iotops
Internet-Draft
Intended status: Best Current Practice
Expires: 24 May 2026

A. Mishra
Inria
A. Losty
A. M. Mandalari
UCL
J. Mozley
Infoblox
M. Cunche
Inria

20 November 2025

IoT DNS Security and Privacy Guidelines
draft-ietf-iotops-iot-dns-guidelines-00

Abstract

This document outlines best current practices for Internet of Things (IoT) device providers regarding the implementation of DNS stub resolvers, with the aim of mitigating security threats, enhancing privacy, and resolving operational challenges. It also provides guidelines for network operators on mitigating the risks identified in this draft.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://miishra.github.io/IoT-DNS-Guidelines/draft-mishra-iotops-iot-dns-guidelines-latest.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-iotops-iot-dns-guidelines/.

Source for this draft and an issue tracker can be found at https://github.com/miishra/IoT-DNS-Guidelines.

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/.

Copyright Notice

Table of Contents

1. Introduction

Research into the DNS behavior of IoT devices shows widespread non-compliance with protocol standards, gaps in protocol support, and security vulnerabilities. This leads to unpredictable operational behavior and exposes devices to fingerprinting and denial-of-service attacks.

While the recommendations in this BCP may apply to all DNS stub resolver behavior, we treat IoT devices as a specific case where targeted recommendations are useful for the following reasons:

* The recommendations address specific IoT-related security concerns not seen in the DNS behavior of general-purpose operating systems

* IoT devices have different resource characteristics from general-purpose devices, such as constrained power consumption, meaning incorrect software implementations can have an increased operational impact

* IoT devices do not typically have security agents installed on them

* There are many DNS RFCs, and this BCP can be used to identify those related to specific security issues observed through research into IoT devices, with the aim of making it easier to address these vulnerabilities

* IoT devices may be deployed at scale on dedicated networks, and these recommendations will be useful to network security teams in mitigating vulnerabilities, especially where device behavior cannot be changed

* Manufacturers may use standard software distributions aimed at IoT devices without considering DNS behavior. This BCP provides recommendations that can be used as part of the criteria to evaluate these distributions

* IoT devices typically perform the same set of DNS queries on start-up, which makes them both more vulnerable because of this predictable behavior and also more prone to fingerprinting

DNS terminology in this draft conforms to RFC 9499. In this context, Stub Resolver refers to the IoT device, and Resolver refers to the DNS server used by the IoT device.

2. Conventions and Definitions

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.

3. Recommendations for IoT Device Stub Resolvers

3.1. Compliance with Encrypted DNS Standards

The majority of IoT devices use unencrypted DNS over port 53, which means this traffic can be captured and is open to interception and manipulation.

IoT devices SHOULD support encrypted DNS protocols such as DNS over TLS (DoT) [RFC7858], DNS over QUIC (DoQ) [RFC9250], or DNS over HTTPS (DoH) [RFC8484] for enhanced privacy, security, and compatibility. To mitigate against fingerprinting IoT devices, DNS queries can be padded as detailed in [RFC7830] and [RFC8467]. Encrypted DNS protocols are not mandated for compliance with DNS standards, but the use of encrypted DNS may be mandated by some regulators and advised by competent authorities in deployment guidelines.

3.2. Configuration of DNS servers used by IoT Stub Resolvers

IoT devices have been observed to fall back to hard-coded IP addresses for DNS resolvers or ignore addresses assigned to them via automated configuration methods such as DHCP Option 6.

DNS resolvers on devices MUST be configurable via network configuration protocols. Stub resolvers MUST NOT fall back to hard-coded resolvers. This may result in an insecure communication channel, and the open resolvers used in these hard-coded configurations may be blocked by network policy, preventing the device from working.

Devices SHOULD use the following priority order for selecting a resolver. The first one that results in a discovered service should be selected.

1. Manual user configuration

2. Device management software

3. IPv6 Router Advertisement (RA) [RFC8106], DHCPv6 [RFC8415] (if M=1 bit in RA), IPv4 DHCP [RFC2132]. When Encrypted resolver options are present [RFC9463], then they SHOULD be used.

If the selected resolver is a plain IP address (e.g. from options 4 or 5) this implies unencrypted DNS. In such cases Discovery of Designated Resolvers [RFC9462] SHOULD be performed to upgrade to encrypted access, where available.

3.3. Source Port and Transaction ID Randomization

Some IoT devices have been observed to have insufficient or no randomization in the source ports of DNS queries or DNS transaction IDs. This leaves them vulnerable to spoofed responses. A combination of Source Port and Transaction ID is used, amongst other criteria, by the stub resolver when accepting a DNS response.

IoT devices MUST undergo adequate Source Port and Transaction ID randomization in their DNS queries as a mitigation against cache poisoning from spoofed responses. Having both of these values correctly randomized decreases the chances of a successful spoofed attack. Stub resolvers MUST follow the recommendations of [RFC5452] as described in Section 4.5 to ensure Source Port randomization and Transaction ID randomization as required by [RFC1035].

3.4. Handling of TTL Values

IoT devices have been observed making unexpectedly high numbers of DNS queries even when DNS record Time-To-Live values (TTLs) would mean this should be unnecessary. Devices have also been observed issuing DNS queries at fixed, highly predictable intervals for the same domain names, regardless of operational changes or TTL values.

Unnecessary queries may lead to a drain of power in resource-constrained IoT devices. Conversely, very high TTLs may impact device operations such as communicating with management servers, receiving software updates, or other changes, which may lead to security issues. Deterministic querying behavior increases the risk of device fingerprinting by adversaries who can profile query timing to identify specific device models or firmware versions.

The ideal operational scenario is for the owners of the authoritative zones used to manage the devices setting TTL values appropriately for the zones and specific records within them. Devices would then query these records only as needed.

IoT devices MUST cache DNS responses and SHOULD honour TTLs when caching. If for operational reasons this is not ideal, such as the case where a management server record could be cached for an extended period preventing failover or change, then minimum and maximum TTLs MAY be configurable on the device but MUST NOT not be hardcoded values. Where IoT stub resolvers cannot be configured with minimum and maximum TTL values, this can be mitigated by setting these on the network resolver.

If certain device operational requirements necessitate periodic revalidation of critical domains (e.g. management servers), these repeated queries SHOULD use non-deterministic inter-query timing to avoid fixed intervals.

In case of unsuccessful resolution, such as when the resolver is unavailable, IoT devices should implement exponential back-off strategies.

3.5. Support of EDNS(0)

Devices have been observed having limited support for EDNS(0), causing them to revert to TCP for queries over 512 bytes, affecting the device’s efficiency. Other research findings include consuming additional processing resources and failing to maintain their network connectivity when responses to DNS requests exceed 512 bytes.

IoT devices MUST support EDNS(0) and send a supported UDP packet size via OPT 41 [RFC6891]. To avoid fragmentation of UDP packets, which may be dropped by intervening networks, the maximum packet size SHOULD be set to 1220 bytes as a default, although device configuration MAY allow this to be configurable. Although the networks to which IoT devices connect may support larger packet sizes than 1220 bytes, the nature of these devices in being deployed on many network types and DNS queries traversing networks controlled by different operators means it is operationally more effective to use this value. In addition, IoT devices MUST support using TCP for queries when a TC bit is returned from the resolver [RFC1035].

3.6. Improve Device Behavior in Response to Resolution Problems

When resolving domain names, IoT devices may be unable to obtain an answer, and as a result, surges in the number of queries and retries have been observed, or an increase in queries using an alternate protocol (more aggressively querying via IPv6 rather than IPv4).

The use of serve-stale [RFC8767] by resolver software on the IoT device may mitigate the impact of failed resolution, such as when authoritative servers are unavailable. If the stub-resolver has this

capability, device manufacturers should consider the benefits and any impact of using this. Network operators SHOULD configure DNS resolvers to use serve-stale for networks supporting IoT devices, especially where these networks are dedicated to this type of device, to limit any operational impact on IoT devices when resolution fails. Network operators MUST support IoT devices with dual-stack resolvers, rather than providing only IPv4 resolvers when devices are configured with both IPv4 and IPv6.

3.7. Use of DNSSEC

IoT devices can be induced to contact an adversary server or make large volumes of DNS queries via spoofed responses to queries. It would be difficult for manufacturers to mitigate this by implementing checks of data received via DNS queries, such as validating IP addresses in the A/AAAA record RDATA. In addition any validation of this type does not address the problem of MiTM attacks that could be the attack vector.

Devices MAY improve security in their DNS stacks by utilizing DNSSEC validation [RFC9364]. Where the stub resolver is not capable of DNSSEC validation, but does support checking that queries are validated, the resolver used by the device as described in section <<configuring-resolvers>> MUST be configured to validate responses. Manufacturers should consider the type of network the device is likely to be deployed on, such as a home network vs. other types, in determining the likelihood of DNSSEC being available on the network and thus deciding if the device should rely on a validating resolver or making the device independently capable of validation.

Manufacturers MUST sign public zones used for device management and services to ensure queries can be validated to support any of their stub-resolvers or more generally resolvers that will use DNSSEC for validation. This will improve security regardless of whether devices can support checking that queries are validated.

IoT stub-resolvers may adopt one of two models for validation: - Resolver validation - stub-resolvers using the Authenticated Data (AD) bit, which is suitable for constrained devices but requires explicit trust in the upstream resolver - Full validation - local cryptographic checks, providing stronger assurance

Both models improve security over unvalidated queries, but manufacturers should weigh the security considerations, such as trust assumptions, against the operational feasibility when considering which approach to take.

3.7.1. Resolver Validation

Where a manufacturer does utilize DNSSEC validation on the device the minimum implementation, from a device perspective, will be a validating resolver and the device supporting the AD bit. Relying solely on the AD bit assumes that the upstream resolver is trustworthy and uncompromised.

Manufacturers may implement a testing mechanism to determine if the network resolver supports DNSSEC so that it can utilize validation in a network that supports it, or fall back to unvalidated queries. Any test of the resolver will only validate that it supports DNSSEC, given that the resolver is performing the validation it must be explicitly trusted.

In order to check that a DNS query has been validated a stub-resolver MUST check the Authenticated Data (AD) bit [RFC4035] in responses to determine whether data was validated by the resolver it is using. When checking for the AD bit stub-resolvers MUST treat DNSSEC validation failures as fatal. Responses that fail validation MUST NOT be used for name resolution.

3.7.2. Full Validation

Device stub-resolvers can perform validation themselves in cases where the network resolver does not validate queries or the device does not trust the network resolver to do so.

Considerations for device manufacturers in implementing full validation include:

* Devices performing local validation gain end-to-end trust but at higher computational cost

* Devices should cache results including intermediate validation results to reduce repeated computation

* Devices will need to be shipped with a root trust anchor and have a mechanism to securely update this

To implement full local validation stub-resolvers MUST conform to [RFC4035] section 4.9. In practice it is likely to be easier for manufacturers to implement a minimum footprint validating recursive server on the device, configured to forward queries to a network resolver if necessary, rather than develop this capability in any stub resolver.

4. Security Considerations

This BCP discusses security considerations for IoT devices in section Recommendations for IoT Device Stub Resolvers and mitigations that can be implemented on DNS resolvers. More general DNS security considerations in managing networks with IoT devices are detailed here.

Most IoT devices do not have specific security software agents installed on them, as is typically the case with general-purpose operating systems and supply chain vulnerabilities may mean that these devices are compromised before reaching the consumer. Network operators can use DNS resolvers to mitigate these risks.

Private network operators MAY block DNS traffic to any resolvers other than those managed by the operator, so that traffic is not bypassing any DNS security controls such as response policy zones or DNS traffic logging. This is more likely to be the case on enterprise or other private networks rather than service providers that don’t want to limit customers using alternate resolvers.

Providers SHOULD alter resolver configuration to mitigate some of the security risks or operational issues identified in this BCP where it does not impact the operation of other types of DNS clients. For instance the use of serve-stale [RFC8767] is likely to benefit all stub resolvers on a network.

Where operators have networks dedicated to IoT devices, they MAY limit DNS resolution to only domain names used by those IoT devices to mitigate any impact in the event of a compromise to the device. Manufacturers SHOULD provide domain names used for communication to facilitate this and other security measures used to secure devices and identify those that are compromised. Manufacturer Usage Descriptions (MUDs) could provide details of domain names used in device operations that can then be added to DNS security controls.

DNS queries are most commonly carried over UDP and compromised devices have been used in DoS attacks by sending queries with forged source addresses, hence network operators MUST implement [RFC2827] network ingress filtering. Network operators should implement DNS Response Rate Limiting (RRL) on resolvers to mitigate high query volumes from devices causing DoS to the DNS infrastructure.

5. IANA Considerations

This document has no IANA actions.

6. References

6.1. Normative References

[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>.

[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>.

6.2. Informative 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>.

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, DOI 10.17487/RFC2132, March 1997, <https://www.rfc-editor.org/rfc/rfc2132>.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <https://www.rfc-editor.org/rfc/rfc2827>.

[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/rfc/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/rfc/rfc5452>.

[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/rfc/rfc6891>.

[RFC7830] Mayrhofer, A., "The EDNS(0) Padding Option", RFC 7830, DOI 10.17487/RFC7830, May 2016, <https://www.rfc-editor.org/rfc/rfc7830>.

[RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <https://www.rfc-editor.org/rfc/rfc7858>.

[RFC8106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 8106, DOI 10.17487/RFC8106, March 2017, <https://www.rfc-editor.org/rfc/rfc8106>.

[RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <https://www.rfc-editor.org/rfc/rfc8415>.

[RFC8467] Mayrhofer, A., "Padding Policies for Extension Mechanisms for DNS (EDNS(0))", RFC 8467, DOI 10.17487/RFC8467, October 2018, <https://www.rfc-editor.org/rfc/rfc8467>.

[RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://www.rfc-editor.org/rfc/rfc8484>.

[RFC8767] Lawrence, D., Kumari, W., and P. Sood, "Serving Stale Data to Improve DNS Resiliency", RFC 8767, DOI 10.17487/RFC8767, March 2020, <https://www.rfc-editor.org/rfc/rfc8767>.

[RFC9250] Huitema, C., Dickinson, S., and A. Mankin, "DNS over Dedicated QUIC Connections", RFC 9250, DOI 10.17487/RFC9250, May 2022, <https://www.rfc-editor.org/rfc/rfc9250>.

[RFC9364] Hoffman, P., "DNS Security Extensions (DNSSEC)", BCP 237, RFC 9364, DOI 10.17487/RFC9364, February 2023, <https://www.rfc-editor.org/rfc/rfc9364>.

[RFC9462] Pauly, T., Kinnear, E., Wood, C. A., McManus, P., and T. Jensen, "Discovery of Designated Resolvers", RFC 9462, DOI 10.17487/RFC9462, November 2023, <https://www.rfc-editor.org/rfc/rfc9462>.

[RFC9463] Boucadair, M., Ed., Reddy.K, T., Ed., Wing, D., Cook, N., and T. Jensen, "DHCP and Router Advertisement Options for the Discovery of Network-designated Resolvers (DNR)", RFC 9463, DOI 10.17487/RFC9463, November 2023, <https://www.rfc-editor.org/rfc/rfc9463>.

Acknowledgments

We thank the researchers, reviewers, and engineers who contributed to the analysis and testing process.

The authors thank Mohamed Boucadair, Chris Box, Eliot Lear, Martine Sophie Lenders, Jim Reid, Michael Richardson and Hannes Tschofenig for their contributions, questions and comments.

Authors' Addresses

Abhishek Mishra
Inria
Email: abhishek.mishra@inria.fr

Andrew Losty
UCL
Email: andrew.losty.23@ucl.ac.uk

Anna Maria Mandalari
UCL
Email: a.mandalari@ucl.ac.uk

Jim Mozley
Infoblox
Email: jmozley@infoblox.com

Mathieu Cunche
Inria
Email: mathieu.cunche@inria.fr