add
Internet-Draft
Intended status: Standards Track
Expires: 25 October 2025

J. Todd
Quad9
T. Jensen
Microsoft
C. Mosher
Innate, Inc.

23 April 2025

Encrypted DNS Server Redirection
draft-ietf-add-encrypted-dns-server-redirection-02

Abstract

This document defines Encrypted DNS Server Redirection (EDSR), a mechanism for encrypted DNS servers to redirect clients to other encrypted DNS servers. This enables dynamic routing to geo-located or otherwise more desirable encrypted DNS servers without modifying DNS client endpoint configurations or the use of anycast by the DNS server.

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://ietf-wg-add.github.io/EncDNSServerRedirect/draft-add-encrypted-dns-server-redirection.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-add-encrypted-dns-server-redirection/.

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

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

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/.

Copyright Notice

Table of Contents

1. 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.

2. Introduction

Encrypted DNS Server Redirection (EDSR) is a protocol that allows an encrypted DNS resolver whose configuration is well known to clients to redirect them to other, more desirable resolvers without having to support anycast and without having to configure clients with these other resolvers ahead of time. It uses a similar mechanism to the one defined by Section 4 of [RFC9462] to redirect an encrypted DNS client from one encrypted DNS resolver to another encrypted DNS resolver. Where DDR uses a threat model that presumes the initial DNS traffic could be unencrypted, EDSR only ever applies when the initial DNS traffic is already encrypted.

One example of what makes redirection to another resolver desirable is geolocation. A DNS service may document one or a few well known resolver configurations even though it routes traffic to hundreds or thousands of resolvers that are closer to the client, reducing latency and making DNS resolutions more applicable to the client.

3. DNS client behavior

3.1. Discovering redirections

When a DNS client first opens a connection to an encrypted DNS server, it MUST use the Discovery Using Resolver Names mechanism defined in Section 5 of [RFC9462] to send a SVCB query for the name of the resolver to discover its encrypted DNS configuration. The DNS client SHOULD open a connection to the server returned in the SVCB query using the same domain name as the original server and one of the IP addresses returned in additional A/AAAA records for the same name. Once a connection has been successfully opened, as subsequently described by reaching a suitable server at the end of the redirection chain, the client SHOULD close the first connection.

3.1.1. Use of Delegated Credentials

If the DNS client’s TLS dependency supports Delegated Credentials [RFC9345], it SHOULD present the "delegated_credential" TLS extension in its ClientHello as described in Section 4.1.1 of [RFC9345] to maximize compatibility with EDSR-supporting servers. This is because some server operators MAY redirect to servers controlled by other entities which do not have access to its private key but which nevertheless have the ability to terminate TLS connections for the server’s name.

3.1.2. Redirection after Discovery Using Resolver IP Addresses

EDSR assumes that the original server is identified by domain name from the client’s perspective. Examples include when the client was configured with the resolver through endpoint management or DNR discovery [RFC9463]. However, when the server was discovered using DDR’s Discovery Using Resolver IP Addresses Section 4 of [RFC9462], this is not the case. Due to the threat model that mode of DDR operates under, where it has to start from an unencrypted resolver, the identity of the server used for verification is its IP address. The risks involved with using the domain name of resolvers discovered by Discovery Using Resolver IP Addresses are further explored in the Security Considerations section Section 6.

When clients use EDSR with a resolver discovered using DDR’s Discovery Using Resolver IP Addresses Section 4 of [RFC9462], the only difference is that the destination server it was redirected to MUST be able to claim the IP address of the previous server in its SAN field.

This section applies to both the Verified and Opportunistic forms of DDR’s Discovery Using Resolver IP Addresses mechanism.

3.2. Identifying self-redirections

If the set of IP addresses that are valid for the server being redirected to include the IP address of the current server, the client SHOULD ignore the redirection, treating it the same as receiving no response or a NODATA response from the SVCB query. However, clients receiving preferable encryption parameters as part of the SVCB response MAY choose to reconnect to negotiate to upgrade to the preferred encryption method. When doing so, the client SHOULD NOT immediately repeat EDSR as the redirection from the server to itself has terminated the redirection chain.

3.3. Waiting for redirections

The client does not need to wait for the results of the redirection discovery query before sending other DNS queries on the connection, though they SHOULD gracefully close the connection as soon as it has successfully established a connection to the server it was redirected to and received or timed out the outstanding queries on the original connection.

See the Deployment Considerations section for reasons a client MAY choose to decline a redirection.

3.4. Refreshing redirections

If a chain of redirections was followed, the effective TTL of the redirection is the minimum of the TTLs encountered along the chain. If the effective TTL of the redirection is considered to be too short for the client’s performance (because it would require frequent repetition of EDSR), clients MAY refuse to follow the redirection. Servers SHOULD NOT redirect clients in a way that results in short effective TTLs.

When the redirection TTL expires or connectivity to the server the client was redirected to fails, the client MUST close the connection and return to using the servers it is currently configured to use by its local configuration before using EDSR again. This allows the client to honor the intention of whatever configuration method was used to provide it with a set of DNS servers to use.

If the client DNS server remains the same, it SHOULD repeat the EDSR mechanism before the effective TTLS expires so that if the same redirection is valid, it can avoid needing to tear down the current connection by refreshing its effective TTL.

3.5. Multiple redirections

When clients receive more than one valid SVCB response, they SHOULD prefer using the redirections that match their configuration (such as supported IP address family or desired encrypted DNS protocol) in ascending order of the SVCB priority. Once a successful connection is made to a redirected destination, clients MUST discard results for other servers. Entries returned for the same IP address MAY be retained for multi-protocol path diversity to what is presumed to be the same server. Later unsuitability of all connections to the server MUST result in restarting EDSR.

Redirections are considered to be a one-to-one relationship (starting with one recursive resolver and following its redirections should result in one replacement recursive resolver). It is not expected that a stub resolver ends up using more recursive resolvers than it was originally configured with when using EDSR.

3.6. Network changes

When a client device changes what network it is connected to, it SHOULD forget pre-existing redirections and start EDSR over with the originally configured resolvers. This ensures that a resolver which redirects clients based on their source network can behave accordingly.

Note that this is unrelated to what resolvers a client is originally configured with. For example, a client which is configured to always used the resolvers advertised by DHCP will likely start with different original resolvers when changing networks. How a client is configured with DNS resolvers is out of scope for this document. EDSR only provides a mechanism for clients to discover redirections from resolvers they were previously configured to use.

4. DNS server behavior

DNS resolvers who want to redirect clients to other resolvers MUST respond to SVCB [RFC9461] queries for their own domain names with records that describe the configuration of the destination server.

Guidance in Section 5 of [RFC9460] to improve performance by including additional A/AAAA records with the SVCB response SHOULD be followed.

If the server knows it supports Discovery Using Resolver IP Addresses, or does not know for sure, it MUST be prepared for clients to connect without an SNI because clients might have discovered the server that way. Otherwise, if the server knows it does not support Discovery Using Resolver IP Addresses, it MAY refuse connections without an SNI instead.

The destination server MAY use Delegated Credentials [RFC9345] if the DNS client advertises its support for Delegated Credentials as described in Section 4.1.1 of [RFC9345]. This is valid so long as the delegated credential is valid for the same domain name used by the referring server.

Delegated Credentials are one approach for servers which need to redirect clients to servers owned by other entities, as is the case with CDN contracts. Another approach is using [RFC9115] to delegate Short-Term Automatically Renewed (STAR) certificates to the servers that need to serve a name on behalf of a name's owner. This approach would not require protocol changes for EDSR peers communicating with one another, unlike Delegated Credentials. Other trade-offs between these approaches are beyond the scope of this document.

4.1. Ensuring compatibility

Redirections MUST only redirect to resolvers which support at least the same protocol, address family, port, and TLS minimum versions as the referring resolver. This ensures that redirections do not lead clients to resolvers that are not compatible with the client. In addition, servers SHOULD avoid redirecting to servers which will also redirect clients unless they are actively coordinating to ensure a positive client experience. See the Deployment Considerations section for more details.

4.2. Dealing with persistent clients

Servers SHOULD be prepared for clients to not follow the redirection immediately as connection failures, other technical issues, or even client policy affecting server choice may lead to clients being unable to follow the redirection promptly or at all. Servers who are redirecting due to being overloaded MAY respond as they normally would to overwhelming traffic.

4.3. Redirection to servers controlled by third parties

Server operators ought to consider using delegated credentials [RFC9345] when they wish to redirect general clients to other servers operated by other entities. This allows the server operator to avoid giving access to their domain's private key to third parties but also ensure general clients have a secured, same-origin redirection experience.

5. Deployment Considerations

5.1. Large trees of redirections

It is possible for DNS servers to redirect clients to DNS servers which also redirect clients. Clients which are presented with long chains of redirections MAY choose to stop following redirections to reduce connection thrashing. DNS server operators SHOULD deploy redirection behavior mindfully to avoid long chains of redirection.

Servers SHOULD ensure their redirections do not create loops, where clients are redirected to a server it already encountered earlier in the process. Clients MAY stop following redirections when they detect this, but MAY also take a simpler approach, following only a maximum number of redirections.

5.2. Redirection TTLs

Servers SHOULD provide sufficiently long TTLs for clients to avoid the need to constantly repeat EDSR queries. Server operators should be mindful of redirection chains because unless they collaboratively control the TTLs of one another's redirections, redirection chains will end up with greatly reduced effective TTLs because the client will always use the lowest. When they do collaboratively control the TTLs of one another's redirections, there is probably a way to do a single-hop redirection instead.

5.3. Including IP addresses in EDSR responses

If a recursive resolver does not include additional A/AAAA records per Section 5 of [RFC9460], stub resolvers might end up failing the redirection if the redirection destination name cannot be resolved. Additionally, the recursive resolver SHOULD ensure records conntaining the same IP version as the existing connection are returned (if the stub is currently connected over IPv4, one or more A records SHOULD be included, and if the stub is currently connected over IPv6, one or more AAAA records SHOULD be included).

5.4. Determining suitability of destinations for a given client

Because servers are required to ensure redirections are to servers that at least support the same protocols as the current connection, server operators will often need to know at run-time which of the potential redirection destinations are appropriate for the client beyond whatever business logic requires the redirection in the first place. While out of scope for protocol design, it is worth calling out that implementors need to consider how they will handle this. A straightforward example would be a cache of the potential redirection destinations that map to their capabilities, with consideration for how that table is populated and updated (example: TLS 1.3 support is rolled out to server which previously only served TLS 1.2). Any such out-of-band lookup would be much better than attempting just-in-time checking with the potential destinations of their capabilities, which would negatively impact the client experience when done during its redirection.

Note that even if there is only one redirection candidate to choose from, the server still needs to know when to not offer the redirection due to compatibility issues.

5.5. Comparison to Discovery Using Resolver Names

Discovery Using Resolver Names as defined in Section 5 of [RFC9345] describes discovery of the encrypted DNS configuration for a server using its domain name. The technical mechanism described there and EDSR are the same in the context of on-wire behavior; they differ by how clients and servers deploy them.

For Discovery Using Resolver Names, servers are free to return whatever subset of valid configurations for the domain name provided by the client it wishes, such as those it sees as valid for the client’s apparent geolocation. In the case of EDSR, servers are expected to only return configurations if it wants the client to be redirected to another resolver. Servers implementing EDSR SHOULD only answer SVCB queries for its own domain name in the EDSR context following its requirements.

For Discovery Using Resolver Names, clients are querying for encrypted DNS configurations available for a given server domain name. EDSR does not restrict clients from issuing these queries whenever they want. However, clients ought to consider that querying an encrypted DNS server for its own configuration that supports EDSR (which is not inherently discoverable by the client) might only return configuration it is ok with the client using to immediately reconnect.

6. Security Considerations

6.1. Trusting the redirected connection

EDSR does not provide novel authentication or security mechanisms. Redirection is trusted by virtue of the server authentication via PKI through TLS [RFC5280]. The DNS stub resolver implementing EDSR SHOULD use whatever policies it uses for other TLS connections for encrypted DNS traffic to determine if a given TLS cert chain is trustworthy before proceeding with EDSR.

EDSR MUST NOT be used with encrypted DNS protocols that are not based on TLS. This scenario will require future standards work.

EDSR should not introduce any additional security considerations beyond use of the original encrypted resolver prior to redirection. Because the original connection was trusted, information sent over it about a new connection to use should be as trusted. However, clients that wish to time bound vulnerabilities to attackers who compromise the original resolver MAY choose to implement a maximum TTL to honor on SVCB records that redirect to other servers.

6.2. Use with unencrypted DNS

EDSR MUST NOT be used to redirect unencrypted DNS traffic to any other resolver. This use case is called "designation" and is covered by Discovery of Designated Resolvers (DDR) as defined in [RFC9462], specifically Section 4: "Discovery Using Resolver IP Addresses". Not following that security guidance opens up a DNS client to malicious redirection to an attacker-controlled DNS server. For more information, see Section 7 of [RFC9462].

EDSR also MUST NOT be used to redirect encrypted DNS traffic to a resolver that advertises support for unencrypted DNS. This would reduce the security posture of the client. Clients MUST NOT follow an encrypted DNS redirection and then send unencrypted DNS traffic to the new resolver.

6.3. Use with DDR discovery from IP addresses

When a resolver is discovered using DDR’s Discovery Using Resolver IP Addresses mechanism defined in Section 4 of [RFC9462], the server’s identity used for TLS purposes is its IP address, not its domain name. This means servers and clients MUST use the original server’s IP address, not the IP address of the previous server in the event of redirection chains, in the SAN field of destination servers to validate the redirection.

The reason for this is due to an attack where the DDR SVCB query response is modified by an active attacker to have a different domain name in its "dohpath" SVCB key. When the client uses it to issue the EDSR query to the (valid) DDR-designated resolver, it will innocently forward the query upstream and return the result. The result may even be DNSSEC signed since it was issued by the valid owner of the attacker's domain name. If this redirection is then followed and validated with the attacker's domain name, it will succeed and the client will have been maliciously redirected to use an attacker's server at the low cost of a port 53 attack without breaking encryption or compromising the encrypted DNS server DDR designated.

There is no harm in using the name of the server for the EDSR query so long as the validation of the destination server is performed using the original IP address and not the name. This ensures EDSR clients can consistently use the domain name of a server for redirection discovery. Use of the DDR-defined SUDN "resolver.arpa" was considered and rejected because this would conflate DDR configuration and EDSR configuration by placing them in the same zone, using the same DNS record type.

7. Privacy Considerations

A client MAY choose to not send other name queries until redirection is complete, but there should be no privacy issue with sending queries to intermediate resolvers before redirection takes place. This is because the intermediate resolvers are considered to be appropriate destinations by the client even if the resolver wants to substitute another resolver for reasons other than name resolution results such as latency optimization or load balancing.

8. Data Flow Considerations

8.1. Data Scope

EDSR does not result in any additional data being shared by the end user, as the DNS queries going to the new resolver were already going to go to the original resolver.

8.2. Data Visibility

EDSR results in a 1:1 replacement of DNS resolvers used (future queries sent to the new resolver will not be sent to the original resolver anymore). This means the number of servers which see any given query remain the same.

This is only true if clients only use one redirected DNS server per original DNS server. If the DNS server offers more than one redirection, and the client validates and uses two or more of those redirections, then there will be greater data visibility (more destinations). This is however entirely within the client's choice following their own policy as a redundancy versus volume of exhausted data trade-off.

EDSR requires the redirection to another server to also use encrypted DNS, so no third-party will be introduced to the data flow unless the encryption is broken.

8.3. Data centralization

EDSR can only increase data centralization if multiple resolver operators choose to redirect DNS clients to the same, other DNS resolver. To prevent the reduction of their resolution redundancy, DNS clients MAY choose to ignore redirections if they find that they point to resolvers they are already configured to use, by a previous redirection or some other configuration.

9. IANA Considerations

This draft has no IANA considerations.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC9345] Barnes, R., Iyengar, S., Sullivan, N., and E. Rescorla, "Delegated Credentials for TLS and DTLS", RFC 9345, DOI 10.17487/RFC9345, July 2023, <https://www.rfc-editor.org/rfc/rfc9345>.

[RFC9460] Schwartz, B., Bishop, M., and E. Nygren, "Service Binding and Parameter Specification via the DNS (SVCB and HTTPS Resource Records)", RFC 9460, DOI 10.17487/RFC9460, November 2023, <https://www.rfc-editor.org/rfc/rfc9460>.

[RFC9461] Schwartz, B., "Service Binding Mapping for DNS Servers", RFC 9461, DOI 10.17487/RFC9461, November 2023, <https://www.rfc-editor.org/rfc/rfc9461>.

[RFC9462] Pauly, T., Kinnear, E., Wood, C. A., McManus, P., and T. Jensen, "Discovery of Designated Resolvers", RFC 9462, DOI 10.17487/RFC9462, November 2023, <https://www.rfc-editor.org/rfc/rfc9462>.

10.2. Informative References

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/rfc/rfc5280>.

[RFC9115] Sheffer, Y., L坦pez, D., Pastor Perales, A., and T. Fossati, "An Automatic Certificate Management Environment (ACME) Profile for Generating Delegated Certificates", RFC 9115, DOI 10.17487/RFC9115, September 2021, <https://www.rfc-editor.org/rfc/rfc9115>.

[RFC9463] Boucadair, M., Ed., Reddy.K, T., Ed., Wing, D., Cook, N., and T. Jensen, "DHCP and Router Advertisement Options for the Discovery of Network-designated Resolvers (DNR)", RFC 9463, DOI 10.17487/RFC9463, November 2023, <https://www.rfc-editor.org/rfc/rfc9463>.

Appendix A. Acknowledgments

The authors would like to thank the following individuals for their invaluable feedback to this document: Ben Schwartz, Eric Orth, Erik Nygren, Manu Bretelle, Med Boucadair, Ralph Weber, Ted Hardie, Tommy Pauly, Viktor Dukhovni, and Vittorio Bertola.

Appendix B. Appendix

This document describes only one mode of redirection. Previous versions of this draft defined an additional mode of redirection that allowed servers to redirect to servers that presented a different domain name than the original server. While the scenario’s validity has some interest, there is no consensus in the WG for how it can be addressed in an acceptably secure fashion.

Authors’ Addresses

J. Todd
Quad9
Email: jtodd@quad9.net

T. Jensen
Microsoft
Email: tojens.ietf@gmail.com

C. Mosher
Innate, Inc.
Email: cmosher@gmail.com