

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: 28 January 2022

S. Huque
Salesforce
27 July 2021

Empty Non-Terminal Sentinel for Black Lies
draft-huque-dnsop-blacklies-ent-01

Abstract

The Black Lies method of providing compact DNSSEC denial of existence proofs has some operational implications. Depending on the specific implementation, it may provide no way to reliably distinguish Empty Non-Terminal names from names that actually do not exist. This draft describes the use of a synthetic DNS resource record type to act as an explicit signal for Empty Non-Terminal names and which is conveyed in an NSEC type bitmap.

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 28 January 2022.

Copyright Notice

Copyright (c) 2021 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 and Motivation	2
2. Synthetic Type for Empty Non-Terminal Names	3
3. Status of Black Lies specification	4
4. Implementation Status	5
5. Acknowledgements	5
6. IANA Considerations	5
7. Security Considerations	5
8. References	5
8.1. Normative References	5
8.2. Informative References	5
Author's Address	5

1. Introduction and Motivation

One of the functions of the Domain Name System Security Extensions (DNSSEC) [RFC4033] [RFC4034] [RFC4035] is "Authenticated Denial of Existence", i.e. proving that a DNS name or record type does not exist. Normally, this is done by means of NSEC or NSEC3 records. In the precomputed signature model, these records chain together existing names, or cryptographic hashes of them in the zone. In the online signing model, they are used to dynamically compute an epsilon function around the queried name. A 'type bitmap' in the data field of the NSEC or NSEC3 record asserts which resource record types are present at the associated name.

An alternative method, Black Lies [BLACKLIES], described in an expired Internet draft, provides more compact denial of existence proofs for online signers by relying on a clever hack. For non-existent names, it claims that the name exists, but has no resource records associated with the queried type, i.e. it returns a NODATA response rather than an NXDOMAIN response. A NODATA response (which has a response code of NOERROR, and an empty ANSWER section) requires only one NSEC record matching the queried name. This has two advantages: the DNS response sizes are smaller, and it reduces the online cryptographic work involved in generating the responses. By contrast, an NXDOMAIN response requires multiple records (up to 2 when using NSEC, and up to 3 when using NSEC3) to prove that (1) the name did not explicitly exist in the zone, and (2) that it could not have been synthesized by a wildcard.

The Black Lies method has some operational implications. Tools that rely on the correctness of the DNS response code (e.g. obtaining NXDOMAIN for non-existent domains) no longer work. Arguably, we should not be doing this anyway, since the response code in the DNS header cannot be authenticated. This means that NXDOMAIN has to be "inferred" from signed records in the DNS response. Whether this inference can be reliably drawn depends on other details of the Black Lies implementation. A Black Lies NODATA response contains only "NSEC" and "RRSIG" in the NSEC type bitmap. This is not sufficient to infer NXDOMAIN though, because Empty Non-Terminal (ENT) responses (which positively exist) will return the exact same response. DNS operators often rely on precisely distinguishing NXDOMAIN from NODATA, including ENT responses (such as tools that prevent the creation of zone cuts or DNAME records at ENTs to avoid accidentally occluding names underneath them - these have been critical safety features of our DNS record provisioning systems).

Of the 3 implementations I've examined, NS1 (previously) and Amazon Route53 suffer from this NXDOMAIN/ENT indistinguishability. Cloudflare avoids this problem by synthesizing the NSEC type bitmap for ENTs to include all (?) RR Types they support, except for the queried type. This has the side effect though of no longer being able to reliably determine the existence of ENTs.

2. Synthetic Type for Empty Non-Terminal Names

This document proposes the use of a synthetic Resource Record type to signal the presence of an Empty Non-Terminal name. This RR type is added to the NSEC type bitmap for responses to ENTs. Currently, the deployed examples of this scheme are using the private RR type code 65281. So the resulting type bitmap would have "NSEC RRSIG TYPE65281". Should this document be published, a formal request for an RR type number could be made.

NS1 has implemented this scheme in their Managed DNS platform. The following is an example of a response to an Empty Non-Terminal name hosted on their service:

```
$ dig +dnssec +multi ent1.sfdcsd.net. A

; <<>> DiG 9.16.15 <<>> +dnssec +multi ent1.sfdcsd.net. A
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53091
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 1232
;; QUESTION SECTION:
;ent1.sfdcsd.net.          IN A

;; AUTHORITY SECTION:
ent1.sfdcsd.net.          3592 IN RRSIG NSEC 13 3 3600 (
                           20210712120255 20210710120255 44688 sfdcsd.net.
                           1G/EJq0M1cs6vw0ragtvMV+B/Sd2CAPsxo1/WIOT6BZt
                           /QxukD5k8AeygmWYKnrR9jdb2SnXBxFEQss/mTSAOw== )
ent1.sfdcsd.net.          3592 IN NSEC \000.ent1.sfdcsd.net. RRSIG NSEC TYPE65281
sfdcsd.net.               3592 IN SOA dns1.p08.nsone.net. hostmaster.nsone.net. (
                           1619363158 ; serial
                           43200      ; refresh (12 hours)
                           7200       ; retry (2 hours)
                           1209600    ; expire (2 weeks)
                           3600       ; minimum (1 hour)
                           )
sfdcsd.net.               3592 IN RRSIG SOA 13 2 3600 (
                           20210712120255 20210710120255 44688 sfdcsd.net.
                           m2J7Q6mk6Y8lNxXEWNw2/cVJPIeHZMAAeYglTgyob3s
                           mXV5hTtOpydytWFynIjdKf8YeGOpZm3zqoyLyPgMbg== )
```

3. Status of Black Lies specification

Despite the fact that Black Lies is not standardized or even formally published as a protocol specification, it seems to be gaining in popularity and deployment. At least 3 major DNS providers (Cloudflare, NS1 and Amazon Route53) have deployed it. Due to the fact that Black Lies relies on contorting existing semantics of the DNS protocol, it seems unlikely that it could be published as a "Standards Track" specification. But given deployment realities, it seems desirable to have a stable specification published for it, even if its status is Informational.

4. Implementation Status

NS1 has implemented the scheme described in this document. Example code to infer NXDOMAIN from Black Lies NODATA responses can be found here: <https://github.com/shuque/blrcode>

5. Acknowledgements

Jan Vcelak of NS1.

6. IANA Considerations

TBD based on DNSOP working group deliberations.

7. Security Considerations

The method proposed in this document addresses a potential security issue, namely reliably determining NXDOMAIN in Black Lies implementations.

8. References

8.1. Normative 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>>.
- [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>>.

8.2. Informative References

- [BLACKLIES] Valsorda, F. and O. Gudmundsson, "Compact DNSSEC Denial of Existence or Black Lies", <<https://tools.ietf.org/html/draft-valsorda-dnsop-black-lies>>.

Author's Address

Shumon Huque
Salesforce
415 Mission Street, 3rd Floor
San Francisco, CA 94105
United States of America

Email: shuque@gmail.com

Network Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: 27 June 2022

K. Fujiwara
JPRS
P. Vixie
none
24 December 2021

Fragmentation Avoidance in DNS
draft-ietf-dnsop-avoid-fragmentation-06

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 June 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

Table of Contents

1. Introduction	2
2. Terminology	3
3. Proposal to avoid IP fragmentation in DNS	3
3.1. Recommendations for UDP responders	4
3.2. Recommendations for UDP requestors	4
3.3. Default Maximum DNS/UDP payload size	4
4. Incremental deployment	6
5. Request to zone operators and DNS server operators	6
6. Considerations	6
6.1. Protocol compliance	6
7. IANA Considerations	7
8. Security Considerations	7
9. Acknowledgments	7
10. References	7
10.1. Normative References	7
10.2. Informative References	8
Appendix A. Weaknesses of IP fragmentation	9
Appendix B. Details of maximum DNS/UDP payload size discussions	10
Appendix C. How to retrieve path MTU value to a destination from applications	11
Appendix D. How to retrieve minimal MTU value to a destination	11
Appendix E. Minimal-responses	11
Authors' Addresses	12

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.

Fragmented DNS UDP responses have systemic weaknesses, which expose the requestor to DNS cache poisoning from off-path attackers. (See Appendix A for references and details.)

[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 messages 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].

IP_DONTFRAG option is not defined by any RFCs. It is similar to IPV6_DONTFRAG option defined in [RFC3542]. IP_DONTFRAG option is used on BSD systems to set the Don't Fragment bit [RFC0791] when sending IPv4 packets. On Linux systems this is done via IP_MTU_DISCOVER and IP_PMTUDISC_DO.

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.
- * UDP responders SHOULD compose UDP responses that result in IP packets that do not exceed the path MTU to the requestor. If the path MTU discovery failed or is impossible, UDP responders SHOULD compose UDP responses that result in IP packets that do not exceed the default maximum DNS/UDP payload size described in Section 3.3.

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

3.2. Recommendations for UDP requestors

- * UDP requestors SHOULD send DNS requests with IP_DONTFRAG / IPV6_DONTFRAG [RFC3542] options.
- * UDP requestors MAY probe to discover the real MTU value per destination. Then, calculate their maximum DNS/UDP payload size as the reported path MTU minus IPv4/IPv6 header size (20 or 40) minus UDP header size (8). If the path MTU discovery failed or is impossible, use the default maximum DNS/UDP payload size described in Section 3.3.
- * UDP requestors SHOULD use the requestor's payload size as the calculated or the default maximum DNS/UDP payload size.
- * 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.

3.3. Default Maximum DNS/UDP payload size

Fragmentation avoidance is achieved with the IP(V6)_DONTFRAG option. The purpose of packet size limitation is to decrease packet loss due to the effects of the IP(V6)_DONTFRAG option.

Default maximum DNS/UDP payload size depends on the connectivity of each node, it cannot be determined unconditionally. However, there are good proposed values.

Operators MAY select a good number from Table 1. Details of proposed values are described in Appendix B.

Source	IPv4	IPv6
Minimal: RFC 4035 MUST	1220	1220
Software developers / DNSFlagDay2020 propose	1232	1232 (1280-40-8)
Authors' recommendation	1400	1400 (1500 -40 -8 - some headers)
Maximum: Ethernet MTU 1500 [Huston2021]	1472 (1500-20-8)	1452 (1500-40-8)
Measured	MTU -20-8	MTU -40-8

Table 1: Default maximum DNS/UDP payload size

However, operators of DNS servers SHOULD measure their path MTU to the Internet at setting up DNS servers (and when network configuration changes).

How to measure path MTU is described in Appendix D.

Operators of authoritative servers (that offer global DNS zones) and full-service resolvers (that access authoritative servers of the global DNS) SHOULD measure their path MTU to well-known locations on the Internet, such as [a-m].root-servers.net or [a-m].gtld-servers.net.

Operators of full-service resolvers 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. Or, operators of ISPs know their customers' connectivity and customers' MTU to ISPs' servers. 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.

DNS clients (stub resolvers) need to specify an appropriate requestor's payload size when supporting EDNS0. In case of CPEs, embedded devices, and user devices, network operators can not control them, developers may choose small values such as 1220 and 1232.

Other DNS servers are out-of-scope of this document. (For example, Forwarding only resolvers, or private DNS).

4. 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.

5. 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 E).
- * Use smaller signature / public key size algorithm for DNSSEC. Notably, the signature size of ECDSA or EdDSA is smaller than RSA.

6. Considerations

6.1. 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.

7. IANA Considerations

This document has no IANA actions.

8. Security Considerations

9. Acknowledgments

The author would like to specifically thank Paul Wouters, Mukund Sivaraman, Tony Finch, Hugo Salgado, Peter van Dijk, Brian Dickson, Puneet Sood and Jim Reid for extensive review and comments.

10. References

10.1. Normative References

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.
- [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>>.
- [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>>.
- [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>>.

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

10.2. 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>>.
- [Huston2021] Huston, G. and J. Damas, "Measuring DNS Flag Day 2020", OARC 34 Workshop , February 2021.
- [RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <<https://www.rfc-editor.org/info/rfc1122>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.

Appendix A. Weaknesses of IP fragmentation

"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.

Appendix B. Details of maximum DNS/UDP payload size discussions

There are many discussions for default path MTU size and maximum DNS/UDP payload size.

- * 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. The corresponding minimum MTU for an IPv4 interface is 68 (60 + 8) [RFC0791].
- * Most of the Internet and especially the inner core has an MTU of at least 1500 octets. Maximum DNS/UDP payload size for IPv6 on MTU 1500 ethernet is 1452 (1500 minus 40 (IPv6 header size) minus 8 (UDP header size)). To allow for possible IP options and distant tunnel overhead, authors' recommendation of default maximum DNS/UDP payload size is 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.

- * [Huston2021] analyzed the result of [DNSFlagDay2020], reported that their measurements suggest that in the interior of the Internet between recursive resolvers and authoritative servers the prevailing MTU is at 1,500 and there is no measurable signal of use of smaller MTUs in this part of the Internet, and proposed that their measurements suggest setting the EDNS0 Buffer size to IPv4 1472 octets and IPv6 1452 octets.

Appendix C. 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))

Section 3.4 of [RFC1122] specifies FIND_MAXSIZES() as one of "INTERNET/TRANSPORT LAYER INTERFACES".

Appendix D. How to retrieve minimal MTU value to a destination

The Linux tool "tracert" can be used to measure the path MTU to a destination.

Or, "ping/ping6" command with "-D" Don't Fragment bit set / Disable IPv6 fragmentation options.

Appendix E. 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
none
11400 La Honda Road
Woodside, CA, 94062
United States of America

Phone: +1 650 393 3994
Email: paul@redbarn.org

Network Working Group
Internet-Draft
Updates: 5155, 6014, 8624 (if approved)
Intended status: Standards Track
Expires: 10 April 2022

P. Hoffman
ICANN
7 October 2021

Revised IANA Considerations for DNSSEC
draft-ietf-dnsop-dnssec-iana-cons-05

Abstract

This document changes the review requirements needed to get DNSSEC algorithms and resource records added to IANA registries. It updates RFC 6014 to include hash algorithms for DS (Delegation Signer) records and NSEC3 (Hashed Authenticated Denial of Existence) parameters. It also updates RFC 5155 and RFC 6014, which have requirements for DNSSEC algorithms, and updates RFC 8624 to say that algorithms that are described in RFCs that are not on standards track are only at the "MAY" level of implementation recommendation. The rationale for these changes is to bring the requirements for DS records and for the hash algorithms used in NSEC3 in line with the requirements for all other DNSSEC algorithms.

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 10 April 2022.

Copyright Notice

Copyright (c) 2021 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. Update to RFC 6014	3
3. Update to RFC 8624	3
4. IANA Considerations	3
5. Security Considerations	4
6. References	4
6.1. Normative References	4
6.2. Informative References	5
Appendix A. Acknowledgements	5
Author's Address	5

1. Introduction

DNSSEC is primarily described in [RFC4033], [RFC4034], and [RFC4035]. DNSSEC commonly uses another resource record beyond those defined in RFC 4034: NSEC3 [RFC5155]. DS resource records were originally defined in [RFC3658], and that definition was obsoleted by RFC 4034.

[RFC6014] updated the requirements for how DNSSEC cryptographic algorithm identifiers in the IANA registries are assigned, reducing the requirements from being "Standards Action" to "RFC Required". However, the IANA registry requirements for hash algorithms for DS records [RFC3658] and for the hash algorithms used in NSEC3 records [RFC5155] are still "Standards Action". This document updates those IANA registry requirements. (For reference on how IANA registries can be updated in general, see [RFC8126].)

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. Update to RFC 6014

Section 4 updates RFC 6014 to bring the requirements for DS records and NSEC3 hash algorithms in line with the rest of the DNSSEC cryptographic algorithms by allowing any DS hash algorithms, NSEC3 hash algorithms, NSEC3 parameters, and NSEC3 flags that are fully described in an RFC to have identifiers assigned in the IANA registries. This is an addition to the IANA considerations in RFC 6014.

3. Update to RFC 8624

This document updates [RFC8624] for all DNSKEY and DS algorithms that are not on standards track.

The second paragraph of Section 1.2 of RFC 8624 currently says:

This document only provides recommendations with respect to mandatory-to-implement algorithms or algorithms so weak that they cannot be recommended. Any algorithm listed in the [DNSKEY-IANA] and [DS-IANA] registries that are not mentioned in this document MAY be implemented. For clarification and consistency, an algorithm will be specified as MAY in this document only when it has been downgraded from a MUST or a RECOMMENDED to a MAY.

That paragraph is now replaced with the following:

This document provides recommendations with respect to mandatory-to-implement algorithms, algorithms so weak that they cannot be recommended, and algorithms that are defined in RFCs that are not on standards track. Any algorithm listed in the [DNSKEY-IANA] and [DS-IANA] registries that are not mentioned in this document MAY be implemented. For clarification and consistency, an algorithm will be specified as MAY in this document only when it has been downgraded from a MUST or a RECOMMENDED to a MAY.

This update is also reflected in the IANA considerations in Section 4.

4. 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".

5. 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.

Administrators of DNSSEC-signed zones, and of validating resolvers, may have been making security decisions based on the contents of the IANA registries. This was a bad idea in the past, and now is an even worse idea because there will be more algorithms in those registries that may not have gone through IETF review. Security decisions about which algorithms are safe and not safe should be made by reading the security literature, not by looking in IANA registries.

6. References

6.1. 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>>.
- [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>>.
- [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>>.
- [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>>.

6.2. Informative 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>>.

Appendix A. Acknowledgements

Donald Eastlake, Murray Kucherawy, Dan Harkins, Martin Duke, and Benjamin Kaduk contributed to this document.

Author's Address

Paul Hoffman
ICANN

Email: paul.hoffman@icann.org

DNSOP
Internet-Draft
Updates: 1034 (if approved)
Intended status: Standards Track
Expires: 24 October 2022

M. Andrews
ISC
S. Huque
Salesforce
P. Wouters
Aiven
D. Wessels
Verisign
22 April 2022

DNS Glue Requirements in Referral Responses
draft-ietf-dnsop-glue-is-not-optional-05

Abstract

The DNS uses glue records to allow iterative clients to find the addresses of name servers that are contained within a delegated zone. Authoritative Servers are expected to return all available in-domain glue records in a referral response. If message size constraints prevent the inclusion of all in-domain glue records, the server MUST set the TC flag to inform the client that the response is incomplete, and that the client SHOULD use another transport to retrieve the full response. This document updates RFC 1034 to clarify correct server behavior.

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 24 October 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
1.1. Reserved Words	3
2. Types of Glue in Referral Responses	3
2.1. Glue for In-Domain Name Servers	3
2.2. Glue for Sibling Domain Name Servers	4
2.3. Glue for Cyclic Sibling Domain Name Servers	5
2.4. Missing Glue	6
3. Requirements	7
3.1. Glue for In-Domain Name Servers	7
3.2. Glue for Sibling Domain Name Servers	8
3.3. Updates to RFC 1034	8
4. Security Considerations	8
5. Operational Considerations	8
6. IANA Considerations	9
7. Acknowledgements	9
8. Changes	9
9. Normative References	10
10. Informative References	10
Authors' Addresses	11

1. Introduction

The Domain Name System (DNS) [RFC1034], [RFC1035] uses glue records to allow iterative clients to find the addresses of name servers that are contained within a delegated zone. Glue records are added to the parent zone as part of the delegation process and returned in referral responses, otherwise a resolver following the referral has no way of finding these addresses. Authoritative servers are expected to return all available in-domain glue records in a referral response. If message size constraints prevent the inclusion of all in-domain glue records over the chosen transport, the server **MUST** set the TC (Truncated) flag to inform the client that the response is incomplete, and that the client **SHOULD** use another transport retrieve the full response. This document clarifies that expectation.

DNS responses sometimes contain optional data in the additional section. In-domain glue records, however, are not optional. Several other protocol extensions, when used, are also not optional. This includes TSIG [RFC2845], OPT [RFC6891], and SIG(0) [RFC2931].

At the time of this writing, addresses (A or AAAA records) for a delegation's authoritative name servers are the only type of glue defined for the DNS.

Note that this document only clarifies requirements of name server software implementations. It does not introduce or change any requirements on data placed in DNS zones or registries. In other words, this document only makes requirements on "available glue records" (i.e., those given in a zone), but does not make requirements regarding their presence in a zone. If some glue records are absent from a given zone, an authoritative name server may be unable to return a useful referral response for the corresponding domain. The IETF may want to consider a separate update to the requirements for including glue in zone data, beyond those given in [RFC1034] and [RFC1035].

1.1. Reserved Words

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].

2. Types of Glue in Referral Responses

This section describes different types of glue that may be found in DNS referral responses. Note that the type of glue depends on the QNAME. A particular record can be in-domain glue for one response and sibling glue for another.

2.1. Glue for In-Domain Name Servers

The following is a simple example of glue records present in the delegating zone "test" for the child zone "foo.test". The name servers for foo.test (ns1.foo.test and ns2.foo.test) are both below the delegation point. They are configured as glue records in the "test" zone:

foo.test.	86400	IN NS	ns1.foo.test.
foo.test.	86400	IN NS	ns2.foo.test.
ns1.foo.test.	86400	IN A	192.0.2.1
ns2.foo.test.	86400	IN AAAA	2001:db8::2:2

A referral response from "test" for "foo.test" with glue for in-domain name servers looks like this:

```
;; QUESTION SECTION:
;www.foo.test.      IN      A

;; AUTHORITY SECTION:
foo.test.           86400    IN      NS      ns1.foo.test.
foo.test.           86400    IN      NS      ns2.foo.test.

;; ADDITIONAL SECTION:
ns1.foo.test.       86400    IN      A        192.0.2.1
ns2.foo.test.       86400    IN      AAAA     2001:db8::2:2
```

2.2. Glue for Sibling Domain Name Servers

Sibling domain name servers are NS records that are not contained in the delegated zone itself, but in another zone delegated from the same parent. In many cases, glue for sibling domain name servers are not strictly required for resolution, since the resolver can make follow-on queries to the sibling zone to resolve the name server addresses (after following the referral to the sibling zone). However, most name server implementations today provide them as an optimization to obviate the need for extra traffic from iterative resolvers.

Here the delegating zone "test" contains two delegations for the child zones "bar.test" and "foo.test":

```
bar.test.           86400    IN      NS      ns1.bar.test.
bar.test.           86400    IN      NS      ns2.bar.test.
ns1.bar.test.       86400    IN      A        192.0.2.1
ns2.bar.test.       86400    IN      AAAA     2001:db8::2:2

foo.test.           86400    IN      NS      ns1.bar.test.
foo.test.           86400    IN      NS      ns2.bar.test.
```

A referral response from "test" for "foo.test" with glue for sibling domain name servers looks like this:

```
;; QUESTION SECTION:
;www.foo.test.      IN      A

;; AUTHORITY SECTION:
foo.test.           86400      IN      NS      ns1.bar.test.
foo.test.           86400      IN      NS      ns2.bar.test.

;; ADDITIONAL SECTION:
ns1.bar.test.       86400      IN      A      192.0.2.1
ns2.bar.test.       86400      IN      AAAA    2001:db8::2:2
```

2.3. Glue for Cyclic Sibling Domain Name Servers

The use of sibling domain name servers can introduce cyclic dependencies. This happens when one domain specifies name servers from a sibling domain, and vice versa. This type of cyclic dependency can only be broken when the delegating name server includes glue for the sibling domain in a referral response.

Here the delegating zone "test" contains two delegations for the child zones "bar.test" and "foo.test", and each use name servers under the other:

```
bar.test.           86400      IN      NS      ns1.foo.test.
bar.test.           86400      IN      NS      ns2.foo.test.
ns1.bar.test.       86400      IN      A      192.0.2.1
ns2.bar.test.       86400      IN      AAAA    2001:db8::2:2

foo.test.           86400      IN      NS      ns1.bar.test.
foo.test.           86400      IN      NS      ns2.bar.test.
ns1.foo.test.       86400      IN      A      192.0.2.3
ns2.foo.test.       86400      IN      AAAA    2001:db8::2:4
```

A referral response from "test" for "bar.test" with glue for sibling domain name servers looks like this:

```
;; QUESTION SECTION:
;www.bar.test.      IN      A

;; AUTHORITY SECTION:
bar.test.           86400      IN      NS      ns1.foo.test.
bar.test.           86400      IN      NS      ns2.foo.test.

;; ADDITIONAL SECTION:
ns1.foo.test.       86400      IN      A      192.0.2.3
ns2.foo.test.       86400      IN      AAAA    2001:db8::2:4
```

In late 2021 the authors analyzed zone file data available from ICANN's Centralized Zone Data Service [CZDS] and found 222 out of approximately 209,000,000 total delegations that had only sibling domain NS RRs in a cyclic dependency as above.

2.4. Missing Glue

An example of missing glue is included here, even though it can not be considered as a type of glue. While not common, real examples of responses that lack required glue, and with TC=0, have been shown to occur and cause resolution failures.

The example below is based on a response observed in June 2020. The names have been altered to fall under documentation domains. It shows a case where none of the glue records present in the zone fit into the available space of the UDP response, and the TC flag was not set. While this example shows a referral with DNSSEC records [RFC4033], [RFC4034], [RFC4035], this behavior has been seen with plain DNS responses as well. Some records have been truncated for display purposes. Note that at the time of this writing, the servers originally responsible for this example have been updated and now correctly set the TC flag.

```
% dig +nored +dnssec +bufsize=512 +ignore @ns.example.net \
    rh202ns2.355.foo.example

; <<>> DiG 9.15.4 <<>> +nored +dnssec +bufsize +ignore \
    @ns.example.net rh202ns2.355.foo.example
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 8798
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 9, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;rh202ns2.355.foo.example.          IN A

;; AUTHORITY SECTION:
foo.example.      86400    IN NS      rh120ns2.368.foo.example.
foo.example.      86400    IN NS      rh202ns2.355.foo.example.
foo.example.      86400    IN NS      rh120ns1.368.foo.example.
foo.example.      86400    IN NS      rh202ns1.355.foo.example.
foo.example.      3600     IN DS      51937 8 1 ...
foo.example.      3600     IN DS      635 8 2 ...
foo.example.      3600     IN DS      51937 8 2 ...
foo.example.      3600     IN DS      635 8 1 ...
foo.example.      3600     IN RRSIG    DS 8 2 3600 ...
```

3. Requirements

This section describes updated requirements for including glue in DNS referral responses.

3.1. Glue for In-Domain Name Servers

This document clarifies that when a name server generates a referral response, it **MUST** include all available glue records for in-domain name servers in the additional section, or **MUST** set TC=1 if constrained by message size.

At the time of writing, most iterative clients send initial queries over UDP and retry over TCP upon receiving a response with the TC flag set. UDP responses are generally limited to between 1232 and 4096 bytes, due to values commonly used for the EDNS0 UDP Message Size field [RFC6891], [FLAGDAY2020]. TCP responses are limited to 65,535 bytes.

3.2. Glue for Sibling Domain Name Servers

This document clarifies that when a name server generates a referral response, it SHOULD include all available glue records in the additional section. If, after adding glue for all in-domain name servers, the glue for all sibling domain name servers does not fit due to message size constraints, the name server is NOT REQUIRED to set TC=1.

Note that users may experience resolution failures for domains with cyclically-dependent sibling name servers when the delegating name server chooses to omit the corresponding glue in a referral response. As described in Section 2.3, such domains are rare.

3.3. Updates to RFC 1034

Replace

"Copy the NS RRs for the subzone into the authority section of the reply. Put whatever addresses are available into the additional section, using glue RRs if the addresses are not available from authoritative data or the cache. Go to step 4."

with

"Copy the NS RRs for the subzone into the authority section of the reply. Put whatever NS addresses are available into the additional section, using glue RRs if the addresses are not available from authoritative data or the cache. If all glue RRs for in-domain name servers do not fit, set TC=1 in the header. Go to step 4."

4. Security Considerations

This document clarifies correct DNS server behavior and does not introduce any changes or new security considerations.

5. Operational Considerations

At the time of this writing, the behavior of most DNS server implementations is to set the TC flag only if none of the available glue records fit in a response over UDP transport. The updated requirements in this document might lead to an increase in the fraction of UDP responses with the TC flag set, and consequently an increase in the number of queries to over TCP transport.

6. IANA Considerations

There are no actions for IANA.

7. Acknowledgements

The authors wish to thank Joe Abley, David Blacka, Brian Dickson, Kazunori Fujiwara, Paul Hoffman, Geoff Huston, Jared Mauch, George Michaelson, Yasuhiro Orange Morishita, Benno Overeinder, John R Levine, Hugo Salgado, Shinta Sato, Puneet Sood, Petr Spacek, Ralf Weber, Tim Wicinski, Suzanne Woolf, and other members of the DNSOP working group for their input.

8. Changes

RFC Editor: Please remove this section before publication.

This section lists substantial changes to the document as it is being worked on.

From -01 to -02:

- * Clarified that "servers" means "authoritative servers".
- * Clarified that "available glue" means "all available glue".
- * Updated examples and placed before RFC 1034 update.

From -02 to -03:

- * Clarified scope to focus only on name server responses, and not zone/registry data.
- * Reorganized with section 2 as Types of Glue and section 3 as Requirements.
- * Removed any discussion of promoted / orphan glue.
- * Use appropriate documentation addresses and domain names.
- * Added Sibling Cyclic Glue example.

From -03 to -04:

- * Use "referral glue" on the assumption that other types of glue may be defined in the future.
- * Added Operational Considerations section.

- * Note many current implementations set TC=1 only when no glue RRs fit. New requirements may lead to more truncation and TCP.
- * Sibling glue can be optional. Only require TC=1 when all in-domain glue RRs don't fit.
- * Avoid talking about requirements for UDP/TCP specifically, and talk more generically about message size constraints regardless of transport.

From -04 to -05:

- * Reverting the -04 change to use the phrase "referral glue".
- * Rephrase "in-domain glue" as "glue for in-domain name servers".
- * Rephrase "sibling glue" as "glue for sibling domain name servers".
- * Expand paragraph noting this document does not make requirements about presence of glue in zones.

9. 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>>.
- [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>>.

10. Informative References

- [CZDS] ICANN, "Centralized Zone Data Service", January 2022, <<https://czds.icann.org/>>.
- [FLAGDAY2020] Various DNS software and service providers, "DNS Flag Day 2020", October 2020, <<https://dnsflagday.net/2020/>>.

- [RFC2845] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, DOI 10.17487/RFC2845, May 2000, <<https://www.rfc-editor.org/info/rfc2845>>.
- [RFC2931] Eastlake 3rd, D., "DNS Request and Transaction Signatures (SIG(0)s)", RFC 2931, DOI 10.17487/RFC2931, September 2000, <<https://www.rfc-editor.org/info/rfc2931>>.
- [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>>.
- [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>>.

Authors' Addresses

M. Andrews
ISC
Email: marka@isc.org

Shumon Huque
Salesforce
Email: shuque@gmail.com

Paul Wouters
Aiven
Email: paul.wouters@aiven.io

Duane Wessels
Verisign

Email: dwessels@verisign.com

Network Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: 18 October 2022

W. Hardaker
USC/ISI
V. Dukhovni
Bloomberg, L.P.
16 April 2022

Guidance for NSEC3 parameter settings
draft-ietf-dnsop-nsec3-guidance-08

Abstract

NSEC3 is a DNSSEC mechanism providing proof of non-existence by asserting that there are no names that exist between two domain names within a zone. Unlike its counterpart NSEC, NSEC3 avoids directly disclosing the bounding domain name pairs. This document provides guidance on setting NSEC3 parameters based on recent operational deployment experience.

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 18 October 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
1.1. Requirements Notation	3
2. NSEC3 Parameter Value Considerations	3
2.1. Algorithms	3
2.2. Flags	4
2.3. Iterations	4
2.4. Salt	5
3. Recommendations for Deploying and Validating NSEC3 Records .	5
3.1. Best-practice for Zone Publishers	6
3.2. Recommendation for Validating Resolvers	6
3.3. Recommendation for Primary / Secondary Relationships . .	7
4. Security Considerations	8
5. Operational Considerations	8
6. IANA Considerations	8
7. References	8
7.1. Normative References	8
7.2. Informative References	9
Appendix A. Deployment measurements at time of publication . . .	9
Appendix B. Computational burdens of processing NSEC3 iterations	9
Appendix C. Acknowledgments	10
Appendix D. Github Version of This Document	10
Appendix E. Implementation Notes	11
E.1. OpenDNSSEC	11
E.2. PowerDNS	11
E.3. Knot DNS and Knot Resolver	11
E.4. Google Public DNS Resolver	11
E.5. Google Cloud DNS	11
Authors' Addresses	11

1. Introduction

As with NSEC [RFC4035], NSEC3 [RFC5155] provides proof of non-existence that consists of signed DNS records establishing the non-existence of a given name or associated Resource Record Type (RRTYPE) in a DNSSEC [RFC4035] signed zone. In the case of NSEC3, however, the names of valid nodes in the zone are obfuscated through (possibly multiple iterations of) hashing (currently only SHA-1 is in use within the Internet).

NSEC3 also provides "opt-out support", allowing for blocks of unsigned delegations to be covered by a single NSEC3 record. Use of the opt-out feature allows large registries to only sign as many NSEC3 records as there are signed DS or other RRsets in the zone; with opt-out, unsigned delegations don't require additional NSEC3 records. This sacrifices the tamper-resistance proof of non-existence offered by NSEC3 in order to reduce memory and CPU overheads.

NSEC3 records have a number of tunable parameters that are specified via an NSEC3PARAM record at the zone apex. These parameters are the hash algorithm, processing flags, the number of hash iterations and the salt. Each of these has security and operational considerations that impact both zone owners and validating resolvers. This document provides some best-practice recommendations for setting the NSEC3 parameters.

1.1. Requirements Notation

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. NSEC3 Parameter Value Considerations

The following sections describe recommendations for setting parameters for NSEC3 and NSEC3PARAM.

2.1. Algorithms

The algorithm field is not discussed by this document.

2.2. Flags

The NSEC3PARAM flags field currently contains no flags, but individual NSEC3 records contain the "Opt-Out" flag [RFC5155], which specifies whether or not that NSEC3 record provides proof of non-existence or not. In general, NSEC3 with the Opt-Out flag enabled should only be used in large, highly dynamic zones with a small percentage of signed delegations. Operationally, this allows for fewer signature creations when new delegations are inserted into a zone. This is typically only necessary for extremely large registration points providing zone updates faster than real-time signing allows or when using memory-constrained hardware. Smaller zones, or large but relatively static zones, are encouraged to use a flags value of 0 (zero) and take advantage of DNSSEC's proof-of-non-existence support.

2.3. Iterations

NSEC3 records are created by first hashing the input domain and then repeating that hashing algorithm a number of times based on the iteration parameter in the NSEC3PARAM and NSEC3 records. The first hash is typically sufficient to discourage zone enumeration performed by "zone walking" an NSEC or NSEC3 chain. Only determined parties with significant resources are likely to try and uncover hashed values, regardless of the number of additional iterations performed. If an adversary really wants to expend significant CPU resources to mount an offline dictionary attack on a zone's NSEC3 chain, they'll likely be able to find most of the "guessable" names despite any level of additional hashing iterations.

Most names published in the DNS are rarely secret or unpredictable. They are published to be memorable, used and consumed by humans. They are often recorded in many other network logs such as email logs, certificate transparency logs, web page links, intrusion detection systems, malware scanners, email archives, etc. Many times a simple dictionary of commonly used domain names prefixes (www, ftp, mail, imap, login, database, etc) can be used to quickly reveal a large number of labels within a zone. Because of this, there are increasing performance costs yet diminishing returns associated with applying additional hash iterations beyond the first.

Although Section 10.3 of [RFC5155] specifies upper bounds for the number of hash iterations to use, there is no published guidance for zone owners about good values to select. Recent academic studies have shown that NSEC3 hashing provides only moderate protection [GPUNSEC3][ZONEENUM].

2.4. Salt

NSEC3 records provide an additional salt value, which can be combined with an FQDN to influence the resulting hash, but properties of this extra salt are complicated.

In cryptography, salts generally add a layer of protection against offline, stored dictionary attacks by combining the value to be hashed with a unique "salt" value. This prevents adversaries from building up and remembering a single dictionary of values that can translate a hash output back to the value that it derived from.

In the case of DNS, the situation is different because the hashed names placed in NSEC3 records are always implicitly "salted" by hashing the fully-qualified domain name from each zone. Thus, no single pre-computed table works to speed up dictionary attacks against multiple target zones. An attacker is always required to compute a complete dictionary per zone, which is expensive in both storage and CPU time.

To understand the role of the additional NSEC3 salt field, we have to consider how a typical zone walking attack works. Typically, the attack has two phases - online and offline. In the online phase, an attacker "walks the zone" by enumerating (almost) all hashes listed in NSEC3 records and storing them for the offline phase. Then, in the offline cracking phase, the attacker attempts to crack the underlying hash. In this phase, the additional salt value raises the cost of the attack only if the salt value changes during the online phase of the attack. In other words, an additional, constant salt value does not change the cost of the attack.

Changing a zone's salt value requires the construction of a complete new NSEC3 chain. This is true both when resigning the entire zone at once, and when incrementally signing it in the background where the new salt is only activated once every name in the chain has been completed. As a result, re-salting is a very complex operation, with significant CPU time, memory, and bandwidth consumption. This makes very frequent re-salting impractical, and renders the additional salt field functionally useless.

3. Recommendations for Deploying and Validating NSEC3 Records

The following subsections describe recommendations for the different operating realms within the DNS.

3.1. Best-practice for Zone Publishers

First, if the operational or security features of NSEC3 are not needed, then NSEC SHOULD be used in preference to NSEC3. NSEC3 requires greater computational power (see Appendix B) for both authoritative servers and validating clients. Specifically, there is a nontrivial complexity in finding matching NSEC3 records to randomly generated prefixes within a DNS zone. NSEC mitigates this concern. If NSEC3 must be used, then an iterations count of 0 MUST be used to alleviate computational burdens. Note that extra iteration counts other than 0 increase the impact of CPU-exhausting DoS attacks, and also increase the risk of interoperability problems.

Note that deploying NSEC with minimally covering NSEC records [RFC4470] also incurs a cost, and zone owners should measure the computational difference in deploying both RFC4470 or NSEC3.

In short, for all zones, the recommended NSEC3 parameters are as shown below:

```
; SHA-1, no extra iterations, empty salt:
;
bcp.example. IN NSEC3PARAM 1 0 0 -
```

For small zones, the use of opt-out based NSEC3 records is NOT RECOMMENDED.

For very large and sparsely signed zones, where the majority of the records are insecure delegations, opt-out MAY be used.

Operators are encouraged to forgo using a salt entirely by using a zero-length salt value instead (represented as a "-" in the presentation format).

If salts are used, note that since the NSEC3PARAM RR is not used by validating resolvers (see [RFC5155] section 4), the iterations and salt parameters can be changed without the need to wait for RRsets to expire from caches. A complete new NSEC3 chain needs to be constructed and the zone resigned.

3.2. Recommendation for Validating Resolvers

Because there has been a large growth of open (public) DNSSEC validating resolvers that are subject to compute resource constraints when handling requests from anonymous clients, this document recommends that validating resolvers change their behavior with respect to large iteration values. Specifically, validating resolver operators and validating resolver software implementers are

encouraged to continue evaluating NSEC3 iteration count deployments and lower their default acceptable limits over time. Similarly, because treating a high iterations count as insecure leaves zones subject to attack, validating resolver operators and validating resolver software implementers are further encouraged to lower their default and acceptable limit for returning SERVFAIL when processing NSEC3 parameters containing large iteration count values. See Appendix A for measurements taken near the time of publication and potential starting points.

Validating resolvers MAY return an insecure response to their clients when processing NSEC3 records with iterations larger than 0. Note also that a validating resolver returning an insecure response MUST still validate the signature over the NSEC3 record to ensure the iteration count was not altered since record publication (see [RFC5155] section 10.3).

Validating resolvers MAY also return a SERVFAIL response when processing NSEC3 records with iterations larger than 0. Validating resolvers MAY choose to ignore authoritative server responses with iteration counts greater than 0, which will likely result in returning a SERVFAIL to the client when no acceptable responses are received from authoritative servers.

Validating resolvers returning an insecure or SERVFAIL answer to their client after receiving and validating an unsupported NSEC3 parameter from the authoritative server(s) SHOULD return an Extended DNS Error (EDE) [RFC8914] EDNS0 option of value (RFC EDITOR: TBD). Validating resolvers that choose to ignore a response with an unsupported iteration count (and do not validate the signature) MUST NOT return this EDE option.

Note that this specification updates [RFC5155] by significantly decreasing the requirements originally specified in Section 10.3 of [RFC5155]. See the Security Considerations for arguments on how to handle responses with non-zero iteration count.

3.3. Recommendation for Primary / Secondary Relationships

Primary and secondary authoritative servers for a zone that are not being run by the same operational staff and/or using the same software and configuration must take into account the potential differences in NSEC3 iteration support.

Operators of secondary services should advertise the parameter limits that their servers support. Correspondingly, operators of primary servers need to ensure that their secondaries support the NSEC3 parameters they expect to use in their zones. To ensure reliability,

after primaries change their iteration counts, they should query their secondaries with known non-existent labels to verify the secondary servers are responding as expected.

4. Security Considerations

This entire document discusses security considerations with various parameters selections of NSEC3 and NSEC3PARAM fields.

The point where a validating resolver returns insecure vs the point where it returns SERVFAIL must be considered carefully. Specifically, when a validating resolver treats a zone as insecure above a particular value (say 100) and returns SERVFAIL above a higher point (say 500), it leaves the zone subject to attacker-in-the-middle attacks as if it was unsigned between these values. Thus, validating resolver operators and software implementers SHOULD set the point above which a zone is treated as insecure for certain values of NSEC3 iterations counts to the same as the point where a validating resolver begins returning SERVFAIL.

5. Operational Considerations

This entire document discusses operational considerations with various parameters selections of NSEC3 and NSEC3PARAM fields.

6. IANA Considerations

This document requests a new allocation in the "Extended DNS Error Codes" of the "Domain Name System (DNS) Parameters" registration table with the following characteristics:

- * INFO-CODE: (RFC EDITOR: TBD)
- * Purpose: Unsupported NSEC3 iterations value
- * Reference: (RFC EDITOR: this document)

7. References

7.1. 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>>.

- [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>>.
- [RFC4470] Weiler, S. and J. Ihren, "Minimally Covering NSEC Records and DNSSEC On-line Signing", RFC 4470, DOI 10.17487/RFC4470, April 2006, <<https://www.rfc-editor.org/info/rfc4470>>.
- [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>>.

7.2. Informative References

- [GPUNSEC3] Wander, M., Schwittmann, L., Boelmann, C., and T. Weis, "GPU-Based NSEC3 Hash Breaking", DOI 10.1109/NCA.2014.27, 2014, <<https://doi.org/10.1109/NCA.2014.27>>.
- [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>>.
- [ZONEENUM] Wang, Z., Xiao, L., and R. Wang, "An efficient DNSSEC zone enumeration algorithm", n.d..

Appendix A. Deployment measurements at time of publication

At the time of publication, setting an upper limit of 100 iterations for treating a zone as insecure is interoperable without significant problems, but at the same time still enables CPU-exhausting DoS attacks.

At the time of publication, returning SERVFAIL beyond 500 iterations appears to be interoperable without significant problems.

Appendix B. Computational burdens of processing NSEC3 iterations

The queries per second (QPS) of authoritative servers will decrease due to computational overhead when processing DNS requests for zones containing higher NSEC3 iteration counts. The table (Appendix C) below shows the drop in QPS for various iteration counts.

Iterations	QPS [% of 0 iterations QPS]
0	100 %
10	89 %
20	82 %
50	64 %
100	47 %
150	38 %

Appendix C. Acknowledgments

The authors would like to thank the dns-operations discussion participants, which took place on mattermost hosted by DNS-OARC.

Additionally, the following people contributed text or review comments to the draft:

- * Vladimir Čunat
- * Tony Finch
- * Paul Hoffman
- * Warren Kumari
- * Alexander Mayrhofer
- * Matthijs Mekking
- * Florian Obser
- * Petr Špaček
- * Paul Vixie
- * Tim Wicinski

Appendix D. Github Version of This Document

While this document is under development, it can be viewed, tracked, issued, pushed with PRs, ... here:

<https://github.com/hardaker/draft-hardaker-dnsop-nsec3-guidance>

Appendix E. Implementation Notes

The following implementations have implemented the guidance in this document. They have graciously provided notes about the details of their implementation below.

E.1. OpenDNSSEC

The OpenDNSSEC configuration checking utility will alert the user about nsec3 iteration values larger than 100.

E.2. PowerDNS

PowerDNS 4.5.2 changed the default value of nsec3-max-iterations to 150.

E.3. Knot DNS and Knot Resolver

Knot DNS 3.0.6 warns when signing with more than 20 NSEC3 iterations. Knot Resolver 5.3.1 treats NSEC3 iterations above 150 as insecure.

E.4. Google Public DNS Resolver

Google Public DNS treats NSEC3 iterations above 100 as insecure since September 2021.

E.5. Google Cloud DNS

Google Cloud DNS uses 1 iteration and 64-bits of fixed random salt for all zones using NSEC3. These parameters cannot be adjusted by users.

Authors' Addresses

Wes Hardaker
USC/ISI
Email: ietf@hardakers.net

Viktor Dukhovni
Bloomberg, L.P.
Email: ietf-dane@dukhovni.org

Independent Submission
Internet-Draft
Intended status: Best Current Practice
Expires: October 12, 2021

R. Arends
ICANN
J. Abley
Public Interest Registry
E. Lisse
Namibian Network Information Center (Pty) Ltd
April 10, 2021

Top-level Domains for Private Internets
draft-ietf-dnsop-private-use-tld-01

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.

This document describes a particular set of code points which, by virtue of the way they have been designated in the ISO 3166 standard, are thought to be plausible choices for the implementation of private namespaces that are anchored in top-level domains.

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.

In order to avoid the operational and security consequences of collisions between private and global use of these code elements as top-level domains, this document specifies that such top-level domains should never be deployed in the global namespace, and reserves them accordingly in the Special-Use Names Registry [RFC6761].

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 October 12, 2021.

Copyright Notice

Copyright (c) 2021 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
2.1.	ISO 3166-1 alpha-2 User-assigned Code Elements
3.	Private-use top-level Domains
4.	Domain Name Reservation Considerations
5.	IAB Considerations
6.	IANA Considerations
7.	Security Considerations
8.	Acknowledgements
9.	Informative References
	Appendix A. Examples of Current Uses of the User-assigned Code Elements.
	Authors' Addresses

1. Introduction

The Domain Name System (DNS) [RFC1034] is used to map names to services, systems and other devices that are accessible across networks. Many network operators configure such name mappings in such a way that names referring to private resources, such as services that are intended for use within private networks, are not published in the DNS for general use over the Internet. Collections of such names form a private namespace.

Private namespaces can be considered to be local sub-trees of the familiar, global DNS namespace. An operator can choose where their private namespace is anchored. Since it is useful for applications to be able to make use of both private and global names, it is important that the private and global namespaces do not overlap. Some operators are known to have chosen top-level domains that do not exist in the global namespace as anchors for their private namespaces. 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.

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.

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.

The ISO 3166 standard is used for the implementation of country-code Top-Level Domains in the DNS. This standard is maintained by the ISO 3166 Maintenance Agency and includes a set of code elements designated "user-assigned". Such user-assigned code points are in use for a variety of applications where it is useful to avoid conflict with codes assigned to countries or regions.

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.

2.1. 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].

Appendix A contains examples of the usage of ISO 3166-1 user-assigned alpha-2 code elements in various organisations.

3. 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."

The ISO 3166 user-assigned code elements are hence plausible choices for network operators who have decided to use a top-level domain as an anchor for their private namespace. They are safer choices than some other labels that do not currently exist as top-level domains, since new top-level domains are assigned from time to time.

The ISO 3166 standard is not maintained by the IETF, and it is possible that the standard will change in the future. However, the use of ISO 3166 alpha-2 user-assigned code elements as top-level domain anchors for private namespaces under the current standard is well-known. Regardless of any future changes to the ISO 3166 standard, choosing to add a top-level domain in the global namespace that conflicted with any of these code points in the future could have negative operational effects and pose security risks.

To avoid these negative operational consequences, this document directs that the top-level domains corresponding to these ISO 3166 alpha-2 user-assigned code elements should never be deployed in the global namespace; that is, they must never exist as an owner name in the root zone of the DNS.

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].

4. Domain Name Reservation Considerations

The information that follows is intended to satisfy the requirements of [RFC6761]. The top-level domains corresponding to the ISO 3166 User-Assigned code elements are special in the following ways:

1. Users are free to use these names as they would any other top-level domain. However, since this document specifies that these names **MUST** never be deployed in the global DNS, users **SHOULD** be aware that these names are likely to yield different results on different networks.
2. Application software **SHOULD NOT** recognise these names as special and **SHOULD** use these names as they would any other name.
3. Name resolution APIs and libraries **SHOULD NOT** recognise these names as special and **SHOULD NOT** treat them differently. Name resolution APIs **SHOULD** send queries for these names to their configured caching DNS server(s).
4. Caching DNS servers **MAY** recognise these names as special and **SHOULD NOT**, by default, attempt to look up NS records for the, or otherwise query authoritative DNS servers on the global Internet in an attempt to resolve these domains. Instead, caching DNS servers **SHOULD**, by default, generate immediate (positive or negative) responses for all such queries. This is to avoid unnecessary load on the root name servers and other name servers. Caching DNS servers **SHOULD** offer a configuration option (disabled by default) to enable upstream resolution of such names, for use in private networks where these domains are known to be handled by an authoritative DNS server in said private network.
5. Authoritative DNS servers **SHOULD** recognise these names as special and **SHOULD**, by default, generate immediate negative responses for all such queries, unless explicitly configured by the administrator to give positive answers from a private namespace.
6. DNS server operators **SHOULD**, if they are using private namespaces anchored at these names, configure their authoritative DNS servers to act as authoritative for these names.
7. DNS Registries/Registrars **MUST NOT** grant requests to register any of these names in the normal way to any person or entity. These

names are reserved due to their use in private namespaces, and fall outside the set of names available for allocation by registries/registrars. Attempting to allocate one of these names as if it were a normal DNS domain name will probably not work as desired, for reasons 4, 5 and 6 above.

5. IAB Considerations

This document specifies that various two-character codes should never be used in the global DNS as top-level domains, for technical, operational and security reasons. This technical restriction has implications for root zone management in the DNS, policy for which is developed at ICANN.

As part of its review process for this document, the authors suggest that the IAB exercise its relevant liaisons to ICANN (the organisation and the community) to ensure that the content of this document does not raise any concerns that the IAB feels are important. The authors further suggest that the text in this section be replaced prior to publication by a record of the IAB's review.

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.

This document directs that the following rows be added to the Special-Use Names Registry:

Name	Reference
AA.	(this document)
QM.	(this document)
QN.	(this document)
QO.	(this document)
QP.	(this document)
QQ.	(this document)
QR.	(this document)
QS.	(this document)
QT.	(this document)
QU.	(this document)
QV.	(this document)
QW.	(this document)
QX.	(this document)

QY.	(this document)
QZ.	(this document)
XA.	(this document)
XB.	(this document)
XC.	(this document)
XD.	(this document)
XE.	(this document)
XF.	(this document)
XG.	(this document)
XH.	(this document)
XI.	(this document)
XJ.	(this document)
XK.	(this document)
XL.	(this document)
XM.	(this document)
XN.	(this document)
XO.	(this document)
XP.	(this document)
XQ.	(this document)
XR.	(this document)
XS.	(this document)
XT.	(this document)
XU.	(this document)
XV.	(this document)
XW.	(this document)
XX.	(this document)
XY.	(this document)
XZ.	(this document)
ZZ.	(this document)

Use of private-use identifiers of any sort is known to result in unexpected collisions. 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 or expects.

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

Appendix A. 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."

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

Eberhard W. Lisse
Namibian Network Information Center (Pty) Ltd

Email: el@lisse.na

DNSOP WG
Internet-Draft
Intended status: Standards Track
Expires: December 6, 2021

T. Reddy
McAfee
N. Cook
Open-Xchange
D. Wing
Citrix
M. Boucadair
Orange
June 4, 2021

DNS Access Denied Error Page
draft-reddy-dnsop-error-page-08

Abstract

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

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

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 December 6, 2021.

Copyright Notice

Copyright (c) 2021 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	6
3. Error Page URI EDNS0 Option Format	6
4. Error Page URI Processing	7
4.1. Mitigating EDNS0 Forgery	8
5. Error Page	9
6. Usability Considerations	10
7. Security Considerations	10
8. IANA Considerations	11
8.1. A New Error Page URI EDNS Option	11
9. Acknowledgements	11
10. References	11
10.1. Normative References	11
10.2. Informative References	13
Authors' Addresses	14

1. Introduction

DNS filters are deployed for a variety of reasons, including endpoint security, parental filtering, and filtering required by law enforcement. Some of these reasons 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 upon 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 their DNS responses in real time in order 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 within 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 Internet Service Providers (ISPs) typically block access to some DNS domains due to a requirement imposed by an external entity (e.g., Law Enforcement Agency). Such blocking is performed using DNS-based content filtering.

DNS responses can be filtered by sending a bogus (also called, "forged") A or AAAA response, NXDOMAIN error or empty answer, or an extended DNS error (EDE) code defined in [RFC8914]. Each of these methods have advantages and disadvantages that are discussed below:

1. The DNS response is forged to provide a list of IP addresses that points to an HTTP(S) server alerting the end user about the reason for blocking access to the requested domain (e.g., malware). When an HTTP(S) enabled domain name is blocked, the network security device (e.g., CPE, firewall) presents a block page instead of the HTTP response from the content provider hosting that domain. 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 while the network security device(s) stores 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 using the private key from the local root certificate.
 - * However, configuring the local root certificate on endpoints is not a 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 connected devices are managed by the IT team or Mobile Device Management (MDM) devices, especially in the quite common Bring Your Own Device (BYOD) 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 reach the domain but with no success. Frustrated, the end user may switch to an alternate network that offers no DNS-level protection against malware and phishing, potentially compromising both security and privacy. Furthermore, certificate errors train users to click through certificate errors, which is a bad 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 on a host device (e.g. [Chrome-Install-Cert]). Doing so, however, is also a bad security practice as it creates a security vulnerability that may be exploited by a MITM attack. When a 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 reach the domain but with no success. Frustrated, the end user may use insecure connections to reach the domain, potentially compromising both security and privacy.
 3. The extended error codes Blocked, Censored, and Filtered defined in Section 4 of [RFC8914] can be returned by a DNS server to provide additional information about the cause of an DNS error. If the extended error code "Forged Answer" defined in Section 4.5 of [RFC8914] is returned by the DNS server, the client can identify the DNS response is forged together with the reason for HTTPS certificate error.

These extended error codes do not suffer from the limitations discussed in bullets (1) and (2), but the user still does not know the exact reason nor he/she is aware of the exact entity

blocking the access to the domain. For example, a DNS server may block access to a domain based on the content category such as "Adult Content" to enforce parental control, "Violence & Terrorism" due to an external requirement imposed by an external entity (e.g., Law Enforcement Agency), etc. These content categories 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. Furthermore, the threat data used to categorize domains may sometimes misclassify domains (e.g., domains wrongly classified as Domain Generation Algorithm (DGA) by deep learning techniques, domain wrongly classified as phishing due to crowd sourcing, new domains not categorized by the threat data). A user needs to know the contact details of the IT/InfoSec team to raise a complaint.

4. The EXTRA-TEXT field of the EDE option defined in Section 2 of [RFC8914] can include additional textual information about the cause of the error, but the information could be provided in a language that is not understood by the user. When a resolver or forwarder forwards the received EDE option, the EXTRA-TEXT field only conveys the source of the error (Section 3 of [RFC8914]) and does not provide additional textual information about the cause of the error. Most importantly, EDE option does not offer authenticated information; it can thus be spoofed by an attacker. In addition, the additional textual information may not be able to convey all of the required information about the cause of the DNS error because lengthy EXTRA-TEXT content would be truncated to prevent fragmentation (Section 3 of [RFC8914]).

No matter which type of response is generated (forged IP address(es), NXDOMAIN or empty answer, or an extended error code), the user who triggered the DNS 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 an 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. This is achieved as the block page no longer needs to create a signed certificate when blocking a destination. This approach 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].

'Encrypted DNS' refers to any encrypted scheme to convey DNS messages, for example, 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 provides additional information in a DNS response about the cause of blocking access to a requested domain. This option is structured as depicted in Figure 1.

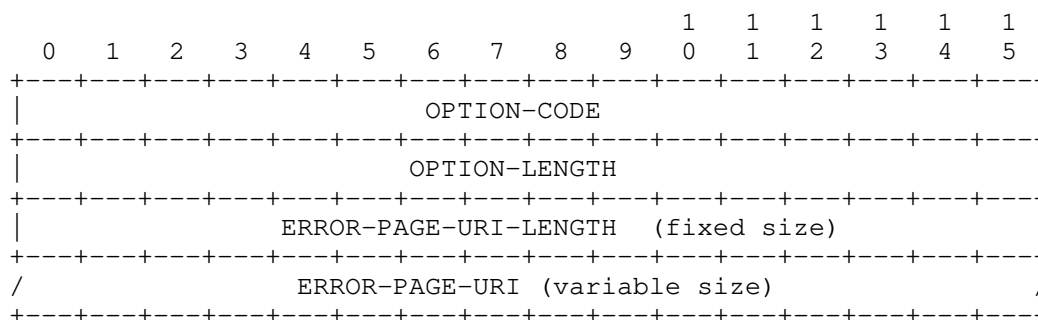


Figure 1: Error Page URI EDNS0 Option Format

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 field.
- 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.

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 the HTTPS client on the endpoint encodes an 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 provided as the variable value for "target-domain" to expand the URI Template into an URI reference in the HTTP GET request.

An example is illustrated below:

If the URI Template is "https://resolver.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" sent in an HTTP GET request. In the above example, the expansion of the above URI Template is "https://resolver.example.net/block-page?target-domain=example.com".

HTTP/2 [RFC7540] is the minimum RECOMMENDED HTTP version to use to retrieve the error page. The HTTPS client retrieving the error page MUST verify the entire certification path as per [RFC5280]. The HTTPS client additionally uses validation techniques 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 rules below to process the Error Page URI EDNS0 option:

- 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 The DNS client MUST reject the error page URI if the scheme is not "https".

4.1. Mitigating EDNS0 Forgery

The Error Page URI EDNS0 option is susceptible to forgery. An attacker (e.g., a man in the middle (MITM)) could insert an extended Error Page URI EDNS0 option into the DNS response causing a client to attempt to visit that URI. For instance, the attacker can be located between the stub resolver and DNS recursive server or between the DNS proxy and the upstream resolver. To mitigate that attack, the following measures are enforced:

- o 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 a 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. Both of these fallback mechanisms adversely impacts security and privacy. If the DNS client has enabled opportunistic privacy profile for DoT, the DNS client MUST ignore Error Page URI EDNS0 option in responses, but SHOULD process other parts of the response.
- o If a 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 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 can learn whether the encrypted DNS resolver performs DNS-based content filtering or not by retrieving resolver information using the method defined in [I-D.reddy-add-resolver-info].
- o DNS forwarders (or DNS proxies) are supposed to propagate unknown EDNS0 options (Sections 4.1 and 4.4.1 of [RFC5625]), which means the Error Page URI EDNS0 option may get propagated by such a DNS server. To detect this scenario, the DNS client MUST verify the domain name in the Error Page URI matches the domain name of the encrypted DNS resolver. If this match fails, the DNS client MUST ignore Error Page URI EDNS0 option in the response, but SHOULD process other parts of the response.

5. Error Page

The following 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.

6. Usability Considerations

The error page SHOULD be returned in the user's preferred language as expressed by the Accept-Language HTTP header.

7. Security Considerations

Security considerations in Section 6 of [RFC8914] 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. However, this specification mandates the DNS client to not process DNS response with Error Page URI EDNS0 option if domain name in the Error Page URI does not match the domain name of the encrypted DNS server. The validation ensures both the servers are operated by the same entity and have the same origin (similar to the Same Origin Policy (SOP)).

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.

8. IANA Considerations

8.1. 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]

9. Acknowledgements

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

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

- [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>>.

10.2. Informative References

- [Chrome-Install-Cert]
"How to manually install the Securlly SSL certificate in Chrome", <support.securlly.com/hc/en-us/articles/206081828-How-to-manually-install-the-Securlly-SSL-certificate-in-Chrome>.
- [Facebook-Container]
"Facebook container for Firefox", <<https://www.mozilla.org/en-US/firefox/facebookcontainer/>>.
- [I-D.ietf-dprive-dnsoquic]
Huitema, C., Mankin, A., and S. Dickinson, "Specification of DNS over Dedicated QUIC Connections", draft-ietf-dprive-dnsoquic-02 (work in progress), February 2021.
- [I-D.reddy-add-resolver-info]
Reddy, T. and M. Boucadair, "DNS Resolver Information", draft-reddy-add-resolver-info-03 (work in progress), April 2021.
- [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>>.
- [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

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 8 September 2022

S. Sahib
Brave Software
S. Huque
Salesforce
P. Wouters
Aiven
7 March 2022

Survey of Domain Verification Techniques using DNS
draft-sahib-domain-verification-techniques-03

Abstract

Many services on the Internet need to verify ownership or control of a domain in the Domain Name System (DNS) [RFC1034] [RFC1035]. This verification is often done by requesting a specific DNS record to be visible in the domain. This document surveys various techniques in wide use today, the pros and cons of each, and proposes some practises to avoid known problems.

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/ShivanKaul/draft-sahib-domain-verification-techniques>.

Status of This Memo

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

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

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

This Internet-Draft will expire on 8 September 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. Conventions and Definitions	3
3. Verification Techniques	3
3.1. TXT based	3
3.1.1. Examples	4
3.2. CNAME based	5
3.2.1. Examples	5
3.3. Common Patterns	6
3.3.1. Name	6
3.3.2. RDATA	6
4. Recommendations	6
4.1. Targeted Domain Verification	6
4.2. Targeted Service Verification	7
4.3. TXT vs CNAME	7
4.4. Time-bound checking	8
5. Email sending authorization	9
6. Security Considerations	9
7. Operational Considerations	9
8. IANA Considerations	10
9. References	10
9.1. Normative References	10
9.2. Informative References	10
Acknowledgments	11
Authors' Addresses	11

1. Introduction

Many providers of internet services need domain owners to prove that they control a particular domain before they can operate a services or grant some privilege to the associated domain. For instance, certificate authorities like Let's Encrypt [LETSencrypt] ask requesters of TLS certificates to prove that they operate the domain they are requesting the certificate for. Providers generally allow for several different ways of proving domain control. This document describes and recommends common practises with using DNS based techniques for domain verification. Other techniques such as email or HTTP(S) based verification are out-of-scope.

In practice, DNS-based verification takes the form of the provider generating a random value visible only to the requester, and then asking the requester to create a DNS record containing this random value and placing it at a location within the domain that the provider can query for. Generally only one temporary DNS record is sufficient for proving domain ownership, although sometimes the DNS record must be kept in the zone to prove continued ownership of the domain.

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.

Provider: an internet-based provider of a service, for e.g., Let's Encrypt provides a certificate authority service or GitHub provides code-hosting services. These services often require a user to verify that they control a domain.

3. Verification Techniques

3.1. TXT based

TXT record-based DNS domain verification is usually the default option for DNS verification. The service provider asks the user to add a DNS TXT record (perhaps through their domain host or DNS provider) at the domain with a certain value. Then, the service provider does a DNS TXT query for the domain being verified and checks that the value exists. For example, this is what a DNS TXT verification record could look like:

```
example.com.    IN      TXT      "foo-verification=bar-237943648324687364"
```

Here, the value "bar-237943648324687364" for the attribute "foo-verification" serves as the randomly-generated TXT value being added to prove ownership of the domain to Foo provider. Although the original DNS protocol specifications did not associate any semantics with the DNS TXT record, [RFC1464] describes how to use them to store attributes in the form of ASCII text key-value pairs for a particular domain. In practice, there is wide variation in the content of DNS TXT records used for domain verification, and they often do not follow the key-value pair model. Even so, the rdata portion of the DNS TXT record has to contain the value being used to verify the domain. The value is usually a randomly-generated token in order to guarantee that the entity who requested that the domain be verified (i.e. the person managing the account at Foo provider) is the one who has (direct or delegated) access to DNS records for the domain. The generated token typically expires in a few days. The TXT record is placed at the domain being verified ("example.com" in the example above). After a TXT record has been added, the service provider will usually take some time to verify that the DNS TXT record with the expected token exists for the domain.

The same domain name can have multiple distinct TXT records (a TXT Record Set), where each TXT record may be associated with a distinct service. Having many of these may cause operational issues, and it is RECOMMENDED that providers use a prefix (eg "_foo.example.com") instead of using the top of the domain ("APEX") directly, such as:

```
_foo.example.com. IN TXT "bar-237943648324687364"
```

3.1.1. Examples

3.1.1.1. Let's Encrypt

Let's Encrypt [LETSencrypt] has a challenge type DNS-01 that lets a user prove domain ownership in accordance with the ACME protocol [RFC8555]. In this challenge, Let's Encrypt asks you to create a TXT record with a randomly-generated token at _acme-challenge.<YOUR_DOMAIN>. For example, if you wanted to prove domain ownership of example.com, Let's Encrypt could ask you to create the DNS record:

```
_acme-challenge.example.com. IN TXT "cE3A8qQpEzAIYq-T9DWNdLJ1_YRXamdxcjGTbz  
rOH5L"
```

[RFC8555] (section 8.4) places requirements on the random value.

3.1.1.2. Google Workspace

[GOOGLE-WORKSPACE-TXT] asks the user to sign in with their administrative account and obtain their verification token as part of the setup process for Google Workspace. The verification token is a 68-character string that begins with "google-site-verification=", followed by 43 characters. Google recommends a TTL of 3600 seconds. The owner name of the TXT record is the domain or subdomain name being verified.

3.1.1.3. GitHub

GitHub asks you to create a DNS TXT record under `_github-challenge-ORGANIZATION-<YOUR_DOMAIN>`, where ORGANIZATION stands for the GitHub organization name [GITHUB-TXT]. The code is a numeric code that expires in 7 days.

3.2. CNAME based

Less commonly than TXT record verification, service providers also provide the ability to verify domain ownership via CNAME records. One reason for using CNAME is for the case where the user cannot create TXT records. One common reason is that the domain name may already have CNAME record that aliases it to a 3rd-party target domain. CNAMEs have a technical restriction that no other record types can be placed along side them at the same domain name ([RFC1034], Section 3.6.2).. The CNAME based domain verification method typically uses a randomized label prepended to the domain name being verified.

3.2.1. Examples

3.2.1.1. Google

[GOOGLE-WORKSPACE-CNAME] lets you specify a CNAME record for verifying domain ownership. The user gets a unique 12-character string that is added as "Host", with TTL 3600 (or default) and Destination an 86-character string beginning with "gv-" and ending with ".domainverify.googlehosted.com.".

To verify a subdomain, the unique 12-character string is appended with the subdomain name for "Host" field for e.g. `JLKDER712AFP.subdomain` where subdomain is the subdomain being verified.

3.2.1.2. AWS Certificate Manager (ACM)

To get issued a certificate by AWS Certificate Manager (ACM), you can create a CNAME record to verify domain ownership [ACM-CNAME]. The record name for the CNAME looks like:

```
`\_<random-token1>.example.com.    IN    CNAME \_RANDOM-TOKEN.acm-validations.aws.
```

Note that if there are more than 5 CNAMEs being chained, then this method does not work.

3.3. Common Patterns

3.3.1. Name

ACME and GitHub have a suffix of `_PROVIDER_NAME-challenge` in the Name field of the TXT record challenge. For ACME, the full Host is `_acme-challenge.<YOUR_DOMAIN>`, while for GitHub it is `_github-challenge-ORGANIZATION-<YOUR_DOMAIN>`. Both these patterns are useful for doing targeted domain verification, as discussed in section (#targeted-domain-verification) because if the provider knows what it is looking for (domain in the case of ACME, organization name + domain in case of GitHub) it can specifically do a DNS query for that TXT record, as opposed to having to do a TXT query for the apex.

ACME does the same name construction for CNAME records.

3.3.2. RDATA

One pattern that quite a few providers follow (Dropbox, Atlassian) is constructing the rdata of the TXT DNS record in the form of `PROVIDER-SERVICE-domain-verification=` followed by the random value being checked for. This is in accordance with [RFC1464] which mandates that attributes must be stored as key-value pairs.

4. Recommendations

4.1. Targeted Domain Verification

The TXT record being used for domain verification is most commonly placed at the domain name being verified. For example, if `example.com` is being verified, then the DNS TXT record will have `example.com` in the Name section. Unfortunately, this practise does not scale very well.

Many services are now attempting to verify domain names, causing many of these TXT records to be placed at that same location at the top of the domain (the APEX).

When a DNS administrator sees 15 DNS TXT records for their domain based on only random letters, they can no longer determine for which service or vendor the DNS TXT records were added. This causes administrators to leave all DNS TXT records in there, as they want to avoid breaking a service. Over time, the domain ends up with a lot of no longer needed, unknown and untracable DNS TXT records.

An operational issue arises from the DNS protocol only being able to query for "all TXT records" at a single location. If multiple services all require TXT records, this can cause the DNS answer for TXT records to become very large. It has been observed that some well known domains had so many services deployed that their DNS TXT answer did not fit in a single UDP DNS packet. This results in fragmentation which is known to be vulnerable to various attacks draft-ietf-dnsop-avoid-fragmentation-06. It can also lead to UDP packet truncation, causing a retry over TCP. Not all networks properly transport DNS over TCP and some DNS software mistakenly believe TCP support is optional draft-ietf-dnsop-dns-tcp-requirements-15.

4.2. Targeted Service Verification

One malicious service that promises to deliver something after domain verification could surreptitiously ask another service provider to start processing or sending mail for the target domain and then present the victim domain administrator with this DNS TXT record pretending to be for their service. Once the administrator has added the DNS TXT record, instead of getting their service, their domain is now certifying another service of which they are not aware they are now a consumer.

If services use a clear description and name attribution in the required DNS TXT record, this can be avoided. For example by requiring a DNS TXT record at `_vendorname.example.com` instead of at `example.com`, a malicious service could no longer replay this without the DNS administrator noticing this. The LetsEncrypt ACME challenge uses this method.

4.3. TXT vs CNAME

The inherent problem of a CNAME is that it cannot co-exist with any other data. What happens when both a CNAME and other data such as a TXT record or NS record exist depends on the DNS implementation. But most likely, either the CNAME or the other records will be silently ignored. The user interface for adding a record might not check for this. It might also break in unexpected ways. If a CNAME is added for continuous authorization, and for another service a TXT record is added, the TXT record might work but the CNAME record might break.

Operational experience has also shown a vendor that provides two difference services, one requiring a CNAME and one requiring a TXT record for authorization that needed to be deployed at the same location. If both services would have used a TXT record, this would not have caused any problems.

Another issues with CNAME records is that they MUST NOT point to another CNAME. But where this might be true in an initial deployment, if the target that the CNAME points to is changed from a non-CNAME record to a CNAME record, some DNS software might no longer resolve this as expected.

Early web based DNS administration tools did not always have the TXT record available in a pulldown menu for DNS record types, while CNAME would be available. However as many anti-spam meassures now require TXT records, this support is now generally available. It is recommended that the CNAME method is only used for delegating authorization to an actual subdomain, for example:

```
recrutement.example.com.    IN    CNAME    example.recrutement-vendor.com.
```

4.4. Time-bound checking

After domain verification is done, there is no need for the TXT or CNAME record to continue to exist as the presence of the domain-verifying DNS record for a service only implies that a user with access to the service also has DNS control of the domain at the time the code was generated. It should be safe to remove the verifying DNS record once the verification is done and the service provider doing the verification should specify how long the verification will take (i.e. after how much time can the verifying DNS record be deleted). However, despite this, some services ask the record to exist in perpetuity [ATLASSIAN-VERIFY].

If a provider will use the DNS TXT record only for a one-time verification, it is RECOMMENDED that they clearly indicate this in the RDATA of the TXT record, so a DNS administrator at the target domain can easilly spot an obsolete record in the future. For example:

```
_provider-token.example.com.  IN  TXT  "type=activation_only  
expiry=2023-10-12 token=TOKENDATA"
```

If a provider requires the continued precense of the TXT record as proof that the domain owner is still authorizing the service, this should also be clear from the TXT record RDATA. For example:

```
_provider-service.example.com. IN TXT "type=continued_service  
expiry=never token=TOKENDATA"
```

5. Email sending authorization

Some vendors use a hosted service that wants to generate emails that appear to be from the customer. When a customer has deployed anti-spam measures such as DKIM [RFC6376], DMARC [RFC7489] or SPF [RFC7208], the vendor's mail service needs to be added to the list of allowed mail servers. However, some customers might not want to give permission for a vendor to send emails from their entire domain. It is recommended that a vendor uses a subdomain. If the vendor's domain is example-vendor.com, and the customer domain is example-customer.com, the vendor could use the subdomain example-customer.example-vendor.com to send emails. Alternatively, the customer could delegate a subdomain example-vendor.example-customer.com to the vendor for email sending, as those email addresses would have a stronger origin appearance of being emails sent by the customer to their clients.

Besides requiring proof of ownership of the domain, the customer needs to authorize the hosted service to send email on their behalf.

6. Security Considerations

Both the provider and the service being authenticated and authorized should be obvious from the TXT content to prevent malicious services from misleading the domain owner into certifying a different provider or service.

It is RECOMMENDED that DNSSEC [RFC4033] is employed by the domain owner. A service provider MUST enable DNSSEC validation when verifying domain name challenges to protect against domain name spoofing.

7. Operational Considerations

It is often consumers of the provider services that are not DNS experts that need to relay information from a provider's website to their local DNS administrators. The exact DNS record type, content and location is often not clear when the DNS administrator receives the information. It is RECOMMENDED that providers offer extremely detailed help pages, that are accessible without needing a login on the provider website, as the DNS administrator often has no login account on the provider service website. It is recommended that any instructions given by the provider contains the entire DNS record using a Fully Qualified Domain Name (FQDN).

8. IANA Considerations

This document has no IANA actions.

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/rfc/rfc1034>>.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<https://www.rfc-editor.org/rfc/rfc1035>>.
- [RFC1464] Rosenbaum, R., "Using the Domain Name System To Store Arbitrary String Attributes", RFC 1464, DOI 10.17487/RFC1464, May 1993, <<https://www.rfc-editor.org/rfc/rfc1464>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.
- [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/rfc/rfc4033>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/rfc/rfc8174>>.

9.2. Informative References

- [ACM-CNAME] AWS, "Option 1: DNS Validation", n.d., <<https://docs.aws.amazon.com/acm/latest/userguide/dns-validation.html>>.
- [ATLASSIAN-VERIFY] Atlassian, "Verify over DNS", n.d., <<https://support.atlassian.com/user-management/docs/verify-a-domain-to-manage-accounts/#Verifyadomainforyourorganization-VerifyoverDNS>>.

[GITHUB-TXT]

GitHub, "Verifying your organization's domain", n.d.,
<<https://docs.github.com/en/github/setting-up-and-managing-organizations-and-teams/verifying-your-organizations-domain>>.

[GOOGLE-WORKSPACE-CNAME]

Google, "CNAME record values", n.d.,
<<https://support.google.com/a/answer/112038>>.

[GOOGLE-WORKSPACE-TXT]

Google, "TXT record values", n.d.,
<<https://support.google.com/a/answer/2716802>>.

[LETSENCRYPT]

Let's Encrypt, "Challenge Types: DNS-01 challenge", 2020,
<<https://letsencrypt.org/docs/challenge-types/#dns-01-challenge>>.

[RFC6376] Crocker, D., Ed., Hansen, T., Ed., and M. Kucherawy, Ed.,
"DomainKeys Identified Mail (DKIM) Signatures", STD 76,
RFC 6376, DOI 10.17487/RFC6376, September 2011,
<<https://www.rfc-editor.org/rfc/rfc6376>>.

[RFC7208] Kitterman, S., "Sender Policy Framework (SPF) for
Authorizing Use of Domains in Email, Version 1", RFC 7208,
DOI 10.17487/RFC7208, April 2014,
<<https://www.rfc-editor.org/rfc/rfc7208>>.

[RFC7489] Kucherawy, M., Ed. and E. Zwicky, Ed., "Domain-based
Message Authentication, Reporting, and Conformance
(DMARC)", RFC 7489, DOI 10.17487/RFC7489, March 2015,
<<https://www.rfc-editor.org/rfc/rfc7489>>.

[RFC8555] Barnes, R., Hoffman-Andrews, J., McCarney, D., and J.
Kasten, "Automatic Certificate Management Environment
(ACME)", RFC 8555, DOI 10.17487/RFC8555, March 2019,
<<https://www.rfc-editor.org/rfc/rfc8555>>.

Acknowledgments

TODO

Authors' Addresses

Shivan Sahib
Brave Software
Email: shivankaulsahib@gmail.com

Shumon Huque
Salesforce
Email: shuque@gmail.com

Paul Wouters
Aiven
Email: paul.wouters@aiven.io

DNSOP WG
Internet-Draft
Updates: 8914 (if approved)
Intended status: Standards Track
Expires: 29 October 2022

D. Wing
Citrix
T. Reddy
Akamai
N. Cook
Open-Xchange
M. Boucadair
Orange
27 April 2022

Structured Data for Filtered DNS
draft-wing-dnsop-structured-dns-error-page-03

Abstract

DNS filtering is widely deployed for network security, but filtered DNS responses lack information for the end user to understand the reason for the filtering. Existing mechanisms to provide detail to end users cause harm especially if the blocked DNS response is to an HTTPS website.

This document updates RFC8914's EXTRA-TEXT field to provide information on DNS filtering. This information can be parsed by the client and displayed, logged, or used for other purposes.

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 29 October 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Terminology	5
3. I-JSON in EXTRA-TEXT field	6
4. Protocol Operation	6
4.1. Client Generating Request	6
4.2. Server Generating Response	7
4.3. Client Processing Response	7
5. Examples	8
6. Security Considerations	9
7. IANA Considerations	9
8. Changes	10
9. References	11
9.1. Normative References	11
9.2. Informative References	12
Authors' Addresses	13

1. Introduction

DNS filters are deployed for a variety of reasons including endpoint security, parental filtering, and filtering required by law enforcement. Network-based security solutions such as firewalls and Intrusion Prevention Systems (IPS) rely upon network traffic inspection to implement perimeter-based security policies and operate by filtering DNS responses. In a home, DNS filtering is used for the same reasons as above and additionally for parental control. Internet Service Providers typically block access to some DNS domains due to a requirement imposed by an external entity (e.g., law enforcement agency) also performed using DNS-based content filtering.

Users of DNS services which perform filtering may wish to receive more information about such filtering to resolve problems with the filter -- for example to contact the administrator to allowlist a domain that was erroneously filtered or to understand the reason a particular domain was filtered. With that information, the user can choose another network, open a trouble ticket with the DNS administrator to resolve erroneous filtering, log the information, or other uses.

DNS responses can be filtered by sending a bogus (also called, "forged") A or AAAA response, NXDOMAIN error or empty answer, or an extended DNS error (EDE) code defined in [RFC8914]. Each of these methods have advantages and disadvantages that are discussed below:

1. The DNS response is forged to provide a list of IP addresses that points to an HTTP(S) server alerting the end user about the reason for blocking access to the requested domain (e.g., malware). When an HTTP(S) enabled domain name is blocked, the network security device (e.g., CPE, firewall) presents a block page instead of the HTTP response from the content provider hosting that domain. 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 while the network security device(s) stores 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 using the private key from the local root certificate.
 - * However, configuring the local root certificate on endpoints is not a 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 filtered DNS response points to a server that will display the block page. If the client is using HTTPS (via web browser or another application) this results in a certificate validation error which gives no information to the end-user about the reason for the DNS filtering. Browsers will display errors such as "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). Applications might display even more cryptic error messages.
 - * Enterprise networks do not assume that all the connected devices are managed by the IT team or Mobile Device Management (MDM) devices, especially in the quite common Bring Your Own Device (BYOD) 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 prevented and, consequently, may repeatedly try to reach the domain but with no success. Frustrated, the end user may switch to an alternate network that offers no DNS filtering against malware and phishing, potentially compromising both security and privacy. Furthermore, certificate errors train users to click through certificate errors, which is a bad 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 on a host device. Doing so, however, is also a bad security practice as it creates a security vulnerability that may be exploited by a MITM attack. When a 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 reach the domain but with no success. Frustrated, the end user may use insecure connections to reach the domain, potentially compromising both security and privacy.
 3. The extended error codes Blocked, Censored, and Filtered defined in Section 4 of [RFC8914] can be returned by a DNS server to provide additional information about the cause of a DNS error. If the extended error code "Forged Answer" defined in Section 4.5 of [RFC8914] is returned by the DNS server, the client can identify the DNS response is forged together with the reason for HTTPS certificate error.
 4. These extended error codes do not suffer from the limitations discussed in bullets (1) and (2), but the user still does not know the exact reason nor he/she is aware of the exact entity blocking the access to the domain. For example, a DNS server may block access to a domain based on the content category such as "Adult Content" to enforce parental control, "Violence & Terrorism" due to an external requirement imposed by an external entity (e.g., Law Enforcement Agency), etc. These content categories 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. Furthermore, the threat data used to categorize domains may sometimes misclassify domains (e.g., domains wrongly classified as Domain Generation Algorithm (DGA) by deep learning techniques, domain wrongly classified as phishing due to crowd sourcing, new domains not categorized by the threat data). A user needs to know the contact details of the IT/InfoSec team to raise a complaint.

5. When a resolver or forwarder forwards the received EDE option, the EXTRA-TEXT field only conveys the source of the error (Section 3 of [RFC8914]) and does not provide additional textual information about the cause of the error.

For both DNS filtering mechanisms described above, the DNS server can return extended error codes Blocked, Censored, Filtered, or Forged Answer defined in Section 4 of [RFC8914]. However, these codes only explain that filtering occurred but lack detail for the user to diagnose erroneous filtering.

No matter which type of response is generated (forged IP address(es), NXDOMAIN or empty answer, even with an extended error code), the user who triggered the DNS 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 such detail.

One of the other benefits of this approach is to eliminate the need to "spooof" block pages for HTTPS resources. This is achieved since clients implementing this approach would be able to display a meaningful error message, and would not need to connect to such a block page. This approach thus avoids the need to install a local root certificate authority on those IT-managed devices.

This document describes a format for computer-parsable data in the EXTRA-TEXT field of Extended DNS Errors [RFC8914].

This document does not recommend DNS filtering but provides a mechanism for better transparency to explain to the users why some DNS queries are filtered.

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 uses terms defined in DNS Terminology [RFC8499].

"Requestor" refers to the side that sends a request. "Responder" refers to an authoritative, recursive resolver or other DNS component that responds to questions. Other terminology is used here as defined in the RFCs cited by this document.

"Encrypted DNS" refers to any encrypted scheme to convey DNS messages, for example, DNS-over-HTTPS [RFC8484], DNS-over-TLS [RFC7858], or DNS-over-QUIC [I-D.ietf-dprive-dnssoquic].

3. I-JSON in EXTRA-TEXT field

Servers compliant with this specification send I-JSON data in the EXTRA-TEXT field [RFC8914] using the Internet JSON (I-JSON) message format [RFC7493].

Note that [RFC7493] was based on [RFC7159], but [RFC7159] was replaced by [RFC8259].

This document defines the following JSON names:

- c: (contact) The contact details of the IT/InfoSec team to report mis-classified DNS filtering. This field is structured as an array of contact URIs (e.g., tel, sips, https). At least one contact URI MUST be included. This field is mandatory.
- j: (justification) the textual justification for this particular DNS filtering. This field is mandatory.
- o: (organization) human-friendly name of the organization that filtered this particular DNS query. This field is optional.

New JSON names MUST be defined in the IANA registry (Section 7), consist only of lower-case ASCII characters, digits, and hyphens (that is, Unicode characters U+0061 through 007A, U+0030 through U+0039, and U+002D). These names MUST be 63 characters or shorter and it is RECOMMENDED they be as short as possible.

To reduce packet overhead the generated JSON SHOULD be as short as possible: short domain names, concise text in the values for the "j" and "o" names, and minified JSON (that is, without spaces or line breaks between JSON elements).

The JSON data can be parsed to display to the user, logged, or otherwise used to assist the end-user or IT staff with troubleshooting and diagnosing the cause of the DNS filtering.

4. Protocol Operation

4.1. Client Generating Request

When generating a DNS query, the client MUST include the OPT pseudo-RR [RFC6891] to elicit the Extended DNS Error option [RFC8914] in the DNS response.

4.2. Server Generating Response

When the DNS server filters its DNS response to an A or AAAA record query, the DNS response MAY contain an empty answer, NXDOMAIN, or a forged A or AAAA response, as desired by the DNS server. In addition, if the query contained the OPT pseudo-RR the DNS server MAY return more detail in the EXTRA-TEXT field as described in Section 4.3.

Servers may decide to return small TTL values in filtered DNS responses (e.g., 2 seconds) to handle domain category and reputation updates.

4.3. Client Processing Response

On receipt of a DNS response with an Extended DNS Error option, the following actions are performed if the EXTRA-TEXT field contains valid JSON:

- * The response MUST be received over an encrypted DNS channel. If not, the requestor MUST discard data in the EXTRA-TEXT field.
- * The response MUST be received from a DNS server which advertised EDE support via RESINFO [I-D.reddy-add-resolver-info].
- * Servers which don't support this specification might use plain text in the EXTRA-TEXT field so that requestors SHOULD properly handle both plaintext and JSON text in the EXTRA-TEXT field.
- * The DNS response MUST also contain an extended error code of "Censored", "Blocked", "Filtered" or "Forged" [RFC8914], otherwise the EXTRA-TEXT field is discarded.
- * If either of the mandatory JSON names "c" and "j" are missing or have empty values in the EXTRA-TEXT field, the entire JSON is discarded.
- * If a 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. Both of these fallback mechanisms adversely impacts security and privacy. If the DNS client has enabled opportunistic privacy profile for DoT, the DNS client MUST ignore the EXTRA-TEXT field of the EDE responses, but SHOULD process other parts of the response.

- * If a 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 MAY process the EXTRA-TEXT field of the DNS response. Note that the strict and opportunistic privacy profiles as defined in [RFC8310] only apply to DoT; there has been no such distinction made for DoH.
- * If the DNS client determines that the encrypted DNS server does not offer DNS filtering service, it MUST discard the EXTRA-TEXT field of the EDE response. For example, the DNS client can learn whether the encrypted DNS resolver performs DNS-based content filtering or not by retrieving resolver information using the method defined in [I-D.reddy-add-resolver-info].
- * When a forwarder receives an EDE option, whether or not (and how) to pass along JSON information in the EXTRA-TEXT on to their client is implementation dependent [RFC5625]. Implementations MAY choose to not forward the JSON information, or they MAY choose to create a new EDE option that conveys the information in the "c" and "j" fields encoded in the JSON object.

5. Examples

An example showing the nameserver at 'ns.example.net' that filtered a DNS "A" record query for 'example.org' is shown in Figure 1.

```
{
  "c": ["tel:+358-555-1234567", "sips:bob@bobphone.example.com",
        "https://ticket.example.com?d=example.org&t=1650560748"],
  "j": "malware present for 23 days",
  "o": "example.net Filtering Service"
}
```

Figure 1: JSON returned in EXTRA-TEXT field of Extended DNS Error response

In Figure 2 the same content is shown with minified JSON (no whitespace, no blank lines) with '\\' line wrapping per [RFC8792].

```
===== NOTE: '\ ' line wrapping per RFC 8792 =====  
  
{ "c": ["tel:+358-555-1234567", "sips:bob@bobphone.example.com", \  
  "https://ticket.example.com?d=example.org&t=1650560748"], \  
  "j": "malware present for 23 days", "o": "example.net Filtering \  
  Service" }
```

Figure 2: Minified response

6. Security Considerations

Security considerations in Section 6 of [RFC8914] apply to this document.

To minimize impact of active on-path attacks on the DNS channel, the client validates the response as described in Section 4.3.

A client might choose to display the information in the EXTRA-TEXT field if and only if the encrypted resolver has sufficient reputation, according to some local policy (e.g. user configuration, administrative configuration, or a built-in list of respectable resolvers). This limits the ability of a malicious encrypted resolver to cause harm. If the client decides not to display the all of the information in the EXTRA-TEXT field, it can be logged for diagnostics purpose and the client can only display the resolver hostname that blocked the domain and error description for the EDE code to the end-user.

When displaying the free-form text of "c" and "j", the browser SHOULD NOT make any of those elements into actionable (clickable) links.

An attacker might inject (or modify) the EDE EXTRA-TEXT field with an DNS proxy or DNS forwarder that is unaware of EDE. Such a DNS proxy or DNS forwarder will forward that attacker-controlled EDE option. To prevent such an attack, clients supporting this document MUST discard the EDE option if their DNS server does not signal EDE support via RESINFO [I-D.reddy-add-resolver-info]. As recommended in [I-D.reddy-add-resolver-info], RESINFO should be retrieved over an encrypted DNS channel or integrity protected with DNSSEC.

7. IANA Considerations

This document requests IANA to register the "application/json+structured-dns-error" media type in the "Media Types" registry [IANA-MediaTypes]. This registration follows the procedures specified in [RFC6838]:

Type name: application

Subtype name: json+structured-dns-error

Required parameters: N/A

Optional parameters: N/A

Encoding considerations: as defined in Section NN of [RFCXXXX].

Security considerations: See Section NNN of [RFCXXXX].

Interoperability considerations: N/A

Published specification: [RFCXXXX]

Applications that use this media type: Section NNNN of [RFCXXXX].

Fragment identifier considerations: N/A

Additional information: N/A

Person & email address to contact for further information: IETF,
iesg@ietf.org

Intended usage: COMMON

Restrictions on usage: none

Author: See Authors' Addresses section.

Change controller: IESG

Provisional registration? No

8. Changes

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

8.1. Changes from 02 to 03

- * Require using RESINFO [I-D.reddy-add-resolver-info] in client processing and added discussion of attack mitigation of using RESINFO.

- * Removed validation of URI domain suffix, which we can't do for some URLs (e.g., tel:), is difficult/impossible for others when 3rd party is handling level one support (e.g., sips:). Instead rely on RESINFO telling us if EDE is supported by the DNS server and, if so, expect it to properly support EDE rather than blindly forward an unknown DNS option.
- * Removed 'partial URI' text

8.2. Changes from 01 to 02

- * repurpose Extended DNS Error's EXTRA-TEXT field to carry JSON, which also means this document updates RFC8914
- * clarified DNS forwarders might forward EXTRA-TEXT without change or might rewrite "j" and "d"

8.3. Changes from 00 to 01

- * removed support for multiple responsible parties
- * one-character JSON names to minimize JSON length
- * partial URI sent in "c" and "r" names, combined with "d" name sent in JSON to minimize attack surface and minimize JSON length
- * moved EDNS(0) forgery-mitigation text, some Security Considerations text, and some other text from [I-D.reddy-dnsop-error-page] to this document

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <<https://www.rfc-editor.org/info/rfc6838>>.
- [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>>.

- [RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <<https://www.rfc-editor.org/info/rfc7159>>.
- [RFC7493] Bray, T., Ed., "The I-JSON Message Format", RFC 7493, DOI 10.17487/RFC7493, March 2015, <<https://www.rfc-editor.org/info/rfc7493>>.
- [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>>.

9.2. Informative References

- [I-D.ietf-dprive-dnssoquic]
Huitema, C., Dickinson, S., and A. Mankin, "DNS over Dedicated QUIC Connections", Work in Progress, Internet-Draft, draft-ietf-dprive-dnssoquic-12, 20 April 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-dprive-dnssoquic-12>>.
- [I-D.reddy-add-resolver-info]
Reddy, T. and M. Boucadair, "DNS Resolver Information", Work in Progress, Internet-Draft, draft-reddy-add-resolver-info-05, 13 April 2022, <<https://datatracker.ietf.org/doc/html/draft-reddy-add-resolver-info-05>>.
- [I-D.reddy-dnsop-error-page]
Reddy, T., Cook, N., Wing, D., and M. Boucadair, "DNS Access Denied Error Page", Work in Progress, Internet-Draft, draft-reddy-dnsop-error-page-08, 4 June 2021, <<https://datatracker.ietf.org/doc/html/draft-reddy-dnsop-error-page-08>>.
- [IANA-MediaTypes]
IANA, "Media Types", <<https://www.iana.org/assignments/media-types>>.
- [RFC5625] Bellis, R., "DNS Proxy Implementation Guidelines", BCP 152, RFC 5625, DOI 10.17487/RFC5625, August 2009, <<https://www.rfc-editor.org/info/rfc5625>>.

- [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>>.
- [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/info/rfc8259>>.
- [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>>.
- [RFC8792] Watsen, K., Auerswald, E., Farrel, A., and Q. Wu, "Handling Long Lines in Content of Internet-Drafts and RFCs", RFC 8792, DOI 10.17487/RFC8792, June 2020, <<https://www.rfc-editor.org/info/rfc8792>>.
- [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

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

Tirumaleswar Reddy
Akamai
Bangalore
Karnataka
India
Email: kondtir@gmail.com

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

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

Domain Name System Operations (dnsop)	U. Wisser
Internet-Draft	The Swedish Internet Foundation
Intended status: Standards Track	S. Huque
Expires: 7 September 2022	Salesforce
	6 March 2022

DNSSEC automation
draft-wisser-dnssec-automation-03

Abstract

This document describes an algorithm and a protocol to automate DNSSEC Multi-Signer [RFC8901] "Multi-Signer DNSSEC Models" setup, operations and decommissioning. Using Model 2 of the Multi-Signer specification, where each operator has their own distinct KSK and ZSK sets (or CSK sets), [RFC8078] "Managing DS Records from the Parent via CDS/CDNSKEY" and [RFC7477] "Child-to-Parent Synchronization in DNS" to accomplish this.

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 7 September 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
1.1. Out-Of-Scope	3
1.2. Notation	3
1.3. Requirements Language	3
2. Use Cases	3
2.1. Maintaining a Multi-Signer group	4
2.2. Secure Nameserver Operator Transition	4
3. Automation Models	4
3.1. Centralized	4
3.2. Decentralized	4
3.3. Capabilities	5
4. Algorithms	5
4.1. Prerequisites	5
4.2. Definitions	5
4.2.1. DS Waiting Time	5
4.2.2. DNSKEY Waiting Time	6
4.2.3. NS Waiting Time	6
4.3. Setting up a new Multi-Signer group	6
4.4. A Signer joins the Multi-Signer group	6
4.5. A signer leaves the Multi-Signer group	7
4.6. A Signer performs a ZSK rollover	7
4.7. A Signer performs a CSK or KSK rollover	8
4.8. Algorithm rollover for the whole Multi-Signer group. . .	8
5. Signers with different algorithms in one Multi-Signer group	9
6. Acknowledgements	10
7. IANA Considerations	10
8. Implementation Status	10
9. Security Considerations	10
10. Normative References	10
11. Informative References	11
Appendix A. Change History	11
A.1. Change from 01 to 02	11
A.2. Change from 02 to 03	11
Authors' Addresses	12

1. Introduction

[RFC8901] describes the necessary steps and API for a Multi-Signer DNSSEC configuration. In this document we will combine [RFC8901] with [RFC8078] and [RFC7477] to define an automatable algorithm for setting up, operating and decommissioning of a Multi-Signer DNSSEC configuration.

One of the special cases of Multi-Signer DNSSEC is the secure change of DNS operator. Using Multi-Signer Model 2 the secure change of DNS operator can be accomplished.

1.1. Out-Of-Scope

In order for any Multi-Signer group to give consistent answers across all nameservers, the data contents of the zone also have to be synchronized (in addition to infrastructure records like NS, DNSKEY, CDS etc). This content synchronization is out-of-scope for this document (although there are a number of methods that can be used, such as making the the same updates to each operator using their respective APIs, using zone transfer in conjunction with "inline signing" at each operator, etc.)

1.2. Notation

Short definitions of expressions used in this document

Signer

An entity signing a zone

Multi-Signer Group

A group of signers that sign the same zone

Controller

An entity controlling the multi-signer group. Used in the decentralized model.

1.3. Requirements Language

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].

2. Use Cases

2.1. Maintaining a Multi-Signer group

As described in [RFC8901] a Multi-Signer DNSSEC configuration has some challenges that can be overcome with the right infrastructure and following a number of steps for setup and operation.

In this document we describe, except for the initial trust, how the steps in the Multi-Signer DNSSEC setup can be automated.

2.2. Secure Nameserver Operator Transition

Changing the nameserver operator of a DNSSEC signed zone can be challenging. Currently the most common method is temporarily "going insecure". This is poor for security, and for users relying on the security of the zone. Furthermore, when DNSSEC is being used for application security functions like DANE [RFC6698], it is critical that the DNSSEC chain of trust remain unbroken during the transfer.

Multi-Signer DNSSEC Model 2 provides a mechanism for transitioning from one nameserver operator to another without "going insecure". A new operator joins the current operator in a temporary Multi-Signer group. Once that is accomplished and stable the old operator leaves the Multi-Signer group completing the transition.

3. Automation Models

Automation of the necessary steps can be categorized into two main models, centralized and decentralized. Both have pros and cons, and a zone operator should carefully choose the model that works best.

3.1. Centralized

In a centralized model the zone operator will run controller that executes all steps necessary and controls all signers.

A centralized controller needs to have authorized access to all signers. This can be achieved in a variety of different ways. For example will many service providers offer access through a REST API. Another possibility is access through Dynamic Update [RFC2136] with TSIG authentication.

3.2. Decentralized

In the decentralized models all signers will communicate with each other and execute the necessary steps on their instance only. For this signers need a specialized protocol to communicate configuration details that are not part of the zone data.

3.3. Capabilities

In order for any of the models to work the signer must support the following capabilities.

1. Add DNSKEY records (without the private key)
2. Remove (previously added) DNSKEY record(s)
3. Add CDS and CDNSKEY records for keys not in the DNSKEY set
4. Remove (previously added) CDS and CDNSKEY records
5. Add CSYNC record
6. Remove CSYNC record

4. Algorithms

4.1. Prerequisites

Each Signer to be added, including the initial Signer, must meet the following prerequisites before joining the Multi-Signer Group

1. A working setup of the zone, including DNSSEC signing.
2. Uses the same algorithm for DNSSEC signing as the Multi-Signer group uses or will use.
3. Signer or controller must be able to differentiate between its own keys and keys from others signers
4. Signer controller must be able to differentiate between NS records that are updated by itself and NS records that receive updates from other signers.
5. The domain must be covered by a CDS/CDNSKEY scanner and a CSYNC scanner. Otherwise updates to the parent zone have to be made manually.

4.2. Definitions

4.2.1. DS Waiting Time

Once the parent has picked up and published the new DS record set, the any further changes MUST to be delayed until the new DS set has propagated.

The minimum DS Waiting Time is the TTL of the DS RRset.

4.2.2. DNSKEY Waiting Time

Once the DNSKEY sets of all signers are updated, any further changes MUST to be delayed until the new DNSKEY set has propagated.

The minimum DNSKEY Waiting Time is the maximum of all DNSKEYS TTL values from all signers plus the time it takes to publish the zone on all secondaries.

4.2.3. NS Waiting Time

Once the parent has picked up and published the new NS record set, any further changes MUST be delayed until the new NS set has propagated.

The minimum NS Waiting Time is the maximum of the TTL value of the NS set in the parent zone and all NS sets from all signers.

4.3. Setting up a new Multi-Signer group

The zone is already authoritatively served by one DNS operator and is DNSSEC signed. For full automation both the KSK and ZSK or CSK must be online.

This would be a special case, a Multi-Signer group with only one signer.

4.4. A Signer joins the Multi-Signer group

1. Confirm that the incoming Signer meets the prerequisites.
2. Establish a trust mechanism between the Multi-Signer group and the Signer.
3. Add ZSK for each signer to all other Signers.
4. Calculate CDS/CDNSKEY Records for all KSKs/CSKs represented in the Multi-Signer group.
5. Configure all Signers with the compiled CDS/CDNSKEY RRSET.
6. Wait for Parent to publish the combined DS RRset.
7. Remove CDS/CDNSKEY Records from all Signers. (optional)
8. Wait maximum of DS-Wait-Time and DNSKEY-Wait-Time

9. Compile NS RRSET including all NS records from all Signers.
 10. Configure all Signers with the compiled NS RRSET.
 11. Compare NS RRSET of the Signers to the Parent, if there is a difference publish CSYNC record with NS and A and AAAA bit set on all signers.
 12. Wait for Parent to publish NS.
 13. Remove CSYNC record from all signers. (optional)
- 4.5. A signer leaves the Multi-Signer group
1. Remove exiting Signer's NS records from remaining Signers
 2. Compare NS RRSET of the Signers to the Parent, if there is a difference publish CSYNC record with NS and A and AAAA bit set on remaining signers.
 3. Wait for Parent to publish NS RRSET.
 4. Remove CSYNC record from all signers. (optional)
 5. Wait NS-Wait-Time
 6. Stop the exiting Signer from answering queries.
 7. Calculate CDS/CDNSKEY Records for KSKs/CSKs published by the remaining Signers.
 8. Configure remaining Signers with the compiled CDS/CDNSKEY RRSET.
 9. Remove ZSK of the exiting Signer from remaining Signers.
 10. Wait for Parent to publish the updated DS RRset.
 11. Remove CDS/CDNSKEY set from all signers. (Optional)
- 4.6. A Signer performs a ZSK rollover
1. The signer introduces the new ZSK in its own DNSKEY RRset.
 2. Update all signers with the new ZSK.
 3. Wait DNSKEY-Wait-Time
 4. Signer can start using the new ZSK.

5. When the old ZSK is not used in any signatures by the signer, the signer can remove the old ZSK from its DNSKEY RRset.
 6. Remove ZSK from DNSKEY RRset of all signers.
- 4.7. A Signer performs a CSK or KSK rollover
1. Signer publishes new CSK / KSK in its own DNSKEY RRset.
 2. In case of CSK, add CSK to DNSKEY set of all other Signers.
 3. Signer signs DNSKEY RRset with old and new CSK / KSK.
 4. Calculate new CDS/CDNSKEY RRset and publish on all signers.
 5. Wait for parent to pickup and publish new DS RR set.
 6. Wait DS-Wait-Time + DNSKEY-Wait-Time
 7. Signer removes old CSK/KSK from its DNSKEY RR set. And removes all signatures done with this key.
 8. In case of CSK, remove old CSK from DNSKEY set of all other signers.
 9. Calculate new CDS/CDNSKEY RRset and publish on all signers.
 10. Wait for parent to pickup and publish new DS RR set.
 11. Remove CDS/CDNSKEY RR sets from all signers.
- 4.8. Algorithm rollover for the whole Multi-Signer group.
1. All signers publish KSK and ZSK or CSK using the new algorithm.
 2. All signers sign all zone data with the new keys.
 3. Wait until all signers have signed all data with the new key(s).
 4. Add new ZSK of each signer to all other Signers.
 5. Calculate new CDS/CDNSKEY RRset and publish on all signers.
 6. Wait for parent to pickup and publish new DS RR set.
 7. Wait DS-Wait-Time + DNSKEY-Wait-Time

8. Removes all keys and signatures which are using the old algorithm.
 9. Calculate new CDS/CDNSKEY RRset and publish on all signers.
 10. Wait for parent to pickup and publish new DS RR set.
 11. Remove CDS/CDNSKEY RR sets from all signers.
5. Signers with different algorithms in one Multi-Signer group

Section 2.2 of [RFC4035] states that a signed zone MUST include a DNSKEY for each algorithm present in the zone's DS RRset and expected trust anchors for the zone.

A setup where different signers use different key algorithms therefore violates [RFC4035].

According to Section 5.11 of [RFC6840] validators SHOULD NOT insist that all algorithms signaled in the DS RRset work, and they MUST NOT insist that all algorithms signaled in the DNSKEY RRset work.

So a Multi-Signer setup where different signers use different key algorithms should still validate.

This could be an acceptable risk in a situation where going insecure is not desirable or impossible and name servers have to be changed between operators which only support distinct set of key algorithms.

We have to consider the following scenarios

Validator supports both algorithms

Validation should be stable through all stages of the multi-signer algorithms.

Validator supports none of the algorithms

The validator will treat the zone as unsigned. Resolution should work through all stages of the multi-signer algorithms.

Validator supports only one of the algorithms

The validator will not be able to validate the DNSKEY RR set or any data from one of the signers. So in some cases the validator will consider the zone bogus and reply with a SERVFAIL response code.

The later scenario can be mitigated, but not fully eliminated, by selecting two well supported algorithms.

6. Acknowledgements

The authors would like to thank the following for their review of this work and their valuable comments: Steve Crocker, Eric Osterweil, Roger Murray, Jonas Andersson, Peter Thomassen.

7. IANA Considerations

8. Implementation Status

One implementation of a centralized controller which supports updates through Dynamic DNS or REST API's of several vendors has been implemented by the Swedish Internet Foundation.

The code can be found as part of the Multi-Signer project on Github <https://github.com/DNSSEC-Provisioning/multi-signer-controller>

9. Security Considerations

Every step of the multi-signer algorithms has to be carefully executed at the right time and date. Any failure could resolve in the loss of resolution for the domain.

Independently of the chosen model, it is crucial that only authorized entities will be able to change the zone data. Some providers or software installation allow to make more specific configuration on the allowed changes. All extra steps to allows as little access to change zone data as possible should be taken.

If used correctly the multi-signer algorithm will strengthen the DNS security by avoiding "going insecure" at any stage of the domain life cycle.

10. 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>>.
- [RFC2136] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, DOI 10.17487/RFC2136, April 1997, <<https://www.rfc-editor.org/info/rfc2136>>.

- [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>>.
- [RFC6840] Weiler, S., Ed. and D. Blacka, Ed., "Clarifications and Implementation Notes for DNS Security (DNSSEC)", RFC 6840, DOI 10.17487/RFC6840, February 2013, <<https://www.rfc-editor.org/info/rfc6840>>.
- [RFC7477] Hardaker, W., "Child-to-Parent Synchronization in DNS", RFC 7477, DOI 10.17487/RFC7477, March 2015, <<https://www.rfc-editor.org/info/rfc7477>>.
- [RFC8078] Gudmundsson, O. and P. Wouters, "Managing DS Records from the Parent via CDS/CDNSKEY", RFC 8078, DOI 10.17487/RFC8078, March 2017, <<https://www.rfc-editor.org/info/rfc8078>>.

11. Informative References

- [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <<https://www.rfc-editor.org/info/rfc6698>>.
- [RFC8901] Huque, S., Aras, P., Dickinson, J., Vcelak, J., and D. Blacka, "Multi-Signer DNSSEC Models", RFC 8901, DOI 10.17487/RFC8901, September 2020, <<https://www.rfc-editor.org/info/rfc8901>>.

Appendix A. Change History

A.1. Change from 01 to 02

1. Trying to fix wording to be more precise
2. Added algorithm for ZSK rollover
3. Added algorithm for KSK rollover
4. Added algorithm for algorithm rollover

A.2. Change from 02 to 03

1. Fix sequence of steps in the joining procedure
2. Explicit handling of CSK cases in CSK/ KSK rollover

Authors' Addresses

Ulrich Wisser
The Swedish Internet Foundation
Box 92073
SE-12007 Stockholm
Sweden
Email: ulrich@wisser.se
URI: <https://www.internetstiftelsen.se>

Shumon Huque
Salesforce
415 Mission Street, 3rd Floor
San Francisco, CA 94105
United States of America
Email: shuque@gmail.com