the zone that contains the record being looked up.

This document encourages the first strategy, to minimize timeouts or accidental delays.

A timeout (accidental delay) is most likely to happen when the recursive client believes that the authoritative server offers encrypted transport, but the actual server reached declines encrypted transport (or worse, filters the incoming traffic and does not even respond with an ICMP port closed message).

By associating state with the IP address, the recursive client is most able to avoid reaching a heterogeneous deployment.
For example, consider an authoritative server named ns0.example.com that is served by two installations (with two A records), one at 192.0.2.7 that follows this guidance, and one at 192.0.2.8 that is a legacy (cleartext port 53-only) deployment. A recursive client who associates state with the NS name and reaches .7 first will "learn" that ns0.example.com supports encrypted transport. A subsequent query over encrypted transport dispatched to .8 would fail, potentially delaying the response.

By associating the state with the authoritative IP address, the client can minimize the number of accidental delays introduced (see also Section 4.4.1 and Section 3.1).

4.6. Probing Policy

When a recursive resolver discovers the need for an authoritative lookup to an authoritative DNS server using IP address X, it retrieves the records associated with X from its cache.

The following sections presume that the time of the discovery of the need for lookup is time T0.

If any of the records discussed here are absent, they are treated as null.

The recursive resolver must decide whether to initially send a query over Do53, or over any of the supported encrypted transports (DoT or DoQ).

Note that a resolver might initiate this query via any or all of the known transports. When multiple queries are sent, the initial packets for each connection can be sent concurrently, similar to "Happy Eyeballs" ([RFC8305]). However, unlike Happy Eyeballs, when one transport succeeds, the other connections do not need to be terminated, but can instead be continued to establish whether the IP address X is capable of communicating on the relevant transport.

4.6.1. Sending a Query over Do53

For any of the supported encrypted transports E, if either of the following holds true, the resolver SHOULD NOT send a query to X over Do53:

* E-session[X] is in the established state, or

* E-status[X] is success, and (T0 - E-last-response[X]) < persistence
Otherwise, if there is no outstanding session for any encrypted transport, and the last successful encrypted transport connection was long ago, the resolver sends a query to X over Do53. When it does so, it inserts a handle for the query in Do53-queries[X].

4.6.2. Receiving a Response over Do53

When a response R for query Q arrives at the recursive resolver in cleartext sent over Do53 from authoritative server with IP address X, the recursive resolver should:

If Q is not in Do53-queries[X]:
* Discard R and process it no further (do not respond to a cleartext response to a query that is not outstanding)

Otherwise:
* Remove Q from Do53-queries[X]

If R is successful:
* If Q is in Do53-queries[X]:
  - Return R to the requesting client
* For each supported encrypted transport E:
  - If Q is in E-queries[X]:
    o Remove Q from E-queries[X]

But if R is unsuccessful (e.g. SERVFAIL):
* if Q is not in any of *-queries[X]:
  - Return SERVFAIL to the client

FIXME: What response should be sent to the client in the case that an extended DNS error ([RFC8914]) is offered in an authoritative’s response?
4.6.3. Initiating a Connection over Encrypted Transport

If any E-session[X] is in the established state, the recursive resolver SHOULD NOT initiate a new connection to X over Do53 or E, but should instead send queries to X through the existing session (see Section 4.6.8). If the recursive resolver has a preferred encrypted transport, but only a different transport is in the established state, it MAY also initiate a new connection to X over its preferred transport while concurrently sending the query over the established transport E.

Before considering whether to initiate a new connection over an encrypted transport, the timer should examine and possibly refresh its state for encrypted transport E to authoritative IP address X:

* if E-session[X] is in state pending, and
* T0 - E-initiated[X] > E-timeout, then
  - set E-session[X] to null and
  - set E-status[X] to timeout

When resources are available to attempt a new encrypted transport, the resolver should only initiate a new connection to X over E as long as one of the following holds true:

* E-status[X] is success, or
* E-status[X] is fail or timeout and (T0 - E-completed[X]) > damping, or
* E-status[X] is null and E-initiated[X] is null

When initiating a session to X over encrypted transport E, if E-resumptions[X] is not empty, one ticket should be popped off the stack and used to try to resume a previous session. Otherwise, the initial Client Hello handshake should not try to resume any session.

When initiating a connection, the resolver should take the following steps:

* set E-initiated[X] to T0
* store a handle for the new session (which should have pending state) in E-session[X]
* insert a handle for the query that prompted this connection in 
  E-queries[X], with status unsent or early, as appropriate (see 
  below).

4.6.3.1. Early Data

Modern encrypted transports like TLS 1.3 offer the chance to store 
"early data" from the client into the initial Client Hello in some 
contexts. A resolver that initiates a connection over a encrypted 
transport according to this guidance in a context where early data is 
possible SHOULD send the DNS query that prompted the connection in 
the early data, according to the sending guidance in Section 4.6.8.

If it does so, the status of Q in E-queries[X] should be set to early 
instead of unsent.

4.6.3.2. Resumption Tickets

When initiating a new connection (whether by resuming an old session 
or not), the recursive resolver SHOULD request a session resumption 
ticket from the authoritative server. If the authoritative server 
supplies a resumption ticket, the recursive resolver pushes it into 
the stack at E-resumptions[X].

4.6.3.3. Server Name Indication

For modern encrypted transports like TLS 1.3, most client 
implementations expect to send a Server Name Indication (SNI) in the 
Client Hello.

There are two complications with selecting or sending SNI in this 
unilateral probing:

* Some authoritative servers are known by more than one name; 
  selecting a single name to use for a given connection may be 
  difficult or impossible.

* In most configurations, the contents of the SNI field is exposed 
  on the wire to a passive adversary. This potentially reveals 
  additional information about which query is being made, based on 
  the NS of the query itself.

To avoid additional leakage and complexity, a recursive resolver 
following this guidance SHOULD NOT send SNI to the authoritative when 
attempting encrypted transport.
If the recursive resolver needs to send SNI to the authoritative for some reason not found in this document, it is RECOMMENDED that it implements Encrypted Client Hello ([I-D.ietf-tls-esni]) to reduce leakage.

4.6.3.4. Authoritative Server Authentication

Because this probing policy is unilateral and opportunistic, the client connecting under this policy MUST accept any certificate presented by the server. If the client cannot verify the server’s identity, it MAY use that information for reporting, logging, or other analysis purposes. But it MUST NOT reject the connection due to the authentication failure, as the result would be falling back to cleartext, which would leak the content of the session to a passive network monitor.

4.6.4. Establishing an Encrypted Transport Connection

When an encrypted transport connection actually completes (e.g., the TLS handshake completes) at time T1, the resolver sets E-completed[X] to T1 and does the following:

If the handshake completed successfully:

* update E-session[X] so that it is in state established
* set E-status[X] to success
* set E-last-response[X] to T1
* set E-completed[X] to T1
* for each query Q in E-queries[X]:
  - if early data was accepted and Q is early,
    o set the status of Q to sent
  - otherwise:
    o send Q through the session (see Section 4.6.8), and set the status of Q to sent

4.6.5. Failing to Establish an Encrypted Transport Connection

If, at time T2 an encrypted transport handshake completes with a failure (e.g. a TLS alert),
* set E-session[X] to null
* set E-status[X] to fail
* set E-completed[X] to T2
* for each query Q in E-queries[X]:
  - if Q is not present in any other *-queries[X] or in Do53-queries[X], add Q to Do53-queries[X] and send query Q to X over Do53.

Note that this failure will trigger the recursive resolver to fall back to cleartext queries to the authoritative server at IP address X. It will retry encrypted transport to X once the damping timer has elapsed.

4.6.6. Encrypted Transport Failure

Once established, an encrypted transport might fail for a number of reasons (e.g., decryption failure, or improper protocol sequence).

If this happens:
* set E-session[X] to null
* set E-status[X] to fail
* for each query Q in E-queries[X]:
  - if Q is not present in any other *-queries[X] or in Do53-queries[X], add Q to Do53-queries[X] and send query Q to X over Do53. FIXME: should a resumption ticket be used here for this previously successful connection?

Note that this failure will trigger the recursive resolver to fall back to cleartext queries to the authoritative server at IP address X. It will retry encrypted transport to X once the damping timer has elapsed.

FIXME: are there specific forms of failure that we might handle differently? For example, What if a TCP timeout closes an idle DoT connection? What if a QUIC stream ends up timing out but other streams on the same QUIC connection are going through? Do the described scenarios cover the case when an encrypted transport’s port is made unavailable/closed?
4.6.7. Handling Clean Shutdown of an Encrypted Transport Connection

At time T3, the recursive resolver may find that authoritative server X cleanly closes an existing outstanding connection (most likely due to resource exhaustion, see Section 3.4).

When this happens:

* set E-session[X] to null
* for each query Q in E-queries[X]:
  - if Q is not present in any other *-queries[X] or in Do53-queries[X], add Q to Do53-queries[X] and send query Q to X over Do53.

Note that this premature shutdown will trigger the recursive resolver to fall back to cleartext queries to the authoritative server at IP address X. Any subsequent query to X will retry the encrypted connection promptly.

4.6.8. Sending a Query over Encrypted Transport

When sending a query to an authoritative server over encrypted transport at time T4, the recursive resolver should take a few reasonable steps to ensure privacy and efficiency.

After sending query Q, the recursive resolver should ensure that Q’s state in E-queries[X] is set to sent.

The recursive resolver also sets E-last-activity[X] to T4.

In addition, the recursive resolver should consider the guidance in the following sections.

4.6.8.1. Avoid EDNS Client Subnet

To protect the privacy of the client, the recursive resolver SHOULD NOT send EDNS(0) Client Subnet information to the authoritative server ([RFC7871]) unless explicitly authorized to do so by the client.
4.6.8.2. Pad Queries to Mitigate Traffic Analysis

To increase the anonymity set for each query, the recursive resolver SHOULD use a sensible padding mechanism for all queries it sends. For example, an implementation might use EDNS(0) padding [RFC7830] within an encrypted transport, or a DoQ client might make use of the PADDING frames found in Section 19.1 of [QUIC]). How much to pad is out of scope of this document, but a reasonable suggestion can be found in [RFC8467].

4.6.8.3. Send Queries in Separate Channels

When multiple queries are multiplexed on a single encrypted transport to a single authoritative server, the recursive resolver MUST separate queries clearly and be capable of receiving responses out of order. For guidance on how to best achieve this on a given encrypted transport, see [RFC7766] (for DoT) and [I-D.ietf-dprive-dnsoquic] (for DoQ).

To the extent that the encrypted transport can avoid head-of-line blocking (e.g. QUIC can use a separate stream per query) the recursive resolver SHOULD avoid head-of-line blocking.

4.6.9. Receiving a Response over Encrypted Transport

When a response R for query Q arrives at the recursive resolver over encrypted transport E from authoritative server with IP address X at time T5, the recursive resolver should:

If Q is not in E-queries[X]:

* Discard R and process it no further (do not respond to a encrypted response to a query that is not outstanding)

Otherwise:

* Remove Q from E-queries[X]
* Set E-last-activity[X] to T5
* Set E-last-response[X] to T5

If R is successful:

* Return R to the requesting client
* For each supported encrypted transport N other than E:
- If Q is in N-queries[X]:
  o Remove Q from N-queries[X]
* If Q is in Do53-queries[X]:
  - Remove Q from Do53-queries[X]

But if R is unsuccessful (e.g. SERVFAIL):
* If Q is not in Do53-queries[X] or in any of *-queries[X]:
  - Return SERVFAIL to the requesting client

FIXME: What response should be sent to the client in the case that an extended DNS error ([RFC8914]) is offered in an authoritative’s response?

4.6.10. Resource Exhaustion

To keep resources under control, a recursive resolver should proactively manage outstanding encrypted connections. Section 6.5 of [I-D.ietf-dprive-dnsoquic] ("Connection Handling") offers useful guidance for clients managing DoQ connections. Section 3.4 of [RFC7858] offers useful guidance for clients managing DoT connections.

Even with sensible connection management, a recursive resolver doing unilateral probing may find resources unexpectedly scarce, and may need to close some outstanding connections.

In such a situation, the recursive resolver SHOULD use a reasonable prioritization scheme to close outstanding connections.

One reasonable prioritization scheme would be:
* close outstanding established sessions based on E-last-activity[X]
  (oldest timestamp gets closed first)

Note that when resources are limited, a recursive resolver following this guidance may also choose not to initiate new connections for encrypted transport.

4.6.11. Maintaining Connections

Some recursive resolvers looking to amortize connection costs, and to minimize latency MAY choose to synthesize queries to a particular resolver to keep an encrypted transport session active.
A recursive resolver that adopts this approach should try to align the synthesized queries with other optimizations. For example, a recursive resolver that "pre-fetches" a particular resource record to keep its cache "hot" can send that query over an established encrypted transport session.

5. IANA Considerations

IANA does not need to do anything for implementers to adopt the guidance found in this document.

6. Privacy Considerations

6.1. Server Name Indication

A recursive resolver querying an authoritative server over DoT or DoQ that sends Server Name Indication (SNI) in the clear in the cryptographic handshake leaks information about the intended query to a passive network observer.

In particular, if two different zones refer to the same nameserver IP addresses via differently-named NS records, a passive network observer can distinguish queries to one zone from the queries to the other.

Omitting SNI entirely, or using Encrypted Client Hello to hide the intended SNI, avoids this additional leakage. However, a series of queries that leak this information is still an improvement over the all-cleartext status quo at the time of this document.

7. Security Considerations

The guidance in this document provides defense against passive network monitors for most queries. It does not defend against active attackers. It can also leak some queries and their responses due to "happy eyeballs" optimizations when the resolver’s cache is cold.

Implementation of the guidance in this document should increase deployment of opportunistic encrypted DNS transport between recursive resolvers and authoritative servers at little operational risk.

However, implementers cannot rely on the guidance in this document for robust defense against active attackers, but should treat it as a stepping stone en route to stronger defense.

In particular, a recursive resolver following this guidance can easily be forced by an active attacker to fall back to cleartext DNS queries. Or, an active attacker could position itself as a machine-
in-the-middle, which the recursive resolver would not defend against or detect due to lack of server authentication. Defending against these attacks without risking additional unexpected protocol failures would require signalling and coordination that are out of scope for this document.

This guidance is only one part of operating a privacy-preserving DNS ecosystem. A privacy-preserving recursive resolver should adopt other practices as well, such as QNAME minimization ([RFC9156]), local root zone ([RFC8806]), etc, to reduce the overall leakage of query information that could infringe on the client’s privacy.

8. Acknowledgements

Many people contributed to the development of this document beyond the authors, including Alexander Mayrhofer, Brian Dickson, Christian Huitema, Eric Nygren, Jim Reid, Kris Shrishak, Ralf Weber, Robert Evans, Sara Dickinson, and the DPRIVE working group.

9. References

9.1. Normative References


9.2. Informative References


[I-D.ietf-dprive-dnsoquic]


Appendix A. Defense Against Active Attackers

The protocol described in this document provides no defense against active attackers. A future protocol for recursive-to-authoritative DNS might want to provide such protection.

This appendix assumes that the use case for that future protocol is a recursive resolver that wants to prevent an active attack on communication between it and an authoritative server that has committed to offering encrypted DNS transport. An inherent part of this use case is that the recursive resolver would want to respond with a SERVFAIL response to its client if it cannot make an authenticated encrypted connection to any of the authoritative nameservers for a name.

However, an authoritative server that merely offers encrypted transport (for example, by following the guidance in Section 3) has made no such commitment, and no recursive resolver that prioritizes delivery of DNS records to its clients would want to "fail closed" unilaterally.

So such a future protocol would need at least three major distinctions from the protocol described in this document:

* A signaling mechanism that tells the resolver that the authoritative server intends to offer authenticated encryption

* Authentication of the authoritative server

* A way to combine defense against an active attacker with the defenses described in this document

This can be thought of as a DNS analog to [MTA-STS] or [DANE-SMTP].

A.1. Signalling Mechanism Properties

To defend against an active attacker, the signalling mechanism needs to be able to indicate that the recursive resolver should "fail closed" if it cannot authenticate the server for a particular query.

The signalling mechanism itself would have to be resistant to downgrade attacks from active attackers.

One open question is how such a signal should be scoped. While this document scopes opportunistic state about encrypted transport based on the IP addresses of the client and server, signalled intent to offer encrypted transport is more likely to be scoped by queried zone in the DNS, or by nameserver name than by IP address.
A reasonable authoritative server operator or zone administrator probably doesn’t want to risk breaking anything when they first enable the signal. Therefore, a signalling mechanism should probably also offer a means to report problems to the authoritative server operator without the client failing closed. Such a mechanism is likely to be similar to [TLSRPT] or [DNS-Error-Reporting].

A.2. Authentication of Authoritative Server

Forms of server authentication might include:

* an X.509 Certificate issued by a widely-known certification authority associated with the common NS names used for this authoritative server

* DANE authentication (to avoid infinite recursion, the DNS records necessary to authenticate could be transmitted in the TLS handshake using the DNSSEC Chain Extension (see [RFC9102]))

A recursive resolver would have to verify the server’s identity. When doing so, the identity would presumably be based on the NS name used for a given query or the IP address of the server.

A.3. Combining Protocols

If this protocol gains reasonable adoption, and a newer protocol that can offer defense against an active attacker were available, deployment is likely to be staggered and incomplete. This means that an operator that want to maximize confidentiality for their users will want to use both protocols together.

Any new stronger protocol should consider how it interacts with the opportunistic protocol defined here, so that operators are not faced with the choice between widespread opportunistic protection against passive attackers (this document) and more narrowly-targeted protection against active attackers.

Appendix B. Document Considerations

[ RFC Editor: please remove this section before publication ]

B.1. Document History

B.1.1. draft-ietf-dprive-unilateral-probing Substantive Changes from -00 to -01

* Moved discussion of non-opportunistic encryption to an appendix
* Clarify state transitions when sending over encrypted transport

* Introduced new field E-last-response[X] for comparison with persistence

B.1.2. Substantive Changes from -01 to -02 (now draft-ietf-dprive-unilateral-probing-00)

* Clarify that deployment to a pool does not need to be strictly simultaneous

* Explain why authoritatives need to serve the same records regardless of SNI

* Defer to external, protocol-specific references for resource management

* Clarify that probed connections must not fail due to authentication failure

B.1.3. draft-dkgjsal-dprive-unilateral-probing Substantive Changes from -00 to -01

* Fallback to cleartext when encrypted transport fails.

* Reduce default timeout to 4s

* Clarify SNI guidance: OK for selecting server credentials, not OK for changing answers

* Document ALPN and port numbers

* Justify sorting recursive resolver state by authoritative IP address

Authors’ Addresses

Daniel Kahn Gillmor (editor)
American Civil Liberties Union
125 Broad St.
New York, NY, 10004
United States of America
Email: dkg@fifthhorseman.net
DNS Resolver Information
draft-reddy-add-resolver-info-05

Abstract

This document specifies a method for DNS resolvers to publish information about themselves. Clients can use the resolver information to identify the capabilities of DNS resolvers.

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 15 October 2022.

Copyright Notice

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.
1. Introduction

Historically, DNS stub resolvers communicated with recursive resolvers without needing to know anything about the features supported by these recursive resolvers. As more and more recursive resolvers expose different features that may impact the delivered DNS service, means to help stub resolvers to identify the capabilities of the resolver are valuable. Typically, stub resolvers can discover and authenticate encrypted DNS servers provided by a local network, for example, using the techniques specified in [I-D.ietf-add-dnr] and [I-D.ietf-add-ddr]. However, these stub resolvers need a means to retrieve information from the discovered recursive resolvers about their capabilities.

This document fills that void by specifying a method for stub resolvers to retrieve such information. To that aim, a new RRtype is defined for stub resolvers to query the recursive resolvers. The information that a resolver might want to give is defined in Section 5.

Retrieved information can be used to feed the server selection procedure.

2. 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.
This document makes use of the terms defined in [RFC8499].

'Encrypted DNS' refers to a DNS protocol that provides an encrypted channel between a DNS client and server (e.g., DoT, DoH, or DoQ).

3. Retrieving Resolver Information

A stub resolver that wants to retrieve the resolver information may use the RRtype "RESINFO" defined in this document (see Section 7.1).

The content of the RDATA in a response to RRtype query is defined in Section 5. If the resolver understands the RESINFO RRtype, the RRset in the Answer section MUST have exactly one record.

The client can retrieve the resolver information using the RESINFO RRtype and QNAME of the domain name that is used to authenticate the DNS server (referred to as ADN in [I-D.ietf-add-dnr]).

If the special use domain name "resolver.arpa" defined in [I-D.ietf-add-ddr] is used to discover the Encrypted DNS server, the client can retrieve the resolver information using the RESINFO RRtype and QNAME of the designated resolver.

4. Format of the Resolver Information

The resolver information is returned as a JSON object. Precisely, the JSON object MUST use the I-JSON message format [RFC7493].

Note that [RFC7493] was based on [RFC7159], but [RFC7159] was replaced by [RFC8259]. Requiring the use of I-JSON instead of more general JSON format greatly increases the likelihood of interoperability.

The JSON object returned by a DNS query may contain any name/value pairs. All names MUST consist only of lower-case ASCII characters, digits, and hyphens (that is, Unicode characters U+0061 through U+007A, U+0030 through U+0039, and U+002D). These names MUST be 63 characters or shorter.

All names in the returned object MUST either be defined in the IANA registry Section 7.2 or begin with the substring "temp-" for names defined for local use only.

5. Resolver Information

The resolver information includes the following attributes:

qnamememinimization: If the DNS server supports QNAME minimisation
[RFC7816] to improve DNS privacy, the parameter value is set to true. This is a mandatory attribute.

extendeddnserror: If the DNS server supports extended DNS error (EDE) [RFC8914] to return additional information about the cause of DNS errors, the parameter lists the possible extended DNS error codes that can be returned by the DNS server. This is an optional attribute.

resinfourl: An URL that points to the generic unstructured resolver information (e.g., DoH APIs supported, possible HTTP status codes returned by the DoH server, how to report a problem) for troubleshooting purpose. The server MUST support the content-type ‘text/html’. The DNS client MUST reject the URL if the scheme is not "https". The client MUST validate that both the encrypted DNS server and the resolver information server are owned and managed by the same entity by establishing a TLS connection to the domain name in the URL and checking if the subjectAltName entry in the server certificate includes the name of the encrypted DNS server. If this match fails, the client MUST ignore the resolver information. As such, the URL should be treated only as diagnostic information for IT staff. This is a mandatory attribute.

New attributes can be defined as per the procedure defined in Section 7.2.

As specified in [RFC7493], the I-JSON object is encoded as UTF8. [RFC7493] explicitly allows the returned objects to be in any order.

Figure 1 shows an example of resolver information.

```json
{
  "qnameminimization": true,
  "extendeddnserror": [15, 16, 17],
  "resinfourl": "https://resolver.example.com/guide",
}
```

Figure 1: An Example of Resolver Information
6. Security Considerations

Unless a DNS request to retrieve the resolver information is encrypted (e.g., sent over DNS-over-TLS (DoT) [RFC7858] or DNS-over-HTTPS (DoH)) [RFC8484], the response is susceptible to forgery. The DNS resolver information can be retrieved after the encrypted connection is established to the DNS server or retrieved before the encrypted connection is established to the DNS server by using local DNSSEC validation.

7. IANA Considerations

Note to the RFC Editor: Please update [RFCXXXX] with the RFC number to be assigned to this document.

7.1. RESINFO RRtype

This document requests IANA to register a new value from the "Resource Record (RR) TYPES" subregistry of the "Domain Name System (DNS) Parameters" registry available at [RRTYPE]:

Type: RESINFO
Value: TBD
Meaning: Resolver Information as an I-JSON
Reference: [RFCXXXX]

7.2. DNS Resolver Information Registration

This document requests IANA to create a new registry entitled "DNS Resolver Information". This registry contains definitions of the names that can be used to provide the resolver information.

The registration procedure is Specification Required (Section 4.6 of [RFC8126]).

The structure of the registry is as follows:

Name: The name to be used in the JSON object. The name MUST conform to the definition of "string" in I-JSON message format. The IANA registry MUST NOT register names that begin with "temp-", so these names can be used freely by any implementer.

Value Type: The type of data to be used in the JSON object.

Description: Provides a description of the attribute

Specification: The reference specification for the registered element.
The initial content of this registry is provided in Table 1.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value Type</th>
<th>Specification</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>qnameminimization</td>
<td>boolean</td>
<td>Indicates whether qnameminimization is enabled or not</td>
<td>[RFCXXXX]</td>
</tr>
<tr>
<td>extendeddnserror</td>
<td>number</td>
<td>Lists the set of extended DNS errors</td>
<td>[RFCXXXX]</td>
</tr>
<tr>
<td>resinfourl</td>
<td>string</td>
<td>Provides an unstructured resolver information that is used for troubleshooting</td>
<td>[RFCXXXX]</td>
</tr>
</tbody>
</table>

Table 1: Initial RESINFO Registry

8. Acknowledgments

This specification leverages the work that has been documented in [I-D.pp-add-resinfo].

Thanks to Tommy Jensen, Vittorio Bertola, Vinny Parla, Chris Box, Ben Schwartz, Tony Finch, Daniel Kahn Gillmor, Eric Rescorla and Shashank Jain for the discussion and comments.

9. References

9.1. Normative References


9.2. Informative References

[I-D.ietf-add-ddr]

[I-D.ietf-add-dnr]

[I-D.pp-add-resinfo]


Authors’ Addresses

Tirumaleswar Reddy
Akamai
Embassy Golf Link Business Park
Bangalore 560071
Karnataka
India
Email: kondtir@gmail.com

Mohamed Boucadair
Orange
35000 Rennes
France
Email: mohamed.boucadair@orange.com
Abstract

This draft describes an extension to the Discovery of Designated Resolvers (DDR) standard, enabling use of encrypted DNS in the presence of legacy DNS forwarders.

Discussion Venues

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

Discussion of this document takes place on the mailing list (add@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/add/.

Source for this draft and an issue tracker can be found at https://github.com/bemasc/ddr-forwarders.

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 9 January 2023.

Copyright Notice

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction ............................................. 3
1.1. Scope ................................................. 4
2. Conventions and Definitions .............................. 4
3. Reputation Verified Selection (RVS) ....................... 4
3.1. Reputation systems ..................................... 6
3.2. Using resolvers of intermediate reputation .......... 6
4. Management of local blocking functionality .............. 7
4.1. Local implementation with DNR ......................... 7
4.2. Local implementation with DDR ........................ 7
4.3. Move upstream ......................................... 7
5. Compatibility issues that can arise from cross-forwarder upgrade ..................................... 7
5.1. Split-horizon namespaces ................................ 8
5.1.1. Mitigation: NXDOMAIN Fallback ...................... 8
5.2. Interposable domains ................................... 8
5.2.1. Mitigation: Exemption list .......................... 9
5.3. Caching ................................................ 9
5.3.1. Mitigation: Stub caches .............................. 9
6. Privacy Considerations ................................... 9
6.1. Privacy gains ......................................... 9
6.2. Privacy losses ........................................ 9
6.2.1. Mitigation: Open multiple connections .......... 10
7. Security Considerations ................................ 10
7.1. Redirection .......................................... 10
7.1.1. Possible weakness: Stale reputation .............. 10
7.1.2. Possible weakness: Inappropriate reputation .... 10
7.2. Forensic logging ...................................... 11
7.2.1. Network-layer logging ............................. 11
7.2.2. DNS-layer logging .................................. 11
8. References .............................................. 11
8.1. Normative References ................................ 11
8.2. Informative References ................................ 11
Acknowledgments .............................................. 13
Authors’ Addresses ........................................ 13
1. Introduction

The Discovery of Designated Resolvers specification [DDR] describes a mechanism for clients to learn about the encrypted protocols supported by a DNS server. It also describes a client validation policy that has strong security properties.

Recent estimates suggest that a large fraction, perhaps a majority, of residential internet users in the United States and Europe rely on local DNS forwarders that are not compatible with DDR. This is because they are accessed via a private IP address, which TLS certificates cannot normally prove ownership of. Many such devices also face significant hurdles in being upgraded to support encrypted DNS, so it is likely that a large installed base of legacy DNS forwarders, providing Do53 on a private IP address, will remain for some years.

A client in such a network that wants to use the network’s DNS resolver is forced to use Do53. It is therefore vulnerable to passive surveillance both on the local network, and between this network and the upstream provider, even if the upstream DNS resolver supports encrypted DNS.

Many of these attacks can be mitigated by using the method described in this document. In a nutshell the process is as follows.

1. The client begins DDR discovery, querying for _dns.resolver.arpa.
2. The legacy DNS forwarder, since it does not understand DDR, forwards this query upstream.
3. The upstream recursive resolver, which supports DDR, replies with details of how to access its encrypted DNS service.
4. The client receives this response and performs Reputation Verified Selection (see Section 3).
5. On successful completion, the client may commence using encrypted DNS towards the upstream resolver. This is known as Cross-Forwarder Upgrade.

By this process, Do53 is replaced with encrypted DNS for most queries. The client may wish to continue to send locally-relevant queries (e.g. .local) towards the legacy DNS forwarder.
1.1. Scope

This document describes the interaction between DDR and legacy DNS forwarders.

DNS forwarders and resolvers that are implemented with awareness of DDR are out of scope, as they are not affected by this discussion (although see Security Considerations, Section 7).

IPv6-only networks whose default DNS server has a Global Unicast Address are out of scope, even if this server is actually a simple forwarder. If the DNS server does not use a private IP address, it is not a "legacy DNS forwarder" under this draft’s definition.

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.

Private IP Address - Any IP address reserved for loopback [RFC1122], link-local [RFC3927], private [RFC1918], local [RFC4193], or Carrier-Grade NAT [RFC6598] use.

Legacy DNS Forwarder - An apparent DNS resolver, known to the client only by a private IP address, that forwards the client’s queries to an upstream resolver, and has not been updated with any knowledge of DDR.

Cross-Forwarder Upgrade - Establishment and use of a direct, encrypted connection between the client and the upstream resolver.

3. Reputation Verified Selection (RVS)

Reputation Verified Selection (RVS) is a method for validating whether connection using DDR is allowed. Clients MAY use RVS when (a) the local DNS server is identified by a Private IP address and (b) the DDR SVCB resolution process does not produce any Encrypted DNS endpoints that have this IP address in their A or AAAA records. RVS then proceeds as follows:

1. The client connects to one of the indicated Encrypted DNS endpoints.

2. The client receives a certificate, which it verifies to a suitable root of trust.
3. For each identity (e.g. SubjectAltName) in the certificate, the client constructs a Resolver Identity:

* For DNS over TLS and DNS over QUIC, the Resolver Identity is an IP address or hostname and the port number used for the connection.

* For DNS over HTTPS, the Resolver Identity is a URI Template in absolute form, containing the port number used for the connection and path indicated by dohpath.

4. The client determines the reputation of each Resolver Identity derived from the certificate.

5. The maximum (i.e. most favorable) reputation is the reputation of this connection.

Successful validation then permits cross-forwarder upgrade.

OPEN QUESTION: Would it be better to use the SVCB TargetName to select a single Resolver Identity? This would avoid the need to enumerate the certificate’s names, but it would require the use of SNI (unlike standard DDR), and would not be compatible with all upstream encrypted resolvers.

OPEN QUESTION: Can we simplify the resolver identity to just a domain name? This would make reputation systems easier, but it would not allow distinct reputation for different colocated resolution services, so reputation providers would have to be sure that no approved resolver has other interesting colocated services.

This process MUST be repeated whenever a new TLS session is established, but reputation scores for each resolver endpoint MAY be cached.

For DNS over HTTPS, the :authority pseudo-header MUST reflect the Resolver Identity with the most favorable reputation, to ensure that the HTTP requests are well-formed and are directed to the intended service. If the Resolver Identity is a wildcard, the reputation system MUST replace it with a valid hostname that matches the wildcard.

Assessing reputation limits the ability of a DDR forgery attack to cause harm, as it will only allow an attacker to direct clients to a resolver they consider trustworthy. Major DoH client implementations already include lists of known or trusted resolvers [CHROME-DOH][MICROSOFT-DOH][MOZILLA-TRR].
Clients SHOULD start by checking the resolver endpoint with the numerically lowest SVCB SvcPriority. Clients MAY wait until a DNS query triggers an Encrypted DNS connection attempt before performing this verification.

If RVS encounters an error or rejects the server, the client MUST NOT send encrypted DNS queries to that server. If RVS rejects all compatible ServiceMode records, the client MUST fall back to the unencrypted resolver (i.e. plaintext DNS on port 53).

3.1. Reputation systems

Embedding a list of known trusted resolvers in a client is only one possible model for assessing the reputation of a resolver. In future a range of online reputation services might be available to be queried, each returning an answer according to their own specific criteria. These might involve answers on other properties such as jurisdiction, or certification by a particular body. It is out of scope for this document to define these query methods, other than to note that designers should be aware of bootstrapping problems. It is the client’s decision as to how to combine these answers, possibly using additional metadata (e.g. location), to make a determination of reputation.

3.2. Using resolvers of intermediate reputation

If the determined reputation is a binary "definitely trustworthy" or "definitely malicious", the client’s recommended action is clear. However, intermediate trust levels are also possible (e.g. "probably safe", "newly launched"). In these cases there are some options clients can consider:

* The client can simply decline to use the encrypted service. In this case, unless there is another option, the client will fall back to Do53.

* The client can ask the user about a specific domain names that appear in the certificate. These names might be recognizable to the user, e.g. as that of an ISP. It’s also possible to present more details about why a Resolver Identity lacks some element of reputation.

* The client can use the encrypted service for a limited time, as a means of mitigating interception attacks. For example, if the client limits the DDR response TTL to 5 minutes, this ensures that any attacker can continue to redirect queries for at most 5 minutes after they have left the local network.
4. Management of local blocking functionality

Certain local DNS forwarders block access to domains associated with malware and other threats. Others block based on the category of service provided by those domains, e.g. domains hosting services that are not appropriate for a work or school environment. In the short term to ensure this service is not lost due to a cross-forwarder upgrade, the maintainers can simply add "resolver.arpa" to their actively curated list of domains to block. This pattern has been deployed by Mozilla, with the domain "use-application-dns.net" [MOZILLA-CANARY].

In the long term, it is best for filtering DNS forwarders to implement support for encrypted DNS. The following subsections describe some ways to implement this.

4.1. Local implementation with DNR

The local forwarder can be upgraded to one that implements an encrypted DNS service discoverable through DNR. This requires a TLS certificate on the local device, proving ownership of the chosen Authentication Domain Name (ADN). Onward queries to the internet SHOULD also be protected with encryption.

4.2. Local implementation with DDR

If the local forwarder can be upgraded to offer an encrypted DNS service, this can then be made discoverable through classic DDR. If the device has a private IP (as presumed for RVS), a self-signed certificate is sufficient as long as the client supports the Opportunistic Discovery mode of DDR. Onward queries to the internet SHOULD also be protected with encryption.

4.3. Move upstream

The blocking functionality can be moved to the upstream resolver. Cross-forwarder upgrade then enables the service to continue, as long as the upstream resolver has sufficient reputation.

5. Compatibility issues that can arise from cross-forwarder upgrade

Legacy DNS forwarders sometimes provide various additional services that would be lost in the event of a cross-forwarder upgrade. For all of these, a possible general mitigation is to provide users or administrators with the ability to control whether DDR is used with legacy forwarders. For example, this control could be provided via a preference, or via a notification upon discovering a new upstream resolver. Specific mitigations are also described below.
5.1. Split-horizon namespaces

Some local network resolvers contain additional names that are not resolvable in the global DNS. A simple cross-forwarder upgrade might lose access to these local names. Clients SHOULD be aware of well-known suffixes (e.g., .local, .home.arpa.) that require local resolution. Dynamic discovery of local prefixes would help this issue. To address any remaining ones, the following mitigation can be used.

5.1.1. Mitigation: NXDOMAIN Fallback

In "NXDOMAIN Fallback", the client repeats a query to the unencrypted resolver if the encrypted resolver returns NXDOMAIN. This allows the resolution of local names, provided they do not collide with globally resolvable names (as required by [RFC2826]).

This is similar to the fallback behavior currently deployed in Mozilla Firefox [FIREFOX-FALLBACK].

NXDOMAIN Fallback results in slight changes to the security and privacy properties of encrypted DNS. Queries for nonexistent names no longer have protection against a local passive adversary, and local names are revealed to the upstream resolver.

NXDOMAIN Fallback is only applicable when a legacy DNS forwarder might be present, i.e. the unencrypted resolver has a private IP address, and the encrypted resolver has a different IP address. In other DDR configurations, any local names are expected to resolve similarly on both resolvers.

5.2. Interposable domains

An "interposable domain" is a domain whose owner deliberately allows resolvers to forge certain responses. This arrangement is most common for search engines, which often support a configuration where resolvers forge a CNAME record to direct all clients to a child-appropriate instance of the search engine [DUCK-CNAME][BING-CNAME][GOOGLE-CNAME].

Future deployments of interposable domains can instruct administrators to enable or disable DDR when adding the forged record, but forged records in legacy DNS forwarders could be lost due to a cross-forwarder upgrade.
5.2.1. Mitigation: Exemption list

There are a small number of pre-existing interposable domains, largely of interest only to web browsers. Clients can maintain a list of relevant interposable domains and resolve them only via the network’s resolver.

5.3. Caching

Many legacy DNS forwarders also provide a shared cache for all network users. Cross-forwarder upgrades will bypass this cache, resulting in slower DNS resolution for some queries.

5.3.1. Mitigation: Stub caches

Clients can compensate partially for any loss of shared caching by implementing local DNS caches. This mitigation is already widely deployed in browsers and operating systems.

6. Privacy Considerations

6.1. Privacy gains

The conservative validation policy results in no encryption when a legacy DNS forwarder is present. This leaves the user’s query activity vulnerable to passive monitoring [RFC7258], either on the local network or between the user and the upstream resolver.

Reputation Verified Selection enables the use of encrypted transport in these configurations, reducing exposure to a passive surveillance adversary.

6.2. Privacy losses

In some legacy DNS forwarder implementations, the upstream resolver is not able to determine whether two queries were issued by the same client inside the network. It can only see aggregated queries being made by the forwarder. [DDR] to a non-local resolver requires individual encrypted DNS connections from each device, revealing which queries were made by the same client. RVS shares this property.
6.2.1. Mitigation: Open multiple connections

If the above issue is a concern, clients MAY open multiple connections to the designated encrypted resolver with separate local state (e.g. TLS session tickets), and distribute queries among them. This may reduce the upstream resolver’s ability to link queries that came from a single client.

7. Security Considerations

When the client uses the conservative validation policy described in [DDR], the client can establish a secure DDR connection only in the absence of an active attacker. An on-path attacker can impersonate the resolver and intercept all queries, by preventing the DDR upgrade.

This basic security analysis also applies if the client uses Reputation Verified Selection. However, the detailed security properties differ, as discussed in this section.

7.1. Redirection

An on-path attacker might be located on the local network, or between the local network and the upstream resolver. In either case, the attacker can redirect the client to a resolver of the attacker’s choice, as long as that resolver meets the client’s requirements for reputation. Hence the reputation system is essential to the security of the user.

Weaknesses in the reputation system could reopen this class of vulnerabilities.

7.1.1. Possible weakness: Stale reputation

If a previously-reputable resolver is compromised, users can be redirected to it while this reputation remains high. Once an attack has been detected, it should be reported to relevant reputation services so that they can revise their assessment of this resolver.

7.1.2. Possible weakness: Inappropriate reputation

The reputation of a resolver might depend on aspects of the client’s connection context, e.g. their geographic location. For example, a local ISP’s resolver could be reputable for clients in its service area, but suspicious for clients on distant continent. Accordingly, very large reputation systems may need to customize their results based on the context.
7.2. Forensic logging

7.2.1. Network-layer logging

With the conservative validation policy, a random sample of IP packets is likely sufficient for manual retrospective detection of a DNS redirection attack.

With Reputation Verified Selection, local forensic logs must capture a specific packet (the attacker’s DDR designation response) to enable retrospective detection of a redirection attack.

7.2.1.1. Additional Mitigation: Log all DDR responses

Redirection attacks are largely mitigated by RVS, but the loss of network-layer logging for such attacks can be mitigated by logging all DDR responses, or more generally all DNS responses. This makes retrospective attack detection straightforward, as the attacker’s DDR response will indicate an unexpected server.

7.2.2. DNS-layer logging

DNS-layer forensic logging conducted by a legacy DNS forwarder would be lost in a cross-forwarder upgrade.

7.2.2.1. Solution: Plan to upgrade

Forwarders that want to observe all queries from RVS clients should plan to implement DDR or DNR. In the short term it is possible for the forwarder to disable DDR by responding negatively to _dns.resolver.arpa, but this is not recommended long-term as it prevents confidentiality protection.

8. References

8.1. Normative References


8.2. Informative References


Acknowledgments

Thanks to Anthony Lieuallen and Eric Orth for early reviews of a previous draft.

Authors’ Addresses

Benjamin Schwartz
Google LLC
Email: bemasc@google.com

Chris Box
BT
Email: chris.box@bt.com