Network Working Group K. Fujiwara
Internet-Draft JPRS
Intended status: Best Current Practice P. Vixie
Expires: January 1, 2021 Farsight
June 30, 2020

Fragmentation Avoidance in DNS
draft-ietf-dnsop-avoid-fragmentation-00

Abstract

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.

Table of Contents

1. Introduction

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

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

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

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

A DNS message receiver cannot trust fragmented UDP datagrams primarily due to the small amount of entropy provided by UDP port numbers and DNS message identifiers, each of which being only 16 bits in size, and both likely being in the first fragment of a packet, if fragmentation occurs. By comparison, TCP is considered resistant against IP fragmentation attacks because TCP has a 32-bit sequence number and 32-bit acknowledgment number in each segment. In TCP, fragmentation should be avoided for performance reasons, whereas for UDP, fragmentation should be avoided for resiliency and authenticity reasons.

[I-D.ietf-intarea-frag-fragile] 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.

This document proposes to avoid IP fragmentation in DNS/UDP.

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])

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

3. Proposal to avoid IP fragmentation in DNS

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.

The minimum MTU for an IPv4 interface is 68 octets, and all receivers must be able to receive and reassemble datagrams at least 576 octets in size (see Section 2.1, NOTE 1 of [I-D.ietf-intarea-frag-fragile]). The minimum MTU for an IPv6 interface is 1280 octets (see Section 5 of [RFC8200]). These are theoretic limits and no modern networks implement them. In practice, the smallest MTU witnessed in the operational DNS community is 1500 octets, the Ethernet maximum payload size. While many non-Ethernet networks exist such as Packet on SONET (PoS), Fiber Distributed Data Exchange (FDDI), and Ethernet Jumbo Frame, there is currently no reliable way of discovering such links in an IP transmission path. Absent some kind of path MTU discovery result or a static configuration by the server or system operator, a conservative estimate must be chosen, even if it is less efficient than the path MTU would have been had that been discoverable.

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

- UDP requestors and responders SHOULD send DNS responses with IP_DONTFRAG / IPV6_DONTFRAG [RFC3542] options, which will yield either a silent timeout, or a network (ICMP) error, if the path MTU is exceeded. Upon a timeout, UDP requestors may retry using TCP or UDP, per local policy.

- The estimated maximum DNS/UDP payload size SHOULD be the discovered or estimated path MTU minus the estimated header space. Path MTU discovery [RFC1191], [RFC8201] and [I-D.ietf-tsvwg-datagram-plpmtud] may discover real path MTU value to destinations. One method to retrieve path MTU value is described in Appendix A. When discovered path MTU information is not available, a message sender SHOULD use the default maximum DNS/UDP payload size described in following section.

- The maximum buffer size offered by an EDNS0 initiator SHOULD be no larger than the estimated maximum DNS/UDP payload size. If the desired response cannot be reasonably expected to fit into a buffer of that size, the initiator should use TCP instead of UDP.

- Responders SHOULD compose UDP responses that result in IP packets that do not exceed the path MTU to the requestor. Thus, if the requestor offers a buffer size larger than responder's discovered

or estimated maximum DNS/UDP payload size, then the responder will behave as though the requestor had specified a buffer size equal to the responder's estimated maximum DNS/UDP payload size.

- Fragmented DNS/UDP messages may be dropped without IP reassembly. An ICMP error should be sent in this case, with rate limiting to prevent this logic from becoming a DDoS amplification vector. If rate limiting is not possible, then no ICMP error should be sent. (This is a countermeasure against DNS spoofing attacks using IP fragmentation.)

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

4. Maximum DNS/UDP payload size

- Most of the Internet and especially the inner core has an MTU of at least 1500 octets. An operator of a full resolver would be well advised to measure their path MTU to several authority name servers and to a random sample of their expected stub resolver client networks, to find the upper boundary on IP/UDP packet size in the average case. This limit should not be exceeded by most messages received or transmitted by a full resolver, or else fallback to TCP will occur too often. An operator of authoritative servers would also be well advised to measure their path MTU to several full-service resolvers. The Linux tool "tracepath" can be used to measure the path MTU to well known authority name servers such as [a-m].root-servers.net or [a-m].gtld-servers.net. If the reported path MTU is for example no smaller than 1460, then the maximum DNS/UDP payload would be 1432 for IP4 (which is 1460 - IP4 header(20) - UDP header(8)) and 1412 for IP6 (which is 1460 - IP6 header(40) - UDP header(8)). To allow for possible IP options and distant tunnel overhead, a useful default for maximum DNS/UDP payload size would be 1400.

- [RFC4035] defines that "A security-aware name server MUST support the EDNS0 message size extension, MUST support a message size of at least 1220 octets". Then, the smallest number of the maximum DNS/UDP payload size is 1220.

- DNS flag day 2020 proposed 1232 as an EDNS buffer size. [DNSFlagDay2020] By the above reasoning, this proposal is either too small or too large.

5. Incremental deployment

The proposed method supports incremental deployment.

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

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

6. Request to zone operators and DNS server operators

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

- o Use smaller number of name servers (13 may be too large)

- o Use smaller number of A/AAAA RRs for a domain name

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

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

7. Considerations

7.1. Protocol compliance

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

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

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

8. IANA Considerations

This document has no IANA actions.

9. Security Considerations

10. Acknowledgments

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

11. References

11.1. Normative References

[I-D.ietf-intarea-frag-fragile]
Bonica, R., Baker, F., Huston, G., Hinden, R., Troan, O., and F. Gont, "IP Fragmentation Considered Fragile", draft-ietf-intarea-frag-fragile-17 (work in progress), September 2019.

[I-D.ietf-tsvwg-datagram-plpmtud]
Fairhurst, G., Jones, T., Tuexen, M., Ruengeler, I., and T. Voelker, "Packetization Layer Path MTU Discovery for Datagram Transports", draft-ietf-tsvwg-datagram-plpmtud-22 (work in progress), June 2020.

[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>.

[RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, DOI 10.17487/RFC5155, March 2008, <https://www.rfc-editor.org/info/rfc5155>.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <https://www.rfc-editor.org/info/rfc6891>.

[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>.

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

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

[RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <https://www.rfc-editor.org/info/rfc8201>.

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

11.2. Informative References

[Brandt2018] Brandt, M., Dai, T., Klein, A., Shulman, H., and M. Waidner, "Domain Validation++ For MitM-Resilient PKI", Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security , 2018.

[DNSFlagDay2020] "DNS flag day 2020", n.d., <https://dnsflagday.net/2020/>.

[Fujiwara2018]
Fujiwara, K., "Measures against cache poisoning attacks using IP fragmentation in DNS", OARC 30 Workshop , 2019.

[Herzberg2013]
Herzberg, A. and H. Shulman, "Fragmentation Considered Poisonous", IEEE Conference on Communications and Network Security , 2013.

[Hlavacek2013]
Hlavacek, T., "IP fragmentation attack on DNS", RIPE 67 Meeting , 2013, <https://ripe67.ripe.net/presentations/240-ipfragattack.pdf>.

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

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

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

Appendix B. Minimal-responses

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

Under the minimal-responses configuration, DNS servers compose response messages using only RRSets corresponding to queries. In case of delegation, DNS servers compose response packets with delegation NS RRSet in authority section and 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 RRSet and signatures are added in authority section. Details are defined in [RFC4035] and [RFC5155].

Authors' Addresses

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

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

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

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