Internet Engineering Task Force (IETF)
Request for Comments: 9715
Category: Informational
ISSN: 2070-1721

K. Fujiwara
JPRS
P. Vixie
AWS Security

January 2025

IP Fragmentation Avoidance in DNS over UDP

Abstract

The widely deployed Extension Mechanisms for DNS (EDNS(0)) feature in the DNS enables a DNS receiver to indicate its received UDP message size capacity, which supports the sending of large UDP responses by a DNS server. Large DNS/UDP messages are more likely to be fragmented, and IP fragmentation has exposed weaknesses in application protocols. It is possible to avoid IP fragmentation in DNS by limiting the response size where possible and signaling the need to upgrade from UDP to TCP transport where necessary. This document describes techniques to avoid IP fragmentation in DNS.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9715.

Table of Contents

1. Introduction
2. Terminology
3. How to Avoid IP Fragmentation in DNS
   3.1. Proposed Recommendations for UDP Responders
   3.2. Proposed Recommendations for UDP Requestors
4. Proposed Recommendations for DNS Operators
5. Protocol Compliance Considerations
6. IANA Considerations

7. Security Considerations
7.1. On-Path Fragmentation on IPv4
7.2. Small MTU Network
7.3. Weaknesses of IP Fragmentation
7.4. DNS Security Protections
7.5. Possible Actions for Resolver Operators
8. References
8.1. Normative References
8.2. Informative References
Appendix A. Details of Requestor's Maximum UDP Payload Size Discussions
Appendix B. Minimal Responses
Appendix C. Known Implementations
C.1. BIND 9
C.2. Knot DNS and Knot Resolver
C.3. PowerDNS Authoritative Server, PowerDNS Recursor, and PowerDNS dnsdist
C.4. PowerDNS Authoritative Server
C.5. Unbound
Acknowledgments
Authors' Addresses

1. Introduction

This document was originally intended to be a Best Current Practice, but due to operating system and socket option limitations, some of the recommendations have not yet gained real-world experience; therefore, this document is Informational. It is expected that, as operating systems and implementations evolve, we will gain more experience with the recommendations and will publish an updated document as a Best Current Practice in the future.

DNS has an EDNS(0) mechanism [RFC6891]. The widely deployed EDNS(0) feature in the DNS enables a DNS receiver to indicate its received UDP message size capacity, which supports the sending of large UDP responses by a DNS server. DNS over UDP invites IP fragmentation when a packet is larger than the Maximum Transmission Unit (MTU) of some network in the packet's path.

Fragmented DNS UDP responses have systemic weaknesses, which expose the requestor to DNS cache poisoning from off-path attackers (see Section 7.3 for references and details).

[RFC8900] states 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 by segmenting data into packets that are smaller than or equal to the Maximum Segment Size (MSS). For each transmitted segment, the size of the IP and TCP headers is known, and the IP packet size can be chosen to keep it within the estimated MTU and the MSS. This takes advantage of the elasticity of the TCP's packetizing process, depending on 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, so we must make more conservative estimates about available UDP payload space.

[RFC7766] states that all general-purpose DNS implementations MUST support both UDP and TCP transport.

DNS transaction security [RFC8945] [RFC2931] does protect against the security risks of fragmentation, and it protects delegation responses. But [RFC8945] has limited applicability due to key distribution requirements, and there is little if any deployment of [RFC2931].

This document describes various techniques to avoid IP fragmentation of UDP packets in DNS. This document is primarily applicable to DNS use on the global Internet.

In contrast, a path MTU that deviates from the recommended value might be obtained through static configuration, server routing hints, or a future discovery protocol. However, addressing this falls outside the scope of this document and may be the subject of future specifications.

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.

The definitions of "requestor" and "responder" are per [RFC6891]:

> "Requestor" refers to the side that sends a request. "Responder" refers to an authoritative, recursive resolver or other DNS component that responds to questions.

The definition of "path MTU" is per [RFC8201]:

> path MTU [is] the minimum link MTU of all the links in a path between a source node and a destination node.

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" [RFC9499].

3. How 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. Proposed Recommendations for UDP Responders

R1. UDP responders should not use IPv6 fragmentation [RFC8200].

R2. UDP responders should configure their systems to prevent fragmentation of UDP packets when sending replies, provided it can be done safely. The mechanisms to achieve this vary across different operating systems.

For BSD-like operating systems, the IP Don’t Fragment (DF) flag bit [RFC0791] can be used to prevent fragmentation. In contrast, Linux systems do not expose a direct API for this purpose and require the use of Path MTU socket options (IP_MTU_DISCOVER) to manage fragmentation settings. However, it is important to note that enabling IPv4 Path MTU Discovery for UDP in current Linux versions is considered harmful and dangerous. For more details, see Appendix C.

R3. UDP responders should compose response packets that fit in the minimum of the offered requestor’s maximum UDP payload size [RFC6891], the interface MTU, the network MTU value configured

by the knowledge of the network operators, and the RECOMMENDED maximum DNS/UDP payload size 1400. For more details, see Appendix A.

R4. If the UDP responder detects an immediate error indicating that the UDP packet exceeds the path MTU size, the UDP responder may recreate response packets that fit in the path MTU size or with the TC bit set.

The cause and effect of the TC bit are unchanged [RFC1035].

3.2. Proposed Recommendations for UDP Requestors

R5. UDP requestors should limit the requestor's maximum UDP payload size to fit in the minimum of the interface MTU, the network MTU value configured by the network operators, and the RECOMMENDED maximum DNS/UDP payload size 1400. A smaller limit may be allowed. For more details, see Appendix A.

R6. UDP requestors should drop fragmented DNS/UDP responses without IP reassembly to avoid cache poisoning attacks (at the firewall function).

R7. DNS responses may be dropped by IP fragmentation. It is recommended that requestors eventually try alternative transport protocols.

4. Proposed Recommendations for DNS Operators

Large DNS responses are typically the result of zone configuration. People who publish information in the DNS should seek configurations resulting in small responses. For example:

R8. Use a smaller number of name servers.

R9. Use a smaller number of A/AAAA RRs for a domain name.

R10. Use minimal-responses configuration: Some implementations have a 'minimal responses' configuration option that causes DNS servers to make response packets smaller by containing only mandatory and required data (Appendix B).

R11. Use a smaller signature / public key size algorithm for DNSSEC. Notably, the signature sizes of the Elliptic Curve Digital Signature Algorithm (ECDSA) and Edwards-curve Digital Signature Algorithm (EdDSA) are smaller than those of equivalent cryptographic strength using RSA.

It is difficult to determine a specific upper limit for R8, R9, and R11, but it is sufficient if all responses from the DNS servers are below the size of R3 and R5.

5. Protocol Compliance Considerations

Some authoritative servers deviate from the DNS standard as follows:

* Some authoritative servers ignore the EDNS(0) requestor's maximum UDP payload size and return large UDP responses [Fujiwara2018].

* 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) flag bit is not set, on-path fragmentation may happen on IPv4, and it can lead to vulnerabilities as shown in Section 7.3. To avoid this, R6 needs to be used to discard the fragmented responses and retry using TCP.

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 may lead to TCP fallback. This would indicate prior reliance upon IP fragmentation, which is 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 in transit, up to and including the network stack of the initiator, it increases the attack window for poisoning the requestor’s cache.

7.3. Weaknesses of IP Fragmentation

"Fragmentation Considered Poisonous" [Herzberg2013] notes 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] note that off-path attackers can intervene in the Path MTU Discovery [RFC1191] to cause authoritative servers to produce fragmented responses. [RFC7739] states the security implications of predictable fragment identification values.

Section 3.2 of [RFC8085] states 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 is only 16 bits in size, and both are likely to be in the first fragment of a packet if fragmentation occurs. By comparison, the 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.

7.4. DNS Security Protections

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.

7.5. Possible Actions for Resolver Operators

Because this document is published as Informational rather than a

Best Current Practice, this section presents steps that resolver operators can take to avoid vulnerabilities related to IP fragmentation.

To avoid vulnerabilities related to IP fragmentation, implement R5 and R6.

Specifically, configure the firewall functions protecting the full-service resolver to discard incoming DNS response packets with a non-zero Fragment Offset (FO) or a More Fragments (MF) flag bit of 1 on IPv4, and discard packets with IPv6 Fragment Headers. (If the resolver's IP address is not dedicated to the DNS resolver and uses UDP communication that relies on IP Fragmentation for purposes other than DNS, discard only the first fragment that contains the UDP header from port 53.)

The most recent resolver software is believed to implement R7.

Even if R7 is not implemented, it will only result in a name resolution error, preventing attacks from leading to malicious sites.

8. References

8.1. Normative References

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[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>.

[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>.

[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>.

[RFC7766] Dickinson, J., Dickinson, S., Bellis, R., Mankin, A., and D. Wessels, "DNS Transport over TCP - Implementation Requirements", RFC 7766, DOI 10.17487/RFC7766, March 2016, <https://www.rfc-editor.org/info/rfc7766>.

[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>.

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

[RFC8945] Dupont, F., Morris, S., Vixie, P., Eastlake 3rd, D.,
Gudmundsson, O., and B. Wellington, "Secret Key
Transaction Authentication for DNS (TSIG)", STD 93,
RFC 8945, DOI 10.17487/RFC8945, November 2020,
<https://www.rfc-editor.org/info/rfc8945>.

[RFC9499] Hoffman, P. and K. Fujiwara, "DNS Terminology", BCP 219,
RFC 9499, DOI 10.17487/RFC9499, March 2024,
<https://www.rfc-editor.org/info/rfc9499>.

8.2. 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, pp. 2060-2076,
DOI 10.1145/3243734.3243790, October 2018,
<https://dl.acm.org/doi/10.1145/3243734.3243790>.

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

[Fujiwara2018]
Fujiwara, K., "Measures against DNS cache poisoning
attacks using IP fragmentation", OARC 30 Workshop, 2019,
<https://indico.dns-
oarc.net/event/31/contributions/692/attachments/660/1115/
fujiwara-5.pdf>.

[Herzberg2013]
Herzberg, A. and H. Shulman, "Fragmentation Considered
Poisonous, or: One-domain-to-rule-them-all.org", IEEE
Conference on Communications and Network Security (CNS),
DOI 10.1109/CNS.2013.6682711, 2013,
<https://ieeexplore.ieee.org/document/6682711>.

[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, <https://indico.dns-
oarc.net/event/37/contributions/806/
attachments/782/1366/2021-02-04-dns-flag.pdf>.

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

[RFC2308] Andrews, M., "Negative Caching of DNS Queries (DNS

NCACHE)", RFC 2308, DOI 10.17487/RFC2308, March 1998, <https://www.rfc-editor.org/info/rfc2308>.

[RFC2671] Vixie, P., "Extension Mechanisms for DNS (EDNS0)", RFC 2671, DOI 10.17487/RFC2671, August 1999, <https://www.rfc-editor.org/info/rfc2671>.

[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/info/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/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>.

[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>.

[RFC9460] Schwartz, B., Bishop, M., and E. Nygren, "Service Binding and Parameter Specification via the DNS (SVCB and HTTPS Resource Records)", RFC 9460, DOI 10.17487/RFC9460, November 2023, <https://www.rfc-editor.org/info/rfc9460>.

[RFC9471] Andrews, M., Huque, S., Wouters, P., and D. Wessels, "DNS Glue Requirements in Referral Responses", RFC 9471, DOI 10.17487/RFC9471, September 2023, <https://www.rfc-editor.org/info/rfc9471>.

Appendix A. Details of Requestor's Maximum UDP Payload Size Discussions

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

* The minimum MTU for an IPv6 interface is 1280 octets (see Section 5 of [RFC8200]). So, it can be used as the default path MTU value for IPv6. The corresponding minimum MTU for an IPv4 interface is 68 (60 + 8) [RFC0791].

* [RFC4035] states that "A security-aware name server MUST support the EDNS0 ([RFC2671]) message size extension, [and it] 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] proposes that UDP requestors set the requestor's payload size to 1232 and 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.

* Most of the Internet, especially the inner core, has an MTU of at least 1500 octets. Maximum DNS/UDP payload size for IPv6 on an 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 recommendation of default maximum DNS/UDP payload size is 1400.

* [Huston2021] analyzes the result of [DNSFlagDay2020] and reports that their measurements suggest that in the interior of the Internet between recursive resolvers and authoritative servers, the prevailing MTU is 1500 and there is no measurable signal of use of smaller MTUs in this part of the Internet. They propose that their measurements suggest setting the EDNS(0) requestor's UDP payload size to 1472 octets for IPv4 and 1452 octets for IPv6.

As a result of these discussions, this document recommends a value of 1400, with smaller values also allowed.

Appendix B. 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 necessary Resource Records (RRs). For delegations, see [RFC9471]. 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 (see Section 2 of [RFC2308]).

Some resource records (MX, SRV, SVCB, and HTTPS) require additional A, AAAA, and Service Binding (SVCB) records in the Additional section defined in [RFC1035], [RFC2782], and [RFC9460].

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 C. Known Implementations

This section records the status of known implementations of the proposed recommendations described in Section 3.

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

C.1. BIND 9

BIND 9 does not implement R1 and R2.

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

When BIND 9 is 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 R3, 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 R4 and the send fails with EMSGSIZE, BIND 9 sets the TC bit and tries to send a minimal answer again.

For R5, BIND 9 uses the edns-buf-size option, with the default of 1232.

For R7, after two UDP timeouts, BIND 9 will fall back to TCP.

C.2. Knot DNS and Knot Resolver

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

Fragments are ignored if they arrive over a Linux 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.

C.3. PowerDNS Authoritative Server, PowerDNS Recursor, and PowerDNS dnsdist

* Use IP_PMTUDISC_OMIT with a fallback to IP_PMTUDISC_DONT.

* The default EDNS buffer size of 1232; no probing for smaller sizes.

* There is no handling of EMSGSIZE.

* Recursor: UDP timeouts do not cause a switch to TCP, but "spoofing near misses" may.

C.4. PowerDNS Authoritative Server

* The default DNSSEC algorithm is 13.

* Responses are minimal; this is not configurable.

C.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 a UDP size of 1232 from peers, by default. The requestor’s size is limited to a max of 1232.

After some timeouts, Unbound retries with a smaller size, if applicable, or at size 1232 for IPv6 and 1472 for IPv4. This does not cause any negative effects due to the "flag day" [DNSFlagDay2020] change to 1232.

Unbound has the "minimal responses" configuration option; set default on.

Acknowledgments

The authors 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, Wouter Wijngaards, Vladimir Cunat, Benno Overeinder, and tpn Nmec for their extensive reviews and comments.

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