Internet Engineering Task Force (IETF)
Request for Comments: 9520
Updates: 2308, 4035, 4697
Category: Standards Track
ISSN: 2070-1721

D. Wessels
W. Carroll
M. Thomas
Verisign

December 2023

Negative Caching of DNS Resolution Failures

Abstract

In the DNS, resolvers employ caching to reduce both latency for end users and load on authoritative name servers. The process of resolution may result in one of three types of responses: (1) a response containing the requested data, (2) a response indicating the requested data does not exist, or (3) a non-response due to a resolution failure in which the resolver does not receive any useful information regarding the data's existence. This document concerns itself only with the third type.

RFC 2308 specifies requirements for DNS negative caching. There, caching of TYPE 2 responses is mandatory and caching of TYPE 3 responses is optional. This document updates RFC 2308 to require negative caching for DNS resolution failures.

RFC 4035 allows DNSSEC validation failure caching. This document updates RFC 4035 to require caching for DNSSEC validation failures.

RFC 4697 prohibits aggressive requerying for NS records at a failed zone's parent zone. This document updates RFC 4697 to expand this requirement to all query types and to all ancestor zones.

Status of This Memo

This is an Internet Standards Track document.

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). Further information on Internet Standards is available in 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/rfc9520.

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

Caching has always been a fundamental component of DNS resolution on the Internet. For example, [RFC0882] states:

> The sheer size of the database and frequency of updates suggest that it must be maintained in a distributed manner, with local caching to improve performance.

The early DNS RFCs ([RFC0882], [RFC0883], [RFC1034], and [RFC1035]) primarily discuss caching in the context of what [RFC2308] calls "positive responses", that is, when the response includes the requested data. In this case, a TTL is associated with each Resource Record (RR) in the response. Resolvers can cache and reuse the data until the TTL expires.

Section 4.3.4 of [RFC1034] describes negative response caching, but notes it is optional and only talks about name errors (NXDOMAIN). This is the origin of using the SOA MINIMUM field as a negative caching TTL.

[RFC2308] updated [RFC1034] to specify new requirements for DNS negative caching, including making it mandatory for caching resolvers

to cache name error (NXDOMAIN) and no data (NODATA) responses when an SOA record is available to provide a TTL. [RFC2308] further specified optional negative caching for two DNS resolution failure cases: server failure and dead/unreachable servers.

This document updates [RFC2308] to require negative caching of all DNS resolution failures and provides additional examples of resolution failures, [RFC4035] to require caching for DNSSEC validation failures, as well as [RFC4697] to expand the scope of prohibiting aggressive requerying for NS records at a failed zone's parent zone to all query types and to all ancestor zones.

1.1. Motivation

Operators of DNS services have known for some time that recursive resolvers become more aggressive when they experience resolution failures. A number of different anecdotes, experiments, and incidents support this claim.

In December 2009, a secondary server for a number of in-addr.arpa subdomains saw its traffic suddenly double, and queries of type DNSKEY in particular increase by approximately two orders of magnitude, coinciding with a DNSSEC key rollover by the zone operator [DNSSEC-ROLLOVER]. This predated a signed root zone, and an operating system vendor was providing non-root trust anchors to the recursive resolver, which became out of date following the rollover. Unable to validate responses for the affected in-addr.arpa zones, recursive resolvers aggressively retried their queries.

In 2016, the Internet infrastructure company Dyn experienced a large attack that impacted many high-profile customers. As documented in a technical presentation detailing the attack (see [RETRY-STORM]), Dyn staff wrote:

> At this point we are now experiencing botnet attack traffic and what is best classified as a "retry storm"
>
> Looking at certain large recursive platforms > 10x normal volume

In 2018, the root zone Key Signing Key (KSK) was rolled over [KSK-ROLLOVER]. Throughout the rollover period, the root servers experienced a significant increase in DNSKEY queries. Before the rollover, a.root-servers.net and j.root-servers.net together received about 15 million DNSKEY queries per day. At the end of the revocation period, they received 1.2 billion per day: an 80x increase. Removal of the revoked key from the zone caused DNSKEY queries to drop to post-rollover but pre-revoke levels, indicating there is still a population of recursive resolvers using the previous root trust anchor and aggressively retrying DNSKEY queries.

In 2021, Verisign researchers used botnet query traffic to demonstrate that certain large public recursive DNS services exhibit very high query rates when all authoritative name servers for a zone return refused (REFUSED) or server failure (SERVFAIL) responses (see [BOTNET]). When the authoritative servers were configured normally, query rates for a single botnet domain averaged approximately 50

queries per second. However, with the servers configured to return SERVFAIL, the query rate increased to 60,000 per second. Furthermore, increases were also observed at the root and Top-Level Domain (TLD) levels, even though delegations at those levels were unchanged and continued operating normally.

Later that same year, on October 4, Facebook experienced a widespread and well-publicized outage [FB-OUTAGE]. During the 6-hour outage, none of Facebook's authoritative name servers were reachable and did not respond to queries. Recursive name servers attempting to resolve Facebook domains experienced timeouts. During this time, query traffic on the .COM/.NET infrastructure increased from 7,000 to 900,000 queries per second [OUTAGE-RESOLVER].

1.2. Related Work

[RFC2308] describes negative caching for four types of DNS queries and responses: name errors, no data, server failures, and dead/unreachable servers. It places the strongest requirements on negative caching for name errors and no data responses, while server failures and dead servers are left as optional.

[RFC4697] is a Best Current Practice that documents observed resolution misbehaviors. It describes a number of situations that can lead to excessive queries from recursive resolvers, including requerying for delegation data, lame servers, responses blocked by firewalls, and records with zero TTL. [RFC4697] makes a number of recommendations, varying from "SHOULD" to "MUST".

[THUNDERING-HERD] describes "The DNS thundering herd problem" as a situation arising when cached data expires at the same time for a large number of users. Although that document is not focused on negative caching, it does describe the benefits of combining multiple identical queries to upstream name servers. That is, when a recursive resolver receives multiple queries for the same name, class, and type that cannot be answered from cached data, it should combine or join them into a single upstream query rather than emit repeated identical upstream queries.

[RFC5452], "Measures for Making DNS More Resilient against Forged Answers", includes a section that describes the phenomenon known as "Birthday Attacks". Here, again, the problem arises when a recursive resolver emits multiple identical upstream queries. Multiple outstanding queries make it easier for an attacker to guess and correctly match some of the DNS message parameters, such as the port number and ID field. This situation is further exacerbated in the case of timeout-based resolution failures. Of course, DNSSEC is a suitable defense to spoofing attacks.

[RFC8767] describes "Serving Stale Data to Improve DNS Resiliency". This permits a recursive resolver to return possibly stale data when it is unable to refresh cached, expired data. It introduces the idea of a failure recheck timer and says:

> Attempts to refresh from non-responsive or otherwise failing authoritative nameservers are recommended to be done no more

| frequently than every 30 seconds.

1.3. 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.

DNS transport: In this document, "DNS transport" means a protocol used to transport DNS messages between a client and a server. This includes "classic DNS" transports, i.e., DNS-over-UDP and DNS-over-TCP [RFC1034] [RFC7766], as well as newer encrypted DNS transports, such as DNS-over-TLS [RFC7858], DNS-over-HTTPS [RFC8484], DNS-over-QUIC [RFC9250], and similar communication of DNS messages using other protocols. Note: at the time of writing, not all DNS transports are standardized for all types of servers but may become standardized in the future.

2. Conditions That Lead to DNS Resolution Failures

A DNS resolution failure occurs when none of the servers available to a resolver client provide any useful response data for a particular query name, type, and class. A response is considered useful when it provides either the requested data, a referral to a descendant zone, or an indication that no data exists at the given name.

It is common for resolvers to have multiple servers from which to choose for a particular query. For example, in the case of stub-to-recursive, the stub resolver may be configured with multiple recursive resolver addresses. In the case of recursive-to-authoritative, a given zone usually has more than one name server (NS record), each of which can have multiple IP addresses and multiple DNS transports.

Nothing in this document prevents a resolver from retrying a query at a different server or the same server over a different DNS transport. In the case of timeouts, a resolver can retry the same server and DNS transport a limited number of times.

If any one of the available servers provides a useful response, then it is not considered a resolution failure. However, if none of the servers for a given query tuple <name, type, class> provide a useful response, the result is a resolution failure.

Note that NXDOMAIN and NOERROR/NODATA responses are not conditions for resolution failure. In these cases, the server is providing a useful response, indicating either that a name does not exist or that no data of the requested type exists at the name. These negative responses can be cached as described in [RFC2308].

The remainder of this section describes a number of different conditions that can lead to resolution failure. This section is not exhaustive. Additional conditions may be expected to cause similar resolution failures.

2.1. SERVFAIL Responses

Server failure is defined in [RFC1035] as: "The name server was unable to process this query due to a problem with the name server." A server failure is signaled by setting the RCODE field to SERVFAIL.

Authoritative servers return SERVFAIL when they don't have any valid data for a zone. For example, a secondary server has been configured to serve a particular zone but is unable to retrieve or refresh the zone data from the primary server.

Recursive servers return SERVFAIL in response to a number of different conditions, including many described below.

Although the extended DNS errors method exists "primarily to extend SERVFAIL to provide additional information," it "does not change the processing of RCODEs" [RFC8914]. This document operates at the level of resolution failure and does not concern particular causes.

2.2. REFUSED Responses

A name server returns a message with the RCODE field set to REFUSED when it refuses to process the query, e.g., for policy or other reasons [RFC1035].

Authoritative servers generally return REFUSED when processing a query for which they are not authoritative. For example, a server that is configured to be authoritative for only the example.net zone may return REFUSED in response to a query for example.com.

Recursive servers generally return REFUSED for query sources that do not match configured access control lists. For example, a server that is configured to allow queries from only 2001:db8:1::/48 may return REFUSED in response to a query from 2001:db8:5::1.

2.3. Timeouts and Unreachable Servers

A timeout occurs when a resolver fails to receive any response from a server within a reasonable amount of time. Additionally, a DNS transport may more quickly indicate lack of reachability in a way that wouldn't be considered a timeout: for example, an ICMP port unreachable message, a TCP "connection refused" error, or a TLS handshake failure. [RFC2308] refers to these conditions collectively as "dead / unreachable servers".

Note that resolver implementations may have two types of timeouts: a smaller timeout that might trigger a query retry and a larger timeout after which the server is considered unresponsive. Section 3.1 discusses the requirements for resolvers when retrying queries.

Timeouts can present a particular problem for negative caching, depending on how the resolver handles multiple outstanding queries for the same <query name, type, class> tuple. For example, consider a very popular website in a zone whose name servers are all unresponsive. A recursive resolver might receive tens or hundreds of

queries per second for that website. If the recursive server implementation joins these outstanding queries together, then it only sends one recursive-to-authoritative query for the numerous pending stub-to-recursive queries. However, if the implementation does not join outstanding queries together, then it sends one recursive-to-authoritative query for each stub-to-recursive query. If the incoming query rate is high and the timeout is large, this might result in hundreds or thousands of recursive-to-authoritative queries while waiting for an authoritative server to time out.

A recursive resolver that does not join outstanding queries together is more susceptible to Birthday Attacks ([RFC5452], Section 5), especially when those queries result in timeouts.

2.4. Delegation Loops

A delegation loop, or cycle, can occur when one domain utilizes name servers in a second domain, and the second domain uses name servers in the first. For example:

```
FOO.EXAMPLE.    NS      NS1.EXAMPLE.COM.
FOO.EXAMPLE.    NS      NS2.EXAMPLE.COM.

EXAMPLE.COM.    NS      NS1.FOO.EXAMPLE.
EXAMPLE.COM.    NS      NS2.FOO.EXAMPLE.
```

In this example, no names under foo.example or example.com can be resolved because of the delegation loop. Note that a delegation loop may involve more than two domains. A resolver that does not detect delegation loops may generate DDoS-levels of attack traffic to authoritative name servers, as documented in the TsuNAME vulnerability [TsuNAME].

2.5. Alias Loops

An alias loop, or cycle, can occur when one CNAME or DNAME RR refers to a second name, which, in turn, is specified as an alias for the first. For example:

```
APP.FOO.EXAMPLE.        CNAME   APP.EXAMPLE.NET.
APP.EXAMPLE.NET.        CNAME   APP.FOO.EXAMPLE.
```

The need to detect CNAME loops has been known since at least [RFC1034], which states in Section 3.6.2:

> Of course, by the robustness principle, domain software should not fail when presented with CNAME chains or loops; CNAME chains should be followed and CNAME loops signalled as an error.

2.6. DNSSEC Validation Failures

For zones that are signed with DNSSEC, a resolution failure can occur when a security-aware resolver believes it should be able to establish a chain of trust for an RRset but is unable to do so, possibly after trying multiple authoritative name servers. DNSSEC validation failures may be due to signature mismatch, missing DNSKEY

RRs, problems with denial-of-existence records, clock skew, or other reasons.

Section 4.7 of [RFC4035] already discusses the requirements and reasons for caching validation failures. Section 3.4 of this document strengthens those requirements.

2.7. FORMERR Responses

A name server returns a message with the RCODE field set to FORMERR when it is unable to interpret the query [RFC1035]. FORMERR responses are often associated with problems processing Extension Mechanisms for DNS (EDNS(0)) [RFC6891]. Authoritative servers may return FORMERR when they do not implement EDNS(0), or when EDNS(0) option fields are malformed, but not for unknown EDNS(0) options.

Upon receipt of a FORMERR response, some recursive clients will retry their queries without EDNS(0), while others will not. Nonetheless, resolution failures from FORMERR responses are rare.

3. Requirements for Caching DNS Resolution Failures

3.1. Retries and Timeouts

A resolver MUST NOT retry a given query to a server address over a given DNS transport more than twice (i.e., three queries in total) before considering the server address unresponsive over that DNS transport for that query.

A resolver MAY retry a given query over a different DNS transport to the same server if it has reason to believe the DNS transport is available for that server and is compatible with the resolver's security policies.

This document does not place any requirements on how long an implementation should wait before retrying a query (aka a timeout value), which may be implementation or configuration dependent. It is generally expected that typical timeout values range from 3 to 30 seconds.

3.2. Caching

Resolvers MUST implement a cache for resolution failures. The purpose of this cache is to eliminate repeated upstream queries that cannot be resolved. When an incoming query matches a cached resolution failure, the resolver MUST NOT send any corresponding outgoing queries until after the cache entries expire.

Implementation details for such a cache are not specified in this document. The implementation might cache different resolution failure conditions differently. For example, DNSSEC validation failures might be cached according to the queried name, class, and type, whereas unresponsive servers might be cached only according to the server's IP address. Developers should document their implementation choices so that operators know what behaviors to expect when resolution failures are cached.

Resolvers MUST cache resolution failures for at least 1 second. Resolvers MAY cache different types of resolution failures for different (i.e., longer) amounts of time. Consistent with [RFC2308], resolution failures MUST NOT be cached for longer than 5 minutes.

The minimum cache duration SHOULD be configurable by the operator. A longer cache duration for resolution failures will reduce the processing burden from repeated queries but may also increase the time to recover from transitory issues.

Resolvers SHOULD employ an exponential or linear backoff algorithm to increase the cache duration for persistent resolution failures. For example, the initial time for negatively caching a resolution failure might be set to 5 seconds and increased after each retry that results in another resolution failure, up to a configurable maximum, not to exceed the 5-minute upper limit.

Notwithstanding the above, resolvers SHOULD implement measures to mitigate resource exhaustion attacks on the failed resolution cache. That is, the resolver should limit the amount of memory and/or processing time devoted to this cache.

3.3. Requerying Delegation Information

Section 2.1 of [RFC4697] identifies circumstances in which:

> ...every name server in a zone's NS RRSet is unreachable (e.g., during a network outage), unavailable (e.g., the name server process is not running on the server host), or misconfigured (e.g., the name server is not authoritative for the given zone, also known as "lame").

It prohibits unnecessary "aggressive requerying" to the parent of a non-responsive zone by sending NS queries.

The problem of aggressive requerying to parent zones is not limited to queries of type NS. This document updates the requirement from Section 2.1.1 of [RFC4697] to apply more generally:

> Upon encountering a zone whose name servers are all non-responsive, a resolver MUST cache the resolution failure. Furthermore, the resolver MUST limit queries to the non-responsive zone's parent zone (and to other ancestor zones) just as it would limit subsequent queries to the non-responsive zone.

3.4. DNSSEC Validation Failures

Section 4.7 of [RFC4035] states:

> To prevent such unnecessary DNS traffic, security-aware resolvers MAY cache data with invalid signatures, with some restrictions.

This document updates [RFC4035] with the following, stronger, requirement:

> To prevent such unnecessary DNS traffic, security-aware resolvers MUST cache DNSSEC validation failures, with some restrictions.

One of the restrictions mentioned in [RFC4035] is to use a small TTL when caching data that fails DNSSEC validation. This is, in part, because the provided TTL cannot be trusted. The advice from Section 3.2 herein can be used as guidance on TTLs for caching DNSSEC validation failures.

4. IANA Considerations

This document has no IANA actions.

5. Security Considerations

As noted in Section 3.2, an attacker might attempt a resource exhaustion attack by sending queries for a large number of names and/or types that result in resolution failure. Resolvers SHOULD implement measures to protect themselves and bound the amount of memory devoted to caching resolution failures.

A cache poisoning attack (see Section 2.2 of [RFC7873]) resulting in denial of service may be possible because failure messages cannot be signed. An attacker might generate queries and send forged failure messages, causing the resolver to cease sending queries to the authoritative name server (see Section 2.6 of [RFC4732] for a similar "data corruption attack" and Section 5.2 of [TuDoor] for a "DNSDoS attack"). However, this would require continued spoofing throughout the backoff period and repeated attacks due to the 5-minute cache limit. As in Section 4.1.12 of [RFC4686], this attack's effects would be "localized and of limited duration".

6. Privacy Considerations

This specification has no impact on user privacy.

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>.

[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>.

[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>.

[RFC4697] Larson, M. and P. Barber, "Observed DNS Resolution Misbehavior", BCP 123, RFC 4697, DOI 10.17487/RFC4697, October 2006, <https://www.rfc-editor.org/info/rfc4697>.

[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>.

7.2. Informative References

[BOTNET] Wessels, D. and M. Thomas, "Botnet Traffic Observed at Various Levels of the DNS Hierarchy", May 2021, <https://indico.dns-oarc.net/event/38/contributions/841/>.

[DNSSEC-ROLLOVER] Michaleson, G., Wallström, P., Arends, R., and G. Huston, "Roll Over and Die?", February 2010, <https://www.potaroo.net/ispcol/2010-02/rollover.html>.

[FB-OUTAGE] Janardhan, S., "More details about the October 4 outage", October 2021, <https://engineering.fb.com/2021/10/05/networking-traffic/outage-details/>.

[KSK-ROLLOVER] Müller, M., Thomas, M., Wessels, D., Hardaker, W., Chung, T., Toorop, W., and R. van Rijswijk-Deij, "Roll, Roll, Roll Your Root: A Comprehensive Analysis of the First Ever DNSSEC Root KSK Rollover", IMC '19: Proceedings of the Internet Measurement Conference, Pages 1-14, DOI 10.1145/3355369.3355570, October 2019, <https://doi.org/10.1145/3355369.3355570>.

[OUTAGE-RESOLVER] Verisign, "Observations on Resolver Behavior During DNS Outages", January 2022, <https://blog.verisign.com/security/facebook-dns-outage/>.

[RETRY-STORM] Sullivan, A., "Dyn, DDoS, and DNS", March 2017, <https://ccnso.icann.org/sites/default/files/file/field-file-attach/2017-04/presentation-oracle-dyn-ddos-dns-13mar17-en.pdf>.

[RFC0882] Mockapetris, P., "Domain names: Concepts and facilities", RFC 882, DOI 10.17487/RFC0882, November 1983, <https://www.rfc-editor.org/info/rfc882>.

[RFC0883] Mockapetris, P., "Domain names: Implementation specification", RFC 883, DOI 10.17487/RFC0883, November

1983, <https://www.rfc-editor.org/info/rfc883>.

[RFC4686] Fenton, J., "Analysis of Threats Motivating DomainKeys Identified Mail (DKIM)", RFC 4686, DOI 10.17487/RFC4686, September 2006, <https://www.rfc-editor.org/info/rfc4686>.

[RFC4732] Handley, M., Ed., Rescorla, E., Ed., and IAB, "Internet Denial-of-Service Considerations", RFC 4732, DOI 10.17487/RFC4732, December 2006, <https://www.rfc-editor.org/info/rfc4732>.

[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>.

[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>.

[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>.

[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/info/rfc7858>.

[RFC7873] Eastlake 3rd, D. and M. Andrews, "Domain Name System (DNS) Cookies", RFC 7873, DOI 10.17487/RFC7873, May 2016, <https://www.rfc-editor.org/info/rfc7873>.

[RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://www.rfc-editor.org/info/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/info/rfc8767>.

[RFC8914] Kumari, W., Hunt, E., Arends, R., Hardaker, W., and D. Lawrence, "Extended DNS Errors", RFC 8914, DOI 10.17487/RFC8914, October 2020, <https://www.rfc-editor.org/info/rfc8914>.

[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/info/rfc9250>.

[THUNDERING-HERD] Sivaraman, M. and C. Liu, "The DNS thundering herd

problem", Work in Progress, Internet-Draft, draft-muks-dnsop-dns-thundering-herd-00, 25 June 2020, <https://datatracker.ietf.org/doc/html/draft-muks-dnsop-dns-thundering-herd-00>.

[TsuNAME] Moura, G. C. M., Castro, S., Heidemann, J., and W. Hardaker, "TsuNAME: exploiting misconfiguration and vulnerability to DDoS DNS", IMC '21: Proceedings of the 21st ACM Internet Measurement Conference, Pages 398-418, DOI 10.1145/3487552.3487824, November 2021, <https://doi.org/10.1145/3487552.3487824>.

[TuDoor] Li, X., Xu, W., Liu, B., Zhang, M., Li, Z., Zhang, J., Chang, D., Zheng, X., Wang, C., Chen, J., Duan, H., and Q. Li, "TuDoor Attack: Systematically Exploring and Exploiting Logic Vulnerabilities in DNS Response Pre-processing with Malformed Packets", IEEE Symposium on Security and Privacy (SP), DOI 10.1109/SP54263.2024.00046, 2024, <https://doi.ieeecomputersociety.org/10.1109/SP54263.2024.00046>.

Acknowledgments

The authors wish to thank Mukund Sivaraman, Petr Spacek, Peter van Dijk, Tim Wicinksi, Joe Abley, Evan Hunt, Barry Leiba, Lucas Pardue, Paul Wouters, and other members of the DNSOP Working Group for their feedback and contributions.

Authors' Addresses

Duane Wessels
Verisign
12061 Bluemont Way
Reston, VA 20190
United States of America
Phone: +1 703 948-3200
Email: dwessels@verisign.com
URI: https://verisign.com

William Carroll
Verisign
12061 Bluemont Way
Reston, VA 20190
United States of America
Phone: +1 703 948-3200
Email: wicarroll@verisign.com
URI: https://verisign.com

Matthew Thomas
Verisign
12061 Bluemont Way
Reston, VA 20190
United States of America
Phone: +1 703 948-3200

Email: mthomas@verisign.com
URI: https://verisign.com