Network Working Group
Internet-Draft
Intended status: Informational
Expires: 14 August 2026

K. Fujiwara
JPRS
Y. Thessalonikefs
NLnet Labs

10 February 2026

Upper limit values for DNS
draft-fujiwara-dnsop-dns-upper-limit-values-05

Abstract

DNS was designed to have as few hard upper limits as possible to allow for future extensibility. However, the lack of a clear upper limit leads to vulnerabilities, and several attack methods have been reported. This document collects upper limit values implemented by DNS software to avoid vulnerabilities.

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 14 August 2026.

Copyright Notice

Copyright (c) 2026 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

DNS was designed to have as few upper limits as possible to allow for future extensibility described in the paper "Development of the Domain Name System" [Mockapetris1988].

If a protocol is implemented without considering the upper limit, it may become vulnerable to DoS attacks.

There are parameters in the DNS protocol that do not have clear upper limits. For example, the number of alias levels using CNAME Resource records and the number of resource records in an RRSet.

This document collects upper limit values implemented by DNS software to avoid vulnerabilities.

This document is intended to serve as a basis for discussions that will lead to the creation of a future Best Current Practice document.

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 specialized terms used in this document are defined in DNS Terminology [RFC9499].

Glueless delegation (Gluelessness) is the term used to describe delegation without glue.

3. Implemented upper limits

3.1. Packet size

There were comments that there are size limitations even if no precise upper limits are set.

The DNS packet format has an upper limit of 65535 octets, so an RRset cannot exceed that size. Also, the upper limit size of a single resource record is 65535 octets minus the DNS header size because RDLENGTH is 16 bits.

Section 4.2.1 UDP usage of [RFC1035] limits the UDP message size to 512.

The size of a DNS response that can be sent using unfragmented UDP is about 1400 octets. [RFC9715]

[flagday2020] proposed 1232 octets and is used as the default value in most DNS software.

3.2. DNS Response Rate Limiting

[RFC5358] describes DNS Reflector Attacks and how to prevent the use of default configured recursive nameservers. Simply preparing a large RRSet can increase the amplification factor of DNS Reflector Attacks. Countermeasures for recursive resolvers were described in [RFC5358], however, countermeasures for authoritative servers have not been standardized as an RFC and are implemented in various software as DNS Response Rate Limiting.

3.3. Number of alias levels using CNAME/DNAME

CNAME aliases are widely used; however, when there are multiple levels of CNAME aliases, full-service resolvers have to redo the name resolution, which increases the load. And more, each stub resolver that receives a response containing multiple CNAME aliases must find the final A, AAAA Resource record that corresponds to the CNAME in each application.

Unbound and BIND 9 introduced the ’max-query-restarts’ parameter, and their default value is 11.

3.4. Number of Resource Records in a RRSet

Currently, many web services that use domain names require that a TXT record containing a token of their choosing be placed at the zone apex to verify the registrant domain name. A large enterprise set 74 TXT records at its zone apex.

CVE-2024-1737, "BIND’s database will be slow if a very large number of RRs exist at the same name" was reported, and BIND 9.18.28 implemented the limit. BIND 9 introduced the ’max-records-per-type’ parameter that limits the number of resource records in an RRSet, and the default value is 100.

BIND 9 also limits the maximum number of RR types that can be stored for an owner name with the ’max-types-per-name’ parameter with the default value 100.

Unbound has the ’iter-scrub-ns’ parameter that limits the number of RRs explicitly for the NS RRSet and the default value is 20.

Unbound has the ’iter-scrub-cname’ parameter that limits the number of CNAME/DNAME records in an upstream reply by clipping off the remainder of the reply. The default value is 11.

3.5. Number of NS records in a delegation

Although we could not find a clear explanation, the upper limit value of the number of name server names that can be registered for gTLDs and ccTLDs is 13.

3.6. Number of RRSIGs/DNSKEYs/DSs in a RRSet

[KeyTrap] is a vulnerability caused by the fact that there is no upper limit on the number of DNSKEY, DS, or RRSIG resource records which could result in CPU exhaustion during DNSSSEC validation attempts.

Unbound introduced, as hard-coded values, the maximum number of RRSIG validations for an RRset (MAX_VALIDATE_RRSIGS) as 8, and the maximum allowed digest match failures per DS, for DNSKEYs with the same properties (MAX_DS_MATCH_FAILURES) as 8.

BIND 9 limits the number of DNSSEC validations that can happen in a single fetch/processing a single cache miss with the 'max-validations-per-fetch' parameter with the default value 16.

BIND 9 limits the number of DNSSEC validation failures that can happen in a single fetch/single cache miss with the 'max-validation-failures-per-fetch' parameter with the default value 1.

3.7. Number of NSEC3 hash calculations

[NSEC3Vulnerability] is a vulnerability that exploits the amount of NSEC3 hash iterations needed when proving negative responses which could result in CPU exhaustion during DNSSEC validation attempts.

Unbound introduced, as a hard-coded value, the maximum number of NSEC3 hash calculations with a default value of 8.

3.8. Number of outgoing queries per incoming query

Unbound limits the number of outgoing queries per incoming query with the 'max-sent-count' parameter with a default value of 32. This counter is reset on query restarts (e.g., delagation points or CNAME/DNAME redirections).

Unbound limits the number of outgoing queries per incoming query with the 'max-global-quota' parameter with a default value of 200. This counter is not reset on query restarts (e.g., delagation points or CNAME/DNAME redirections) and it persists on any internal subqueries spawned.

BIND 9 limits the number of iterative queries while servicing a single recursive query with the 'max-query-count' parameter with a default value of 200.

BIND 9 limits the number of iterative queries while servicing a recursive query while looking up a single name with the 'max-recursion-queries' parameter with the default value 50. CNAME restarts this counter.

BIND 9 limits the number of levels of recursion permitted at any one time while servicing a recursive query with the 'max-recursion-depth' parameter with the default value 7.

BIND 9 limits the total deadline before giving up a single recursive query with the 'resolver-query-timeout' parameter with the default value 10 seconds.

3.9. Number of NS resolution queries

Unbound limits the number of NS resolution queries needed per incoming query to a hard-coded value of 64.

Unbound limits the number of NS resolution queries neeeded per delegation point to a hard-coded value of 16, in response to [NXNS].

Unbound limits the number of acceptable NXDOMAIN replies for NS queries (for a query and its subqueries) to a hard-coded value of 5 in response to [NXNS].

3.10. Number of delegation levels of glueless delegations

Unrelated (or, rarely sibling) name server names are used/required for DNS hosting services.

[RFC9471] states that all in-domain glue records are attached to the delegation response. Therefore, using in-domain name server names for DNS delegation minimizes name resolution costs.

For some domain names, there are multiple layers of dependence on unrelated name server names when resolving the name. If information on unrelated name server names is not in the cache, the recursive resolver must perform A/AAAA name resolution for the unrelated name server names. Frequent use of unrelated name server names can cause unstable name resolution.

Furthermore, there are cases where cyclic dependencies in delegation occur, settings that depend on sibling glue, and cases where the sibling glue disappears or some name servers stop responding, making it impossible to resolve names.

[Tsuname2021] pointed out attacks and countermeasures that use increased load due to cyclic dependencies. Many cyclic delegations are likely due to misconfigurations.

[djbdns] allows three levels of gluelessness.

Unbound limits the maximum number of referral responses accepted per resolution attempt to a hard-coded value of 130.

4. Discussions on future upper limits

Evaluation is necessary before setting common upper limit values. Implementing possible upper limits as configurable parameters that operators can control is useful.

If we set upper limits on authoritative servers in the future, errors should be detected on the primary servers. Secondary servers should not detect errors because they only receive zone transfers.

5. IANA Considerations

This document requests no IANA actions.

6. Security Considerations

7. Acknowledgments

The authors would like to thank discussions of dnsop WG.

Thanks to Petr paek for providing the helpful comments from an implementor’s perspective and the upper limits that BIND 9 implements.

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/rfc/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/rfc/rfc2119>.

[RFC5358] Damas, J. and F. Neves, "Preventing Use of Recursive Nameservers in Reflector Attacks", BCP 140, RFC 5358, DOI 10.17487/RFC5358, October 2008, <https://www.rfc-editor.org/rfc/rfc5358>.

[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>.

[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/rfc/rfc9471>.

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

8.2. Informative References

[djbdns] Bernstein, D. J., "djbdns: Notes on the Domain Name System", n.d., <https://cr.yp.to/djbdns/notes.html>.

[flagday2020] "DNS Flag Day 2020", 2020, <https://www.dnsflagday.net/2020/>.

[KeyTrap] Elias Heftrig, Haya Schulmann, Niklas Vogel, and Michael Waidner, "The KeyTrap Denial-of-Service Algorithmic Complexity Attacks on DNS", 2024.

[Mockapetris1988] Paul V Mockapetris and Kevin J Dunlap, "Development of the domain name system", the Proceedings of SIGCOMM 1988 , August 1988.

[NSEC3Vulnerability] Petr paek, "NSEC3 closest encloser proof can exhaust CPU (CVE-2023-50868)", 2024, <https://www.isc.org/blogs/2024-bind-security-release/#nsec3-closest-encloser-proof-can-exhaust-cpu-cve-2023-50868>.

[NXNS] Yehuda Afek, Lior Shafir, and Anat Bremler-Barr, "NXNS Attack: Recursive DNS Inefficiencies and Vulnerabilities", 2020, <https://dl.acm.org/doi/10.5555/3489212.3489248>.

[RFC9715] Fujiwara, K. and P. Vixie, "IP Fragmentation Avoidance in DNS over UDP", RFC 9715, DOI 10.17487/RFC9715, January 2025, <https://www.rfc-editor.org/rfc/rfc9715>.

[Tsuname2021] Moura, G. M., Sebastian Castro, John S Heidemann, and Wes Hardaker, "TsuNAME: exploiting misconfiguration and vulnerability to DDoS DNS", IMC ’21: Proceedings of the 21st ACM Internet Measurement Conference , 2021.

Authors' Addresses

Kazunori Fujiwara
Japan Registry Services Co., Ltd.
Japan
Email: fujiwara@wide.ad.jp

Yorgos Thessalonikefs
NLnet Labs
Netherlands
Email: yorgos@nlnetlabs.nl