Internet Engineering Task Force
Internet-Draft
Updates: 4035, 6891 (if approved)
Intended status: Standards Track
Expires: 9 October 2025

S. Zhang
Tsinghua University
S. Wang
L. Chen
Zhongguancun Laboratory
D. Li
B. Liu
Tsinghua University

7 April 2025

Handling Unvalidated Data during DNSSEC Troubleshooting
draft-zhang-dnsop-dnssec-unvalidated-data-00

Abstract

Due to the prevalence of DNSSEC (Domain Name System Security Extensions) misconfigurations, many domain administrators troubleshoot the records of DNSSEC-signed domains via queries with CD (Checking Disabled) bit set. However, as DNS resolvers are not forced to perform DNSSEC validation for CD=1 queries, the unvalidated data introduced during troubleshooting could be mixed up with the routine ones in the resolver cache. Recent research has revealed that the reuse of the cached unvalidated data in subsequent resolutions could lead to the risk of Denial-of-Service (DoS). This document clarifies the definition of unvalidated data in the context of DNSSEC. Then, it demonstrates the DoS vulnerabilities of current DNS resolver implementations due to the reuse of cached unvalidated data. Accordingly, it provides several recommendations for DNSSEC-validating resolvers to handle the unvalidated data and mitigate the risk of DoS, so as to improve the availability of DNSSEC-signed domains.

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 9 October 2025.

Table of Contents

1. Introduction and Terminology

Domain Name System Security Extensions (DNSSEC) aim to protect the authenticity and integrity of DNS data against cache poisoning attacks. Over the past decade, DNSSEC has been increasingly deployed by both domains and DNS resolvers throughout the Internet. However, because of the design complexity, various misconfigurations of DNSSEC records frequently occur in the wild, resulting in massive service outages due to DNSSEC validation failure [IANIX24].

To troubleshoot DNSSEC record configurations and manage DNSSEC validation locally, one can leverage the CD (Checking Disabled) bit in the DNS message header as proposed in [RFC4035]. Specifically, the set of the CD bit in a DNS query indicates that the DNS client does not force the target DNS resolver to perform DNSSEC validation on the received records. Typically, DNSSEC-validating resolvers accept troubleshooting queries with CD=1 from arbitrary clients they serve.

However, current DNSSEC specifications are ambiguous about how DNS resolvers should handle the data introduced during troubleshooting, where records that have not passed DNSSEC validation could remain in the resolver’s cache. Even though the unvalidated records are generally associated to the lowest trust level according to [Li23], the resolver could still reuse them in subsequent resolution tasks without re-querying for the validated ones. Previous study has revealed that the reuse of cached unvalidated data could result in persistent validation failures of DNSSEC-signed domains, i.e., Denial-of-Service (DoS).

To this end, this document proposes guidelines on how DNSSEC-validating resolvers should handle the unvalidated data introduced during DNSSEC troubleshooting. First, it clarifies the definition of unvalidated data in the context of DNSSEC. Then, it presents the DoS vulnerabilities in current resolver implementations due to the reuse of cached unvalidated data when resolving DNSSEC-signed domains. Finally, it provides recommendations for DNSSEC-validating resolvers to optimize the caching and reusing of the unvalidated data.

The key words "MUST", "MUST NOT", "SHOULD" and "MAY" in this document are to be interpreted as described in [RFC2119].

2. Clarification of Unvalidated Data in DNSSEC

Section 4.7 of [RFC4035] proposed BAD cache to facilitate the caching of resource records that have been proven invalid, so as to avoid unnecessary query retries when encountering DNSSEC misconfigurations at the domain side. This document uses the term "unvalidated" to further refer to the DNSSEC-related data that have not passed DNSSEC validation in the resolver cache, including resource records that have not been validated, cannot be validated, or have been proven invalid. The unvalidated data also include the EDNS(0) (Extension Mechanisms for DNS) status of a particular nameserver host cached by the resolver, as Section 6.2.2 of [RFC6891] have stated that the DO (DNSSEC OK) bit proposed by [RFC3225] is only signaled through EDNS(0) to indicate DNSSEC support.

3. Vulnerabilities in Current Implementations

This document demonstrates a novel DoS attack surface introduced by the CD bit, where the resolver’s reusing of cached unvalidated data could lead to persistent resolution failures of DNSSEC-signed domains.

In general, an attacker could bypass DNSSEC validation by sending DNS queries for troubleshooting (i.e., CD=1), causing a DNSSEC-validating resolver to accept and cache forged data without validation. Later, when handling legitimate DNS queries that demand DNSSEC validation, the resolver reuses the cached unvalidated records, leading to validation failures due to missing or bogus records along the chain of trust, such as DNSKEY or DS records with their signatures either removed or manipulated. Note that the forged, unvalidated data are not directly queried by clients in routine DNS operations, and thus simply retrying the failed DNS queries (e.g., typically for A records) cannot discard them. As a result, the resolver continuously encounters SERVFAIL responses, even if it has already obtained the valid records queried by the client. Additionally, the attacker can break the DNSSEC chain of trust by tampering with the cached information of nameservers, such as their IP addresses and EDNS(0) capabilities, thereby leading to validation failures for all DNSSEC-signed domains hosted on those nameservers.

Specifically, this document illustrates three variants of this DoS attack as follows.

3.1. V1: Cached Unvalidated DNSKEY/DS Records

First, the attacker sends a DNS query with CD=1 to the target DNSSEC-validating resolver, requesting the DNSKEY/DS record of the victim domain. Next, the attacker returns a forged response, in which the RRSIG record of domain's DNSKEY/DS is either removed or manipulated. Due to the set of the CD bit, the resolver is not required to perform DNSSEC validation. Instead, it caches the received records. Later, when an ordinary client queries the resolver for the victim domain's A record with CD=0, the resolver reuses the unvalidated DNSKEY/DS record in the cache rather than issuing new queries. In the case of RRSIG removed, the validation fails because the RRSIG required to establish the DNSSEC trust chain is missing. In the case of RRSIG manipulated, the manipulated RRSIG causes the validation of the DNSKEY/DS record to fail. However, as the unvalidated DNSKEY/DS cache entries are not directly hit by the client query, the resolver fails to discard them even after retrying the failed routine queries. Instead, it continuously reuses them until their TTLs (Time-to-Live) expire, leading to persistent validation failure for the duration of the caching TTL.

3.2. V2: Cached Unvalidated Nameserver IP Addresses

To prevent the resolver from obtaining necessary records for DNSSEC validation, an attacker can also forge the IP addresses of the nameservers. Particularly, apart from tampering with unsigned glue records stored in the parent zone, the attack is also applicable even if both a domain and all the nameserver zones along its delegation chain are properly signed by DNSSEC. Specifically, the attacker first sends a DNS query with CD=1 to the resolver, requesting the A/AAAA record of the victim domain's nameserver. Next, the attacker returns a response containing forged A/AAAA records, which is cached by the resolver without strict validation. Next, when an ordinary client sends queries with CD=0 to the resolver for domains delegating to this nameserver, the resolver reuses the forged IP address of the nameserver to query. If the forged IP address is inactive, the resolver will either return SERVFAIL or timeout to respond. Similar to the variant in Section 3.1, the resolver does not retry querying for the nameserver IP address after the resolution failure, and the DoS duration lasts long to the caching TTL.

3.3. V3: Cached Unvalidated Nameserver's EDNS(0) Status

Different from the previous variants that exploit resource records, the attacker can also trick the resolver into caching the fact that the nameserver host of the victim domain is not EDNS(0) capable, so as to prevent the resolver's requesting for RRSIG records and break DNSSEC validation. Note that some resolver implementations (e.g., [BIND9]) do not follow Section 3 of [RFC3225] to handle the EDNS(0) capability information, i.e., only fall back to queries without EDNS(0) when receiving a response with error status codes (e.g., FORMERR, NOTIMP) from the target authoritative nameserver. Instead, they simply treat the nameserver as EDNS(0)-incapable when receiving a response with no EDNS(0) OPT records, which makes this attack variant possible.

Specifically, after triggering an arbitrary DNS response from the victim's nameserver to the target resolver, the attacker strips off the EDNS(0) OPT record in the additional section. When the resolver receives the response, it considers that the nameserver IP is not capable of EDNS(0), and caches this information for future queries. Then, when an ordinary client requests for DNSSEC-signed domains served by the same nameserver host and requires validation, the resolver removes OPT in the query to the host. Subsequently, the nameserver host determines that the resolver cannot handle DNSSEC records, hence does not respond with the RRSIGs of the queried records. Hence, the resolver fails DNSSEC validation due to missing RRSIGs. The DoS duration depends on how long the EDNS0 capability information is cached by the resolver, which is usually determined by the resolver's refresh interval to retry adding OPT in queries, e.g., 30 minutes for [BIND9].

4. Caching of Unvalidated Records

4.1. Restricting Caching TTL for CD=1 Queries

This document recommends that the unvalidated resource records obtained for CD=1 queries SHOULD be cached following the rules for the BAD cache specified in Section 4.7 of [RFC4035] and Section 2.6 of [RFC9520]. Specifically, the caching of unvalidated records MUST be assigned with a TTL. This TTL SHOULD be small by default, which typically ranges from 3 to 30 seconds, and MUST NOT exceed 5 minutes.

4.2. Conservative BAD Cache

To prevent DoS attacks triggered by intentional response forging, DNSSEC-validating resolvers SHOULD be conservative to save records into the BAD cache. Specifically, to distinguish between forged responses and domain-side misconfigurations, this document suggests that DNSSEC-validating resolvers SHOULD perform a heuristic number of retries to validate the records, e.g., 5 times of retry queries to each available authoritative nameserver as implemented by [Unbound]. Resolvers SHOULD only move the records to the BAD cache when all the validation attempts fail. Note that the number of retries MUST be upper-bounded, thereby avoiding excess queries in case of misconfigurations at the domain side.

5. Reusing of Cached Unvalidated Records

5.1. Records without Validation

Based on the type of the records that have not been validated, this document assigns different priorities for resolvers to validate them.

5.1.1. Records along DNSSEC Chain of Trust

When receiving CD=0 queries, i.e., DNSSEC validation is demanded, DNSSEC-validating resolvers MUST use the validated version of all records along domain’s DNSSEC chain of trust.

If any record along domain’s DNSSEC chain of trust has existed in cache but has been proven invalid, resolvers MUST expire the record from the cache and attempt to obtain the potentially valid one following the retry policy described in Section 4.2.

5.1.2. Referral Records

As referral records (e.g., NS, CNAME) themselves are not necessarily required in constructing DNSSEC chain of trust, this document allows resolvers to validate these records with a lower priority. Specifically, when receiving CD=0 queries, resolvers MAY use the cached unvalidated referral records to obtain necessary records along domain’s DNSSEC chain of trust, and MAY not validate the referral records if the chain of trust of the queried domain has already passed DNSSEC validation. Otherwise, resolvers MAY expire the cached unvalidated referral records and attempt to obtain the potentially valid ones following the retry policy described in Section 4.2.

5.2. Records in BAD Cache

[RFC4035] stated that DNSSEC-validating resolvers SHOULD answer from the BAD cache upon receiving CD=1 queries, and MUST return RCODE 2 (Server Failure) when CD is not set. However, current RFCs remain ambiguous about whether records in the BAD cache could also be involved in other resolution processes, especially when they are not directly hit by the clients' query. For example, when a client queries for domain's A record and demands DNSSEC validation, the DNSKEY record of the queried zone is involved in the resolution, while the DNSKEY could exist in the BAD cache.

To handles these cases, this document further specifies that the reuse scope of records in the BAD cache SHOULD be restricted to only answering CD=1 queries that directly hit the cache with a matched (QNAME, QCLASS, QTYPE) tuple. When records in the BAD cache are not hit directly, but are involved in other resolution processes where DNSSEC validation is demanded, the resolver SHOULD expire them from the cache and attempt to obtain the potentially valid ones following the retry policy described in Section 4.2.

6. Verification of Nameserver's EDNS(0) Status

6.1. Always Enabling EDNS(0) for DNSSEC Queries

If a nameserver hosts any DNSSEC-signed domains, it must be capable of EDNS(0) to serve DNSSEC records (e.g., RRSIG). Hence, when resolving a DNSSEC-signed domain, it is expected that the corresponding nameserver is always EDNS(0)-capable. Section 6.2.2 of [RFC6891] suggested that if DNSSEC is required, no fallback to non-EDNS(0) queries should be performed.

This document further strengthens this condition that DNSSEC-validating resolvers MUST always enable EDNS(0) for queries to the nameservers of DNSSEC-signed domains. Upon receiving responses with error status codes (e.g., NOTIMP, FORMERR), resolvers SHOULD follow the retry policy described in Section 4.2 and move the nameserver-related records (e.g., domain's NS records or nameserver's A/AAAA records) into the BAD cache if necessary, and cache them in accordance with Section 4.1.

6.2. Verifying EDNS(0) Capability

There could be cases where some nameserver hosts fail to enable EDNS(0) temporarily due to accidental errors, and Section 6.2.2 of [RFC6891] stated that resolvers MAY cache that a nameserver host is EDNS(0)-incapable for a brief time. This document agrees with the caching of the EDNS(0)-incapable information, in order to avoid fallback delays and disrupting the resolution of unsigned domains hosted on the same nameserver.

Nevertheless, this document further recommends that DNSSEC-validating resolvers SHOULD constantly verify the EDNS(0) capability status of nameserver hosts to mitigate the risk of attack variant described in Section 3.3. Specifically, this document suggests that the verification interval SHOULD be aligned with the TTL of the BAD cache, i.e., typically range from 3 to 30 seconds, and MUST NOT be more than 5 minutes. After caching that a nameserver host is EDNS(0)-incapable for the specific interval, DNSSEC-validating resolvers should retry to enable EDNS(0) in subsequent queries and see if the nameserver host returns good responses.

7. Security Considerations

The major security risk introduced in the above recommendations is the risk of traffic amplification-based DoS attacks, which could consume resources to overload the resolver and the queried authoritative nameserver. To flexibly balance the risk of traffic amplification and persistent resolution failure based on specific operating scenarios, this document suggests that resolver implementations SHOULD make the variables configurable by users, including the restricted caching TTL described in Section 4.1, number of validation retries before saving into the BAD cache described in Section 4.2, and the verification interval of EDNS(0) status described in Section 6.2.

Specifically, this document discusses the potential risks of its recommendations as follows.

7.1. Risk of Reduced Caching TTL

Section 5.9 of [RFC6840] suggested that DNSSEC-validating stub or forwarding resolvers always set the CD bit on queries to their upstream recursive resolvers. Hence, the implementation recommended in Section 4.1 could inadvertently reduce the records' caching duration at the recursive upstreams, even if the records are actually legitimate. Given the shortened TTL, there could be potential risks where the recursive upstreams frequently query the authoritative nameservers to refresh their caches, leading to traffic

amplification. However, since the downstream DNSSEC-validating stub or forwarding resolvers are able to cache the records with their original TTL once the DNSSEC validation passes, clients are expected to be served by the cached validated records without frequently re-querying the authoritative nameservers. Also, once the recursive upstreams receive a CD=0 query and perform DNSSEC validation, they can also cache and serve the validated version of the records with their original TTL if the validation passes.

7.2. Risk of Conservative BAD Cache

As stated in Section 4.2, DNSSEC-validating resolvers are forced to implement an upper bound to the number of validation retries before saving records into the BAD cache. Such upper bound is expected to be controllable and prevent the resolver from triggering excess queries.

7.3. Risk of Constant EDNS(0) Status Verification

The verification interval of EDNS(0) status recommended in Section 6.2 of this document is aligned with the BAD cache and negative cache TTL suggested by [RFC9520], which is the state-of-the-art standard to balance the risk of DoS due to the cached failure status and frequent re-querying.

8. IANA considerations

This document contains no actions for IANA.

9. References

[BIND9] ISC, ISC., "BIND9", <https://www.isc.org/bind/>.

[IANIX24] IANIX, "Major DNSSEC Outages and Validation Failures", 2024, <https://ianix.com/pub/dnssec-outages.html>.

[Li23] Li, X., Lu, C., Liu, B., Zhang, Q., Li, Z., Duan, H., and Q. Li, "The Maginot Line: Attacking the Boundary of DNS Caching Protection", USENIX 32nd USENIX Security Symposium, 2023, <https://www.usenix.org/system/files/usenixsecurity23-li-xiang.pdf>.

[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>.

[RFC3225] Conrad, D., "Indicating Resolver Support of DNSSEC", RFC 3225, DOI 10.17487/RFC3225, December 2001, <https://www.rfc-editor.org/info/rfc3225>.

[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>.

[RFC6840] Weiler, S., Ed. and D. Blacka, Ed., "Clarifications and Implementation Notes for DNS Security (DNSSEC)", RFC 6840, DOI 10.17487/RFC6840, February 2013, <https://www.rfc-editor.org/info/rfc6840>.

[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>.

[RFC9520] Wessels, D., Carroll, W., and M. Thomas, "Negative Caching of DNS Resolution Failures", RFC 9520, DOI 10.17487/RFC9520, December 2023, <https://www.rfc-editor.org/info/rfc9520>.

[Unbound] NLnet Labs, "Unbound", <https://www.nlnetlabs.nl/projects/unbound/about/>.

Authors' Addresses

Shuhan Zhang
Tsinghua University
Beijing
China
Email: zhangsh22@mails.tsinghua.edu.cn

Shuai Wang
Zhongguancun Laboratory
Beijing
China
Email: wangshuai@zgclab.edu.cn

Li Chen
Zhongguancun Laboratory
Beijing
China
Email: lichen@zgclab.edu.cn

Dan Li
Tsinghua University
Beijing
China
Email: tolidan@tsinghua.edu.cn

Baojun Liu
Tsinghua University
Beijing
China
Email: lbj@tsinghua.edu.cn