Workgroup: Network Working Group
Internet-Draft:
draft-ietf-dnsop-avoid-fragmentation-15
Published: 15 September 2023
Intended Status: Best Current Practice
Expires: 18 March 2024
Authors: K. Fujiwara
JPRS
P. Vixie
AWS Security

Fragmentation Avoidance in DNS

Abstract

EDNS0 enables a DNS server to send large responses using UDP and is widely deployed. Large DNS/UDP responses are fragmented, and IP fragmentation has exposed weaknesses in application protocols. 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 techniques 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 18 March 2024.

Table of Contents

1. Introduction

DNS has an EDNS0 [RFC6891] mechanism. It enables a DNS server to send large responses using UDP. EDNS0 is now widely deployed, and DNS over UDP relies on IP fragmentation when the EDNS buffer size is set to a value larger than the path MTU.

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 that implementations set the "Don't Fragment (DF) bit" [RFC0791] on IPv4 and not use the "Fragment header" [RFC8200] on IPv6 in DNS/UDP messages in order to avoid IP fragmentation. It also describes how to avoid packet losses due to DF bit and small MTU links.

A path MTU different from the recommended value could be obtained from static configuration, or server routing hints, or some future discovery protocol; that would be the subject of a future specification and is beyond our scope here.

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 server, 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])

In this document, the term "Path MTU discovery" includes both Classical Path MTU discovery [RFC1191], [RFC8201], and Packetization Layer Path MTU discovery [RFC8899].

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

3. Proposal to avoid IP fragmentation in DNS

These recommendations are intended for nodes with global IP addresses on the Internet. Private networks or local networks are out of the scope of this document.

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

3.1. Recommendations for UDP responders

R1. UDP responders SHOULD send DNS responses without "Fragment header" [RFC8200] on IPv6.

R2. UDP responders MAY set IP "Don't Fragment flag (DF) bit" [RFC0791] on IPv4.

R3. UDP responders SHOULD compose response packets that fit in the offered requestor's maximum UDP payload size [RFC6891], the interface MTU, and the RECOMMENDED maximum DNS/UDP payload size 1400.

R4. If the UDP responder detects an immediate error indicating 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 with the TC bit set.

R5. UDP responders SHOULD limit the response size when UDP responders are located on small MTU (<1500) networks.

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

3.2. Recommendations for UDP requestors

R6. UDP requestors SHOULD limit the requestor's maximum UDP payload size to the RECOMMENDED size of 1400 or a smaller size.

R7. UDP requestors MAY drop fragmented DNS/UDP responses without IP reassembly to avoid cache poisoning attacks.

R8. DNS responses may be dropped by IP fragmentation. Upon a timeout, to avoid resolution failures, UDP requestors MAY retry using TCP or UDP with a smaller EDNS requestor's maximum UDP payload size per local policy.

4. Recommendations for 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,

R9. Use a smaller number of name servers (13 may be too large)

R10. Use a smaller number of A/AAAA RRs for a domain name

R11. Use minimal-responses configuration: Some implementations have a 'minimal responses' configuration option that causes DNS servers to make response packets smaller, containing only mandatory and required data (Appendix C).

R12. Use a smaller signature / public key size algorithm for DNSSEC. Notably, the signature sizes of ECDSA and EdDSA are smaller than those usually used for RSA.

5. Considerations

5.1. Protocol compliance

Prior research [Fujiwara2018] has shown that some authoritative servers ignore the EDNS0 requestor's maximum UDP payload size, and return large UDP responses.

It is also well known that some authoritative servers 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.

6. IANA Considerations

This document has no IANA actions.

7. Security Considerations

7.1. On-path fragmentation on IPv4

If the Don't Fragment (DF) bit is not set, on-path fragmentation may happen on IPv4, and be vulnerable as shown in Appendix A. To avoid this, recommendation R7 should be used to discard the fragmented responses and retry by TCP.

In the future, recommendation R2 could be changed from "MAY" to "SHOULD".

7.2. Small MTU network

When avoiding fragmentation, a DNS/UDP requestor behind a small-MTU network may experience UDP timeouts which would reduce performance and which may lead to TCP fallback. This would indicate prior

reliance upon IP fragmentation, which is universally considered to be harmful to both the performance and stability of applications, endpoints, and gateways. Avoiding IP fragmentation will improve operating conditions overall, and the performance of DNS/TCP has increased and will continue to increase.

If a UDP response packet is dropped (for any reason), it increases the attack window for poisoning the requestor's cache.

8. Acknowledgments

The author would like to specifically thank Paul Wouters, Mukund Sivaraman, Tony Finch, Hugo Salgado, Peter van Dijk, Brian Dickson, Puneet Sood, Jim Reid, Petr Spacek, Andrew McConachie, Joe Abley, Daisuke Higashi, Joe Touch and Wouter Wijngaards for extensive review and comments.

9. References

9.1. Normative References

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

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <https://www.rfc-editor.org/rfc/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/rfc/rfc2119>.

[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/rfc/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/rfc/rfc8174>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/rfc/rfc8200>.

[RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201,

DOI 10.17487/RFC8201, July 2017, <https://www.rfc-editor.org/rfc/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/rfc/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/rfc/rfc8899>.

9.2. 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.

[I-D.ietf-dnsop-glue-is-not-optional] Andrews, M. P., Huque, S., Wouters, P., and D. Wessels, "DNS Glue Requirements in Referral Responses", Work in Progress, Internet-Draft, draft-ietf-dnsop-glue-is-not-optional-09, 14 June 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-glue-is-not-optional-09>.

[I-D.ietf-dnsop-svcb-https] Schwartz, B. M., Bishop, M., and E. Nygren, "Service binding and parameter specification via the DNS (DNS SVCB and HTTPS RRs)", Work in Progress, Internet-Draft, draft-ietf-dnsop-svcb-https-12, 11 March 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-svcb-https-12>.

[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>.

[RFC2308] Andrews, M., "Negative Caching of DNS Queries (DNS NCACHE)", RFC 2308, DOI 10.17487/RFC2308, March 1998, <https://www.rfc-editor.org/rfc/rfc2308>.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, DOI 10.17487/RFC2782, February 2000, <https://www.rfc-editor.org/rfc/rfc2782>.

[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/rfc/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/rfc/rfc5155>.

[RFC7739] Gont, F., "Security Implications of Predictable Fragment Identification Values", RFC 7739, DOI 10.17487/RFC7739, February 2016, <https://www.rfc-editor.org/rfc/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/rfc/rfc8085>.

[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/rfc/rfc8900>.

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 avoids 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 requestor's maximum UDP payload size discussions

There are many discussions for default path MTU size and requestor's maximum UDP payload size.

* The minimum MTU for an IPv6 interface is 1280 octets (see Section 5 of [RFC8200]). So, we can use it as the 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, the 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 so they 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] and 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 requestor's UDP payload size to 1472 octets for IPv4, and 1452 octets for IPv6.

Appendix C. Minimal-responses

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

Under the minimal-responses configuration, a DNS server composes responses containing only nessesary RRs. For delegations, see [I-D.ietf-dnsop-glue-is-not-optional]. In case of a non-existent domain name or non-existent type, the authority section will contain an SOA record and the answer section is empty. (defined in Section 2 of [RFC2308]).

Some resource records (MX, SRV, SVCB, HTTTPS) require additional A, AAAA, and SVCB records in the Additional Section defined in [RFC1035], [RFC2782] and [I-D.ietf-dnsop-svcb-https].

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

Appendix D. Known Implementations

This section records the status of known implementations of these best practices defined by this specification at the time of publication, and any deviation from the specification.

Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors.

D.1. BIND 9

BIND 9 does not implement the recommendations 1 and 2 in Section 3.1.

BIND 9 on Linux sets IP_MTU_DISCOVER to IP_PMTUDISC_OMIT with a fallback to IP_PMTUDISC_DONT.

BIND 9 on systems with IP_DONTFRAG (such as FreeBSD), IP_DONTFRAG is disabled.

Accepting PATH MTU Discovery for UDP is considered harmful and dangerous. BIND 9's settings avoid attacks to path MTU discovery.

For recommendation 3, BIND 9 will honor the requestor's size up to the configured limit (max-udp-size). The UDP response packet is bound to be between 512 and 4096 bytes, with the default set to 1232. BIND 9 supports the requestor's size up to the configured limit (max-udp-size).

In the case of recommendation 4, and the send fails with EMSGSIZE, BIND 9 set the TC bit and try to send a minimal answer again.

In the first recommendation of Section 3.2, BIND 9 uses the edns-buf-size option, with the default of 1232.

BIND 9 does implement recommendation 2 of Section 3.2.

For recommendation 3, after two UDP timeouts, BIND 9 will fallback to TCP.

D.2. Knot DNS and Knot Resolver

Both Knot servers set IP_PMTUDISC_OMIT to avoid path MTU spoofing. UDP size limit is 1232 by default.

Fragments are ignored if they arrive over an XDP interface.

TCP is attempted after repeated UDP timeouts.

Minimal responses are returned and are currently not configurable.

Smaller signatures are used, with ecdsap256sha256 as the default.

D.3. PowerDNS Authoritative Server, PowerDNS Recursor, PowerDNS dnsdist

- IP_PMTUDISC_OMIT with fallback to IP_PMTUDISC_DONT
- default EDNS buffer size of 1232, no probing for smaller sizes
- no handling of EMSGSIZE
- Recursor: UDP timeouts do not cause a switch to TCP. "Spoofing nearmisses" do.

D.4. PowerDNS Authoritative Server

*the default DNSSEC algorithm is 13

*responses are minimal, this is not configurable

D.5. Unbound

Unbound sets IP_MTU_DISCOVER to IP_PMTUDISC_OMIT with fallback to IP_PMTUDISC_DONT. It also disables IP_DONTFRAG on systems that have it, but not on Apple systems. On systems that support it Unbound sets IPV6_USE_MIN_MTU, with a fallback to IPV6_MTU at 1280, with a fallback to IPV6_USER_MTU. It also sets IPV6_MTU_DISCOVER to IPV6_PMTUDISC_OMIT with a fallback to IPV6_PMTUDISC_DONT.

Unbound requests UDP size 1232 from peers, by default. The requestors size is limited to a max of 1232.

After some timeouts, Unbound retries with a smaller size, if that is smaller, at size 1232 for IPv6 and 1472 for IPv4. This does not do anything since the flag day change to 1232.

Unbound has minimal responses as an option, default on.

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
AWS Security
11400 La Honda Road
Woodside, CA, 94062
United States of America

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