DNS Operations (dnsop)
Internet-Draft
Intended status: Best Current Practice
Expires: 1 September 2026

Y.Q. Qiu
X.L. Li
Nankai University

28 February 2026

Enhanced Bailiwick Checking for DNS Resolvers to Mitigate Cache Poisoning Vulnerabilities
draft-qiu-dnsop-enhanced-bailiwick-02

Abstract

The Domain Name System (DNS) relies on caching to function efficiently. However, DNS cache poisoning remains a significant threat. The "bailiwick" rule is a fundamental security mechanism intended to prevent resolvers from caching out-of-bailiwick data sent by malicious or misconfigured authoritative servers. Current DNS standards provide high-level guidance on bailiwick checking but lack specific algorithmic definitions, leading to inconsistent implementations across different DNS software. These inconsistencies can be exploited, particularly in DNS servers that operate in multiple modes, such as Conditional DNS Servers (CDNS), which may act as both recursive resolvers and forwarders and often share a common cache.

This document specifies enhanced and more precise rules for bailiwick checking in DNS resolvers, including those operating as forwarders or CDNS. The goal is to provide clearer guidelines for implementers, reduce the attack surface for cache poisoning, and improve the overall security and robustness of the DNS infrastructure. The recommendations herein are informed by recent research highlighting vulnerabilities in existing bailiwick implementations.

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 1 September 2026.

Table of Contents

1. Introduction

1.1. Motivation

Bailiwick checking is a critical defense mechanism in the DNS designed to ensure that a DNS server only accepts and caches data for which the responding server is authoritative. As defined in [RFC1034] and [RFC1035], authoritative nameservers should not return data for zones they do not manage. Resolvers are expected to enforce this by discarding "unsolicited" or out-of-bailiwick data.

However, the precise algorithms for bailiwick checking are not rigorously defined in existing standards, leading to diverse interpretations and implementations by DNS software vendors. Historically, DNS cache poisoning has evolved from simple transaction ID guessing [KAMINSKY08] [HITCHHIKERDNS] to exploiting side channels [SADDNS] and forwarding devices [POISONOVERFWD]. Recent research, such as "The Maginot Line: Attacking the Boundary of DNS Caching Protection" [MAGINOTDNS], has exposed vulnerabilities stemming specifically from inconsistencies in bailiwick logic. These vulnerabilities can allow attackers to inject malicious records into a resolver’s cache, potentially hijacking entire DNS zones, including Top-Level Domains (TLDs).

Conditional DNS Servers (CDNS), which combine recursive resolver and forwarder functionalities and often share a global cache, present a particular challenge. Weaknesses in the bailiwick checks of one mode (e.g., forwarding) can be exploited to poison the cache used by another, more secure mode (e.g., recursive resolution).

This document aims to address these issues by providing more explicit and robust rules for bailiwick checking. The goal is to foster more consistent and secure implementations across DNS resolver software, thereby strengthening the overall resilience of the DNS ecosystem against cache poisoning attacks.

This document acknowledges that the diversity of DNS implementations is a strength and not a weakness. The exact mitigations detailed herein are provided as operational guidance and Best Current Practices, rather than rigid Internet Standards. Implementers are encouraged to adapt these mechanisms to suit their specific architectures.

1.2. Requirements Language

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.

1.3. Terminology

This document uses standard DNS terminology as defined in [RFC1034], [RFC1035], and [RFC9499]. The following terms are central to this document:

Bailiwick:
: The scope of authority a nameserver has. Data is considered "in-bailiwick" if it pertains to the zone(s) for which the responding nameserver is authoritative in the context of the original query.

Out-of-Bailiwick:
: Data that is not within the bailiwick of the responding nameserver.

Q.NAME:
: The domain name in the question section of the current query being processed by the resolver.

Q.TYPE:
: The type of the resource record (RR) in the question section of the current query.

Q.ZONE:
: The closest ancestor zone for which the resolver has (or believes it has) authoritative nameserver information relevant to Q.NAME. This zone defines the current bailiwick against which incoming response records are checked. The determination of Q.ZONE is critical and is further detailed in Section 3.2.

RR (Resource Record):
: A single record in a DNS response. RR.NAME is the owner name of the resource record. RR.TYPE is the type of the resource record. RR.DATA is the data content of the resource record.

CDNS (Conditional DNS Server):
: A DNS server configured to act as both a recursive resolver and a forwarder simultaneously, often directing queries to different upstream servers or resolving them differently based on the queried domain name. CDNS implementations typically use distinct sets of zones:

  Z_R (Recursive DNS Zones): The set of domain names for which the CDNS is configured to perform recursive resolution.

  Z_F (Forwarding DNS Zones): The set of domain names for which the CDNS is configured to forward queries to specific upstream forwarders.

Delegation:
: The process by which a nameserver for a parent zone indicates the authoritative nameservers for a child zone, typically via NS records.

Referral:
: A DNS response from a nameserver that does not have the authoritative answer for a query but provides NS records (and potentially glue records) pointing to nameservers that are authoritative for a zone closer to the Q.NAME.

Sanitize Records:
: The process of examining each RR in a DNS response to determine if it is in-bailiwick and should be processed and potentially cached.

2. Background: Bailiwick Checking and its Challenges

2.1. Current Understanding of Bailiwick

The concept of bailiwick is fundamental to DNS security. [RFC1034] states that nameservers should not add RRs to the additional section unless they are "closely related to an RR in the answer or authority sections". More generally, data returned by a nameserver should be within the scope of the zones for which it holds authority, relative to the query that was made. Resolvers are expected to discard data that violates this principle [RFC2181].

The primary goal of bailiwick checking is to prevent a malicious nameserver authoritative for example.com from providing (and a resolver from caching) records for other-example.net, or worse, for TLDs or the root zone.

2.2. Observed Implementation Inconsistencies

Despite the long-standing principle of bailiwick, its implementation varies. The [MAGINOTDNS] study highlights several critical areas of inconsistency and vulnerability:

1. Q.ZONE Initialization: In forwarding mode, some DNS software initializes Q.ZONE to overly broad zones (e.g., the root zone ".") or an ancestor of Q.NAME that is too high in the DNS hierarchy. This effectively weakens or nullifies bailiwick checks for records in the authority and additional sections, as almost any record would appear to be "in-bailiwick" relative to the root. This is referred to as V1 in [MAGINOTDNS].

2. CNAME Chasing Logic: Flaws in how bailiwick context is maintained or re-evaluated during CNAME chasing can lead to vulnerabilities where a resolver is tricked into accepting out-of- bailiwick data for the CNAME target. This is referred to as V2 in [MAGINOTDNS].

3. Shared Cache in CDNS: When recursive and forwarding modes in a CDNS share a global cache, data cached via the potentially weaker bailiwick checks of the forwarding mode can pollute the cache for the recursive mode.

These inconsistencies underscore the need for more precise normative guidance.

2.3. Conditional DNS Servers (CDNS)

CDNS are increasingly common in enterprise networks and ISP infrastructure. They offer flexible query routing, allowing organizations to resolve internal domains locally while forwarding external domains to public resolvers, or to implement split-horizon DNS, content delivery optimizations, or security policies.

The dual nature of CDNS, coupled with shared caching, makes them a prime target if bailiwick checks are inconsistent between modes. An attacker might exploit a vulnerability in the forwarding path’s bailiwick logic to inject malicious data that subsequently affects recursive resolution for other clients or other domains.

3. Normative Requirements for Bailiwick Checking

3.1. General Principles

The following general principles MUST be applied by DNS resolvers when processing responses and considering records for caching:

GP1:
A resolver MUST NOT cache any RR from a response if that RR is determined to be out-of-bailiwick according to the rules specified in this document.

GP2:
The bailiwick checks applied to data received in forwarding mode MUST be at least as stringent as those applied in recursive resolution mode if the forwarded data can be stored in a cache that is also accessible by the recursive resolution process.

GP3:

The Q.ZONE for a query is established at the time the query is sent and is used to validate the response. It represents the zone for which the queried upstream server is considered authoritative in the context of the current Q.NAME.

3.2. Determining the Query Zone (Q.ZONE)

The correct determination of Q.ZONE is paramount for effective bailiwick checking.

3.2.1. Recursive Resolution Mode

When a resolver is performing recursive resolution and is about to send a query for Q.NAME to a set of selected authoritative nameservers:

RRM1:
Q.ZONE MUST be set to the name of the zone for which the selected upstream nameservers are authoritative. This is typically the closest ancestor of Q.NAME (or Q.NAME itself) for which the resolver has NS records in its cache, or the root zone (".") if no closer NS records are available.

3.2.2. Forwarding Mode

When a resolver is configured to forward queries for domains within a specific forwarding zone (Z_F) to a designated set of upstream forwarders:

FM1:
If the Q.NAME falls within a configured forwarding zone Z_F (e.g., Q.NAME is host.internal.example.com and Z_F is internal.example.com), then Q.ZONE for the query sent to the upstream forwarder SHOULD be set to that Z_F (internal.example.com).

FM2:
If the resolver is acting as a general-purpose forwarder (e.g., forwarding all queries for which it is not authoritative itself) and is configured to forward to a specific recursive resolver (e.g., a public DNS service), Q.ZONE SHOULD be considered the root zone ("."). However, if this forwarder caches responses and that cache is shared with a recursive resolution component (as in a CDNS), this broad Q.ZONE definition is insufficient. See Section 4 for CDNS-specific rules.

FM3:

A resolver in forwarding mode, especially if part of a CDNS that shares its cache with a recursive component, MUST NOT initialize Q.ZONE to the root zone (".") or an overly broad ancestor if a more specific zone context (like a configured Z_F) is available for Q.NAME. Relying on an upstream recursive resolver to perform all bailiwick checks is NOT SUFFICIENT if the forwarding resolver itself caches the results in a shared cache.

FM4:
If a forwarder forwards to an authoritative server directly (not a recursive resolver), Q.ZONE MUST be the zone for which that authoritative server is believed to be authoritative for Q.NAME.

3.3. Sanitizing Records in a Response

Upon receiving a response (R) to a query (Q), each RR in the answer, authority, and additional sections of R MUST be checked against Q.ZONE.

3.3.1. Answer Section

AS1:
For an RR in the answer section to be considered in-bailiwick:

a. RR.NAME MUST be equal to Q.NAME (or an alias of Q.NAME if CNAMEs/DNAMEs are involved, see Section 3.4).

b. RR.TYPE MUST be equal to Q.TYPE (unless Q.TYPE is ANY or CNAME/DNAME processing is occurring).

c. If RR.TYPE is CNAME or DNAME, specific CNAME/DNAME processing rules apply (see Section 3.4 and [RFC6672]).

AS2:
Records in the answer section that do not meet these criteria MUST be discarded and NOT cached, unless they are part of a valid CNAME/DNAME chain leading to the Q.NAME.

3.3.2. Authority Section

The authority section typically contains NS records for delegations or SOA records for negative answers.

AuthS1:
For an NS record in the authority section (RR.TYPE is NS) to be considered in-bailiwick:

a. RR.NAME (the zone being delegated) MUST be a sub-domain of Q.ZONE, or Q.ZONE itself. (e.g., if Q.ZONE is "example.com", RR.NAME can be "sub.example.com" or "example.com").

b. Q.NAME MUST be a sub-domain of RR.NAME, or RR.NAME itself. (e.g., if RR.NAME is "sub.example.com", Q.NAME can be "host.sub.example.com" or "sub.example.com").

c. An NS record for Q.ZONE itself (RR.NAME equals Q.ZONE) is considered in-bailiwick.

d. NS records for zones that are not superdomains of or equal to Q.NAME are generally out-of-bailiwick for positive responses, but may be relevant for referrals if they are closer to Q.NAME than Q.ZONE.

AuthS2:
More precisely, for an NS record (RR.NAME, RR.DATA points to NSDNAME) in the authority section of a response from a server authoritative for Q.ZONE, where the query was for Q.NAME: The NS record is in-bailiwick if RR.NAME is an ancestor of Q.NAME (or Q.NAME itself) and RR.NAME is a descendant of Q.ZONE (or Q.ZONE itself). Example: Q.NAME = www.sub.example.com, Q.ZONE = example.com. An NS record for sub.example.com is in-bailiwick. An NS record for example.com is in-bailiwick. An NS record for com is NOT in-bailiwick if provided by the example.com server.

AuthS3:
For an SOA record in the authority section (typically for negative answers or NXDOMAIN), RR.NAME MUST be a superdomain of or equal to Q.NAME, and also a subdomain of or equal to Q.ZONE.

AuthS4:
Records in the authority section that do not meet these criteria MUST be discarded and NOT cached. This prevents the V1 vulnerability where a response for attacker.com (Q.NAME) with Q.ZONE set to . could inject an NS record for com. into the authority section.

3.3.3. Additional Section

The additional section typically contains address records (A/AAAA) for nameservers listed in the authority or answer sections (glue records), or other records deemed helpful by the authoritative server.

AddS1:

For an RR in the additional section to be considered in-bailiwick:

a. It MUST be "closely related" to an RR that is itself in-bailiwick in the answer or authority sections. "Closely related" typically means RR.NAME is the RDATA of an in- bailiwick NS record (glue for a nameserver name) or an in- bailiwick MX or SRV record.

b. Specifically for glue records (A/AAAA for an NSDNAME appearing in an in-bailiwick NS record's RDATA): RR.NAME (the nameserver's name) MUST be within the zone being delegated by that NS record (i.e., a sub-domain of or equal to the owner name of the NS record) or within Q.ZONE. Glue for out-of-zone nameservers is permissible but the address records themselves are only authoritative if they are also within Q.ZONE. Non-authoritative glue may be cached but should be treated with lower trust [RFC2181].

AddS2:
Records in the additional section that are not closely related to in-bailiwick records in other sections, or whose RR.NAME is not justified by Q.ZONE or a valid delegation, MUST be discarded and NOT cached.

3.4. CNAME Chasing

When a resolver receives a CNAME (or DNAME) record in response to a query for Q.NAME:

CN1:
The CNAME/DNAME record itself MUST be validated as per Section 3.3.1. (i.e., RR.NAME must match Q.NAME).

CN2:
The resolver then initiates a new query for the canonical name (the target of the CNAME/DNAME). The Q.ZONE for this new query MUST be determined based on the new Q.NAME (the canonical name) as per rules in Section 3.2.1. It MUST NOT simply inherit the Q.ZONE of the original query if the canonical name falls into a different bailiwick context.

CN3:
Resolvers MUST validate the entire CNAME chain. Each CNAME in the chain must point to a name for which the subsequent CNAME or the final answer is in-bailiwick with respect to the Q.ZONE established for that specific lookup in the chain. This prevents exploitation of CNAMEs to inject unrelated data (addressing V2 from [MAGINOTDNS]).

4. Bailiwick Checking in Conditional DNS Servers (CDNS)

4.1. Cache Interaction and Integrity

CDNS often use a shared global cache for records obtained via both recursive resolution and forwarding. This presents a significant risk if bailiwick checks for forwarded data are weaker.

CDNS1:
If a CDNS shares a cache between its recursive resolver component and its forwarding component, any RR considered for caching from a forwarded response MUST undergo bailiwick checks that are at least as stringent as those defined for recursive resolution mode (Section 3.2.1 and Section 3.3).

CDNS2:
Specifically, when a response is received from an upstream forwarder, and Q.NAME was part of a Z_F:

a. Q.ZONE for validating this response MUST NOT be assumed to be "." by default if the cache is shared.

b. Q.ZONE SHOULD be determined based on the specific Z_F that Q.NAME matched, or based on knowledge of the upstream forwarder’s authoritative scope if the forwarder is not a full recursive resolver itself.

c. If the upstream forwarder is a recursive resolver, the CDNS MAY rely on the upstream resolver’s bailiwick checks ONLY IF the CDNS itself does not cache the response or if its caching is strictly partitioned from the recursive resolver’s cache. If cached in a shared manner, the CDNS MUST re-validate.

CDNS3:
To prevent the V1 vulnerability in CDNS, if Q.NAME is in Z_F and Q.ZONE is determined (e.g., as Z_F itself or an ancestor), records in the authority and additional sections of the response from the forwarder MUST be validated against this Q.ZONE as per Section 3.3.2 and Section 3.3.3. For example, if Z_F is corp.example.com and a query for printer.corp.example.com is forwarded, the response MUST NOT be allowed to cache an NS record for .com in the authority section if Q.ZONE was effectively corp.example.com.

4.2. Forwarding Fallback Considerations

Some CDNS implement a fallback mechanism where if a forwarded query fails (e.g., timeout), the CDNS might attempt to resolve it recursively.

CDNS4:
: If a query originally intended for forwarding (in Z_F) falls back to recursive resolution, the recursive resolution process MUST determine its Q.ZONE and apply bailiwick checks independently, as per Section 3.2.1 and Section 3.3. It MUST NOT carry over any relaxed bailiwick assumptions from the failed forwarding attempt.

5. Security Considerations

The enhanced bailiwick checking rules specified in this document are intended to mitigate several DNS cache poisoning attack vectors, particularly those identified in [MAGINOTDNS] that exploit ambiguities or weaknesses in current bailiwick implementations.

By requiring a more precise determination of Q.ZONE, especially in forwarding mode and within CDNS, these rules make it harder for an attacker to inject out-of-bailiwick NS records for parent zones (like TLDs) or unrelated zones. Stricter validation of authority and additional section records based on a correctly scoped Q.ZONE is key.

Proper CNAME chain validation with appropriate Q.ZONE adjustments at each step helps prevent attacks that leverage CNAMEs to introduce malicious data for the canonical target.

For CDNS, ensuring that data entering a shared cache from forwarding operations is subject to rigorous bailiwick checks equivalent to those in recursive mode is crucial to prevent cross-mode cache pollution.

While these measures significantly improve resilience against certain cache poisoning techniques and align with the forgery resilience measures of [RFC5452], they do not replace the need for DNSSEC ([RFC4033], [RFC4034], [RFC4035]). DNSSEC provides cryptographic assurance of data origin and integrity and remains the most robust defense against cache poisoning. The rules in this document provide defense-in-depth and are particularly important for zones that are not DNSSEC-signed or for resolvers that do not validate DNSSEC signatures.

Implementation of these stricter rules may cause some currently accepted (but technically out-of-bailiwick) data from misconfigured authoritative servers to be rejected. This is generally desirable for security but operators should be aware of potential compatibility issues with non-compliant servers during rollout.

6. IANA Considerations

This document does not require any IANA actions.

7. References

7.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[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>.

[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>.

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, DOI 10.17487/RFC2181, July 1997, <https://www.rfc-editor.org/info/rfc2181>.

[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>.

[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>.

7.2. Informative References

[MAGINOTDNS] 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", Proceedings of the 32nd USENIX Security Symposium , August 2023, <https://www.usenix.org/conference/usenixsecurity23/presentation/li-xiang>.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <https://www.rfc-editor.org/info/rfc4033>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <https://www.rfc-editor.org/info/rfc4034>.

[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>.

[RFC5452] Hubert, A. and R. van Mook, "Measures for Making DNS More Resilient against Forged Answers", RFC 5452, DOI 10.17487/RFC5452, January 2009, <https://www.rfc-editor.org/info/rfc5452>.

[RFC6672] Rose, S. and W. Wijngaards, "DNAME Redirection in the DNS", RFC 6672, DOI 10.17487/RFC6672, June 2012, <https://www.rfc-editor.org/info/rfc6672>.

[SADDNS] Man, K., Qian, Z., Wang, Z., Zheng, X., Huang, Y., and H. Duan, "DNS Cache Poisoning Attack Reloaded: Revolutions with Side Channels", Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20) , November 2020.

[KAMINSKY08] Kaminsky, D., "It's The End Of The Cache As We Know It", Black Hat USA 2008 , 2008.

[POISONOVERFWD] Zheng, X., Lu, C., Peng, J., Yang, Q., Zhou, D., Liu, B., Man, K., Hao, S., Duan, H., and Z. Qian, "Poison Over Troubled Forwarders: A Cache Poisoning Attack Targeting DNS Forwarding Devices", Proceedings of the 29th USENIX Security Symposium , August 2020, <https://www.usenix.org/conference/usenixsecurity20/presentation/zheng>.

[HITCHHIKERDNS] Son, S. and V. Shmatikov, "The Hitchhiker's Guide to DNS Cache Poisoning", Proceedings of the 2010 ACM SIGSAC Conference on Computer and Communications Security (CCS '10) , October 2010.

Authors' Addresses

Yuqi Qiu
Nankai University
38 Tongyan Road
Tianjin
Tianjin, 300355
China
Email: norahqiu@163.com

Xiang Li
Nankai University
38 Tongyan Road
Tianjin
Tianjin, 300355
China
Email: lixiang@nankai.edu.cn