Authors: K. Fujiwara   P. Vixie
         JPRS   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

Copyright Notice

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 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 using the "Fragment header"
[RFC8200] on IPv6 in DNS/UDP messages in order to avoid IP
fragmentation, and 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, 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

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

*UDP responders are RECOMMENDED to set IP "Don't Fragment flag (DF) bit" [RFC0791] on IPv4.

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

*If the UDP responder detects an 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 SHOULD limit 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

*UDP requestors SHOULD limit the requestor's maximum UDP payload size to the RECOMMENDED size of 1400 or smaller 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, to avoid name resolution fails, UDP requestors MAY retry using TCP or UDP with a smaller 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,

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

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

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

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

   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.

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, Peter van Dijk, Andrew
   McConachie, Joe Abley, Daisuke Higashi and Joe Touch 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/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>.

[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>.

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

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

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

[RFC8899]   Fairhurst, G., Jones, T., Tüxen, M., Rüngeler, I., and T.
            Völker, "Packetization Layer Path MTU Discovery for
            Datagram Transports", RFC 8899, DOI 10.17487/RFC8899,
            September 2020, <https://www.rfc-editor.org/info/
            rfc8899>.

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.

[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>.

[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>.

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 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]). Then, 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 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 Buffer size to
IPv4 1472 octets and IPv6 1452 octets.

Appendix C. Minimal-responses

Some implementations have a '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 the
case of delegation, DNS servers compose response packets with
delegation NS RRSet in the authority section and in-domain (in-zone
and below-zone) glue in the additional data section. In case of a
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 the answer section, or the
corresponding DS RRSet and signatures are added in the 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
AWS Security
11400 La Honda Road
Woodside, CA, 94062
United States of America

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