Workgroup: ADD
Internet-Draft:
draft-ietf-add-split-horizon-authority-07
Published: 6 December 2023
Intended Status: Standards Track
Expires: 8 June 2024
Authors: T. Reddy (Nokia) D. Wing (Citrix) K. Smith (Vodafone) B. Schwartz (Meta)

Establishing Local DNS Authority in Validated Split-Horizon Environments

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 are not configured to use this resolver can use it, but only to resolve these domains. This specification defines a mechanism for domain owners to inform clients about local resolvers that are authorized to answer authoritatively for certain subdomains.

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 8 June 2024.

Copyright Notice

Table of Contents

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/recursive resolvers", and "forwarders" (or "stub resolvers") respectively. 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 [RFC8106]), some networks provide a hybrid resolver instead. If this resolver acts as an authoritative server for some names and provides different answers for those domains depending on the source of the query, 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 (e.g., RDNSS Selection [RFC6731], "Access Network Domain Name" [RFC5986], and "Client FQDN" [RFC4702] [RFC4704] in DHCP, "dnsZones" in Provisioning Domains [RFC8801], and INTERNAL_DNS_DOMAIN [RFC8598] in IKEv2). 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 process requires cooperation between the local DNS zone and the public zone.

This specification relies on securely identified local DNS servers, and checks each local domain hint against a globally valid parent zone.

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., DNS over TLS (DoT), HTTPS (DoH), QUIC (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.

All of the special-use domain names registered with IANA [IANA-SUDN], most notably ".home.arpa", "resolver.arpa.", "ipv4only.arpa." and ".local", are never unique to a specific DNS server's authority. All special-use domain names are outside the scope of this document and MUST NOT be validated using the mechanism described in this document.

Use of this specification is limited to DNS servers that support authenticated encryption and split-horizon DNS names that are rooted in the global DNS.

4. Requirements

This solution seeks to fulfill the following requirements:

* No loss of security: No unauthorized party can impersonate a zone unless they could already do so without use of this specification.

* Least privilege: Local resolvers do not hold any secrets that could weaken the security of the public zone if compromised.

* Local zone confidentiality: The specification does not leak local network subdomains to anyone outside of the network.

* Flexibility: The specification can represent and authorize a typical Split DNS zone structure.

* DNSSEC Compatibility: The specification supports DNSSEC-based object security for local zone contents.

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], DDR [I-D.ietf-add-ddr], or an equivalent mechanism (see Section 9).

To establish local authority, the network MUST convey one or more "Authorization Claims" to the client. An "Authorization Claim" is an abstract structure comprising:

* An Authentication Domain Name (ADN) of a local encrypted resolver.

* The DNS name of the authorizing parent zone.

A set of subdomains of this parent zone that are claimed by the named local resolver (potentially including the entire parent zone). To claim the entire parent zone, the claimed subdomain will be represented as an asterisk symbol "".

*A ZONEMD Hash Algorithm (Section 5.3 of [RFC8976]). For interoperability purposes implementations MUST support the "mandatory to implement" hash algorithms defined in Section 2.2.3 of [RFC8976].

*A high-entropy salt, up to 255 octets.

If the local encrypted resolver is identified by name (e.g., DNR), that identifying name MUST be the one used in any corresponding Authorization Claim. Otherwise (e.g., DDR using IP addresses), the resolver MUST present a validatable certificate containing a subjectAltName that matches the Authorization Claim.

To establish its authority, the network MUST provide each Authorization Claim to the parent zone operator. If the contents are approved, the parent zone operator computes a "Verification Token" according to the following procedure:

1. Convert all subdomains into canonical form and sort them in canonical order (Section 6 of [RFC4034]).

2. Replace the suffix corresponding to the parent zone with a zero byte.

3. Let $X be the concatenation of the resulting pseudo-FQDNs.

4. Let len($SALT) be the number of octets of salt, as a single octet.

5. Let $TOKEN = hash(len($SALT) || $SALT || $X). Where "||" denotes concatenation.

The zone operator then publishes a "Verification Record" with the following structure, following the advice in Sections 5.1 and 5.2 of [I-D.ietf-dnsop-domain-verification-techniques]:

*Type = TXT.

*Owner Name = Concatenation of the ADN, "_splitdns-challenge", and the parent zone name.

*Contents = "token=base64url($TOKEN)" (without padding)

By publishing this record, the parent zone authorizes the local encrypted resolver to serve these subdomains authoritatively.

5.1. Example

Consider the following authorization claim:

- *ADN = "resolver17.parent.example"

- *Parent = "parent.example"

- *Subdomains = "payroll.parent.example",
  "secret.project.parent.example"

- *Hash Algorithm = SHA-384

- *Salt = "example salt bytes (should be random)"

To approve this claim, the zone operator would publish the following record:

NOTE: '\' line wrapping per [RFC8792]

```
resolver17.parent.example._splitdns-challenge.parent.example. \
IN TXT "token=z1qyK7QWwQPkT-ZmVW-tAQbsNyYenTNBPp5ogYB8S1wesVCR\
-KJDv2eFwfJcWQM"
```

5.2. Conveying Authorization Claims

The Authorization Claim is an abstract structure that must be encoded in some concrete syntax in order to convey it from the network to the client. This section defines some encodings of the Authorization Claims.

5.2.1. Using DHCP

In DHCP, each Authorization Claim is encoded as a DHCP Authentication Option ([RFC3118] and Section 21.11 of [RFC8415]), using the Protocol value $TBD1, "Split DNS Authentication". In DHCPv4, the long-options mechanism described in Section 8 of [RFC3396] MUST be used if the authentication option exceeds the maximum DHCPv4 option size of 255 octets. The Algorithm field provides the ZONEMD Hash Algorithm, represented by its registered Value. The Replay Detection Method (RDM) value **MUST** be 0x00. The Authentication Information **MUST** contain the following information, concatenated:

1. The ADN in canonical form.

2. The parent name in canonical form.

3. A one-octet "salt length" field.

4. The salt value.

5. The $X value defined in Section 5.

5.2.2. Using Provisioning Domains

When using Provisioning Domains [RFC8801], the Authorization Claims are represented by the PvD Additional Information key "splitDnsClaims", whose value is a JSON Array. Each entry in the array **MUST** be a JSON object with the following structure:

* "resolver": The ADN as a dot-separated name.

* "parent": The parent zone name as a dot-separated name.

* "subdomains": An array containing the claimed subdomains, as dot-separated names with the parent suffix already removed, in canonical order. To claim the entire parent zone, the claimed subdomain will be represented as an asterisk symbol "*".

* "algorithm": The hash algorithm is represented by its "Mnemonic" string from the ZONEMD Hash Algorithms registry ([RFC8976], Section 5.2).

* "salt": The salt, encoded in base64url.

6. Validating Authority over Local Domain Hints

To validate an Authorization Claim provided by the network, participating clients **MUST** resolve the Verification Record for that name. If the resolution produces an RRSet containing the expected token for this Claim, the client **SHALL** regard the named resolver as authoritative for the claimed subdomains. Clients **MUST** ignore any unrecognized keys in the Verification Record.

Each validation of authority applies only to a specific Authentication Domain Name. If a network offers multiple encrypted resolvers, each claimed subdomain 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" DNS 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 encrypted transport. That resolution strategy is considered "tamperproof" because any actor who could modify the 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 TXT records for this purpose. Alternatively, a client might perform DNSSEC validation for the verification query even if it has disabled DNSSEC validation for other DNS queries.

6.2. Using DNSSEC

The client resolves the Verification 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. Delegating DNSSEC across Split DNS Boundaries

We wish to enable DNSSEC validation of local DNS names without requiring the local resolver to hold DNSSEC private keys that are valid for the parent zone. To support this configuration, parent zones **MAY** add a "ds=..." key to the Verification Record whose value is the RDATA of a single DS record, base64url-encoded. This DS record authorizes a DNSKEY whose Owner Name is "resolver.arpa."

To validate DNSSEC, the client first fetches and validates the Verification Record. If it is valid and contains a "ds" key, the client **MAY** send a DNSKEY query for "resolver.arpa." to the local encrypted resolver. At least one resulting DNSKEY RR **MUST** match the DS RDATA from the "ds" key in the Verification Record. All local resolution results for subdomains in this claim **MUST** offer RRSIGs that chain to one of these approved DNSKEYs.

The "ds" key **MAY** appear multiple times in a single Verification Record, in order to authorize multiple DNSKEYs for this local encrypted resolver. If the "ds" key is not present in a valid

Verification Record, the client **MUST** disable DNSSEC validation when resolving the claimed subdomains via this local encrypted resolver.

```
;; Parent zone
$ORIGIN parent.example.

; Parent zone's public KSK and ZSK
@ IN DNSKEY 257 3 5 ABCD...=
@ IN DNSKEY 256 3 5 DCBA...=

; Verification Record containing DS RDATA for the local
; resolver's KSK.  This is an ordinary public TXT record,
; secured by RRSIGs from the public ZSK.
resolver.example._splitdns-challenge IN TXT "token=abc...,ds=QWE..."

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; Local zone, claiming "subdomain.parent.example".

; The local resolver's KSK, validated by the Verification Record.
resolver.arpa. IN DNSKEY 257 3 5 ASDF...=

; Each claimed subdomain has its own ZSK, which is signed by the
; KSK and is used to sign records at that subdomain and below.
subdomain.parent.example.        IN DNSKEY 256 3 5 FDSA...=
subdomain.parent.example.        IN AAAA 2001:db8::17
deeper.subdomain.parent.example. IN AAAA 2001:db8::18
```

Figure 1: Example use of "ds=..."

8. 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).

8.1. Split-Horizon Entire Zone

Consider an organization that operates "example.com", and runs a different version of its global domain on its internal network.

First, the host and network both need to support one of the discovery mechanisms described in Section 5. Figure 2 shows discovery using DNR and PvD.

Validation is then perfomed using either an external resolver (Section 8.1.1) or DNSSEC (Section 8.1.2).

Steps 1-2: The client determines the network's DNS server (dns.example.net) 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 dns.example.net 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:

```
{
  "identifier": "pvd.example.com",
  "expires": "2025-05-23T06:00:00Z",
  "prefixes": ["2001:db8:1::/48", "2001:db8:4::/48"],
  "splitDnsClaims": [{
    "resolver": "dns.example.net",
    "parent": "example.com",
    "subdomains": ["*"],
    "algorithm": "SHA384",
    "salt": "abc...123"
  }]
}
```

The JSON keys "identifier", "expires", and "prefixes" are defined in [RFC8801].

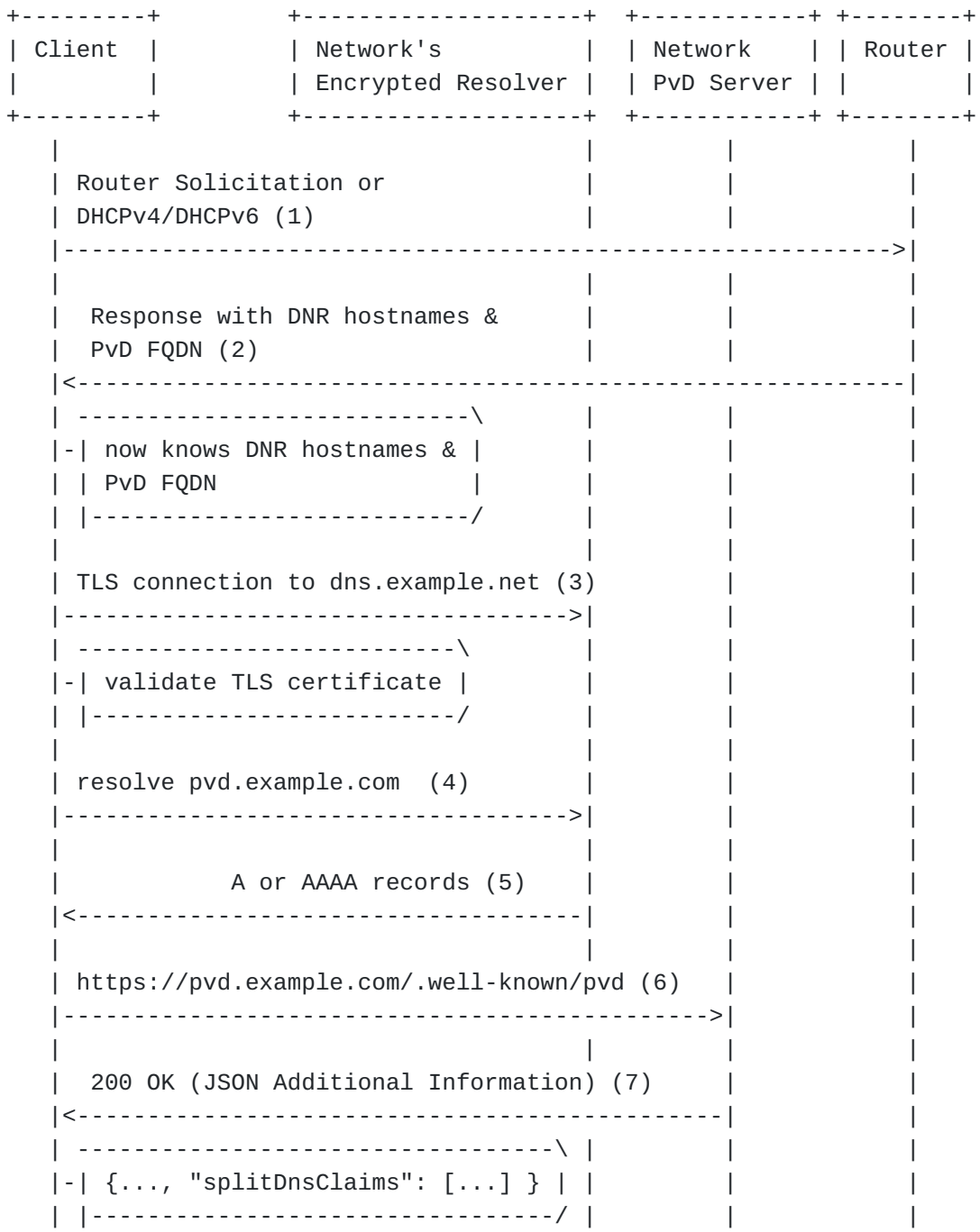

Figure 2: Learning Local Claims of DNS Authority

8.1.1. 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 to issue TXT queries for the Verification Records. The TXT lookup returns a token that matches the claim.

Step 3: The client has validated that example.com has authorized dns.example.net to serve example.com. 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 claimed zones.

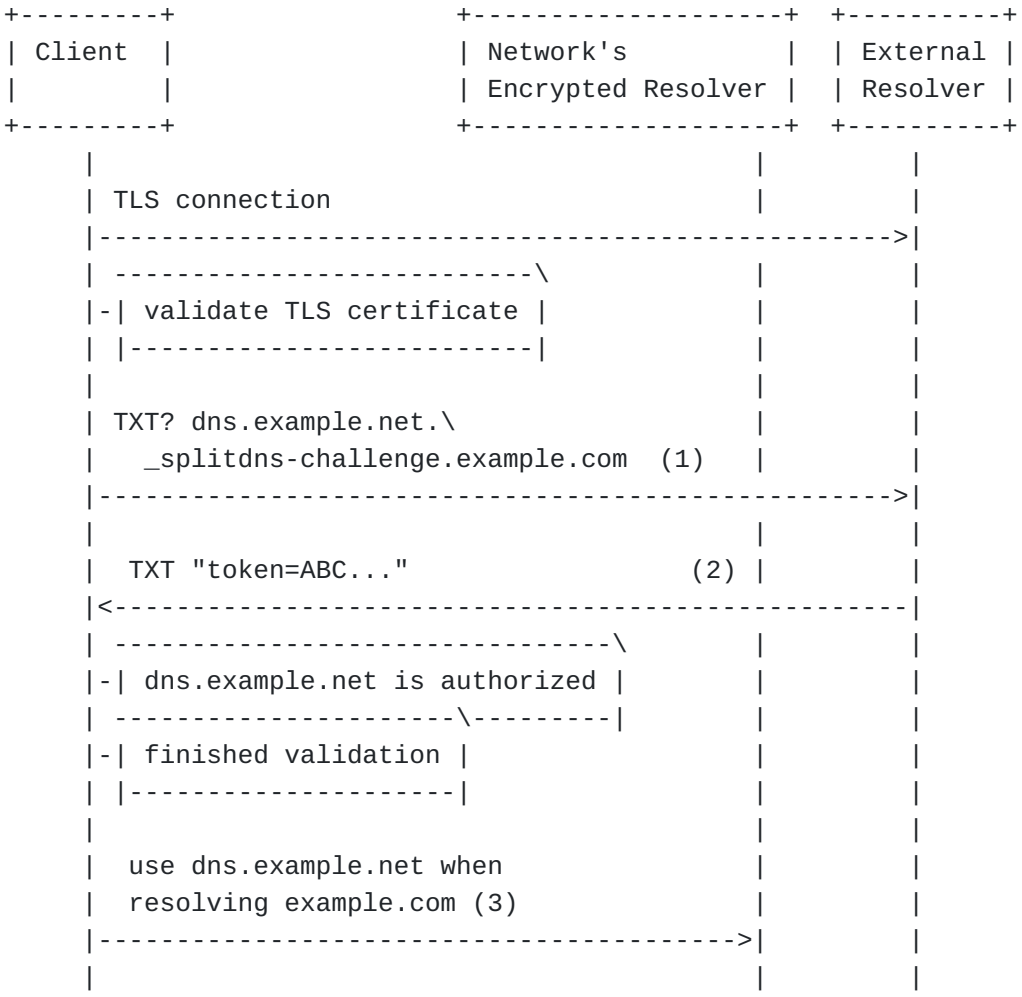

Figure 3: Verifying claims using an external resolver

8.1.2. 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 TXT queries for the Verification Records. The TXT lookup will return a signed response containing the expected token. The client then performs full DNSSEC validation locally.

Step 3: The DNSSEC validation is successful and the token matches, so this Authorization Claim is validated. 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 claimed zones.

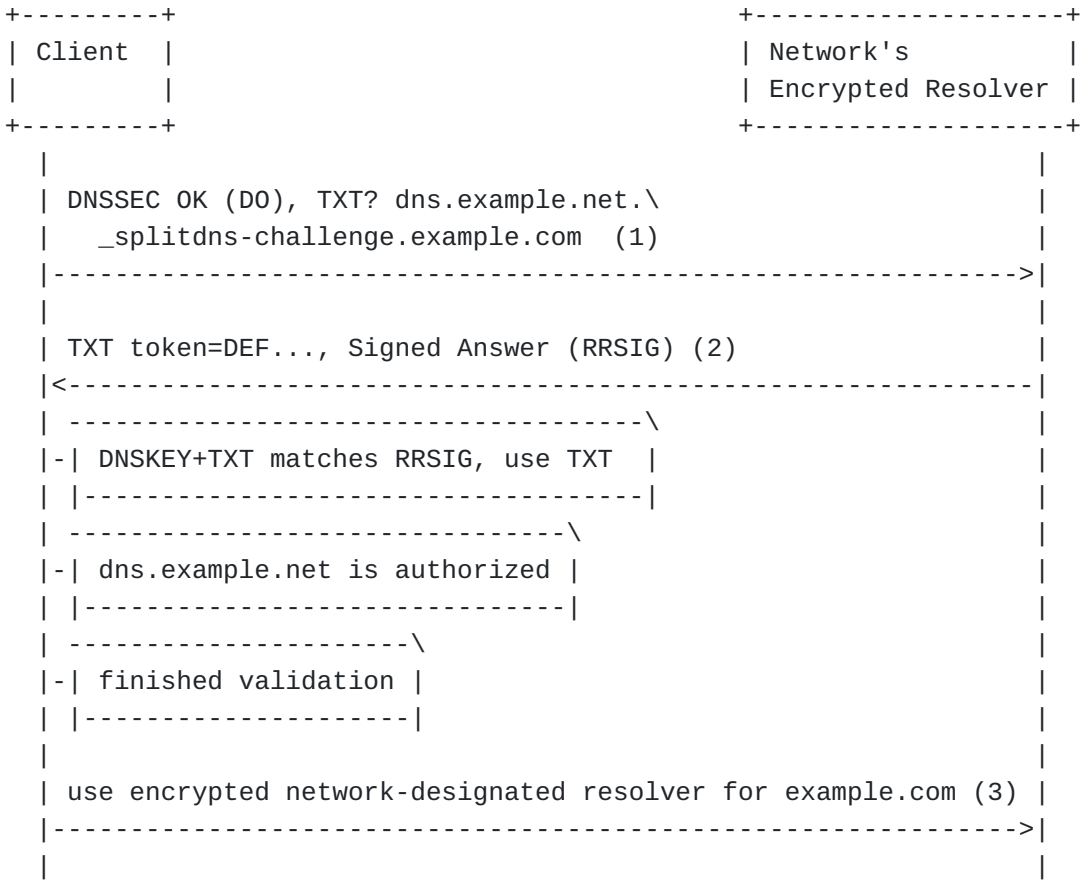

Figure 4: Verifying claims using DNSSEC

8.2. 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 8.1, except that queries for hosts not within the subdomain (e.g., www.example.com) are sent to the external resolver rather than the resolver for internal.example.com.

As in Section 8.1, the internal DNS server will need a certificate signed by a CA trusted by the client.

Although placing internal domains inside a child domain is unnecessary to prevent leakage, such placement reduces the frequency

of changes to the Verification Record, this document recommends the internal domains 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 "private1.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”.

9. 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]. The VPN client can use the mechanism defined in Section 6 to validate that the discovered encrypted DNS resolver is authorized to answer for the claimed subdomains.

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

10. Authorization Claim Update

A verification record is only valid until it expires. Expiry occurs when the Time To Live (TTL) or DNSSEC signature validity period ends. When the verification record expires, clients MUST fetch the verification records again and repeat the verification procedure.

A new verification record must be added to the RRset before the corresponding Authorization Claim is updated. After the claim is updated, the following procedures can be used:

1. DHCP reconfiguration can be initiated by the DHCP server to prompt DHCP clients for dynamically requesting the updated Authorization Claim. This process avoids the need for the client to wait for its current lease to complete and request a new one, enabling the lease renewal to be driven by the DHCP server.

2. The sequence number in the RA (Router Advertisement) PvD (Prefix Delegation) option will be incremented, requiring clients to fetch PvD additional information from the HTTPS server due to the updated sequence number in the new RA ([RFC8801], Section 4.1).

3. The old verification record needs to be maintained until the DHCP lease time or PvD Additional Information expiry.

11. Security Considerations

The Authentication Domain Names of authorized local encrypted resolvers are revealed in the Owner Names of Verification Records. This makes it easier for domain owners to understand which resolvers they are currently authorizing to implement Split DNS, but it could create a confidentiality problem if the local encrypted resolver's name is inside a secret subdomain. To avoid leakage, local resolvers should be given a name that does not reveal any sensitive information (perhaps in addition to the more sensitive name).

The security properties of hashing algorithms are not fixed. Algorithm Agility (see [RFC7696]) is achieved by providing implementations with flexibility to choose hashing algorithms from the ZONEMD Schemes registry ([RFC8976], Section 5.2).

12. IANA Considerations

12.1. DHCP Split DNS Authentication Algorithm

IANA is requested to add the following entry to the "Protocol Name Space Values" registry on the "Dynamic Host Configuration Protocol (DHCP) Authentication Option Name Spaces" page:

*Value: $TBD1

*Description: Split DNS

*Reference: (This Document)

12.2. Provisioning Domains Split DNS Additional Information

IANA is requested to add the following entry to the "Additional Information PvD Keys" registry on the "Provisioning Domains (PvDs)" page:

*JSON key: "splitDnsClaims"

*Description: "Verifiable locally served domains"

*Type: Array of Objects

*Example:

```
[{
  "resolver": "dns.example.net",
  "parent": "example.com",
  "subdomains": ["sub"],
  "algorithm": "SHA384",
  "salt": "abc...123"
}]
```

*Reference: (This document)

12.3. DNS Underscore Name

IANA is requested to add the following entry to the "Underscored and Globally Scoped DNS Node Names" registry on the "Domain Name System (DNS) Parameters" page:

*RR Type: TXT

*_NODE NAME: _splitdns-challenge

*Reference: (This document)

13. Acknowledgements

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

Thanks to Tianran Zhou for the opsdir review, Watson Ladd for the secdir review and Bob Halley for the intdir review.

14. References

14.1. Normative References

[IANA-SUDN] IANA, "Special-Use Domain Names", <https://www.iana.org/assignments/special-use-domain-names/special-use-domain-names.xhtml>.

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

[RFC3118] Droms, R., Ed. and W. Arbaugh, Ed., "Authentication for DHCP Messages", RFC 3118, DOI 10.17487/RFC3118, June 2001, <https://www.rfc-editor.org/info/rfc3118>.

[RFC3396] Lemon, T. and S. Cheshire, "Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)", RFC 3396,

DOI 10.17487/RFC3396, November 2002, <https://www.rfc-editor.org/info/rfc3396>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <https://www.rfc-editor.org/info/rfc4034>.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <https://www.rfc-editor.org/info/rfc4035>.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <https://www.rfc-editor.org/info/rfc6698>.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <https://www.rfc-editor.org/info/rfc6762>.

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

[RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <https://www.rfc-editor.org/info/rfc8415>.

[RFC8801] Pfister, P., Vyncke, É., Pauly, T., Schinazi, D., and W. Shao, "Discovering Provisioning Domain Names and Data", RFC 8801, DOI 10.17487/RFC8801, July 2020, <https://www.rfc-editor.org/info/rfc8801>.

[RFC8976] Wessels, D., Barber, P., Weinberg, M., Kumari, W., and W. Hardaker, "Message Digest for DNS Zones", RFC 8976, DOI 10.17487/RFC8976, February 2021, <https://www.rfc-editor.org/info/rfc8976>.

14.2. Informative References

[I-D.ietf-add-ddr] Pauly, T., Kinnear, E., Wood, C. A., McManus, P., and T. Jensen, "Discovery of Designated Resolvers", Work in Progress, Internet-Draft, draft-ietf-add-ddr-10, 5

August 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-add-ddr-10>.

[I-D.ietf-add-dnr] Boucadair, M., Reddy.K, T., Wing, D., Cook, N., and T. Jensen, "DHCP and Router Advertisement Options for the Discovery of Network-designated Resolvers (DNR)", Work in Progress, Internet-Draft, draft-ietf-add-dnr-16, 27 April 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-add-dnr-16>.

[I-D.ietf-dnsop-domain-verification-techniques] Sahib, S. K., Huque, S., Wouters, P., and E. Nygren, "Domain Control Validation using DNS", Work in Progress, Internet-Draft, draft-ietf-dnsop-domain-verification-techniques-03, 17 October 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-domain-verification-techniques-03>.

[I-D.ietf-ipsecme-add-ike] Boucadair, M., Reddy.K, T., Wing, D., and V. Smyslov, "Internet Key Exchange Protocol Version 2 (IKEv2) Configuration for Encrypted DNS", Work in Progress, Internet-Draft, draft-ietf-ipsecme-add-ike-14, 10 May 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-ipsecme-add-ike-14>.

[RFC4702] Stapp, M., Volz, B., and Y. Rekhter, "The Dynamic Host Configuration Protocol (DHCP) Client Fully Qualified Domain Name (FQDN) Option", RFC 4702, DOI 10.17487/RFC4702, October 2006, <https://www.rfc-editor.org/info/rfc4702>.

[RFC4704] Volz, B., "The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Client Fully Qualified Domain Name (FQDN) Option", RFC 4704, DOI 10.17487/RFC4704, October 2006, <https://www.rfc-editor.org/info/rfc4704>.

[RFC5986] Thomson, M. and J. Winterbottom, "Discovering the Local Location Information Server (LIS)", RFC 5986, DOI 10.17487/RFC5986, September 2010, <https://www.rfc-editor.org/info/rfc5986>.

[RFC6731] Savolainen, T., Kato, J., and T. Lemon, "Improved Recursive DNS Server Selection for Multi-Interfaced Nodes", RFC 6731, DOI 10.17487/RFC6731, December 2012, <https://www.rfc-editor.org/info/rfc6731>.

[RFC7686] Appelbaum, J. and A. Muffett, "The ".onion" Special-Use Domain Name", RFC 7686, DOI 10.17487/RFC7686, October 2015, <https://www.rfc-editor.org/info/rfc7686>.

[RFC7696] Housley, R., "Guidelines for Cryptographic Algorithm Agility and Selecting Mandatory-to-Implement Algorithms", BCP 201, RFC 7696, DOI 10.17487/RFC7696, November 2015, <https://www.rfc-editor.org/info/rfc7696>.

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

[RFC8310] Dickinson, S., Gillmor, D., and T. Reddy, "Usage Profiles for DNS over TLS and DNS over DTLS", RFC 8310, DOI 10.17487/RFC8310, March 2018, <https://www.rfc-editor.org/info/rfc8310>.

[RFC8499] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <https://www.rfc-editor.org/info/rfc8499>.

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

[RFC8792] Watsen, K., Auerswald, E., Farrel, A., and Q. Wu, "Handling Long Lines in Content of Internet-Drafts and RFCs", RFC 8792, DOI 10.17487/RFC8792, June 2020, <https://www.rfc-editor.org/info/rfc8792>.

[RFC8806] Kumari, W. and P. Hoffman, "Running a Root Server Local to a Resolver", RFC 8806, DOI 10.17487/RFC8806, June 2020, <https://www.rfc-editor.org/info/rfc8806>.

[RFC9162] Laurie, B., Messeri, E., and R. Stradling, "Certificate Transparency Version 2.0", RFC 9162, DOI 10.17487/RFC9162, December 2021, <https://www.rfc-editor.org/info/rfc9162>.

Authors' Addresses

Tirumaleswar Reddy
Nokia
India

Email: kondtir@gmail.com

Dan Wing
Citrix Systems, Inc.
4988 Great America Pkwy

Santa Clara, CA 95054
United States of America

Email: danwing@gmail.com

Kevin Smith
Vodafone Group
One Kingdom Street
London
United Kingdom

Email: kevin.smith@vodafone.com

Benjamin Schwartz
Meta Platforms, Inc.

Email: ietf@bemasc.net