

Independent Submission
Internet-Draft
Intended status: Standards Track
Expires: May 1, 2021

R. Arends
M. Larson
ICANN
October 28, 2020

DNS Error Reporting
draft-arends-dns-error-reporting-00

Abstract

DNS Error Reporting is a lightweight error reporting mechanism that provides the operator of an authoritative server with reports on DNS resource records that fail to resolve or validate, that a Domain Owner or DNS Hosting organization can use to improve domain hosting. The reports are based on Extended DNS Errors [RFC8914].

When a domain name fails to resolve or validate due to a misconfiguration or an attack, the operator of the authoritative server may be unaware of this. To mitigate this lack of feedback, this document describes a method for a validating recursive resolver to automatically signal an error to an agent specified by the authoritative server. DNS Error Reporting uses the DNS to report errors.

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 May 1, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction
2. Requirements Notation
3. Terminology
4. Overview
 - 4.1. Managing Caching Optimizations
 - 4.2. Example
5. EDNS0 Option Specification
6. DNS Error Reporting Specification
 - 6.1. Reporting Resolver Specification
 - 6.1.1. Constructing the Reporting Query
 - 6.2. Authoritative Server Specification
 - 6.3. Reporting Agent Specification
 - 6.4. Choosing a Reporting Agent Domain
7. Limitations
8. IANA Considerations
9. Security Considerations
10. Acknowledgements
11. Informative References

Authors' Addresses

1. Introduction

When an authoritative server serves a stale DNSSEC signed zone, the cryptographic signatures over the resource record sets (RRsets) may have lapsed. A validating recursive resolver will fail to validate these resource records.

Similarly, when there is a mismatch between the DS records at a parent zone and the key signing key at the child zone, a validating recursive resolver will fail to authenticate records in the child zone.

These are two of several failure scenarios that may go unnoticed for some time by the operator of a zone.

There is no direct relationship between operators of validating recursive resolvers and authoritative servers. Outages are often noticed indirectly, by end users, and reported via social media, if reported at all.

When records fail to validate there is no facility to report this failure in an automated way. If there is any indication that an error or warning has happened, it is buried in log files of the validating resolver, if these errors are logged at all.

This document describes a facility that can be used by validating recursive resolvers to report errors in an automated way.

It allows an authoritative server to signal a reporting agent where the validating recursive resolver can report issues if it is configured to do so.

The burden of reporting a failure falls on the validating recursive resolver. It is important that the effort needed to report failure is low, with minimal impact to its main functions. To accomplish this goal, the DNS itself is utilized to report the error.

2. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this

document are to be interpreted as described in [RFC2119].

3. Terminology

Reporting Resolver: In the context of this document, the term reporting resolver is used as a shorthand for a validating recursive resolver that supports DNS Error Reporting.

Reporting Query: The DNS query used to report an error is called a reporting query. A reporting query is for DNS resource record type NULL. The details of the error report are encoded in the QNAME of the reporting query.

Reporting Agent: A facility responsible for receiving error reports on behalf of authoritative servers. This facility is indicated by a domain name.

Reporting Agent Domain: a domain name which the reporting resolver includes in the QNAME of the reporting query.

4. Overview

In a query-response exchange, a reporting resolver indicates support for DNS Error Reporting by including an EDNS option with OPTION-CODE [TBD] [RFC Editor: change TBD to the proper code when assigned by IANA.] and OPTION-LENGTH zero. The REPORTING AGENT DOMAIN field in the EDNS option is absent in a query.

An authoritative server indicates support for DNS Error Reporting by including an EDNS0 option with OPTION-CODE [TBD] [RFC Editor: change TBD to the proper code when assigned by IANA.] and the REPORTING AGENT DOMAIN in the option's payload. The authoritative server MUST NOT include this option if the reporting resolver has not signalled support for DNS Error Reporting. The authoritative server MUST NOT include this option in the response if the configured reporting agent domain is empty or the null label (the root).

When a reporting resolver sends a reporting query to report an error, it MUST NOT include the EDNS0 Error Reporting option in the reporting query. This avoids additional compounding error reporting when the reporting agent server is misconfigured.

To report an error, the reporting resolver encodes the error report in the QNAME of the reporting query. The reporting resolver builds this QNAME by concatenating the extended error code [RFC8914], the QTYPE and QNAME that resulted in failure, the label "_er", and the reporting agent domain. See the example in section 4.2. Note that a regular RCODE is not included, as the RCODE is not relevant to the extended error code.

The resulting concatenated domain name is sent as a standard DNS query for DNS resource record type NULL by the reporting resolver. This query MUST NOT have the EDNS0 option code [TBD] set to avoid compounding error notifications.

The query will ultimately arrive at an authoritative server of the reporting agent. A NODATA negative response is returned by the authoritative server of the reporting agent domain, which in turn can be cached by the reporting resolver.

This caching is essential. It ensures that the number of reports sent by a reporting resolver for the same problem is dampened, i.e.

once per TTL, however, certain optimizations such as [RFC8020] and [RFC8198] may reduce the error reporting.

4.1. Managing Caching Optimizations

The reporting resolver may utilize various caching optimizations that inhibit subsequent error reporting by the reporting resolver to the authoritative server for an agent domain.

If the authoritative server for the agent domain were to respond with NXDOMAIN (name error), [RFC8020] rules state that any name at or below that domain should be considered unreachable, and negative caching would prohibit subsequent queries for anything at or below that domain for a period of time, depending on the negative TTL [RFC2308].

Since the authoritative server for an agent domain may not know the contents of all the zones it acts as an agent for, it is crucial that the authoritative does not respond with NXDOMAIN, as that may inhibit subsequent queries. The use of a wildcard domain name [RFC4592] in the zone for the agent domain will ensure the RCODE is consistently NOERROR.

Considering the Resource Record type for this wildcard record, type NULL is prohibited in master zone files [RFC1035]. However, any type that is not special according to [RFC4592] section 4 will do, such as a TXT record with an email address for the reporting agent in the RDATA.

Wildcard expansion occurs, even if the QTYPE is not for the type owned by the wildcard domain name. The response is a "no error, but no data" response ([RFC4592], section 2.2.1.) that contains a NOERROR RCODE and empty answer section. Note that reporting resolvers are not expected to query for this TXT record, since reporting queries use type NULL. This record is solely present to ensure a NODATA response is returned in response to reporting queries.

When the zone for the reporting agent domain is signed, a resolver may utilize aggressive negative caching, discussed in [RFC8198]. This optimization makes use of NSEC and NSEC3 (without opt-out) records and allows the resolver to do the wildcard synthesis. When this happens, the resolver may not send subsequent queries as it will be able to synthesize a response from previously cached material.

A solution is to avoid DNSSEC for the reporting agent domain's zone. Signing the agent domain's zone will incur an additional burden on the reporting resolver, as it has to validate the response. However, this response has no utility to the reporting resolver.

If an operator does sign a reporting agent domain's zone for whatever reason, one option is to use NSEC3 with opt-out, as that configuration precludes wildcard synthesis on the resolver.

4.2. Example

The domain broken.test is hosted on a set of authoritative servers. One of these serves a stale version. This authoritative server has a reporting agent configured: a01.reporting-agent.example.

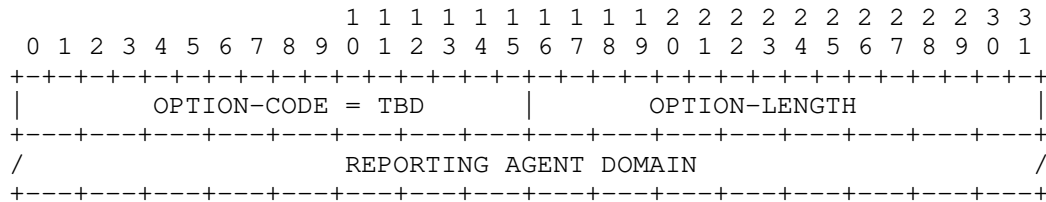
The reporting resolver is unable to validate the broken.test RRSet for type A, due to an RRSIG record with an expired signature.

The reporting resolver constructs the QNAME 7.1.broken.test._er.a01.reporting-agent.example and resolves it. This QNAME indicates extended DNS error 7 occurred while trying to validate broken.test type 1 (A) record.

After this query is received at one of the authoritative servers for the reporting agent domain (a01.reporting-agent.example), the reporting agent (the operators of the authoritative server for a01.reporting-agent.example) determines that the authoritative server for the broken.test zone suffers from an expired signature record (extended error 7) for type A for the domain name broken.test. The reporting agent can contact the operators of broken.test to fix the issue.

5. EDNS0 Option Specification

This method uses an EDNS0 [RFC6891] option to indicate support for sending DNS error reports and responding with the Reporting Agent Domain in DNS messages. The option is structured as follows:



Field definition details:

- o OPTION-CODE, 2-octets/16-bits (defined in [RFC6891]), for indicating error reporting support is TBD. [RFC Editor: change TBD to the proper code when assigned by IANA.]
- o OPTION-LENGTH, 2-octets/16-bits ((defined in [RFC6891]) contains the length of the REPORTING AGENT DOMAIN field in octets.
- o REPORTING AGENT DOMAIN, a Domain name [RFC8499].

6. DNS Error Reporting Specification

The various errors that a reporting resolver may encounter are listed in [RFC8914]. Note that not all listed errors may be supported by the reporting resolver. This document does not specify what is an error and what is not.

The DNS class is not specified in the error report.

6.1. Reporting Resolver Specification

Reporting Resolvers may have a configuration that allows the following:

- o DNS Error Reporting level: warning and / or errors
- o Do nothing: the reporting resolver does not indicate support for DNS Error Reporting.
- o Report to Reporting Agent: Indicate DNS Error Reporting in queries and use the reporting agent specified in the EDNS0 option received from the authoritative server.

- o Report to Configured Agent: Use the reporting agent specified in local configuration. This may override or supplement "Reporting Agent Domain". The use for such an option could be to allow a recursive resolver to report all errors to a reporting agent of its choosing, not just in zones with DNS Error Reporting enabled.

The reporting resolver MUST NOT use DNS error reporting to report a failure in resolving the reporting query.

The reporting resolver MUST NOT use DNS error reporting if the authoritative server has an empty Reporting Agent Domain field in the EDNS Error Reporting option.

6.1.1. Constructing the Reporting Query

The QNAME for the reporting query is constructed by concatenating the following elements, appending each successive element in the list to the right-hand side of the QNAME:

- o The Extended DNS error, presented as a decimal value, in a single DNS label.
- o The QTYPE that was used in the query that resulted in the extended DNS error, presented as a decimal value, in a single DNS label.
- o The QNAME that was used in the query that resulted in the extended DNS error. The QNAME may consist of multiple labels and is concatenated as-is.
- o A label containing the string "_er".
- o The reporting agent domain. The reporting agent domain consists of multiple labels and is concatenated exactly as received in the EDNS option sent by the authoritative server.

If the resulting reporting query QNAME would exceed 255 octets, it MUST NOT be sent.

The purpose of the "_er" label is twofold. First, it allows the reporting agent to quickly differentiate between the agent domain and the faulty query name. Second, if the specified agent domain is empty, or a NULL label (even if it is not allowed in this specification), the reporting query will have "_er" as a top-level domain as a result and not the original query.

6.2. Authoritative Server Specification

The Authoritative Server MUST NOT have multiple reporting agent domains configured for a single zone. To support multiple reporting agents, a single agent can act as a syndicate to subsequently inform additional agents.

An authoritative server for a zone with DNS error reporting enabled MUST NOT also be authoritative for that zone's reporting agent domain's zone.

6.3. Reporting Agent Specification

While there are many zone configurations possible for the reporting agent domain, such as DNAME, CNAME or special delegation structures to redistribute errors, please note that the burden of reporting is on the reporting resolvers and that creating complicated

configurations that cause additional work for the reporting resolver on behalf of misconfigured servers is NOT RECOMMENDED.

It is RECOMMENDED that the reporting agent zone uses a wildcard DNS record of type TXT with an arbitrary string in the RDATA and a TTL of at least one hour.

6.4. Choosing a Reporting Agent Domain

Each authoritative server SHOULD be configured with a unique reporting agent domain. When different authoritative servers share the same reporting agent domain, it is not possible to determine which authoritative server the reported error relates to.

It is RECOMMENDED that the reporting agent domain be kept relatively short to allow for a longer QNAME in the reporting query.

While it may be obvious to use the hostname of the authoritative server as the reporting agent domain, it is not a requirement, as long as the reporting agent is able to map the reporting agent domain to the proper authoritative server. Using the hostname of the authoritative server as the reporting agent domain is NOT RECOMMENDED when the hostname has multiple addresses, or when addresses are anycast.

7. Limitations

The length of the owner name for which errors can be reported is limited due to the requirement to append the reporting agent domain and prepend the Extended Error value and the QTYPE to the reporting query's QNAME.

8. IANA Considerations

IANA is requested to assign the following DNS EDNS0 option code registry:

Value	Name	Status	Reference
TBD	DNS ERROR REPORT	Standard	[this document]

[RFC Editor: change TBD to the proper code when assigned by IANA.]

IANA is requested to assign the following Underscored and Globally Scoped DNS Node Name registry:

RR Type	_NODE NAME	Reference
TXT	_er	[this document]

9. Security Considerations

Use of DNS Error Reporting may expose local configuration mistakes in the reporting resolver, such as stale DNSSEC trust anchors to the reporting agent.

DNS Error reporting SHOULD be done using DNS Query Name Minimization [RFC7816] to improve privacy.

DNS Error Reporting is done without any authentication between the reporting resolver and the authoritative server of the agent domain. Authentication significantly increases the burden on the reporting

resolver without any benefit to the reporting agent, authoritative server or reporting resolver.

The reporting resolver MUST NOT report about queries and responses from an encrypted channel (such as DNS over TLS [RFC7858] and DNS over HTTPS [RFC8484]).

The reporting resolver MUST NOT report about responses that did not match the qname/qtype/qclass and query-id in the original query [RFC5452], section 4.2.

The method described in this document will cause additional queries by the reporting resolver to authoritative servers in order to resolve the reporting query. This additional load is equivalent to the additional load when a resolver resolves the canonical name in a CNAME record.

This method can be abused by deploying broken zones with agent domains that are delegated to servers operated by the intended victim in combination with open resolvers [RFC8499]. This method MUST NOT be deployed by default on reporting resolvers and authoritative servers without requiring an explicit configuration element.

10. Acknowledgements

This document is based on an idea by Roy Arends and David Conrad.

11. Informative References

- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<https://www.rfc-editor.org/info/rfc1035>>.
- [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>>.
- [RFC2308] Andrews, M., "Negative Caching of DNS Queries (DNS NCACHE)", RFC 2308, DOI 10.17487/RFC2308, March 1998, <<https://www.rfc-editor.org/info/rfc2308>>.
- [RFC4592] Lewis, E., "The Role of Wildcards in the Domain Name System", RFC 4592, DOI 10.17487/RFC4592, July 2006, <<https://www.rfc-editor.org/info/rfc4592>>.
- [RFC5452] Hubert, A. and R. van Mook, "Measures for Making DNS More Resilient against Forged Answers", RFC 5452, DOI 10.17487/RFC5452, January 2009, <<https://www.rfc-editor.org/info/rfc5452>>.
- [RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <<https://www.rfc-editor.org/info/rfc6891>>.
- [RFC7816] Bortzmeyer, S., "DNS Query Name Minimisation to Improve Privacy", RFC 7816, DOI 10.17487/RFC7816, March 2016, <<https://www.rfc-editor.org/info/rfc7816>>.
- [RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport

- Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <<https://www.rfc-editor.org/info/rfc7858>>.
- [RFC8020] Bortzmeyer, S. and S. Huque, "NXDOMAIN: There Really Is Nothing Underneath", RFC 8020, DOI 10.17487/RFC8020, November 2016, <<https://www.rfc-editor.org/info/rfc8020>>.
- [RFC8198] Fujiwara, K., Kato, A., and W. Kumari, "Aggressive Use of DNSSEC-Validated Cache", RFC 8198, DOI 10.17487/RFC8198, July 2017, <<https://www.rfc-editor.org/info/rfc8198>>.
- [RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <<https://www.rfc-editor.org/info/rfc8484>>.
- [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>>.
- [RFC8914] Kumari, W., Hunt, E., Arends, R., Hardaker, W., and D. Lawrence, "Extended DNS Errors", RFC 8914, DOI 10.17487/RFC8914, October 2020, <<https://www.rfc-editor.org/info/rfc8914>>.

Authors' Addresses

Roy Arends
ICANN

Email: roy.arends@icann.org

Matt Larson
ICANN

Email: matt.larson@icann.org

dnsop
Internet-Draft
Intended status: Standards Track
Expires: 2 May 2021

E. Bretelle
Facebook
29 October 2020

Recursive Resolvers IP Ranges location distribution and discovery
draft-bretelle-dnsop-recursive-iprange-location-01

Abstract

This document specifies a way for recursive resolvers operators to signal the IP ranges and locations used by their server pools.

Discussion Venues

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

Source for this draft and an issue tracker can be found at <https://github.com/chantra/draft-dns-recursive-iprange-location>.

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 2 May 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components

extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Conventions and Definitions	3
3.	Publishing Resolver pool IP ranges	3
3.1.	TXT Resource Record	3
3.2.	HTTPS Resource Record	4
4.	Security Considerations	4
5.	IANA Consideration	4
5.1.	Underscored Node Name	4
5.2.	URI DNS Service Parameter	5
6.	Acknowledgments	5
7.	References	5
7.1.	Normative References	5
7.2.	Informative References	6
Appendix A.	Document history	6
A.1.	Changes between -00 and -01	6
	Author's Address	6

1. Introduction

Big distributed recursive resolver pools tend to be distributed across the world, operating under multiple countries and possibly using IP ranges for which the country is not necessarily perfectly matching the location of the service. This has lead to sub-optimal answers being returned to those server pools. An solution to this problem has been to use EDNS Client Subnet (ECS) [RFC7871], but this require support from both the recursive resolvers and the name servers authorities, comes with its own Security Considerations, and increased resources usage.

DNS server operators are commonly receiving spoofed DNS traffic over UDP, common techniques have been to reply with TC bit set to force legitimate clients to use TCP, if the load is still too high, they may start to drop traffic from selected subnets. While this may protect their resources, it has the possibility of denying the service to legitimate resolvers.

So far, operators have resorted to ad-hoc mechanism, ranging from exchanging list by email, providing IP ranges and location via webpages, or specific DNS queries, like Google Public DNS (https://developers.google.com/speed/public-dns/faq#locations_of_ip_addresses_ranges_google_public_dns_uses_to_send_queries), or Cloudflare (<https://www.cloudflare.com/ips/>), or OpenDNS

(<https://www.opendns.com/data-center-locations/>). When web pages are available, they are rarely found at consistent locations, neither are they formatted in a uniform way, essentially making name server operators' task rather complicated and brittle.

This document helps providing uniform solutions to let recursive server operators distribute the list of IP ranges under which their servers are operating as well as possibly location up to the postal code granularity by leveraging [RFC8805] format.

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. Publishing Resolver pool IP ranges

An entity willing to share the IP ranges used by their recursive servers would publish a record under the special name "_rdns.example.com". The IP ranges can be distributed either using a "TXT" record or "HTTPS" resource record [I-D.ietf-dnsop-svcb-https]. An entity can share IP information in 2 ways: via an IP Geolocation Feed, or list of IP ranges in a TXT record.

3.1. TXT Resource Record

An entity that wishes to share the IP ranges they are using with their recursive resolvers can distribute it via a TXT record.

The record is expressed as a single line of text found in the RDATA. Multiple TXT resource records for the same owner name may be permitted.

The record is made of a list of space separated IP ranges with optional comma separated Geolocation. The Geolocation field MUST be a 2-letter ISO country code conforming to ISO 3166-1 alpha 2 [ISO.3166.1alpha2]. Parsers SHOULD treat this field case-insensitively.

This example illustrate a record without geolocation:

```
_rdns.example.com. 3600 IN TXT "192.0.2.0/24 198.51.100.0/24 2001:db8::/56 "  
                                "2001:db8:00:ab00::/56"
```

This example illustrate a record with geolocation information:

```
_rdns.example.com. 3600 IN TXT "192.0.2.0/24,xa 198.51.100.0/24,xb "  
                                "2001:db8::/56,xc 2001:db8:00:ab00::/56,xa"
```

As the number of IP ranges increases, the size of the DNS response can become a source for amplification attacks. This is being discussed in Section 4.

3.2. HTTPS Resource Record

Another approach is share an IP geolocation feed [RFC8805] via an HTTPS Resource Record [I-D.ietf-dnsop-svcb-https]. This record has the benefit of providing a format which can provide more granularity if the entity sharing it wishes to, and can scale even when the number of IP ranges increases.

```
_rdns.example.com. 3600 IN HTTPS . (  
                                uri=https://foo.example.com/geofeed )
```

4. Security Considerations

Unless the record is DNSSEC-signed [RFC4033], the answers returned cannot be trusted. In HTTPS Resource Record is requested, the client can possibly trust the content if the URI is within the same zone cut, and HTTPS can authenticate the domain.

When using the TXT Resource Record, the answer returned can quickly become big and the name server operator should aggressively limit the size of the answer it will return to the client, and Truncate it if needed.

5. IANA Consideration

5.1. Underscored Node Name

This document updates the IANA registry "Underscored and Globally Scoped DNS Node Names" at <https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#underscored-globally-scoped-dns-node-names>

The following entries have been added to the registry:

RR Type	HTTPS
Node Name	_rdns
Reference	This document

RR Type	TXT
Node Name	_rdns
Reference	This document

5.2. URI DNS Service Parameter

This document adds a parameter to the "Service Binding (SVCB) Parameter" registry. The allocation request is TBD, taken from the to the First Come First Served range.

If present, this parameters indicates the URI template of an IP Geolocation feed. This is a string encoded as UTF-8 characters.

Name: uri

SvcParamKey	TBD
Meaning	URI to an IP Geolocation feed
Reference	This document

6. Acknowledgments

The authors would like to thank the following individuals for their useful input: John Todd.

7. References

7.1. Normative References

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

Schwartz, B., Bishop, M., and E. Nygren, "Service binding and parameter specification via the DNS (DNS SVCB and HTTPS RRs)", Work in Progress, Internet-Draft, draft-ietf-dnsop-svcb-https-01, 13 July 2020, <<http://www.ietf.org/internet-drafts/draft-ietf-dnsop-svcb-https-01.txt>>.

- [ISO.3166.1alpha2] ISO, "ISO 3166-1 decoding table", <http://www.iso.org/iso/home/standards/country_codes/iso-3166-1_decoding_table.htm>.
- [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>>.
- [RFC8805] Kline, E., Duleba, K., Szamonek, Z., Moser, S., and W. Kumari, "A Format for Self-Published IP Geolocation Feeds", RFC 8805, DOI 10.17487/RFC8805, August 2020, <<https://www.rfc-editor.org/info/rfc8805>>.

7.2. Informative References

- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <<https://www.rfc-editor.org/info/rfc4033>>.
- [RFC7871] Contavalli, C., van der Gaast, W., Lawrence, D., and W. Kumari, "Client Subnet in DNS Queries", RFC 7871, DOI 10.17487/RFC7871, May 2016, <<https://www.rfc-editor.org/info/rfc7871>>.
- [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>>.

Appendix A. Document history

A.1. Changes between -00 and -01

- * Editorial change: making the title more explicit than "Recursive Resolver"
- * Editorial change: Fix examples format to use break lines and fix ascii art table

Author's Address

Emmanuel Bretelle
Facebook

Email: chantra@fb.com

dnsop
Internet-Draft
Intended status: Informational
Expires: May 6, 2021

A. Fidler
BT plc
B. Hubert
OpenXchange
J. Livingood
Comcast
J. Reid
RTFM llp
N. Leymann
Deutsche Telekom AG
November 2, 2020

DNS over HTTPS (DoH) Considerations for Operator Networks
draft-fhllr-dnsop-dohoperator-00

Abstract

The introduction of DNS over HTTPS (DoH), defined in RFC8484, presents a number of challenges to network operators. These are described in this document. The objective is to document the problem space and make suggestions that could help inform network operators on how to take account of DoH deployment. This document also identifies topics that may require further analysis.

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 May 6, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. Contrasting DoH and Conventional DNS	3
4. Regulatory and Policy Considerations	4
4.1. Local Policy Constraints	5
4.2. Regulatory and Legal Impacts	5
4.3. Regulatory Constraints	6
5. Network Operations	6
5.1. Impact on DNS query logging	6
5.2. CDN endpoint selection	6
5.3. Redirection for captive portals	7
5.4. Managed network services	7
5.5. Resolver capacity management	7
5.6. Discovery considerations	8
5.7. Failure recovery	8
5.8. Impact on Network Address Translation	9
5.9. Load balancing and failover	9
6. User Support	9
7. Provisioning	11
8. Privacy Concerns	12
9. Security Considerations	13
10. Human Rights Considerations	15
11. Open Issues for Further Study	15
12. IANA Considerations	16
13. Acknowledgements	16
14. References	16
14.1. Normative References	16
14.2. Informative References	16
Authors' Addresses	17

1. Introduction

Traditional DNS traffic between stub resolvers, recursive servers and authoritative servers is not encrypted. This can pose a privacy challenge for Internet users, because their access to named network resources can potentially be tracked through their DNS queries. In

principle, any network element along the path between the user and resolving or authoritative servers could observe this unencrypted traffic. DoT (DNS over TLS) [RFC7858] is one proposal for providing privacy of DNS queries and DNS over HTTPS (DoH) [RFC8484] is another. Both DoH and DoT encrypt the communications between the end client (user) and recursive resolver. Plaintext DNS traffic between recursive and authoritative servers is generally less of a privacy concern because it usually does not contain information such as the source address of the initial query that could identify the end client.

2. Terminology

DoH Server: A server supporting the DNS over HTTPS is called a "DoH server" to differentiate it from a "DNS server" (one that only provides DNS service over one or more of the other transport protocols standardised for DNS). Similarly, a client that supports the DNS over HTTPS is called a "DoH client".

Do53: DNS over port 53 - conventional plaintext DNS. Do53 server and Do53 client are the respective terms for a server or client that uses conventional port 53 DNS.

Operator: A large Internet service provider, typically a cable company or fixed/mobile telco with a national or international network.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Contrasting DoH and Conventional DNS

With conventional DNS (Do53), using UDP or TCP port 53, most users are assigned the IP addresses of several recursive resolvers via DHCP or similar network bootstrapping mechanism. These are usually the IP addresses of recursive resolvers that are administered by the network operator. Although there is currently no equivalent to this for DoH, the ADD Working Group is developing solutions for DoH server discovery.

Users sometimes also change to third-party recursive resolvers. In some cases, they may even operate their own local resolver. It is not yet clear how or if DoH will be applied in these scenarios more generally. Current DoH behaviour of the most widely used web browsers is documented and reasonably well understood. The same cannot yet be said for operating system software: stub resolver libraries and web toolkits for instance.

RFC 8484 defines the protocol for DNS over HTTPS (DoH). When DoH is used, client and server DNS traffic is encrypted using a TLS RFC 8446 [RFC8446] channel, typically to port 443. DoH clients will have little need for conventional DNS apart from an initial bootstrap query to find the IP addresses of a suitable DoH server. In some cases, this will mean the bulk of the client's DNS resolver traffic bypasses an operator's DNS resolver infrastructure because that traffic uses the resolver service provided by a third-party DoH server.

When DoH is used, the traditional DNS client-server model of clients making queries and waiting for a reply from a server might well change. It can be expected that DoH servers will sometimes use DoH opportunistically. For instance a web server could include application/dns-message elements in the returned HTML data, anticipating the domain names that the web browser might need to resolve before rendering some web page. In this scenario, the browser would not need to lookup those names with DoH or conventional DNS because the relevant DNS data have already been supplied.

DoH is already widely implemented and deployed by browser vendors. All the major web browsers support DoH. Sometimes, DoH is enabled by default. In others, configuration changes are needed to get the browser to use DoH instead of conventional DNS.

Since DoH is not yet natively supported by the most widely-used DNS implementations, DoH servers may need some sort of proxy or "shim" module to convert between application/dns-message elements in HTML and conventional DNS queries and responses. A number of organisations are already offering public DoH resolution service, typically using anycast technology. Some operators have either deployed or are planning to deploy DoH resolver service in their networks.

DoH changes the current, well established business model where an end user (customer) pays for Internet connectivity and recursive DNS service is part of that offering from the ISP. When DoH is used, the customer may be dependent on DoH servers operated by third parties and have no contractual or business relationship with those providers. It also cannot be assumed that these DoH servers will be operating under the same policy and regulatory conditions that are applied by the end user's ISP.

4. Regulatory and Policy Considerations

4.1. Local Policy Constraints

Operator networks often have local policy constraints which require some form of DNS blocking or rewriting - for example to offer customers parental controls, to restrict access to illegal content or to minimize end user exposure to malware, phishing attacks and so on. These tend to be implemented by using data from threat intelligence providers, usually some sort of RPZ feed that is incorporated into the configuration of the operator's DNS resolver infrastructure.

It is not yet clear how or if this functionality can be made available by DoH servers. These protective measures will be less effective once DoH is used because end user DNS traffic will largely bypass the operator's DNS infrastructure, rendering such content and security protections useless. Some of these measures may be offered by some DoH servers, but as yet there is no defined mechanism to ensure that all local policy is implemented.

4.2. Regulatory and Legal Impacts

Operators can also be required to perform DNS blocking and filtering or rewriting for legal reasons: handling takedown notices or complying with court orders. This may also be necessary for operational and/or security reasons such as dealing with botnets and DDoS attacks. [CSRIC]

As before, it is not yet clear how or if DoH servers will provide this functionality. Some of these measures may be offered by some DoH servers, but there is no defined mechanism to ensure that all local policy is implemented, particularly those required in certain jurisdictions today. Current protective measures may be less effective once DoH is used because customer DNS traffic will be able to bypass the operator's DNS infrastructure.

Conventional recursive DNS services are generally located in the country where an operator is based. Since third-party DoH service providers are likely to be based and/or operated from outside those local countries, different protections and regulatory considerations may apply to the protection, storage and processing of user data processed on those servers. Typical regulations that could apply include General Data Protection Regulation (EU) 2016/679 [GDPR] and the EU-US Privacy Shield Framework [USPS]. These can sometimes have global scope - GDPR for instance. Overseas regulations may have lower, higher or even no commitments governing such services compared to those that would apply to a local operator. The potential impact of these regulatory obligations with respect to DoH services is unclear, including whether or not they apply or even could be applied at all.

4.3. Regulatory Constraints

Logs containing individual DNS queries and the IP addresses or other data correlating those queries to specific users or homes may in some legal jurisdictions be considered as Personal Data or PII, Personally Identifying Information. In such jurisdictions detailed DNS query logs may be subject to data protection and retention regulations, or other legal and/or compliance requirements.

Operators can also be subject to regulation or other legal instruments that require DNS query logs to be retained for a certain period of time and made available for law enforcement purposes as needed, such as under a court order or other legal process.

Since DoH potentially bypasses conventional DNS resolvers on which these privacy, regulatory, and legal requirements are imposed, it will reduce or eliminate the potential social value of these rules, and may even be viewed by some countries as a potential breach of regulatory compliance (whether by ISPs, DoH server operators, or others).

5. Network Operations

5.1. Impact on DNS query logging

Analysis of resolver query data is an important task in most operator networks. This can help with traffic management, load balancing and capacity planning as well as network and user security. Widespread uptake of DoH will mean an operator has reduced visibility of the DNS traffic in their network. Query traffic logged by traditional resolving servers will be less representative (or even completely unrepresentative) of the overall DNS activity in an operator's network.

5.2. CDN endpoint selection

End user queries made with DoH could mean that lookups return answers that are sub-optimal. i.e. directing clients to a distant CDN node that is outside the operator's network instead of to the localised CDN node(s) installed inside that network or directly interconnected with that network. Those DNS responses would be keyed on the source IP address of a resolving DoH server, possibly operated by a third party, rather than an address of one of the operator's resolving DNS servers or end client IP address information that those resolving servers might choose to provide through the Client Subnet EDNS0 option RFC 7871 [RFC7871].

The impact to an operator of directing clients to a distant CDN node that is outside the operator's network is not only slower access to resources provided by the CDN. It also incurs higher costs for the operator because traffic is routed over the operator's backbone and peering links rather than remaining within a part of the network that is geographically or topologically close to the end-user.

Additionally, operators have powerful technical, operational and business incentives to provide optimal user experience for their customers, particularly in terms of latency and speed of Internet services. This involves working with multiple CDN and content providers to ensure best performance when delivering those services, for example by providing Client Subnet EDNS0 option information. One risk is that DoH services could be provided by operators or distributors of web content who have different motivations. For instance a provider of DoH service may choose to offer fast access to the content that they host or distribute, but may decide not to offer the geographic information of the end-user (for privacy, policy or business reasons) to competing content providers/distributors.

5.3. Redirection for captive portals

Network operators also often use captive portal DNS to provide customer self-service activation and related customer account provisioning, billing and support activities. For example, captive portal DNS is used extensively to support functions such as self-service provisioning of customer owned and managed Customer Premises Equipment (CPE), service support, mobile pay as you go top up and access to national/regional WiFi hot spots. DoH traffic may bypass these operator-supplied functions that are essential for managing the network. This would significantly disrupt the manner in which networks are operated and managed.

5.4. Managed network services

The provision of managed network services, for instance to corporate or other enterprise clients will be affected by DoH. It could negatively affect bring-your-own-device policies which might introduce devices into these networks that are configured to use third party DoH servers. For instance there is a risk that internal domain names used extensively in such networks could leak to external DoH servers, presenting obvious privacy and security issues.

5.5. Resolver capacity management

Large operator networks are likely to operate their own DoH servers because of local policy or business considerations. This could mean an increase in TCP-based DNS traffic to port 443 as DoH displaces

conventional UDP-based queries to port 53. Transitioning from a primarily UDP-based service to TCP-based DoH would likely require substantial network capacity enhancements to an operator's DNS infrastructure. This might also require changes to existing load balancing and failover architectures. Establishing a DoH service in these environments would absolutely impact operational management and support.

It is unclear how much end-user DNS traffic will migrate to DoH and how quickly that happens since this will depend on the uptake of DoH-capable applications. There is also uncertainty about the default behaviour of these applications, for instance try DoH first then fall back to conventional DNS, use DoH only, try DoH and DNS in parallel and accept whichever answers first, etc. These unknowns have a further obvious impact on capacity planning and network operations.

5.6. Discovery considerations

Some networks offer DNS resolution services on locally scoped addresses that are not globally meaningful - for instance RFC1918 or link-local addresses. This arrangement is commonly found in operator and enterprise networks. Discovery of DoH servers (or other forms of encrypted DNS transport) in these environments is likely to rely on bootstrapping from a locally-addressed Do53 resolver to the chosen DoH server. That DoH server could either be offering resolution service at the same local address as the Do53 resolver, or at a different, possibly global, address. Both options need to be considered. In both cases the DoH server would offer a TLS certificate proving ownership of a name. This name should be meaningful to the end client, conveying the identity of the resolver operator. However given the lack of network authentication it does not currently seem possible to mandate a requirement that the name has to match anything that could be present in the client's configuration.

Many network operators use stub resolvers or proxies in CPE to handle end-user DNS requests. Depending on how the network is organised, these stub resolvers and proxies can either present public or private IP addresses to client devices. When these CPE devices use private IP addresses, it will complicate encrypted DNS discovery.

5.7. Failure recovery

It is not clear how DoH services will affect customers' approach to disaster recovery and fault reporting or influence their business continuity planning. For instance, if a client loses connectivity or access to their chosen DoH provider(s), they may lose Internet service even though they remain connected to the operator's network

and could otherwise use conventional DNS resolution services. It is assumed, but cannot be guaranteed, that DoH-capable applications will fall back to conventional DNS whenever DoH service fails. Applications might however be configured to only use DoH apart from an initial bootstrapping query that uses conventional DNS.

5.8. Impact on Network Address Translation

Techniques such as DNS64 [RFC6147] and NAT64 [RFC6146] are widely used for devices with IPv6-only transport, particularly in mobile networks to ensure continued access to parts of the Internet that are IPv4-only. These generally require the operator's DNS resolver server to carry out some form of IP address mapping. It is not known what impact DoH will have in these environments. It is unlikely that this will work with third party DoH providers because they will not have information about the operator's network that would allow them to map these IPv6 addresses.

In networks where the translator prefix is not the well-known prefix defined by RFC6146, the client's use of a DoH resolver outside the operator's network will prevent access to IPv4-only content, because the resolver will not know the correct prefix to use in its response. Even when the well-known prefix is used, the DoH resolver may not be configured to correctly use it in its response.

5.9. Load balancing and failover

Operator networks make extensive use of DNS-based solutions for load balancing and service failover. These might not work as expected with DoH clients which bypass the operator's DNS resolver infrastructure. Further operational problems may arise if stale DNS data are held in a DoH client's cache.

6. User Support

- o Adoption of DoH is likely to decouple DNS from the provision of Internet connectivity. For most users, DNS resolution is currently part of the service provided by their ISP. With DoH, users can be expected to rely on DoH service providers and are likely to have no business or contractual relationship with those providers.
- o Getting meaningful consent from users - how?
- o The role of user consent and whether it is a necessary factor in the processing of user data is contextual. It depends on the nature of relationships between the involved parties - largely the ISP, the DoH provider(s) and the end user - and how those

relationships were established. Prevailing legislation and regulation such as GDPR can also be an important consideration, albeit one that is obviously out of scope for an IETF document. It is not clear whether reconfiguration of a device or moving it from one network to another would constitute implied consent in a legal sense.

- o In any case, only a fraction of Internet users understand the mechanics of DNS resolution, which makes obtaining informed and meaningful consent difficult. Service providers should seek to explain data use in a way that's understandable to most people. Sustained and collective efforts by service providers to educate users (policymakers, legal scholars, teachers, etc.) about the Internet infrastructure to foster common understanding of these issues would be helpful.
- o How will users be able to opt in/out of DoH services?
- o Users may want to give meaningful consent to use DNS filters. Therefore, there should be an option for users to enable and disable DoH with neither behaviour assumed. Such permissions should also apply to DoH queries made by web-based apps using an API, not just the queries directly entered by the user. When users do provide consent for DoH-related data processing, the architecture must also support the ability for them to withdraw this consent at any time.
- o How do users select their "trusted" DoH Provider? i.e. How is a user or application supplied with a list of DoH providers? How does it choose between them and what are the selection criteria? Presumably these could/should be considerations for the ADD Working Group.
- o Clarification is needed on trusted certificate approach, e.g. is it enforced at application rather than the kernel/operating system layer?
- o Can/should discrete apps be able to choose their own DoH server? Suppose a banking app is configured to use the bank's DoH provider. Can that default be over-ridden? Should it?
- o How does a user get told about (and approve) a change of DoH service for a phone/tablet when they're roaming between mobile telcos or using whatever DoH service is offered in \$coffeeshop?
- o How is an operator expected to support the customer or troubleshoot problems caused by accidental or intentional change of DoH server? If the DoH provider deletes all their historic DoH

traffic, how do they support the ISP customer regarding troubleshooting?

- o How will DoH provisioning take account of existing customer parental control/malware protection settings and flag the consequences of selecting a new provider on these?
- o How will browsers/applications explain DoH/DNS options to customers so that they can make an informed decision, as many will not appreciate what DNS is. If they select a third party DoH provider, that may bypass their existing network operator's content and malware protection controls. The end user will presumably need to set these up again with their new DoH provider.
- o How to explain to customers that they may need to check/contact both their DoH provider(s) and network provider to resolver performance and outage issues.

7. Provisioning

- o If some list or registry of "trusted" DoH servers is needed, who/what is going to maintain this and manage it? What criteria and procedures are needed for adding or removing entries from that list? How does a DoH provider become trusted or become untrusted?
- o What are the requirements to become a DoH trusted recursive resolver? Will browsers or applications only show global or application-specific DoH provider options? How can regional network operators offering DoH just to their customer base be supported? How will browsers and applications know which regional or local options exist and which of these should and should not be honoured?
- o An industry approach for DoH discovery, trust and selection that operates in an open and transparent manner is needed. This should give the customer meaningful consent options.
- o How to configure CPE and other edge devices (e.g. smartphone) to use the operator's chosen DoH provider.
- o Can/should the operator's or application's choice of DoH server be overridden by the customer?
- o How do web applications get to specify the DoH server they want? If web apps get to choose the DoH server, they could be pointing to a malicious server (security issue) or allowing a DNS provider other than that defined by the user to see the DNS queries (privacy issue).

- o How will DoH provisioning and discovery take account of existing customer parental control/malware protection settings and flag the consequences of selecting a new provider on these?
- o If a browser or other edge device can do DoH, what determines if the DoH is the preferred the choice?, e.g. if CPE or set top box devices also supports DNS over TLS, should DoH be an option? If multiple options for DNS resolution are available, what decision process is used to make the customer recommendation and how is this trusted?

8. Privacy Concerns

Compared to traditional DNS, DoH offers more privacy protection against passive surveillance because requests and replies are carried over an encrypted channel. DoH offers an equivalent amount of privacy protection against passive surveillance as DoT does because both rely on TLS for their security properties.

Content Delivery Networks use techniques like EDNS-Client-Subnet (ECS) to return DNS answers that direct a client to an optimal location, for instance the CDN's node in the operator's network which serves the end user. DoH has the potential to be more privacy intrusive than ECS, largely because DoH is based on HTTP and can leverage the rich per-user and per-device tracking that pervades the web today. The implications of that are not yet well understood.

A DoH server will have a direct HTTPS connection to the client, assuming there are no middleboxes in the path between them. That could for example enable DoH servers operated by CDNs to carry out much more fine-grained redirection and content delivery, perhaps even on a per-user or per-user-session basis. They would be able to serve content and advertisements based on the end user's choice of operating system, their browser and that browser's configuration in addition to the client's source IP address, web cookie data, or other factors as is prevalent on the web today.

Global DoH providers will have access to significantly more DNS query data, and therefore be able to perform richer big data analytics, combining these insights with those obtained from other global platforms (search engines, operating systems, browsers, ad trackers, analytics services, web sites, mobile apps, payment systems, e-commerce platforms, social networks, Bluetooth beacons, etc.), potentially leading to a poor privacy outcome for consumers.

The DoH provider may adhere to different privacy policies than the operator's DNS service, particularly where they are located in

different jurisdictions. This may lead to better or worse privacy outcomes for users.

Operators in some jurisdictions are required to perform DNS filtering functions on traditional DNS queries and responses. If this functionality has to be provided using DoH, the only available option may be to fully proxy the HTTPS traffic. That represents more of a privacy intrusion than filtering alone.

It is feasible that individual applications might have the ability to select their own DoH server, bypassing the system- or operator-defined DoH settings. That could lead to privacy violations because DoH queries get sent to an arbitrary DoH server with unknown privacy policies.

If users have no relationship with the DoH provider handling their queries, they may have limited ability to exercise data protection rights (erasure, objection, complaints, etc) or to pursue remedy for breaches. This may be further complicated if the provider is unknown to the end user, can't be easily contacted or is located in another jurisdiction.

9. Security Considerations

DoH will give new opportunities for bad actors to propagate malware, spam and botnets. Once they use DoH, as some botnets have already started doing for command-and-control traffic, their DNS traffic will be encrypted and anonymised, making it hard to deploy countermeasures to protect against and mitigate these serious security threats. This is likely to have an adverse impact on cybersecurity both at a network/country level as well as for end users. Use of DoH could make it slower to identify DNS-based DDoS attacks, more difficult to attribute patient-zero for malware infections and harder to block access to botnet command-and-control nodes. A proof of concept exfiltration channel tool based on DoH [GODOH] already exists and it is reasonable to expect others which are much less benign will emerge in the future.

DoH queries and responses will be intermingled with other HTTPS port 443 traffic. This provides good traffic flow security for the client, because it's not readily clear when a DoH request or reply is taking place (unlike DoT). However network analytics may fail to detect when a malware implant on the client is making DoH requests, which would present a security risk.

Security of DoH relies on the TLS session for the HTTPS connection. Therefore it inherits the security guarantees that TLS provides. There may be interactions between DoH and TLS, for example issues

arising from DoH servers handling large numbers of TLS connections to DoH clients simultaneously, that have not yet been explored.

DNS query traffic is often made available to providers of threat intelligence and reputation services. These providers typically aggregate such data from many operators, process these datasets and then generate whitelists and blocklists which operators can then apply in their networks. DoH is likely to mean there will be a reduced volume of query data readily available for this sort of analysis. Overall DNS query traffic would be spread across a combination of operator-run DNS resolver servers and a number of DoH servers who might (or might not) make their query traffic available to providers of threat intelligence and reputation services.

This will have two unwelcome results. First, threat intelligence and reputation services will have fewer data to analyse and therefore have a significantly less complete perspective of end users' DNS behaviour. Second, the quality and effectiveness of the data provided by threat intelligence and reputation services will be materially diminished. This seems likely to reduce the security of networks and users as a result.

Although DoH uses TLS to provide authentication and data integrity of the channel between client and resolver, this does not guarantee that the resolver is returning correct DNS data to the client. DoH clients may need to perform DNSSEC validation to verify data received from DoH servers.

There is a risk that internal domain names used extensively in managed enterprise networks could leak to external DoH servers, presenting obvious privacy and security issues.

DoH can be implemented within the browser, rather than the kernel or an operating system library. It is not yet clear if that will make endpoint-based malware detection more or less effective.

Browser APIs will allow web applications to make DoH queries. If individual applications have the ability to select their own DoH server, it is not clear if that change would only apply to DoH lookups by that application or if they had broader scope. When these changes over-ride system- or operator-defined DoH settings, they will affect other processes running on the DoH client and effectively hijack their DNS traffic by rerouting it to the application's DoH provider.

The interactions between infrastructure using Network Address Translation (NAT) [RFC3022] and DoH is unclear. In situations where a third party DoH server can return security threat data back to the

operator of the originating network, its value is likely to be diminished due to the IP address sharing inherent in using NAT.

10. Human Rights Considerations

Parental control systems relying on DNS filtering can be bypassed using DoH. This may lead to increased ability of minors to access restricted or otherwise inappropriate content on the Internet, creating a conflict with the UN Convention on the Rights of the Child [Insert Ref to actual treaty text.]

Using DoH to bypass local DNS filtering and provide anonymity for end users is a mixed blessing. Using DoH to bypass country-based DNS filtering may provide end users a way of bypassing censorship mechanisms put in place by restrictive regimes. On the other hand, DoH could also help criminals to evade detection by obscuring the source of their attacks or botnet control nodes, while increasing the commercial tracking of user activity and trade in that data.

In jurisdictions where DNS blocking schemes have been incorporated into law, widespread deployment of DoH could encourage policy approaches that are more restrictive of users' freedom of expression, their ability to access information or limit the generation and availability of online content.

11. Open Issues for Further Study

- o DoH's reliance on TLS raises a number of concerns and unknowns. These include OSCP stapling, certificate life-times, scalability in managing session tickets, handling session resumption and the duration of TLS sessions. The trade-offs between certificate validation and session duration for possibly short-lived DoH transactions are not yet well understood. These factors will need careful analysis, particularly on DoH servers which get queries from large numbers of DoH clients.
- o The impact of DNS traffic migrating from UDP and port 53 to TCP and port 443 needs to be modelled because of the extra packets and round-trip times needed for TCP connection setup and the TLS handshake: performance, capacity planning, network engineering and so on.
- o DoH can leverage the rich per-user and per-device tracking that pervades the web today. Since the implications of that are not yet well understood, further work in this area is needed.
- o How DoH services will develop new functionality to overcome any inherent performance impact from moving the service out of the

operator network. For instance, optimisations to reduce latency in 3/4/5G mobile networks.

- o Clarification is needed around ECS blocking and options to avoid impacting existing network operator on-net caching strategy.
- o What DoH service metrics will be available for users to compare DoH providers?
- o DoH discovery in networks which use private IP addresses for CPE and stub resolvers or proxies could be challenging. Presumably this will be addressed in the add WG.

12. IANA Considerations

This memo includes no request to IANA.

13. Acknowledgements

Fill this in later

14. References

14.1. Normative References

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

14.2. Informative References

- [CSRIC] FCC, "Cybersecurity Risk Management and Best Practices", <<https://transition.fcc.gov/to-be-confirmed>>.
- [GDPR] European Union, "General Data Protection Regulation (EU) 2016/679", <<https://eur-lex.europa.eu/eli/reg/2016/679/oj>>.
- [GODOH] Sensepost, "DNS exfiltration using DoH", <<https://sensepost.com/blog/2018/waiting-for-godoh/>>.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, DOI 10.17487/RFC3022, January 2001, <<https://www.rfc-editor.org/info/rfc3022>>.

- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <<https://www.rfc-editor.org/info/rfc6146>>.
- [RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, DOI 10.17487/RFC6147, April 2011, <<https://www.rfc-editor.org/info/rfc6147>>.
- [RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <<https://www.rfc-editor.org/info/rfc7858>>.
- [RFC7871] Contavalli, C., van der Gaast, W., Lawrence, D., and W. Kumari, "Client Subnet in DNS Queries", RFC 7871, DOI 10.17487/RFC7871, May 2016, <<https://www.rfc-editor.org/info/rfc7871>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <<https://www.rfc-editor.org/info/rfc8484>>.
- [USPS] US Secretary of Commerce, "EU-U.S. Privacy Shield Framework", <<https://www.privacyshield.gov/EU-US-Framework>>.

Authors' Addresses

Andy Fidler
BT plc
BT Adastral Park
Martlesham Heath, Ipswich IP5 3RE
UK

Email: andrew.fidler@bt.com

Bert Hubert
OpenXchange
Rollnerstrasse 14
Nuremberg 90408
Germany

Email: bert.hubert@open-xchange.com

Jason Livingood
Comcast
1800 Arch Street
Philadelphia PA 19118
USA

Email: jason_livingood@comcast.com

Jim Reid
RTFM llp
St Andrews House
382 Hillington Road, Glasgow G51 4BL
Scotland

Email: jim@rfc1035.com

Nic Leymann
Deutsche Telekom AG
Friedrich-Ebert-Allee 140
Bonn 53113
Germany

Email: N.Leymann@telekom.de

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 7, 2021

K. Fujiwara
JPRS
November 03, 2020

Delegation Information (Referrals) Signer for DNSSEC
draft-fujiwara-dnsop-delegation-information-signer-00

Abstract

DNSSEC does not protect delegation information, it contains NS RRSets on the parent side and glue records. This document defines delegation information signer (DiS) resource record for protecting the delegation information, by inserting on the parent side of zone cut to hold a hash of delegation information. The DiS resource record reuses the type code and wire format of DS resource record, and distinguishes it from existing DS RRSets by using a new digest type. This document also describes the usage of DiS resource record and shows the implications on security-aware resolvers. The definition and usage are compatible with current DNSSEC.

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 May 7, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. Specification of the Delegation information Signer	3
3.1. New DS RR Usage: Delegation information signer (DiS)	3
3.2. DiS resource record in a Zone	4
3.3. Change of Authoritative servers	4
3.4. Change of validating resolvers	4
4. Compatibility with the current DNSSEC	4
5. Signing Priming Responses	5
6. IANA Considerations	5
7. Security Considerations	5
8. Acknowledgments	5
9. Normative References	5
Author's Address	6

1. Introduction

The current DNSSEC specifications [RFC4033], [RFC4034], [RFC4035] do not protect the parent side NS RRSet and glue contained in the delegation information.

Recently, the word "in-domain" is defined by [RFC8499]. The in-domain glue is necessary and sufficient glue information for name resolution. [I-D.ietf-dnsop-glue-is-not-optional] proposes that Glue records are expected to be returned as part of a referral and if they cannot be fitted into the UDP response, TC=1 MUST be set to inform the client that the response is incomplete and that TCP SHOULD be used to retrieve the full response.

Then, we can define complete delegation information set that contains the parent side NS RRSet and all in-domain glue. We can generate a hash of the parent side NS RRSet and in-domain glue, and put it in DNS as a parent side information.

The delegation information signer (DiS) resource record (RR) is inserted at a zone cut (i.e., a delegation point) to hold a hash of delegation information (parent side NS RRSet) and required glue. The DiS resource record reuses DS resource record and distinguishes it from DS RRSet by using a new digest type and a new algorithm number.

Recent DNSSEC validators ignore DS resource records whose algorithm and digest type are unknown. Therefore, DiS resource record does not affect current DNSSEC validation.

DNSSEC validators that support DiS resource record can verify NS RRSets and in-domain glue.

This document defines new DS RR usage, gives examples of how it is used and describes the implications on resolvers. This change is compatible with current DNSSEC.

The meaning and processing the delegation information (parent side NS RRSets and glue) are not changed. The delegation information is used for name resolution process, and not used as the result of the name resolution.

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 BCP14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Many of the specialized terms used in this document are defined in DNS Terminology [RFC8499].

3. Specification of the Delegation information Signer

This section defines a new usage of the Delegation Signer (DS) RR type.

3.1. New DS RR Usage: Delegation information signer (DiS)

This document specifies that the new DNSSEC Digest Type XX (it will be assigned by IANA) to the Delegation information Signer with SHA-256 (DISSHA256) for another DS usage.

The key tag and algorithm field may require further discussion.

The digest field is calculated over the parent side NS RRSets corresponding to the owner name of the DiS resource record and whole in-domain glue for its delegation.

```
digest = SHA-256 hash( NS RRSets | in-domain glue RRSets)
```

NS RRSets and in-domain glue RRSets are ordered as [I-D.ietf-dnsop-dns-zone-digest].

Sibling glue and out-of-bailiwick glue are not the data to be signed.

Wire format and Presentation format are the same as DS Resource Record.

3.2. DiS resource record in a Zone

The DiS resource record enables delegation information (parent side NS RRSets and in-domain glue records) signature validation in a validating resolver.

A DiS RRSets is present at all delegation point even if there is no DS RRSets. Since DiS RRSets has the same type code as DS RRSets except for digest type and hash data, details of DiS resource record is the same as DS resource record defined in [RFC4035].

When DNSSEC signer signs a zone, DNSSEC signer

- o Remove all DiS resource records
- o for all delegation points, generate new DiS resource record
- o sign all DS RRSets

3.3. Change of Authoritative servers

Authoritative servers need to support [I-D.ietf-dnsop-glue-is-not-optional]. Then, referral responses MUST contain parent side NS RRSets and whole in-domain glue.

3.4. Change of validating resolvers

When a validating resolver receives a referral response with DS RRSets and the DS RRSets contains a DS resource record that have DISSHA256 digest type, the validating resolver SHOULD validate referral NS RRSets and in-domain glue. First, calculate digest from NS RRSets and in-domain glue from the referral response. Compare the digest and the digest field from the DiS resource record. If the digests differ, the referral is compromised or modified. The validating resolver can drop the referral.

4. Compatibility with the current DNSSEC

Current DNSSEC validators do not know DS resource records with digest type DISSHA256 and these DS records should be ignored. (See Section 5.2 of [RFC4035]).

5. Signing Priming Responses

Another use case for DiS resource record is the protection of priming responses.

The priming response is not a referral. However, it is similar to the referral and the priming response is deterministic.

Then we can put DiS resource record in the root and it can be signed.

The root DiS resource record contains digest consist of the root NS RRSets and all root servers' A and AAAA resource records.

Currently, TTL value of root servers' A/AAAA differ between root servers. Before considering DiS resource record in root, the TTL value of each root server A/AAAA for the root zone and root-servers.net zone must match.

6. IANA Considerations

IANA is requested to allocate new digest type code for DS resource record.

7. Security Considerations

8. Acknowledgments

9. Normative References

[I-D.ietf-dnsop-dns-zone-digest]

Wessels, D., Barber, P., Weinberg, M., Kumari, W., and W. Hardaker, "Message Digest for DNS Zones", draft-ietf-dnsop-dns-zone-digest-14 (work in progress), October 2020.

[I-D.ietf-dnsop-glue-is-not-optional]

Andrews, M., "Glue In DNS Referral Responses Is Not Optional", draft-ietf-dnsop-glue-is-not-optional-00 (work in progress), June 2020.

[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>>.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <<https://www.rfc-editor.org/info/rfc4033>>.

- [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>>.
- [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>>.
- [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>>.

Author's Address

Kazunori Fujiwara
Japan Registry Services Co., Ltd.
Chiyoda First Bldg. East 13F, 3-8-1 Nishi-Kanda
Chiyoda-ku, Tokyo 101-0065
Japan

Phone: +81 3 5215 8451
Email: fujiwara@jprs.co.jp

Network Working Group
Internet-Draft
Updates: RFC 3658, RFC 5155, RFC 6014
(if approved)
Intended status: Standards Track
Expires: January 7, 2021

P. Hoffman
ICANN
July 06, 2020

Revised IANA Considerations for DNSSEC
draft-hoffman-dnssec-iana-cons-01

Abstract

This document changes the review requirements needed to get some DNSSEC algorithms and resource records added to IANA registries. It updates RFC 6014 to include hash algorithms for DS records and NSEC3 parameters.

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 January 7, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. IANA Considerations	2
3. Security Considerations	2
4. Normative References	3
Appendix A. Other Options for Requirements Level	3
Author's Address	4

1. Introduction

DNSSEC is primarily described in [RFC4033], [RFC4034], and [RFC4035]. DNSSEC commonly uses two resource records beyond those defined in RFC 4034: DS [RFC3658] and NSEC3 [RFC5155].

[RFC8126] describes the requirements for listing in the myriad IANA registries.

[RFC6014] updated the requirements for how DNSSEC cryptographic algorithm identifiers in the IANA registries are allocated, reducing the requirements from being "Standards Action" to "RFC Required". However, the IANA registry requirements for hash algorithms for DS records and for the hash algorithms used in NSEC3 are still "Standards Action". This document updates RFC 6014 to bring the requirements for DS records and NSEC3 hash algorithms in line with the rest of the DNSSEC cryptographic algorithms.

2. IANA Considerations

In the "Domain Name System Security (DNSSEC) NextSECure3 (NSEC3) Parameters" registry, the registration procedure for "DNSSEC NSEC3 Flags", "DNSSEC NSEC3 Hash Algorithms", and "DNSSEC NSEC3PARAM Flags" are changed from "Standards Action" to "RFC Required".

In the "Delegation Signer (DS) Resource Record (RR) Type Digest Algorithms" registry, the registration procedure for "Digest Algorithms" is changed from "Standards Action" to "RFC Required".

3. Security Considerations

Changing the requirements for getting security algorithms added to IANA registries as described in this document will make it easier to get good algorithms added to the registries, and will make it easier to get bad algorithms added to the registries. It is impossible to weigh the security impact of those two changes.

4. Normative References

- [RFC3658] Gudmundsson, O., "Delegation Signer (DS) Resource Record (RR)", RFC 3658, DOI 10.17487/RFC3658, December 2003, <<https://www.rfc-editor.org/info/rfc3658>>.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <<https://www.rfc-editor.org/info/rfc4033>>.
- [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>>.
- [RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, DOI 10.17487/RFC5155, March 2008, <<https://www.rfc-editor.org/info/rfc5155>>.
- [RFC6014] Hoffman, P., "Cryptographic Algorithm Identifier Allocation for DNSSEC", RFC 6014, DOI 10.17487/RFC6014, November 2010, <<https://www.rfc-editor.org/info/rfc6014>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.

Appendix A. Other Options for Requirements Level

During an early discussion in the DNSOP Working Group, it was proposed that the requirements for registry allocation for DS resource records be "Specification Required". This would reduce the work required of specification authors, and of the RFC Editor, while still requiring review by an expert reviewer and a long-lived specification.

Author's Address

Paul Hoffman
ICANN

Email: paul.hoffman@icann.org

Network Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: 27 May 2021

K. Fujiwara
JPRS
P. Vixie
Farsight
23 November 2020

Fragmentation Avoidance in DNS
draft-ietf-dnsop-avoid-fragmentation-03

Abstract

EDNS0 enables a DNS server to send large responses using UDP and is widely deployed. Path MTU discovery remains widely undeployed due to security issues, and IP fragmentation has exposed weaknesses in application protocols. Currently, DNS is known to be the largest user of IP fragmentation. It is possible to avoid IP fragmentation in DNS by limiting response size where possible, and signaling the need to upgrade from UDP to TCP transport where necessary. This document proposes to avoid IP fragmentation in 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 27 May 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components

extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	4
3. Proposal to avoid IP fragmentation in DNS	4
3.1. Recommendations for UDP responders	4
3.2. Recommendations for UDP requestors	5
4. Maximum DNS/UDP payload size	5
5. Incremental deployment	6
6. Request to zone operators and DNS server operators	7
7. Considerations	7
7.1. Protocol compliance	7
8. IANA Considerations	7
9. Security Considerations	7
10. Acknowledgments	7
11. References	7
11.1. Normative References	7
11.2. Informative References	9
Appendix A. How to retrieve path MTU value to a destination from applications	10
Appendix B. Minimal-responses	10
Authors' Addresses	10

1. Introduction

DNS has EDNS0 [RFC6891] mechanism. It enables a DNS server to send large responses using UDP. EDNS0 is now widely deployed, and DNS (over UDP) is said to be the biggest user of IP fragmentation.

However, "Fragmentation Considered Poisonous" [Herzberg2013] proposed effective off-path DNS cache poisoning attack vectors using IP fragmentation. "IP fragmentation attack on DNS" [Hlavacek2013] and "Domain Validation++ For MitM-Resilient PKI" [Brandt2018] proposed that off-path attackers can intervene in path MTU discovery [RFC1191] to perform intentionally fragmented responses from authoritative servers. [RFC7739] stated the security implications of predictable fragment identification values.

DNSSEC is a countermeasure against cache poisoning attacks that use IP fragmentation. However, DNS delegation responses are not signed with DNSSEC, and DNSSEC does not have a mechanism to get the correct response if an incorrect delegation is injected. This is a denial-of-service vulnerability that can yield failed name resolutions. If cache poisoning attacks can be avoided, DNSSEC validation failures will be avoided.

In Section 3.2 (Message Side Guidelines) of UDP Usage Guidelines [RFC8085] we are told that an application SHOULD NOT send UDP datagrams that result in IP packets that exceed the Maximum Transmission Unit (MTU) along the path to the destination.

A DNS message receiver cannot trust fragmented UDP datagrams primarily due to the small amount of entropy provided by UDP port numbers and DNS message identifiers, each of which being only 16 bits in size, and both likely being in the first fragment of a packet, if fragmentation occurs. By comparison, TCP protocol stack controls packet size and avoid IP fragmentation under ICMP NEEDFRAG attacks. In TCP, fragmentation should be avoided for performance reasons, whereas for UDP, fragmentation should be avoided for resiliency and authenticity reasons.

[RFC8900] summarized that IP fragmentation introduces fragility to Internet communication. The transport of DNS messages over UDP should take account of the observations stated in that document.

TCP avoids fragmentation using its Maximum Segment Size (MSS) parameter, but each transmitted segment is header-size aware such that the size of the IP and TCP headers is known, as well as the far end's MSS parameter and the interface or path MTU, so that the segment size can be chosen so as to keep the each IP datagram below a target size. This takes advantage of the elasticity of TCP's packetizing process as to how much queued data will fit into the next segment. In contrast, DNS over UDP has little datagram size elasticity and lacks insight into IP header and option size, and so must make more conservative estimates about available UDP payload space.

This document proposes to set IP_DONTFRAG / IPV6_DONTFRAG in DNS/UDP responses in order to avoid IP fragmentation, and describes how to avoid packet losses due to IP_DONTFRAG / IPV6_DONTFRAG.

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 BCP14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

"Requestor" refers to the side that sends a request. "Responder" refers to an authoritative, recursive resolver or other DNS component that responds to questions. (Quoted from EDNS0 [RFC6891])

"Path MTU" is the minimum link MTU of all the links in a path between a source node and a destination node. (Quoted from [RFC8201])

"Path MTU discovery" is defined by [RFC1191], [RFC8201] and [RFC8899].

Many of the specialized terms used in this document are defined in DNS Terminology [RFC8499].

3. Proposal to avoid IP fragmentation in DNS

The methods to avoid IP fragmentation in DNS are described below:

3.1. Recommendations for UDP responders

- * UDP responders SHOULD send DNS responses with IP_DONTFRAG / IPV6_DONTFRAG [RFC3542] options.
- * If the UDP responder detects immediate error that the UDP packet cannot be sent beyond the path MTU size (EMSGSIZE), the UDP responder MAY recreate response packets fit in path MTU size, or TC bit set.
- * UDP responders MAY probe to discover the real MTU value per destination. If the path MTU discovery failed or is impossible, use the default path MTU described in Section 4.
- * UDP responders SHOULD compose UDP responses that result in IP packets that do not exceed the path MTU to the requestor. Of course, as in the conventional case, a specified value (1220 or 1232) as the DNS packet size limit may be used.

The cause and effect of the TC bit is unchanged from EDNS0 [RFC6891].

3.2. Recommendations for UDP requestors

- * UDP requestors SHOULD send DNS responses with IP_DONTFRAG / IPV6_DONTFRAG [RFC3542] options.
- * UDP requestors MAY probe to discover the real MTU value per destination. If the path MTU discovery failed or is impossible, use the default path MTU described in Section 4.
- * UDP requestors SHOULD use the requestor's payload size to limit the path MTU value minus the IP header length and UDP header length. Of course, as in the conventional case, a specified value (1220 or 1232) as the requestor's payload size may be used.
- * UDP requestors MAY drop fragmented DNS/UDP responses without IP reassembly to avoid cache poisoning attacks.
- * DNS responses may be dropped by IP fragmentation. Upon a timeout, UDP requestors may retry using TCP or UDP, per local policy.

4. Maximum DNS/UDP payload size

Default path MTU value for IPv6 is XXXX. Default path MTU value for IPv4 is XXXX.

Discussions under here will be deleted when the discussion is over. There are many discussions for default path MTU values.

- * The minimum MTU for an IPv6 interface is 1280 octets (see Section 5 of [RFC8200]). Then, we can use it as default path MTU value for IPv6.

- * Most of the Internet and especially the inner core has an MTU of at least 1500 octets. An operator of a full resolver would be well advised to measure their path MTU to several authority name servers and to a random sample of their expected stub resolver client networks, to find the upper boundary on IP/UDP packet size in the average case. This limit should not be exceeded by most messages received or transmitted by a full resolver, or else fallback to TCP will occur too often. An operator of authoritative servers would also be well advised to measure their path MTU to several full-service resolvers. The Linux tool "tracert" can be used to measure the path MTU to well known authority name servers such as [a-m].root-servers.net or [a-m].gtld-servers.net. If the reported path MTU is for example no smaller than 1460, then the maximum DNS/UDP payload would be 1432 for IP4 (which is 1460 - IP4 header(20) - UDP header(8)) and 1412 for IP6 (which is 1460 - IP6 header(40) - UDP header(8)). To allow for possible IP options and distant tunnel overhead, a useful default for maximum DNS/UDP payload size would be 1400.
- * [RFC4035] defines that "A security-aware name server MUST support the EDNS0 message size extension, MUST support a message size of at least 1220 octets". Then, the smallest number of the maximum DNS/UDP payload size is 1220.
- * In order to avoid IP fragmentation, [DNSFlagDay2020] proposed that the UDP requestors set the requestor's payload size to 1232, and the UDP responders compose UDP responses fit in 1232 octets. The size 1232 is based on an MTU of 1280, which is required by the IPv6 specification [RFC8200], minus 48 octets for the IPv6 and UDP headers.

By the above reasoning, this proposal is either too small or too large.

5. Incremental deployment

The proposed method supports incremental deployment.

When a full-service resolver implements the proposed method, its stub resolvers (clients) and the authority server network will no longer observe IP fragmentation or reassembly from that server, and will fall back to TCP when necessary.

When an authoritative server implements the proposed method, its full service resolvers (clients) will no longer observe IP fragmentation or reassembly from that server, and will fall back to TCP when necessary.

6. Request to zone operators and DNS server operators

Large DNS responses are the result of zone configuration. Zone operators SHOULD seek configurations resulting in small responses. For example,

- * Use smaller number of name servers (13 may be too large)
- * Use smaller number of A/AAAA RRs for a domain name
- * Use 'minimal-responses' configuration: Some implementations have 'minimal responses' configuration that causes DNS servers to make response packets smaller, containing only mandatory and required data (Appendix B).
- * Use smaller signature / public key size algorithm for DNSSEC. Notably, the signature size of ECDSA or EdDSA is smaller than RSA.

7. Considerations

7.1. Protocol compliance

In prior research ([Fujiwara2018] and dns-operations mailing list discussions), there are some authoritative servers that ignore EDNS0 requestor's UDP payload size, and return large UDP responses.

It is also well known that there are some authoritative servers that do not support TCP transport.

Such non-compliant behavior cannot become implementation or configuration constraints for the rest of the DNS. If failure is the result, then that failure must be localized to the non-compliant servers.

8. IANA Considerations

This document has no IANA actions.

9. Security Considerations

10. Acknowledgments

The author would like to specifically thank Paul Wouters, Mukund Sivaraman and Tony Finch for extensive review and comments.

11. References

11.1. Normative References

- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <<https://www.rfc-editor.org/info/rfc1191>>.
- [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>>.
- [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>>.
- [RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, DOI 10.17487/RFC5155, March 2008, <<https://www.rfc-editor.org/info/rfc5155>>.
- [RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <<https://www.rfc-editor.org/info/rfc6891>>.
- [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<https://www.rfc-editor.org/info/rfc8085>>.
- [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>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <<https://www.rfc-editor.org/info/rfc8201>>.
- [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>>.

- [RFC8899] Fairhurst, G., Jones, T., Tüxen, M., Rüngeler, I., and T. Völker, "Packetization Layer Path MTU Discovery for Datagram Transports", RFC 8899, DOI 10.17487/RFC8899, September 2020, <<https://www.rfc-editor.org/info/rfc8899>>.
- [RFC8900] Bonica, R., Baker, F., Huston, G., Hinden, R., Troan, O., and F. Gont, "IP Fragmentation Considered Fragile", BCP 230, RFC 8900, DOI 10.17487/RFC8900, September 2020, <<https://www.rfc-editor.org/info/rfc8900>>.

11.2. Informative References

- [Brandt2018]
Brandt, M., Dai, T., Klein, A., Shulman, H., and M. Waidner, "Domain Validation++ For MitM-Resilient PKI", Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security , 2018.
- [DNSFlagDay2020]
"DNS flag day 2020", n.d., <<https://dnsflagday.net/2020/>>.
- [Fujiwara2018]
Fujiwara, K., "Measures against cache poisoning attacks using IP fragmentation in DNS", OARC 30 Workshop , 2019.
- [Herzberg2013]
Herzberg, A. and H. Shulman, "Fragmentation Considered Poisonous", IEEE Conference on Communications and Network Security , 2013.
- [Hlavacek2013]
Hlavacek, T., "IP fragmentation attack on DNS", RIPE 67 Meeting , 2013, <<https://ripe67.ripe.net/presentations/240-ipfragattack.pdf>>.
- [RFC3542] Stevens, W., Thomas, M., Nordmark, E., and T. Jinmei, "Advanced Sockets Application Program Interface (API) for IPv6", RFC 3542, DOI 10.17487/RFC3542, May 2003, <<https://www.rfc-editor.org/info/rfc3542>>.
- [RFC7739] Gont, F., "Security Implications of Predictable Fragment Identification Values", RFC 7739, DOI 10.17487/RFC7739, February 2016, <<https://www.rfc-editor.org/info/rfc7739>>.

Appendix A. How to retrieve path MTU value to a destination from applications

Socket options: "IP_MTU (since Linux 2.2) Retrieve the current known path MTU of the current socket. Valid only when the socket has been connected. Returns an integer. Only valid as a getsockopt(2)." (Quoted from Debian GNU Linux manual: ip(7))

"IPV6_MTU getsockopt(): Retrieve the current known path MTU of the current socket. Only valid when the socket has been connected. Returns an integer." (Quoted from Debian GNU Linux manual: ipv6(7))

Appendix B. Minimal-responses

Some implementations have 'minimal responses' configuration that causes a DNS server to make response packets smaller, containing only mandatory and required data.

Under the minimal-responses configuration, DNS servers compose response messages using only RRSets corresponding to queries. In case of delegation, DNS servers compose response packets with delegation NS RRSets in authority section and in-domain (in-zone and below-zone) glue in the additional data section. In case of non-existent domain name or non-existent type, the start of authority (SOA RR) will be placed in the Authority Section.

In addition, if the zone is DNSSEC signed and a query has the DNSSEC OK bit, signatures are added in answer section, or the corresponding DS RRSets and signatures are added in authority section. Details are defined in [RFC4035] and [RFC5155].

Authors' Addresses

Kazunori Fujiwara
Japan Registry Services Co., Ltd.
Chiyoda First Bldg. East 13F, 3-8-1 Nishi-Kanda, Chiyoda-ku, Tokyo
101-0065
Japan

Phone: +81 3 5215 8451
Email: fujiwara@jprs.co.jp

Paul Vixie
Farsight Security Inc
177 Bovet Road, Suite 180
San Mateo, CA, 94402
United States of America

Phone: +1 650 393 3994
Email: vixie@fsi.io

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: March 11, 2021

S. Huque
Salesforce
P. Vixie
Farsight Security
R. Dolmans
NLnet Labs
September 7, 2020

Delegation Revalidation by DNS Resolvers
draft-ietf-dnsop-ns-revalidation-00

Abstract

This document recommends improved DNS [RFC1034] [RFC1035] resolver behavior with respect to the processing of Name Server (NS) resource record sets (RRset) during iterative resolution. When following a referral response from an authoritative server to a child zone, DNS resolvers should explicitly query the authoritative NS RRset at the apex of the child zone and cache this in preference to the NS RRset on the parent side of the zone cut. Resolvers should also periodically revalidate the child delegation by re-querying the parent zone at the expiration of the TTL of the parent side NS RRset.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 11, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Motivation	3
3. Upgrading NS RRset Credibility	4
4. Delegation Revalidation	5
4.1. Using the DS Record TTL	6
5. Optimizations	7
6. Re-delegations and Delegation Removals	7
7. IANA Considerations	7
8. Security Considerations	7
9. References	7
9.1. Normative References	7
9.2. Informative References	8
Acknowledgements	9
Authors' Addresses	9

1. Introduction

RFC EDITOR: PLEASE REMOVE THIS PARAGRAPH BEFORE PUBLISHING: The source for this draft is maintained in GitHub at: <https://github.com/shuque/ns-revalidation>

This document recommends improved DNS resolver behavior with respect to the processing of NS record sets during iterative resolution. The first recommendation is that resolvers, when following a referral response from an authoritative server to a child zone, should explicitly query the authoritative NS RRset at the apex of the child zone and cache this in preference to the NS RRset on the parent side of the zone cut. The second recommendation is to revalidate the delegation by re-querying the parent zone at the expiration of the TTL of the parent side NS RRset.

2. Motivation

There is wide variability in the behavior of deployed DNS resolvers today with respect to how they process delegation records. Some of them prefer the parent NS set, some prefer the child, and for others, what they preferentially cache depends on the dynamic state of queries and responses they have processed. This document aims to bring more commonality and predictability by standardizing the behavior in a way that comports with the DNS protocol.

The delegation NS RRset at the bottom of the parent zone and the apex NS RRset in the child zone are unsynchronized in the DNS protocol. [RFC1034] Section 4.2.2 says "The administrators of both zones should insure that the NS and glue RRs which mark both sides of the cut are consistent and remain so.". But for a variety of reasons they could not be. Officially, a child zone's apex NS RRset is authoritative and thus has a higher cache credibility than the parent's delegation NS RRset, which is non-authoritative glue ([RFC2181], Section 5.4.1. "Ranking data", and Section 6.1. "Zone authority"). Hence the NS RRset "below the zone cut" should immediately replace the parent's delegating NS RRset in cache when an iterative caching DNS resolver crosses a zone boundary. However, this can only happen if (1) the resolver receives the authoritative NS RRset in the Authority section of a response from the child zone, which is not mandatory, or (2) if the resolver explicitly issues an NS RRset query to the child zone as part of its iterative resolution algorithm. In the absence of this, it is possible for an iterative caching resolver to never learn the authoritative NS RRset for a zone, unless a downstream client of the resolver explicitly issues such an NS query, which is not something that normal enduser applications do, and thus cannot be relied upon to occur with any regularity.

Increasingly, there is a trend towards minimizing unnecessary data in DNS responses. Several popular DNS implementations default to such a configuration (see "minimal-responses" in BIND and Unbound). So, they may never include the authoritative NS RRset in the Authority section of their responses.

A common reason that zone owners want to ensure that resolvers place the authoritative NS RRset preferentially in their cache is that the TTLs may differ between the parent and child side of the zone cut. Some DNS Top Level Domains (TLDs) only support long fixed TTLs in their delegation NS sets. In fact, the Extensible Provisioning Protocol (EPP) [RFC5731], that is often used by TLDs to configure delegation parameters has no provision to set the TTL. This inhibits a child zone owner's ability to make more rapid changes to their nameserver configuration using a shorter TTL, if resolvers have no systematic mechanism to observe and cache the child NS RRset.

A child zone's delegation still needs to be periodically revalidated at the parent to make sure that the parent zone has not legitimately re-delegated the zone to a different set of nameservers, or even removed the delegation. Otherwise, resolvers that refresh the TTL of a child NS RRset on subsequent queries or due to pre-fetching, may cling to those nameservers long after they have been re-delegated elsewhere. This leads to the second recommendation in this document, "Delegation Revalidation" - Resolvers should record the TTL of the parent's delegating NS RRset, and use it to trigger a revalidation action.

3. Upgrading NS RRset Credibility

- * When a delegation response is received during iteration, a validation query should be sent in parallel with the resolution of the triggering query, to the delegated nameservers for the newly discovered zone cut. Note that validating resolvers today, when following a secure referral, already need to dispatch a query to the delegated nameservers for the DNSKEY RRset, so this validation query could be sent in parallel with that DNSKEY query.
- * A validation query consists of a query for the child's apex NS RRset, sent to the newly discovered delegation's nameservers. Normal iterative logic applies to the processing of responses to validation queries, including storing the results in cache, trying the next server on SERVFAIL or timeout, and so on. Positive answers to this validation query will be cached with an authoritative data ranking. Successive queries directed to the same zone will be directed to the nameservers listed in the child's apex, due to the ranking of this answer. If the validation query fails, the parent NS RRset will remain the one with the highest ranking and will be used for successive queries.
- * Some resolvers may choose to delay the response to the triggering query until both the triggering query and the validation query have been answered. In practice, we expect many implementations may answer the triggering query in advance of the validation query for performance reasons. An additional reason is that there are number of nameservers in the field that (incorrectly) fail to answer explicit queries for NS records, and thus the revalidation logic may need to be applied lazily and opportunistically to deal with them.

- * If the resolver chooses to delay the response, and there are no nameserver names in common between the child's apex NS RRset and the parent's delegation NS RRset, then the responses received from forwarding the triggering query to the parent's delegated nameservers should be discarded after validation, and this query should be forwarded again to the child's apex nameservers.

4. Delegation Revalidation

This document proposes two mechanisms to perform delegation revalidation: an extensive and a simple mechanism. [TODO: should we keep just the simple mechanism?]

The extensive mechanism:

- * The lowest TTL found in a parent zone's delegating NS RRset should be stored in the cache and used to trigger delegation revalidation as follows: Whenever a cached RRset is being considered for use in a response, the cache should be walked upward toward the root, looking for expired delegations. At the first expired delegation encountered while walking upward toward the root, revalidation should be triggered, putting the processing of dependent queries on hold until validation is complete.
- * To revalidate a delegation, the iterative caching DNS resolver will redo resolution of the triggering query at the closest enclosing zone cut above the revalidation point. If Query-name Minimization [RFC7816] is being employed, this may not be the full name of the triggering query, but the query name with some number of left most labels excised as dictated by the qname minimization algorithm. While searching for these nameservers, additional revalidations may occur, perhaps placing a chain of dependent queries on hold, unwinding in downward order as revalidations closer to the root must be complete before revalidations further from the root can begin.
- * If a delegation can be revalidated at the same node, then the old apex NS RRset should be deleted from cache and then the new delegating NS RRset should be stored in cache. The minimum TTL from the new delegating NS RRset should also be stored in cache to facilitate future revalidations. This order of operations ensures that the RRset credibility rules do not prevent the new delegating NS RRset from entering the cache. It is expected that the child's apex NS RRset will rapidly replace the parent's delegating NS RRset as soon as iteration restarts after the revalidation event.

- * If the new delegating NS RRset cannot be found (RCODE=NXDOMAIN) or if there is a new zone cut at some different level of the hierarchy (insertion or deletion of a delegation point above the revalidation point) or if the new RRset shares no nameserver names in common with the old one (indicating some kind of redelegation, which is rare) then the cache should be purged of all names and RRsets at or below the revalidation point. This facilitates redelegation or revocation of a zone by a parent zone administrator, and also conserves cache storage by deleting unreachable data.

The simple mechanism:

- * Cap the time to cache the child NS RRset to the lower of child and parent NS RRset TTL. The normal iterative resolution algorithm will then cause delegation revalidation to naturally occur at the expiration of the capped child NS TTL, along with dispatching of the validation query to upgrade NS RRset credibility.

4.1. Using the DS Record TTL

If both parent and child zone are DNSSEC [RFC4033] [RFC4034] [RFC4035] signed with a corresponding secure delegation between them, then expiration of the Delegation Signer (DS) record set will cause revalidation of the current child zone's DNSKEY set. According to RFC 4035, Section 2.4, "The TTL of a DS RRSet SHOULD match the TTL of the delegating NS RRset", so this revalidation should be triggered on the same time scale, and thus responses from the stale child nameservers would no longer be trusted. However, delegation revalidation is still necessary to locate the current nameserver addresses to which subsequent DNS queries should be directed.

In practice, the DS TTL often differs from the delegating NS TTL. For example, currently the root zone and the COM and NET TLDs all set an NS RRset TTL of 2 days, while the DS RRset TTL is just 1 day. RSSAC-003 [rssac-003] makes the following observation: "In the root zone, delegating NS records have a 2 day TTL. However, the DS records have a 1 day TTL, against the advice of RFC 4035. This is not particularly surprising since a mistake with a DS record can deny resolution for all names under a TLD. Given the way that DS records are currently used in the root zone (e.g., usually matching just one TLD KSK) it is better for them to have a lower TTL in the event of an emergency change."

If a secure delegation is present, resolvers may use the DS RRset's TTL as the revalidation interval in preference to the delegating NS RRset TTL. (Question: should this be the recommendation instead?; after all the DS is signed so its TTL cannot be spoofed?)

5. Optimizations

TODO: mention possible optimizations: record whether certain nameservers return the child NS set in their authoritative section responses and subsequently forego the extra child NS query for a period of time. Suggest that authoritative servers that do minimal responses return their NS sets in response to DNSKEY queries, and resolvers that see such behavior may also subsequently forego the extra child NS query (from Olafur G.).

6. Re-delegations and Delegation Removals

TODO: mention in more detail what to do when resolvers observe redelegations or removal of delegations at the parent. A quick initial summary follows.

If a delegation is removed (i.e. the parent returns NXDOMAIN), then cache contents should be treated as described in [RFC8020] -- ideally the resolver's cache should be entirely pruned at the delegation point, or the cached contents below the delegation may be allowed to be used until they expire. Similar treatment of the cache should be followed if the child zone has been entirely re-delegated to a new set of nameservers. If only a subset of nameservers have been re-delegated, then no new cache cleanup action is needed.

7. IANA Considerations

This document includes no request to IANA.

8. Security Considerations

Upgrading NS RRset Credibility (Section 3) allows resolvers to cache and utilize the authoritative child apex NS RRset in preference to the non-authoritative parent NS RRset. However, it is important to implement the steps described in Delegation Revalidation (Section 4) at the expiration of the parent's delegating TTL. Otherwise, the operator of a malicious child zone, originally delegated to, but subsequently delegated away from, can cause resolvers that refresh TTLs on subsequent NS set queries, or that pre-fetch NS queries, to never learn of the redelegated zone. This problem has been seen in the wild [include reference to Ghost Domains paper here].

9. References

9.1. Normative References

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <<https://www.rfc-editor.org/info/rfc1034>>.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<https://www.rfc-editor.org/info/rfc1035>>.
- [RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, DOI 10.17487/RFC2181, July 1997, <<https://www.rfc-editor.org/info/rfc2181>>.
- [RFC7816] Bortzmeyer, S., "DNS Query Name Minimisation to Improve Privacy", RFC 7816, DOI 10.17487/RFC7816, March 2016, <<https://www.rfc-editor.org/info/rfc7816>>.
- [RFC8020] Bortzmeyer, S. and S. Huque, "NXDOMAIN: There Really Is Nothing Underneath", RFC 8020, DOI 10.17487/RFC8020, November 2016, <<https://www.rfc-editor.org/info/rfc8020>>.

9.2. Informative References

- [I-D.vixie-dnsex-*resimprove*]
Vixie, P., Joffe, R., and F. Neves, "Improvements to DNS Resolvers for Resiliency, Robustness, and Responsiveness", Work in Progress, Internet-Draft, draft-vixie-dnsex-*resimprove-00*, June 23, 2010, <<https://tools.ietf.org/html/draft-vixie-dnsex-resimprove-00>>.
- [I-D.wijnngaards-dnsex-*resolver-side-mitigation*]
Wijnngaards, W., "Resolver side mitigations", Work in Progress, Internet-Draft, draft-wijnngaards-dnsex-*resolver-side-mitigation-01*, February 24, 2009, <<https://tools.ietf.org/html/draft-wijnngaards-dnsex-resolver-side-mitigation-01>>.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <<https://www.rfc-editor.org/info/rfc4033>>.
- [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>>.
- [RFC5731] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, RFC 5731, DOI 10.17487/RFC5731, August 2009, <<https://www.rfc-editor.org/info/rfc5731>>.
- [rssac-003] RSSAC_Caucus, "RSSAC003 Report on Root Zone TTLs", August 2015, <<https://www.icann.org/en/system/files/files/rssac-003-root-zone-ttls-21aug15-en.pdf>>.

Acknowledgements

Wouter Wijngaards proposed explicitly obtaining authoritative child NS data in [I-D.wijngaards-dnsex-ns-resolver-side-mitigation]. This behavior has been implemented in the Unbound DNS resolver via the "harden-referral-path" option. The combination of child NS fetch and revalidating the child delegation was originally proposed in [I-D.vixie-dnsex-resimprove], by Vixie, Joffe, and Neves.

Authors' Addresses

Shumon Huque
Salesforce

Email: shuque@gmail.com

Paul Vixie
Farsight Security

Email: paul@redbarn.org

Ralph Dolmans
NLnet Labs

Email: ralph@nlnetlabs.nl

Independent Submission
Internet-Draft
Intended status: Best Current Practice
Expires: April 11, 2021

R. Arends
ICANN
J. Abley
Public Interest Registry
October 08, 2020

Top-level Domains for Private Internets
draft-ietf-dnsop-private-use-tld-00

Abstract

There are no defined private-use namespaces in the Domain Name System (DNS). For a domain name to be considered private-use, it needs to be future-proof in that its top-level domain will never be delegated from the root zone. The lack of a private-use namespace has led to locally configured namespaces with a top-level domain that is not future proof.

The DNS needs an equivalent of the facilities provided by BCP 5 (RFC 1918) for private internets, i.e. a range of short, semantic-free top-level domains that can be used in private internets without the risk of being globally delegated from the root zone.

The ISO 3166 standard is used for the definition of eligible designations for country-code top-level Domains. This standard is maintained by the ISO 3166 Maintenance Agency. The ISO 3166 standard includes a set of user-assigned code elements that can be used by those who need to add further names to their local applications.

Because of the rules set out by ISO in their standard, it is extremely unlikely that these user-assigned code elements would ever conflict with delegations in the root zone under current practices. This document explicitly reserves these code elements to be safely used as top-level domains for private DNS resolution.

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 April 11, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction
 2. The ISO 3166-1 alpha-2 and Two-Letter Top-Level Domains
 3. ISO 3166-1 alpha-2 User-assigned Code Elements
 4. Examples of Current Uses of the User-assigned Code Elements.
 5. Private-use top-level Domains
 6. IANA Considerations
 7. Security Considerations
 8. Acknowledgements
 9. Informative References
- Authors' Addresses

1. Introduction

In private networks where a hostname has no utility in the global namespace, it is convenient to have a private-use namespace. Such deployments could theoretically use sub-domains of a domain registered for the specific hosting entity, though not all such configurations have such a domain available. When the hostname is solely used in a private network, it is not necessary that it resolves globally.

Another situation is where applications use identifiers that are similar in appearance to domain names, and may be interpreted by software as domain names, but are not intended to use the global DNS resolution service. Using a private-use namespace helps guard against conflicts with the global DNS resolution system.

Note that a private-use namespace is not a subset of a registered special use namespace [IANA-Special]. There is no facility to register a specific label using the process defined in [RFC6761]. The process in RFC 6761 requires that a label has some kind of special handling in order to be considered special. A private-use namespace can be considered special on a policy level, but not on a technical or protocol level.

Many protocols outside the DNS have a defined set of elements for private use, or an identifier that indicates private use, such as "X-headers" MIME types [RFC2045], addresses for private internets [BCP-5], the "x-" sub-tag in private-use language tags [BCP-47], private-use Autonomous System Numbers [BCP-6], and private-use DNS RRTypes and RCODEs [BCP-42].

There is currently no such facility for the DNS namespace. A user is required to resort to registering a globally unique domain where a locally unique domain would suffice, or may configure a domain name that is not currently delegated from the root zone. Additionally, there are plenty of examples of device vendors that ship networking devices with a default setting for DHCP [RFC2131] option 15 (domain name) [RFC2132], containing a top-level domain that is believed to not be delegated in the root zone.

In practice, the lack of a private-use namespace facility has led to the deployment of arbitrary, unregistered, semantically meaningful top-level domains, such as ".home", ".dhcp", ".lan", ".localdomain", ".internal", ".dlink", ".ip" and ".corp" [ITHI]. These examples of locally configured strings are derived from traffic to the ICANN Managed Root Servers [IMRS] and are part of the most popular observed query names [BCP-219].

While these commonly chosen strings currently do not exist in the root zone, there is no guarantee that these strings will not be delegated in the root zone in the future. Therefore, there is no guarantee that the local use of these strings (or other strings that might be chosen for private use) will be stable, safe, and secure. Similarly, there is no guarantee that any of these strings will be deemed special-use via an application according to [RFC6761].

There are many uses for private-use names. It is not feasible to assign a semantically meaningful, relatively short top-level label to each individual private-use of a namespace in multiple languages. Similar to "X-headers" MIME types, and analogous in concept to address allocation for private internets, this document defines a range of abstract, two-letter labels that are aligned with the user-assigned two-letter code elements in the ISO 3166-1 alpha-2 [ISO3166-1] standard.

2. The ISO 3166-1 alpha-2 and Two-Letter Top-Level Domains

IANA's practice of governing the delegation of ASCII two-letter domain names in the DNS [STD13] root zone is to align it with assignment of two-letter (known as "alpha-2") code elements in the ISO 3166-1 standard [ISO3166-1]. The ISO 3166-1 standard contains many categories of code elements, with the "officially assigned" and some "exceptionally reserved" code elements being used in the DNS to represent entities as country-code top-level domains (ccTLDs) [RFC1591]. The interrelationship is documented in "ICANN and the ISO, A Common Interest in ISO Standard 3166" [ICANNISO].

In addition to the assigned, available, and reserved code elements, there are code elements designated as "user-assigned". The intent of user-assigned code elements is to provide the user with a code element when no other code element satisfies the intended use.

3. ISO 3166-1 alpha-2 User-assigned Code Elements

The ISO 3166-1 standard states in section 5.2:

"In addition, exactly 42 alpha-2 code elements are not used in the ISO 3166, AA, QM to QZ, XA to XZ, ZZ."

And explains in clause 8.1 "Special Provisions":

"Users sometimes need to extend or alter the use of country-code elements for special purposes. The following provisions give guidance for meeting such needs within the framework of this part of ISO 3166. "

And finally, clause 8.1.3 "User assigned code element":

"If users need code elements to represent country names not included in this part of ISO 3166, the series of letters AA, QM to QZ, XA to XZ, and ZZ, and the series AAA to AAZ, QMA to QZZ, XAA

to XZZ, and ZZA to ZZZ respectively and the series of numbers 900 to 999 are available. NOTE Users are advised that the above series of codes are not universals, those code elements are not compatible between different entities."

As shown above, the ISO 3166-1 user-assigned alpha-2 code elements are defined to be AA, QM to QZ, XA to XZ, and ZZ. The ranges ("to") are alphabetic and contain only characters in the US-ASCII definition [STD80].

It is unlikely that the user-assigned range will change.

4. Examples of Current Uses of the User-assigned Code Elements.

Using code elements to represent an entity other than a country name may appear to deviate from the intended use of the ISO 3166-1 standard. However, many organizations, including the IETF and the ISO, have used the user-assigned range to represent entities other than country names. The following list is not exhaustive but illustrates the wide variety of current uses of codes within the ISO 3166-1 user-assigned alpha-2 range.

- o The International Standard Recording Code (ISRC) [ISO3901] uses code element "ZZ" from the User-assigned range for direct registrants independent of any country.
- o The ISO Currency Codes standard [ISO4217] uses code elements "XA" to "XZ" from the user-assigned range for transactions and precious metals.
- o International Securities Identification Numbers [ISO6166] uses the following code elements from the user-assigned range:

QS: internally used by Euroclear France

QT: internally used in Switzerland

QW: internally used in WM Datenservice Germany for historical data

XA: CUSIP Global Services substitute agencies

XB: NSD Russia substitute agencies

XC: WM Datenservice Germany substitute agencies

XD: SIX Telekurs substitute agencies

XF: internally assigned, non-unique numbers

XS: Euroclear and Clearstream international securities

- o The International Civil Aviation Organization [ICAO] Machine Readable Travel Documents standard uses code element "ZZ" from the user-assigned range for UN travel documents.

- o The World Intellectual Property Organization [WIPO] Standard 3 uses the following code elements from the user-assigned range:

QZ: Community Plant Variety Office (European Union) (CPVO).

XN: Nordic Patent Institute (NPI).

XU: International Union for the Protection of New Varieties of Plants (UPOV).

XV: Visegrad Patent Institute (VPI)

XX: recommended to refer to unknown states, other entities or organizations.

- o The United Nations Code for Trade and Transport Locations [UNLOCODE] uses the code element "XZ" from the user-assigned range for international waters in accordance with ISO 3166-1 clause 8.1.3:

"3.2.5 In cases where no ISO 3166 country-code element is available, e.g. installations in international waters or international cooperation zones, the code element "XZ", available for user assignment in accordance with clause 8.1.3 of ISO 3166-1/1997, will be used."

- o The World Bank Country API [WORLDBANK] uses the following code elements from the User-assigned range:

XC: Euro area

XD: High income

XE: Heavily indebted poor countries (HIPC)

XF: International Bank for Reconstruction and Development

XH: Blend

XI: International Development Association

XJ: Latin America and Caribbean (excluding high income)

XL: Least developed countries: UN classification

XM: Low income

XN: Lower middle income

XO: Low & middle income

XP: Middle income

XQ: Middle East & North Africa (excluding high income)

XT: Upper middle income

XU: North America

XX: Not classified

XY: Not Classified

- o The Interpol Implementation data format for the interchange of fingerprint, facial & scar-mark-tattoo information [INTERPOL] uses code element "ZZ" from the user-assigned range as follows: Destination Agency Identifier "ZZ/ALL" is reserved for transactions which shall be distributed by INTERPOL AFIS to all INTERPOL member states."

- o The Certificate Authority Browser Forum Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates [CABForum] states that if a country is not represented by an official ISO 3166-1 alpha-2 country-code, the CA may specify the user-assigned code element "XX" to indicate that an official code element has not been assigned.
- o The UNICODE Common Locale Data Repository (CLDR) [UNICODE] version 37 uses the following code elements from the user-assigned range:

QO: Outlying Oceania; countries in Oceania that do not have a subcontinent.

XA: Pseudo-Accents; special code indicating derived testing locale with English + added accents and lengthened.

XB: Pseudo-Bidi; special code indicating derived testing locale with forced RTL English.

ZZ: Unknown Region; used in APIs or as a replacement for invalid code.

- o The IETF Best Current Practice 47 [BCP-47] contains a section and examples dedicated to private-use subtags, using code elements from the user-assigned range:

"For example, the region subtags 'AA', 'ZZ', and those in the ranges 'QM'-'QZ' and 'XA'-'XZ' (derived from the ISO 3166-1 alpha-2 private use codes) can be used to form a language tag. A tag such as "zh-Hans-XQ" conveys a great deal of public, interchangeable information about the language material"

- o The IETF Proposed Standard "Internationalized Domain Names for Applications" [RFC5890] uses the XN-- prefix. The method that was used to decide on the prefix was explained in an email from the IANA to the IETF IDN Working Group list [XNIDN]:

"The following steps will be used to select the two-character code:

The code will be selected from among a subset of the entries on the ISO 3166-1, clause 8.1.3 User-assigned alpha-2 code elements: AA, QM to QZ, XA to XZ, and ZZ. The selection is limited to these codes because of the following:

The use of ISO 3166-1 User-assigned elements removes the possibility that the code will duplicate a present or future ccTLD code."

5. Private-use top-level Domains

The user-assigned classification of these code elements in the ISO 3166-1 alpha-2 standard allows for the assumption that these code elements will not risk delegation as country-code top-level Domains through future assignments to represent a country or territory. To quote [XNIDN]:

"The use of ISO 3166-1 User-assigned elements removes the possibility that the code will duplicate a present or future ccTLD code."

Using these code elements as top-level domains for the purpose of private-use TLDs is in line with the intended use of these code elements and follows the many examples of other standards and protocols. Furthermore, they are short and free of any semantic meaning.

This document does not recommend any specific ISO 3166-1 alpha-2 user-assigned code as a private use, but instead proposes that any of them can be used by a network or application for private use. That is, there is no attempt to choose just one of the ISO 3166-1 Alpha-2 user-assigned codes for use as private-use TLDs, just as other organizations use multiple user-assigned codes for many internal purposes.

Note that there may be software that treats labels beginning with XN differently due to the use of the XN- prefix in internationalized domain names [RFC5890].

6. IANA Considerations

This document makes the observation that the policy of assigning ccTLD labels is to align with the ISO-3166-1 alpha-2 standard [RFC1591], which includes user-assigned code elements that will never be assigned to a territory [ISO3166-1]. This is then consistent with existing policies that those user-assigned codes will never be delegated from the DNS root zone and, for that reason, will never give rise to collisions with any future new TLD.

7. Security Considerations

Use of private-use identifiers of any sort almost always results in unexpected collisions in practice. This has repeatedly been shown for private-use addresses, private-use identifiers (such as "x-headers") and private-use names in the DNS. These unexpected collisions can easily have security ramifications that are well beyond what the user understands.

8. Acknowledgements

This document is based on an earlier draft by Ed Lewis. David Conrad, Paul Hoffman, Sion Lloyd, Alain Durand, Jaap Akkerhuis, Kal Feher, Andrew Sullivan, Joe Abley, Petr Spacek, Patrick Mevzek and Kim Davies have played a role.

9. Informative References

- [BCP-219] 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>>.
- [BCP-42] Eastlake 3rd, D., "Domain Name System (DNS) IANA Considerations", BCP 42, RFC 6895, DOI 10.17487/RFC6895, April 2013, <<https://www.rfc-editor.org/info/rfc6895>>.
- [BCP-47] Phillips, A., Ed. and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 5646, DOI 10.17487/RFC5646, September 2009, <<https://www.rfc-editor.org/info/rfc5646>>.
- [BCP-5] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <<https://www.rfc-editor.org/info/rfc1918>>.

- [BCP-6] Mitchell, J., "Autonomous System (AS) Reservation for Private Use", BCP 6, RFC 6996, DOI 10.17487/RFC6996, July 2013, <<https://www.rfc-editor.org/info/rfc6996>>.
- [CABForum] "CA/Browser Forum Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates, version 1.6.9", March 2020, <<https://cabforum.org/wp-content/uploads/CA-Browser-Forum-BR-1.6.9.pdf>>.
- [IANA-Special] "Special-Use Domain Names", n.d., <<https://www.iana.org/assignments/special-use-domain-names/special-use-domain-names.xhtml>>.
- [ICANNISO] "ICANN and the International Organization for Standardization (ISO)", n.d., <<https://www.icann.org/resources/pages/icann-iso-3166-2012-05-09-en>>.
- [ICAO] "International Civil Aviation Organization, Machine Readable Travel Documents, Part 3; Specifications Common to all MRTDs", n.d., <https://www.icao.int/publications/Documents/9303_p3_cons_en.pdf>.
- [IMRS] "ICANN Managed Root Server", n.d., <<https://www.dns.icann.org/imrs/>>.
- [INTERPOL] "Interpol Implementation data format for the interchange of fingerprint, facial & smt information", n.d., <<https://www.interpol.int/en/How-we-work/Forensics/Fingerprints>>.
- [ISO3166-1] "ISO 3166-1:2013 Codes for the representation of names of countries and their subdivisions - Part 1: Country codes", 2013, <<https://www.iso.org/standard/63545.html>>.
- [ISO3901] "Information and documentation -- International Standard Recording Code (ISRC)", n.d., <<https://www.iso.org/standard/64817.html>>.
- [ISO4217] "ISO 4217; Codes for the representation of currencies and funds", n.d., <<https://www.iso.org/iso-4217-currency-codes.html>>.
- [ISO6166] "Securities and related financial instruments -- International securities identification numbering system (ISIN)", n.d., <<https://www.iso.org/standard/44811.html>>.
- [ITHI] "ICANN's Identifier Technology Health Indicator; Queries to frequently leaked strings", n.d., <<https://ithi.research.icann.org/graph-m3.html#M332>>.
- [RFC1591] Postel, J., "Domain Name System Structure and Delegation", RFC 1591, DOI 10.17487/RFC1591, March 1994, <<https://www.rfc-editor.org/info/rfc1591>>.
- [RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail

- Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, DOI 10.17487/RFC2045, November 1996, <<https://www.rfc-editor.org/info/rfc2045>>.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <<https://www.rfc-editor.org/info/rfc2131>>.
- [RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, DOI 10.17487/RFC2132, March 1997, <<https://www.rfc-editor.org/info/rfc2132>>.
- [RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, DOI 10.17487/RFC5890, August 2010, <<https://www.rfc-editor.org/info/rfc5890>>.
- [RFC6761] Cheshire, S. and M. Krochmal, "Special-Use Domain Names", RFC 6761, DOI 10.17487/RFC6761, February 2013, <<https://www.rfc-editor.org/info/rfc6761>>.
- [STD13] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <<https://www.rfc-editor.org/info/rfc1034>>.
- [STD80] Cerf, V., "ASCII format for network interchange", STD 80, RFC 20, DOI 10.17487/RFC0020, October 1969, <<https://www.rfc-editor.org/info/rfc20>>.
- [UNICODE] "CLDRv37 - Unicode Common Locale Data Repository version 37", April 2020, <<http://cldr.unicode.org/index/downloads/cldr-37>>.
- [UNLOCODE] "United Nations Code for Trade and Transport Locations; UN/LOCODE Manual", n.d., <https://www.unece.org/fileadmin/DAM/cefact/locode/UNLOCODE_Manual.pdf>.
- [WIPO] "World Intellectual Property Organization; Recommended standard on two-letter codes for the representation of states, other entities and intergovernmental organizations.", n.d., <<https://www.wipo.int/export/sites/www/standards/en/pdf/03-03-01.pdf>>.
- [WORLDBANK] "Worldbank API V2 Country API", n.d..
- [XNIDN] "Results of IANA Selection of IDNA Prefix", February 2003, <<https://psg.com/~randy/lists/iesg/2003/msg01081.html>>.

Authors' Addresses

Roy Arends
ICANN

Email: roy.arends@icann.org

Joe Abley
Public Interest Registry

470 Moore Street
London, true N6C 2C2
Canada

Email: jabley@pir.org

Network Working Group
Internet-Draft
Obsoletes: 5933 (if approved)
Intended status: Standards Track
Expires: April 1, 2021

D. Belyavskiy
TCINET
V. Dolmatov, Ed.
JSC "NPK Kryptonite"
September 28, 2020

Use of GOST 2012 Signature Algorithms in DNSKEY and RRSIG Resource
Records for DNSSEC
draft-ietf-dnsop-rfc5933-bis-02

Abstract

This document describes how to produce digital signatures and hash functions using the GOST R 34.10-2012 and GOST R 34.11-2012 algorithms for DNSKEY, RRSIG, and DS resource records, for use in the Domain Name System Security Extensions (DNSSEC).

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 April 1, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	3
2. DNSKEY Resource Records	3
2.1. Using a Public Key with Existing Cryptographic Libraries	3
2.2. GOST DNSKEY RR Example	4
3. RRSIG Resource Records	4
3.1. RRSIG RR Example	4
4. DS Resource Records	5
4.1. DS RR Example	5
5. Deployment Considerations	5
5.1. Key Sizes	5
5.2. Signature Sizes	5
5.3. Digest Sizes	6
6. Implementation Considerations	6
6.1. Support for GOST Signatures	6
7. Security Considerations	6
8. IANA Considerations	6
9. Acknowledgments	7
10. References	7
10.1. Normative References	7
10.2. Informative References	8
Authors' Addresses	8

1. Introduction

The Domain Name System (DNS) is the global hierarchical distributed database for Internet Naming. The DNS has been extended to use cryptographic keys and digital signatures for the verification of the authenticity and integrity of its data. RFC 4033 [RFC4033], RFC 4034 [RFC4034], and RFC 4035 [RFC4035] describe these DNS Security Extensions, called DNSSEC.

RFC 4034 describes how to store DNSKEY and RRSIG resource records, and specifies a list of cryptographic algorithms to use. This document extends that list with the signature and hash algorithms GOST R 34.10-2012 ([RFC7091]) and GOST R 34.11-2012 ([RFC6986]), and specifies how to store DNSKEY data and how to produce RRSIG resource records with these algorithms.

This document obsoletes RFC5933 [RFC5933]. This document also marks the DNS Security Algorithm GOST R 34.10-2001 as obsolete.

Familiarity with DNSSEC and with GOST signature and hash algorithms is assumed in this document.

1.1. 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.

2. DNSKEY Resource Records

The format of the DNSKEY RR can be found in RFC 4034 [RFC4034].

GOST R 34.10-2012 public keys are stored with the algorithm number TBA1.

According to RFC 7091 [RFC7091], a public key is a point on the elliptic curve $Q = (x,y)$. The wire representation of a public key MUST contain 64 octets, where the first 32 octets contain the little-endian representation of x and the second 32 octets contain the little-endian representation of y .

As RFC 6986 and RFC 7091 allows 2 variants of length of the output hash and signature and many variants of parameters of the digital signature, for the purpose of this document we use 256-bit variant of the digital signature algorithm, corresponding 256-bit variant of the digest algorithm. We select the parameters for the digital signature algorithm to be id-tc26-gost-3410-2012-256-paramSetA in RFC 7836 [RFC7836].

2.1. Using a Public Key with Existing Cryptographic Libraries

At the time of this writing, existing GOST-aware cryptographic libraries are capable of reading GOST public keys via a generic X509 API if the key is encoded according to RFC 7091 [RFC7091], Section 2.3.2.

To make this encoding from the wire format of a GOST public key with the parameters used in this document, prepend the 64 octets of key data with the following 32-byte sequence:

```
0x30 0x5e 0x30 0x17 0x06 0x08 0x2a 0x85 0x03 0x07 0x01 0x01 0x01
0x01 0x30 0x0b 0x06 0x09 0x2a 0x85 0x03 0x07 0x01 0x02 0x01 0x01
0x01 0x03 0x43 0x00 0x04 0x40
```

2.2. GOST DNSKEY RR Example

Given a private key with the following value (the value of the Gost12Asn1 field is split here into two lines to simplify reading; in the private key file, it must be in one line):

```
Private-key-format: v1.2
Algorithm: 23 (ECC-GOST12)
Gost12Asn1: MD4CAQAwFwYIKoUDBwEBAQEwCwYJKoUDBwECAQEBCA0
           zvTDpCSjdRCERkd6WDA2TF/ABQLp9MPZRl7hMXCVGg==
```

The following DNSKEY RR stores a DNS zone key for example:

```
example. 600 IN DNSKEY 256 3 23 (
           XkZ6T+qQ9teOMsA/YK+kTzELhuMPTsYggdy2b+sfzJ6t
           H9eniziMX3gjMnUZIyrnSIchLjup8xpy+UU511Eyjw==
           ) ;{id = 13439 (zsk), size = 512b}
```

3. RRSIG Resource Records

The value of the signature field in the RRSIG RR follows RFC 7091 [RFC7091] and is calculated as follows. The values for the RDATA fields that precede the signature data are specified in RFC 4034 [RFC4034].

hash = GOSTR3411-2012(data)

where "data" is the wire format data of the resource record set that is signed, as specified in RFC 4034 [RFC4034].

The signature is calculated from the hash according to the GOST R 34.10-2012 standard, and its wire format is compatible with RFC 7091 [RFC7091].

3.1. RRSIG RR Example

With the private key from this document, consisting of one MX record:

```
example. 600 IN MX 10 mail.example.
```

Setting the inception date to 2020-01-04 17:25:26 UTC and the expiration date to 2020-02-01 17:25:26 UTC, the following signature RR will be valid:

```
example. 600 IN RRSIG MX 23 1 600 20200201172526 (
                20200104172526 13439 example.
                EtrsAEGsNRf12HKjwNTg8U2HZ5JOSo34UaTcshoElkwd
                5Ror4I7z1tmWAgd4b9OBn80tsajtL0Vuf45u8kEAgA==
        )
```

Note: The ECC-GOST12 signature algorithm uses random data, so the actual computed signature value will differ between signature calculations.

4. DS Resource Records

The GOST R 34.11-2012 digest algorithm is denoted in DS RRs by the digest type TBA2. The wire format of a digest value is compatible with RFC 6986 [RFC6986].

4.1. DS RR Example

For Key Signing Key (KSK):

```
example. IN DNSKEY 257 3 23 (
                hP3ISWPT8ehEEut8ozbqPcmbTAQK0jce7MHmK0geOiRo
                kFALGwsMrBf0H0AK2qrVJCWCJL+50v9UNZAS5mE70g==
                ) ;{id = 7574 (ksk), size = 512b}
```

The DS RR will be:

```
example. IN DS 7574 23 5 (
                990f40dc548a4dbcb4b80a0760f194ac
                0cc18484578834c1ac1f749f70c84103
                )
```

5. Deployment Considerations

5.1. Key Sizes

The key size of GOST public keys conforming to this specification MUST be 512 bits.

5.2. Signature Sizes

The size of a GOST signature conforming to this specification MUST be 512 bits.

5.3. Digest Sizes

The size of a GOST digest conforming to this specification MUST be 256 bits.

6. Implementation Considerations

6.1. Support for GOST Signatures

DNSSEC-aware implementations MAY be able to support RRSIG and DNSKEY resource records created with the GOST algorithms as defined in this document.

7. Security Considerations

Currently, the cryptographic resistance of the GOST R 34.10-2012 digital signature algorithm is estimated as 2^{128} operations of multiple elliptic curve point computations on prime modulus of order 2^{256} .

Currently, the cryptographic collision resistance of the GOST R 34.11-2012 hash algorithm is estimated as 2^{128} operations of computations of a step hash function.

8. IANA Considerations

This document updates the IANA registry "DNS Security Algorithm Numbers". The following entries have been added to the registry:

Value	Algorithm	Mnemonic	Zone Signing	Trans. Sec.	References	Status
TBA1	GOST R 34.10-2012	ECC-GOST12	Y	*	RFC TBA	OPTIONAL

The entry for the Algorithm "GOST R 34.10-2001", number 12 should be updated as such: Description field should be changed to "GOST R 34.10-2001 (deprecated, see TBA1" and Zone Signing field should be changed to "N".

This document updates the RFC IANA registry "Delegation Signer (DS) Resource Record (RR) Type Digest Algorithms" by adding an entry for the GOST R 34.11-2012 algorithm:

Value	Algorithm	Status
TBA2	GOST R 34.11-2012	OPTIONAL

The entry for Value 3, GOST R 34.11-94 should be updated to have its Status changed to '-'.

This paragraph should be removed before the publication of RFC: For the purpose of example computations, the following values were used: TBA1 = 23, TBA2 = 5.

9. Acknowledgments

This document is a minor extension to RFC 4034 [RFC4034]. Also, we tried to follow the documents RFC 3110 [RFC3110], RFC 4509 [RFC4509], and RFC 5933 [RFC5933] for consistency. The authors of and contributors to these documents are gratefully acknowledged for their hard work.

The following people provided additional feedback, text, and valuable assistance: Alexander Venedyukhin, Valery Smyslov, Tim Wicinski, TODO

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/info/rfc2119>>.
- [RFC3110] Eastlake 3rd, D., "RSA/SHA-1 SIGs and RSA KEYS in the Domain Name System (DNS)", RFC 3110, DOI 10.17487/RFC3110, May 2001, <<https://www.rfc-editor.org/info/rfc3110>>.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <<https://www.rfc-editor.org/info/rfc4033>>.
- [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>>.
- [RFC6986] Dolmatov, V., Ed. and A. Degtyarev, "GOST R 34.11-2012: Hash Function", RFC 6986, DOI 10.17487/RFC6986, August 2013, <<https://www.rfc-editor.org/info/rfc6986>>.

- [RFC7091] Dolmatov, V., Ed. and A. Degtyarev, "GOST R 34.10-2012: Digital Signature Algorithm", RFC 7091, DOI 10.17487/RFC7091, December 2013, <<https://www.rfc-editor.org/info/rfc7091>>.
- [RFC7836] Smyshlyaev, S., Ed., Alekseev, E., Oshkin, I., Popov, V., Leontiev, S., Podobaev, V., and D. Belyavsky, "Guidelines on the Cryptographic Algorithms to Accompany the Usage of Standards GOST R 34.10-2012 and GOST R 34.11-2012", RFC 7836, DOI 10.17487/RFC7836, March 2016, <<https://www.rfc-editor.org/info/rfc7836>>.
- [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>>.

10.2. Informative References

- [RFC4509] Hardaker, W., "Use of SHA-256 in DNSSEC Delegation Signer (DS) Resource Records (RRs)", RFC 4509, DOI 10.17487/RFC4509, May 2006, <<https://www.rfc-editor.org/info/rfc4509>>.
- [RFC5933] Dolmatov, V., Ed., Chuprina, A., and I. Ustinov, "Use of GOST Signature Algorithms in DNSKEY and RRSIG Resource Records for DNSSEC", RFC 5933, DOI 10.17487/RFC5933, July 2010, <<https://www.rfc-editor.org/info/rfc5933>>.

Authors' Addresses

Dmitry Belyavskiy
TCINET
8 marta st
Moscow
Russian Federation

Phone: +7 916 262 5593
Email: beldmit@gmail.com

Vasily Dolmatov (editor)
JSC "NPK Kryptonite"
Spartakovskaya sq., 14, bld 2, JSC "NPK Kryptonite"
Moscow 105082
Russian Federation

Email: vdolmatov@gmail.com

DNSOP WG
Internet-Draft
Intended status: Standards Track
Expires: April 16, 2021

T. Reddy
McAfee
N. Cook
Open-Xchange
D. Wing
Citrix
M. Boucadair
Orange
October 13, 2020

DNS Access Denied Error page
draft-reddy-dnsop-error-page-05

Abstract

When a DNS server filters a query, the response conveys no detailed explanation of why that query was blocked, leading thus to end-user confusion. A solution is needed to enhance the user experience.

This document defines a method to return an URI that explains the reason why a DNS query was filtered.

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 April 16, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	5
3. Error page URI EDNS0 option format	5
4. Error page URI Processing	7
5. Sign and Verify	9
6. ERROR Page	9
7. Usability Considerations	10
8. Security Considerations	10
9. IANA Considerations	11
9.1. A New Error Page URI EDNS Option	12
10. Acknowledgements	12
11. References	12
11.1. Normative References	12
11.2. Informative References	14
Authors' Addresses	15

1. Introduction

DNS filters are deployed for a variety of reasons including endpoint security, parental filtering, and filtering required by law enforcement. These are discussed in more detail below:

- o Various network security services are provided by Enterprise networks to protect endpoints (e.g., Hosts including IoT devices). Network-based security solutions such as firewalls and Intrusion Prevention Systems (IPS) rely on network traffic inspection to implement perimeter-based security policies. The network security services may, for example, prevent malware download, block known malicious domains, block phishing sites, etc. These network security services act on DNS queries originating from endpoints. For example, DNS firewalls, a method of expressing DNS response policy information inside specially constructed DNS zones, known as Response Policy Zones (RPZs) allows DNS servers to modify DNS responses in real time to stop access to malware and phishing domains. Note that some of the commonly known types of malware are viruses, worms, trojans, bots, ransomware, backdoors, spyware, and adware.

- o Network devices in a home network offer network security to protect the devices connected to the home network by performing DNS-based content filtering. The network security service may, for example, block access to specific domains to enforce parental control, block access to malware sites, etc.
- o ISPs typically block access to some domains due to a requirement imposed by an external entity (e.g., Law Enforcement Agency) by performing DNS-based content filtering.

DNS responses can be filtered by sending a bogus ("forged") A or AAAA response, NXDOMAIN error or empty answer, or an extended error code defined in [I-D.ietf-dnsop-extended-error]. Each of these have advantages and disadvantages, discussed below:

1. The DNS response is forged providing IP addresses that point to a HTTP(S) server alerting the end user of the reason for blocking access to the domain (e.g., malware). When a HTTP(S) enabled domain name is blocked, the network security device presents a block page instead of the HTTP response from the content provider. If an HTTP enabled domain name is blocked, the network security device intercepts the HTTP request and returns a block page over HTTP. If an HTTPS enabled domain is blocked, the block page is also served over HTTPS. In order to return a block page over HTTPS, man in the middle (MITM) is enabled on endpoints by generating a local root certificate and an accompanying (local) public/private key pair. The local root certificate is installed on the endpoint, and the network security device(s) store a copy of the private key. During the TLS handshake, the network security device modifies the certificate provided by the server and (re)signs it with the private key from the local root certificate.
 - * However, configuring the local root certificate on endpoints is not viable option in several deployments like Home networks, Schools, Small Office/Home Office (SOHO), and Small/Medium Enterprise (SME). In these cases, the typical behavior is that the forged DNS response directs the user towards a server hosted to display the block page which breaks the TLS connection. For web-browsing this then results in an HTTPS certificate error message indicating that a secure connection could not be established, which gives no information to the end-user about the reason for the error. The typical errors are "The security certificate presented by this website was not issued by a trusted certificate authority" (Internet Explorer/Edge), "The site's security certificate is not trusted" (Chrome), "This Connection is Untrusted" (Firefox),

"Safari can't verify the identity of the website..." (Safari on MacOS)".

- * Enterprise networks do not assume that all the devices connected to their network are managed by the IT team or Mobile Device Management (MDM) devices, especially in the quite common BYOD ("Bring Your Own Device") scenario. In addition, the local root certificate cannot be installed on IoT devices without a device management tool.
 - * An end user does not know why the connection was reset and, consequently, may repeatedly try to unsuccessfully reach the domain. Frustrated, the end user may use insecure interfaces to reach the domain, potentially compromising both security and privacy. Furthermore, certificate errors train users to click through certificate errors, which is poor security practice. To eliminate the need for an end user to click through certificate errors, an end user may manually install a local root certificate [Chrome-Install-Cert] on a host device. Doing so, however, is also poor security practice as it creates a security vulnerability that may be exploited by a MITM attack. When the manually installed local root certificate expires, the user has to (again) manually install the new local root certificate.
2. The DNS response is forged to provide a NXDOMAIN response to cause the DNS lookup to terminate in failure. In this case, an end user does not know why the domain cannot be reached, and may repeatedly try to unsuccessfully reach the domain. Frustrated, the end user may use insecure interfaces to reach the domain, potentially compromising both security and privacy.
 3. The extended error codes Blocked, Censored, and Filtered defined in [I-D.ietf-dnsop-extended-error] can be returned by the DNS server to provide additional information about the cause of an DNS error. If the extended error code "Forged answer" defined in [I-D.ietf-dnsop-extended-error] is returned by the DNS server, the client can identify the DNS response is forged and the reason for HTTPS certificate error. These extended error codes do not suffer from the limitations discussed in (1) and (2) but the user still does not know the exact reason nor the user is aware of the exact entity blocking the access to the domain. For example, a DNS server may block access domain based on the content category like "Adult Content" to enforce parental control, "Violence & Terrorism" due to an external requirement imposed by an external entity (e.g., Law Enforcement Agency), etc. The content categories for domains cannot be standardized because the classification of domains into content categories is vendor

specific, typically ranges from 40 to 100 types of categories depending on the vendor and the categories keep evolving. Further, the threat data used to categorize domains may sometimes mis-classify domains (e.g., Domains wrongly classified as DGA (Domain Generation Algorithm) by deep learning techniques, domain wrongly classified as phishing due to crowd sourcing, new domains not categorized by the threat data). The end user needs to know the contact details of the IT/InfoSec team to raise a complaint.

No matter which type of response is generated (forged IP address, NXDOMAIN or empty answer, or an extended error code), the user who generated the query has little chance to understand which entity filtered the query, how to report a mistake in the filter, or why the entity filtered it at all. This document describes a mechanism to provide a URI which, when accessed, provides such information to the user.

One of the other benefits of this approach is to eliminate the need to "spoof" block pages for HTTPS resources, as the block page no longer needs to create a signed certificate when blocking a destination. This avoids the need to install a local root certificate authority on those IT-managed devices.

2. Terminology

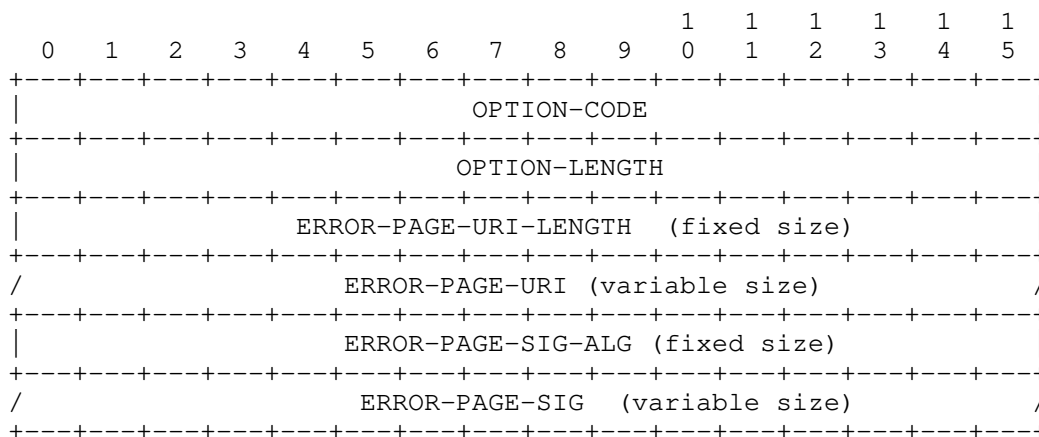
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

This document makes use of the terms defined in [RFC8499] and [I-D.ietf-dnsop-terminology-ter].

'Encrypted DNS' refers to DNS-over-HTTPS [RFC8484], DNS-over-TLS [RFC7858], or DNS-over-QUIC [I-D.ietf-dprive-dnsquic].

3. Error page URI EDNS0 option format

This document uses an EDNS0 [RFC6891] option to include the URI that gives additional information in a DNS response about the cause of blocking access to a domain. The option is structured as follows:



The description of the fields is as follows:

- o OPTION-CODE: TBD, indicates the code assigned for Error page URI (Section 6.1.2 of [RFC6891]). [RFC Editor: change TBD to the proper code once assigned by IANA.]
- o OPTION-LENGTH: See Section 6.1.2 of [RFC6891]. This field contains the length of the payload (everything after OPTION-LENGTH) in octets. The variability of the option length stems from the variable-length ERROR-PAGE-URI and ERROR-PAGE-SIG fields.
- o ERROR-PAGE-URI-LENGTH: This 16-bit field indicates the length of ERROR-PAGE-URI. It MUST NOT be set to 0.
- o ERROR-PAGE-URI: a variable length UTF-8 encoded [RFC5198] text field containing the URI Template [RFC6570] that gives additional information about the cause of blocking access to a domain. The ERROR-PAGE-URI field MUST NOT be zero octets in length.
- o ERROR-PAGE-SIG-ALG: A 16-bits field that contains the algorithm used to generate the signature for the Error page URI Template. The values are defined in the TLS SignatureScheme [TLS-SIG-SCHEME] with limitations described in Section 5.
- o ERROR-PAGE-SIG, a variable length field containing the signature of the Error page URI Template. The signature generation process is discussed in Section 5.

The Error page URI option can be included in any response (SERVFAIL, NXDOMAIN, REFUSED, and even NOERROR, etc) to a query that includes OPT Pseudo-RR [RFC6891].

The URI Template defined in ERROR-PAGE-URI describes how to construct the URL to fetch the error page. The agent acting as HTTPS client on the endpoint encodes a FQDN to which access is denied into an HTTP GET request to retrieve the error page. The HTTPS server returning the error page defines the URI used by the HTTP GET request through the use of a URI Template. The URI Template is processed with a defined variable "target-domain" whose value is set to the FQDN to which access is denied. The FQDN is encoded using base64url [RFC4648] and then provided as the variable value for "target-domain" to expand the URI Template into a URI reference in the HTTP GET request. Padding characters for base64url MUST NOT be included.

An example is illustrated below:

If the URI Template is "https://block.example.net/block-page{?target-domain}" for the HTTPS server returning the error page and access to the target domain "example.com" is blocked by the encrypted DNS server, the variable "target-domain" has the value "example.com" base64url encoded into an HTTP GET request. In the above example, the expansion of the above URI Template is "https://block.example.net/block-page?target-domain=ZXhhbXBsZS5jb20".

HTTP/2 [RFC7540] is the minimum RECOMMENDED version of HTTP to use to retrieve the error page. The HTTPS client retrieving the error page MUST verify the entire certification path per [RFC5280]. The HTTPS client additionally uses validation techniques as described in [RFC6125] to compare the domain name in the error page URI to the server certificate provided in TLS handshake. See [RFC7525] for additional TLS recommendations.

4. Error page URI Processing

The DNS client MUST follow the following rules to process the Error page URI EDNS0 option:

- o The Error page URI EDNS0 option is susceptible to forgery. In order to defend against this attack the DNS client MUST NOT process the DNS response with Error page URI EDNS0 option unless DNS messages exchanged are cryptographically protected using encrypted DNS.
- o If an DNS client has enabled opportunistic privacy profile (Section 5 of [RFC8310]) for DoT, the DNS client will either fallback to an encrypted connection without authenticating the DNS server provided by the local network or fallback to clear text DNS, and cannot exchange encrypted DNS messages. The fallback adversely impacts security and privacy. If the DNS client has

enabled opportunistic privacy profile for DoT, the client MUST NOT process the DNS response with Error page URI EDNS0 option.

- o If an DNS client has enabled strict privacy profile (Section 5 of [RFC8310]) for DoT, the DNS client requires an encrypted connection and successful authentication of the DNS server; this mitigates both passive eavesdropping and client redirection (at the expense of providing no DNS service if an encrypted, authenticated connection is not available). If the DNS client has enabled strict privacy profile for DoT, the client can process the DNS response with Error page URI EDNS0 option. Note that the strict and opportunistic privacy profiles as defined in [RFC8310] only applies to DoT protocol, there has been no such distinction made for DoH protocol.
- o If the DNS response contains more than one Error page URI EDNS0 option, the DNS client MUST discard all Error page URI EDNS0 options in the DNS response.
- o The Error page URI EDNS0 option MUST be processed by the DNS client for a "Censored", "Blocked", "Filtered" or "Forged" extended error codes and MUST be ignored for any other type of extended DNS error code. When "Censored", "Blocked", "Filtered" or "Forged" extended error code is returned in conjunction with an Error page URI EDNS0 option, any other resource records in the answer MUST be ignored by clients supporting this specification.
- o If the DNS client determines that the encrypted DNS server does not offer DNS filtering service, it MUST reject the Error page URI EDNS0 option. For example, the DNS client knows whether the pre-configured encrypted DNS resolver performs DNS-based content filtering or not.
- o The DNS client MUST reject the error page URI if the scheme is not "https".
- o The DNS client verifies the signature in the ERROR-PAGE-SIG field following the mechanism discussed in Section 5. If the signature is valid, the client can positively identify that the Error page URI EDNS0 option has been generated by the encrypted DNS server and the encrypted DNS server did not forward the Error Page URI EDNS0 option from an upstream resolver. If signature validation fails, the DNS client MUST reject the Error page URI EDNS0 option.

Because EDNS is a hop-by-hop extension to DNS and EDNS responses are not cached (Section 6.2.1 of [RFC6891]), a DNS resolver MUST NOT propagate a received Error Page URI EDNS0 option, because an attacker could insert a bogus URI; instead, if access to a domain is denied,

the DNS resolver MUST generate its own Error Page URI. Because DNS forwarders (or DNS proxies) are supposed to propagate unknown EDNS0 options (Section 4.1 and Section 4.4.1 of [RFC5625]), the Error Page URI EDNS0 option may get propagated. To detect both of these scenarios, the Error Page URI Template is protected with an object signature as described in Section 5.

5. Sign and Verify

The algorithms for generating signature for DNS resource record sets (RRsets) are defined in [DNSKEY-IANA]. The "mandatory-to-implement" algorithms are RSA, Elliptic Curve Digital Signature Algorithm (ECDSA), and Edwards-curve Digital Security Algorithm (EdDSA) [RFC8624]. Along similar lines, the encrypted DNS server's end-entity certificate's public key and the signature algorithm with which the key can be used are RSA, ECDSA, and EdDSA [RFC8446]. If ECDSA is used, it is RECOMMENDED to use the deterministic digital signature generation procedure of the ECDSA, specified in [RFC6979].

The signature is generated by the encrypted DNS server using the Error page URI Template, private key of the encrypted DNS server's end-entity certificate as inputs to the signature algorithm. The signature algorithm in the ERROR-PAGE-SIG-ALG field MUST be compatible with the key in the DNS server's end-entity certificate. The implementation MUST support the same set of algorithms in the TLS client for validating the signature in the CertificateVerify message from the server in the TLS handshake and in the DNS client to validate the signature for the Error page URI Template. As a reminder, the server's end-entity certificate's public key will be compatible with the selected authentication algorithm from the client's "signature_algorithms" TLS extension (Section 4.4.2.2 of [RFC8624]).

If the signature algorithm in the ERROR-PAGE-SIG-ALG field is not compatible with the key in the DNS server's end-entity certificate, the DNS client MUST reject the Error page URI EDNS0 option. The DNS client verifies the signature using the signature in the ERROR-PAGE-SIG field, error page URI Template and DNS server's end-entity certificate's public key as inputs to the signature algorithm. For example, if Ed25519 is used, Ed25519 signature algorithm and verification of the Ed25519 signature are described in Sections 5.1.6 and 5.1.7 of [RFC8032], respectively.

6. ERROR Page

The following text outlines the RECOMMENDED contents of an error page to assist the operator developing the error page.

- o The exact reason for blocking access to the domain. If the domain is blocked based on some threat data, the threat type associated with the blocked domain can be provided/displayed to the end user. For example, the reason can indicate the type of malware blocked like spyware and the damage it can do the security and privacy of the user.
- o The domain name blocked.
- o If query was blocked by regulation, a pointer to a regulatory text that mandates this query block.
- o The entity (or organization) blocking the access to the domain and contact details of the IT/InfoSec team to raise a complaint.
- o The blocked error page to not include Ads and dynamic content.

The content of the error page discussed above is non-normative, the above text only provides the guidelines and template for the error page and.

- o Does not attempt to offer an exhaustive list for the contents of an error page.
- o It is not intended to form the basis of any legal/compliance for developing the error page.

7. Usability Considerations

The error page SHOULD be returned in the user's preferred language as expressed by the Accept-Language header. If the error page is displayed in a language not known to the end user and assuming Internationalization features failed, browser extensions to translate to user's native language can be used. For example, "Google Translate" extension [Chrome-Translate] provided by Google on Chrome can be used by the user to translate the error page. The "Google Translate" extension automatically detects whether the language of a page is different from the language the user has selected. If it is in a different language, a banner appears at the top of the page. The user can click on the Translate button in the banner to have all the text on the page appear in the language selected by the user.

8. Security Considerations

Security considerations in [I-D.ietf-dnsop-extended-error] and [RFC8624] need to be taken into consideration.

The Error page URI EDNS0 option causes an HTTPS retrieval by the client. To prevent forgery of the Error page URI EDNS0 option, this specification requires it only be sent only over an encrypted DNS channel with an authorized DNS server.

The client knows it is connecting to a HTTPS server returning the error page. To reduce threat surface the client can retrieve the Error page URL using, for example, an isolated environment and take other precautions such as clearly labeling the page as untrusted or prevent user interaction with the page. Such isolation should prevent transmitting cookies, block JavaScript, block auto-fill of credentials or personal information, and be isolated from the user's normal environment.

Browsers perform some of the above restrictions when accessing captive portals (Section 5 of [RFC8910] or [Safari-Cookie]), during private browsing, or using containerization [Facebook-Container].

Note that the means to use a sandbox environment and a user interface presenting the error page are not covered in this document. By its nature, these aspects are implementation specific and best left to the application and user interface designers.

The encrypted DNS session provides transport security for the interaction between the DNS client and server, but DNSSEC signing and validation is not possible for the Error Page URI EDNS0 option returning the error page URI template. For example, if access to a domain is blocked, the encrypted DNS resolver can rewrite the response to send NXDOMAIN error and Error page URI EDNS0 option in the DNS response, it will omit DNSSEC RRsets, because the modified responses cannot be verified by DNSSEC signatures. However, the signature in the Error Page URI EDNS0 option provides data origin authentication of the Error Page URI EDNS0 option. Nevertheless, the Error Page URI EDNS0 option returning the error page URI Template should be treated only as diagnostic information and MUST NOT alter DNS protocol processing.

By design, the object referenced by the error page URL potentially exposes additional information about the DNS resolution process that may leak information. An example of this is the reason for blocking the access to the domain name and the entity blocking access to the domain.

9. IANA Considerations

9.1. A New Error Page URI EDNS Option

This document defines a new EDNS(0) option, entitled "Error Page URI", assigned a value of TBD from the "DNS EDNS0 Option Codes (OPT)" registry [to be removed upon publication:
<http://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-11>]

Value	Name	Status	Reference
TBD	Error Page URI	Standard	[This document]

10. Acknowledgements

Thanks to Vittorio Bertola, Wes Hardaker, Ben Schwartz, Erid Orth, Viktor Dukhovni, Warren Kumari and Bob Harold for the comments.

11. References

11.1. Normative References

- [I-D.ietf-dnsop-extended-error]
 Kumari, W., Hunt, E., Arends, R., Hardaker, W., and D. Lawrence, "Extended DNS Errors", draft-ietf-dnsop-extended-error-16 (work in progress), May 2020.
- [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>>.
- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <<https://www.rfc-editor.org/info/rfc4648>>.
- [RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, DOI 10.17487/RFC5198, March 2008, <<https://www.rfc-editor.org/info/rfc5198>>.
- [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/info/rfc5280>>.
- [RFC5625] Bellis, R., "DNS Proxy Implementation Guidelines", BCP 152, RFC 5625, DOI 10.17487/RFC5625, August 2009, <<https://www.rfc-editor.org/info/rfc5625>>.

- [RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March 2011, <<https://www.rfc-editor.org/info/rfc6125>>.
- [RFC6570] Gregorio, J., Fielding, R., Hadley, M., Nottingham, M., and D. Orchard, "URI Template", RFC 6570, DOI 10.17487/RFC6570, March 2012, <<https://www.rfc-editor.org/info/rfc6570>>.
- [RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <<https://www.rfc-editor.org/info/rfc6891>>.
- [RFC6979] Pornin, T., "Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)", RFC 6979, DOI 10.17487/RFC6979, August 2013, <<https://www.rfc-editor.org/info/rfc6979>>.
- [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <<https://www.rfc-editor.org/info/rfc7525>>.
- [RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, DOI 10.17487/RFC7540, May 2015, <<https://www.rfc-editor.org/info/rfc7540>>.
- [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>>.
- [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>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

[RFC8624] Wouters, P. and O. Sury, "Algorithm Implementation Requirements and Usage Guidance for DNSSEC", RFC 8624, DOI 10.17487/RFC8624, June 2019, <<https://www.rfc-editor.org/info/rfc8624>>.

[TLS-SIG-SCHEME]
"IANA, TLS SignatureScheme",
<<https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml#tls-signaturescheme>>.

11.2. Informative References

[Chrome-Install-Cert]
"How to manually install the Securly SSL certificate in Chrome", <support.securly.com/hc/en-us/articles/206081828-How-to-manually-install-the-Securly-SSL-certificate-in-Chrome>.

[Chrome-Translate]
"Google Translate",
<https://chrome.google.com/webstore/detail/google-translate/aapbdbdomjkkjkaonfhkkikfgjllcleb/RK%3D2/RS%3DBBFW_pnWkPY0xPMYsAZI5xOgQEE->.

[DNSKEY-IANA]
"IANA, Domain Name System Security (DNSSEC) Algorithm Numbers",
<<http://www.iana.org/assignments/dns-sec-alg-numbers>>.

[Facebook-Container]
"Facebook container for Firefox",
<<https://www.mozilla.org/en-US/firefox/facebookcontainer/>>.

[I-D.ietf-dnsop-terminology-ter]
Hoffman, P., "Terminology for DNS Transports and Location", draft-ietf-dnsop-terminology-ter-02 (work in progress), August 2020.

[I-D.ietf-dprive-dnsquic]
Huitema, C., Mankin, A., and S. Dickinson, "Specification of DNS over Dedicated QUIC Connections", draft-ietf-dprive-dnsquic-00 (work in progress), April 2020.

[RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <<https://www.rfc-editor.org/info/rfc7858>>.

- [RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <<https://www.rfc-editor.org/info/rfc8032>>.
- [RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <<https://www.rfc-editor.org/info/rfc8484>>.
- [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>>.
- [RFC8910] Kumari, W. and E. Kline, "Captive-Portal Identification in DHCP and Router Advertisements (RAs)", RFC 8910, DOI 10.17487/RFC8910, September 2020, <<https://www.rfc-editor.org/info/rfc8910>>.
- [Safari-Cookie] "Isolated cookie store (CVE-2016-1730)", <<https://support.apple.com/en-us/HT205732>>.

Authors' Addresses

Tirumaleswar Reddy
McAfee, Inc.
Embassy Golf Link Business Park
Bangalore, Karnataka 560071
India

Email: kondtir@gmail.com

Neil Cook
Open-Xchange
UK

Email: neil.cook@noware.co.uk

Dan Wing
Citrix Systems, Inc.
USA

Email: dwing-ietf@fuggles.com

Mohamed Boucadair
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com