

Internet Engineering Task Force
Internet-Draft
Updates: 4034, 4035 (if approved)
Intended status: Standards Track
Expires: 31 August 2025

S. Huque
Salesforce
C. Elmerot
Cloudflare
O. Gudmundsson
Retired / Unaffiliated
27 February 2025

Compact Denial of Existence in DNSSEC
draft-ietf-dnsop-compact-denial-of-existence-07

Abstract

This document describes a technique to generate a signed DNS response on demand for a non-existent name by claiming that the name exists but doesn't have any data for the queried record type. Such answers require only one minimally covering NSEC or NSEC3 record, allow online signing servers to minimize signing operations and response sizes, and prevent zone content disclosure.

This document updates RFC 4034 and 4035.

Discussion Venues

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

Source for this draft and an issue tracker can be found at <https://github.com/shuque/id-dnssec-compact-lies>.

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 31 August 2025.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction and Motivation	3
1.1. Requirements Language	3
2. Distinguishing non-existent names	4
3. Generating Responses with NSEC	4
3.1. Responses for Non-Existent Names	4
3.2. Responses for Non-Existent Types	5
3.3. Responses for Wildcard Matches	5
3.4. Responses for Unsigned Referrals	6
3.5. Responses to explicit queries for NXNAME	6
4. Generating Responses with NSEC3	7
5. Response Code Restoration	8
5.1. Signaled Response Code Restoration	8
6. Operational Implications	9
7. Updates to RFCs	9
7.1. Updates to RFC 4034	9
7.2. Updates to RFC 4035	10
8. Security Considerations	10
9. Acknowledgements	11
10. IANA Considerations	12
11. References	12
11.1. Normative References	12
11.2. Informative References	13
Appendix A. Other Approaches	14
Appendix B. Historical Implementation Notes	15
Authors' Addresses	15

1. Introduction and Motivation

One of the functions of the Domain Name System Security Extensions (DNSSEC) [RFC9364] is "Authenticated Denial of Existence", i.e., proving that a DNS name or record type does not exist. Normally, this is done by means of signed NSEC or NSEC3 records. In the precomputed signature model, these records chain together existing names, or cryptographic hashes of them in the zone. In the online signing model, described in NSEC and NSEC3 "White Lies" [RFC4470] [RFC7129], they are used to dynamically compute an epsilon function around the queried name. A "Type Bit Maps" field in the data of the NSEC or NSEC3 record asserts which resource record types are present at the name.

The response for a non-existent name requires up to 2 signed NSEC records or up to 3 signed NSEC3 records (and for online signers, the associated cryptographic computation), to prove that (1) the name did not explicitly exist in the zone, and (2) that it could not have been synthesized by a wildcard.

This document describes an alternative technique, "Compact Denial of Existence" or "Compact Answers", to generate a signed DNS response on demand for a non-existent name by claiming that the name exists but has no resource records associated with the queried type, i.e., it returns a NODATA response rather than an NXDOMAIN response. A NODATA response, which has a response code (RCODE) of NOERROR and an empty ANSWER section, requires only one NSEC or NSEC3 record matching the queried name. This has two advantages: the DNS response size is smaller, and it reduces the online cryptographic work involved in generating the response.

The use of minimally covering NSEC or NSEC3 records also prevents adversaries from enumerating the entire contents of DNS zones by walking the NSEC chain, or by performing an offline dictionary attack on the hashes in the NSEC3 chain.

This document assumes a reasonable level of familiarity with DNS operations and protocol terms. Much of the terminology is explained in further detail in "DNS Terminology" [RFC9499].

1.1. 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.

2. Distinguishing non-existent names

This method generates NODATA responses for non-existent names that don't match a DNS wildcard. Since there are clearly no record types for such names, the NSEC Type Bit Maps field in the response will only contain the "NSEC" and "RRSIG" types (and in the case of NSEC3 the Type Bit Maps field will be empty). Tools that need to accurately identify non-existent names in responses cannot rely on this specific type bitmap because Empty Non-Terminal (ENT) names (which positively exist) also have no record types at the name and will return exactly the same Type Bit Maps field.

This specification defines the use of a synthetic Resource Record type to signal the presence of a non-existent name. The mnemonic for this RR type is "NXNAME", chosen to clearly distinguish it from the response code mnemonic NXDOMAIN.

Type	Value	Meaning
NXNAME	128	NXDOMAIN indicator for Compact Denial of Existence

This RR type is added to the NSEC Type Bit Maps field for responses to non-existent names, in addition to the required RRSIG and NSEC types. If NSEC3 is being used, this RR type is the sole entry in the Type Bit Maps field. It is a "Meta-Type", as defined in [RFC6895], stores no data in a DNS zone, and cannot be usefully queried. Section 3.5 describes what a DNS resolver or authoritative server should do if it receives an explicit query for NXNAME.

No special handling of this RR type is required on the part of DNS resolvers. However, resolvers may optionally implement the behavior described in Section 5.1 (Response Code Restoration) to better restore NXDOMAIN visibility to various applications that may remain oblivious to the new NXNAME signal.

3. Generating Responses with NSEC

This section describes various types of answers generated by authoritative servers implementing Compact Denial of Existence using NSEC. Section 4 describes changes needed to support NSEC3.

3.1. Responses for Non-Existent Names

When the authoritative server receives a query for a non-existent name in a zone that it serves, a NODATA response (response code NOERROR, empty Answer section) is generated with a dynamically constructed NSEC record with the owner name matching the queried name (QNAME) placed in the Authority section.

The Next Domain Name field SHOULD be set to the immediate lexicographic successor of the QNAME. This is accomplished by adding a leading label with a single null (zero-value) octet. The Type Bit Maps field MUST only have the bits set for the following RR Types: RRSIG, NSEC, and NXNAME.

For example, a request for the non-existing name "a.example.com." would result in the following NSEC record to be generated (in DNS presentation format):

```
a.example.com. 3600 IN NSEC \000.a.example.com. RRSIG NSEC NXNAME
```

The NSEC record MUST have corresponding RRSIGs generated.

3.2. Responses for Non-Existent Types

When the authoritative server receives a query for a name that exists, but has no resource record sets associated with the queried type, it generates a NODATA response, with a dynamically constructed signed NSEC record in the Authority Section. The owner name of the NSEC record matches the queried name. The Next Domain Name field is set to the immediate lexicographic successor of the QNAME. The Type Bit Maps field lists the available Resource Record types at the name.

An Empty Non-Terminal is a special subset of this category, where the name has no resource record sets of any type (but has descendant names that do). For a query for an Empty Non-Terminal, the NSEC Type Bit Maps field will only contain RRSIG and NSEC. (Note that this is substantially different than the ENT response in precomputed NSEC, where the NSEC record has the same type bitmap, but "covers" rather than matches the ENT, and has the Next Domain Name field set to the next lexicographic descendent of the ENT in the zone.)

3.3. Responses for Wildcard Matches

For wildcard matches, the authoritative server will provide a dynamically signed response that claims that the queried name exists explicitly. Specifically, the answer RR set will have an RRSIG record demonstrating an exact match (i.e., the label count in the RRSIG RDATA will be equal to the number of labels in the query name minus the root label). This obviates the need to include an NSEC record in the Authority section of the response that shows that no closer match than the wildcard was possible.

For a Wildcard NODATA match (where the queried name matches a wildcard but no data for the queried type exists), a response akin to a non-wildcard NODATA is returned. The Answer section is empty, and the Authority section contains a single NSEC record that matches the query name with a Type Bit Maps field representing the list of types available at the wildcard.

3.4. Responses for Unsigned Referrals

Per the DNSSEC protocol, a referral to an unsigned subzone includes an NSEC record whose Owner Name matches the subzone, and which proves the delegation is unsigned by the absence of the DS RRtype bit in the Type Bit Maps field.

With Compact Denial of Existence, the Next Domain Name field for this NSEC record is computed with a slightly different epsilon function than the immediate lexicographic successor of the Owner Name, as that name would then fall under the delegated subzone. Instead, the Next Domain Name field is formed by appending a zero octet to the first label of the Owner Name.

For example, a referral response for the subzone sub.example.com would include an NSEC record like the following:

```
sub.example.com. 3600 IN NSEC sub\000.example.com. NS RRSIG NSEC
```

3.5. Responses to explicit queries for NXNAME

NXNAME is a meta type which should not appear anywhere in a DNS message apart from the NSEC type bitmap of a Compact Answer response for a non-existent name. If however a DNS server implementing this specification receives an explicit query for the NXNAME RR type, this section describes what the response should be.

If an explicit query for the NXNAME RR type is received, the DNS server MUST return a Format Error (response code FORMERR). A resolver should not forward these queries upstream or attempt iterative resolution. Many DNS server implementations already return errors for all types in the meta and q-type range except those types that are already defined to support queries.

Optionally, a DNS server MAY also include the following [RFC8914] Extended DNS Error (EDE) Code in the response:

INFO-CODE	Purpose
30	Invalid Query Type

Note that this EDE code is generally applicable to any RR type that should not appear in DNS queries.

4. Generating Responses with NSEC3

The NSEC3 protocol [RFC5155] uses hashed names to provide zone enumeration defense. This protection is already better provided by minimally covering NSEC records. NSEC3's Opt-Out feature also provides no specific benefit for online signing implementations (minimally covering NSEC3 records provide no useful Opt-Out span). Hence, there is no known advantage to implementing Compact Denial of Existence with NSEC3. However, an existing implementation of traditional NSEC3 online signing migrating to Compact Denial of Existence may find it simpler to do so with NSEC3 than implementing NSEC from scratch.

For NSEC3, the functional details of the protocol remain as described in Section 3, with the following changes:

NSEC3 records and their signatures are dynamically generated instead of NSEC.

The NSEC3 parameters should be set to algorithm 1, a flags field of 0, an additional hash iteration count of 0, and an empty salt. In DNS presentation format, this is "1 0 0 -".

The Owner Name of the NSEC3 resource record is the NSEC3 hash of the relevant domain name, encoded in Base32 with Extended Hex Alphabet ([RFC4648], Section 7), prepended as a single label to the name of the zone. The Next Hashed Owner Name is the immediate name successor of the unencoded binary form of the previous hash, which can be computed by adding one to the binary hash value.

In responses for non-existent names, the Type Bit Maps field will contain only the NXNAME meta-type. In responses to Empty Non-Terminal names, the Type Bit Maps field will be empty.

For example, a request for the non-existent name "a.example.com." would result in the following NSEC3 record to be generated:

```
H64KFA4P1ACER2EBPS9QSDK6DNP8B3JQ.example.com. IN NSEC3 1 0 0 - (
H64KFA4P1ACER2EBPS9QSDK6DNP8B3JR NXNAME )
```

Unlike Compact Denial of Existence with NSEC, no special form of the next name field for unsigned referrals is needed. The Hashed Next Owner Name field remains the NSEC3 hash of original owner name plus one.

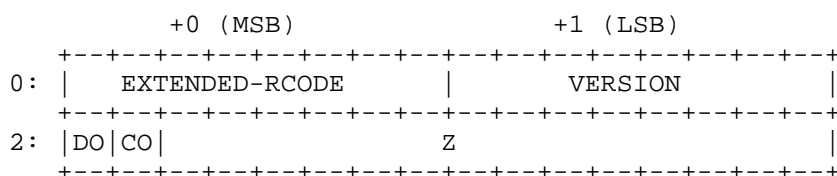
5. Response Code Restoration

For non-existent names, implementations should try wherever possible, to preserve the response code value of 3 (NXDOMAIN). This is generally possible for non-DNSSEC enabled queries, namely those which do not set the DNSSEC_OK EDNS flag (DO bit). For such queries, authoritative servers implementing Compact Denial of Existence could return a normal NXDOMAIN response. There may be limited benefit to doing this however, since most modern DNS resolvers are DNSSEC-aware, and by [RFC3225] Section 3, DNSSEC-aware recursive servers are required to set the DO bit on outbound queries, regardless of the status of the DO bit on incoming requests.

A validating resolver that understands the NXNAME signal from an authoritative server could modify the response code from NOERROR to NXDOMAIN in responses they return to downstream queriers that have not set the DO bit in their requests.

5.1. Signaled Response Code Restoration

This section describes an optional but recommended scheme to permit signaled restoration of the NXDOMAIN RCODE for DNSSEC enabled responses. A new EDNS0 [RFC6891] header flag is defined in the 2nd most significant bit of the flags field in the EDNS0 OPT header. This flag is referred to as the "Compact Answers OK (CO)" flag.



When this flag is sent in a query by a resolver, it indicates that the resolver will accept a signed NXNAME enhanced NODATA response for a non-existent name together with the response code field set to NXDOMAIN (3).

In responses to such queries, a Compact Denial authoritative server implementing this signaling scheme, will set the Compact Answers OK EDNS header flag, and for non-existent names will additionally set the response code field to NXDOMAIN.

EDNS is a hop by hop signal, so resolvers will need to record the presence of this flag in associated cache data and respond to downstream DNSSEC enabled queriers appropriately. If the querier does not set the Compact Answers OK flag, the resolver will need to reset the response code back to NOERROR (0) for an NXNAME response.

6. Operational Implications

For DNSSEC enabled queries, a signed zone at an authoritative server implementing Compact Answers will never return a response with a response code of NXDOMAIN, unless they have implemented the optional response code restoration feature described in Section 5.1. Similarly, resolvers not implementing this feature will also not be able to return NXDOMAIN. In the absence of this, tools that rely on accurately determining non-existent names will need to infer them from the presence of the NXNAME RR type in the Type Bit Maps field of the NSEC record in NODATA responses from these servers.

Address lookup functions typically invoked by applications may need to do more work when dealing with implementations of Compact Answers. For example, a NODATA response to the lookup of an AAAA record for a non-existent name, can cause such functions to issue another query at the same name for an A record. Whereas an NXDOMAIN response to the first query could suppress additional queries for other types at that name. Address lookup functions could be enhanced to issue DNSSEC enabled queries and to examine the NSEC Type Bit Maps field in responses to accurately determine non-existent names. Note that this is less of a concern with Happy Eyeballs [RFC8305] style of connection functions that typically issue back to back DNS queries without waiting for individual responses.

Protocol optimizations that enable DNS resolvers to synthesize NXDOMAIN or wildcard responses, like [RFC8020] and [RFC8198], cannot be realized from responses that use Compact Denial of Existence. In general, no online signing scheme that employs minimally covering NSEC or NSEC3 records (including this one) permits RFC 8198 style NXDOMAIN or wildcard response synthesis. Additionally, this protocol also precludes RFC 8020 style NXDOMAIN synthesis for DNSSEC enabled responses.

7. Updates to RFCs

7.1. Updates to RFC 4034

[RFC4034] Section 4.1.2, The Type Bit Maps Field, states the following:

- * Bits representing pseudo-types MUST be clear, as they do not appear in zone data. If encountered, they MUST be ignored upon being read.

This paragraph is updated to the following:

- * Bits representing pseudo-types MUST be clear, as they do not appear in zone data. If encountered, they MUST be ignored upon being read. There is one exception to this rule for Compact Denial of Existence (RFC TBD), where the NXNAME pseudo-type is allowed to appear in responses to non-existent names.

Note: as a practical matter, no known resolver insists that pseudo-types must be clear in the NSEC Type Bit Maps, so this restriction (prior to its revision) has posed no problem for the deployment of this mechanism.

7.2. Updates to RFC 4035

[RFC4035] Section 2.3, Including NSEC RRs in a Zone, states the following:

- * An NSEC record (and its associated RRSIG RRset) MUST NOT be the only RRset at any particular owner name. That is, the signing process MUST NOT create NSEC or RRSIG RRs for owner name nodes that were not the owner name of any RRset before the zone was signed. The main reasons for this are a desire for namespace consistency between signed and unsigned versions of the same zone and a desire to reduce the risk of response inconsistency in security oblivious recursive name servers.

This paragraph is updated to the following::

- * An NSEC record (and its associated RRSIG RRset) MUST NOT be the only RRset at any particular owner name. That is, the signing process MUST NOT create NSEC or RRSIG RRs for owner name nodes that were not the owner name of any RRset before the zone was signed. The main reasons for this are a desire for namespace consistency between signed and unsigned versions of the same zone and a desire to reduce the risk of response inconsistency in security oblivious recursive name servers. This concern only applies to implementations of DNSSEC that employ pre-computed signatures. There is an exception to this rule for online signing implementations of DNSSEC (e.g., Minimally Covering NSEC, and Compact Denial of Existence), where dynamically generated NSEC records can be produced for owner names that don't exist or are empty non-terminals.

8. Security Considerations

Online signing of DNS records requires authoritative servers for the DNS zone to have access to the private signing keys. Exposing signing keys on Internet reachable servers makes them more vulnerable to attack.

Additionally, generating signatures on-demand is more computationally intensive than returning pre-computed signatures. Although the Compact Answers scheme reduces the number of online signing operations compared to previous techniques like White Lies, it still may make authoritative servers more vulnerable to computational denial of service attacks than pre-computed signatures. The use of signature algorithms (like those based on Elliptic Curves) that have a comparatively low cost for signing is recommended.

Some security tools rely on detection of non-existent domain names by examining the response code field of DNS response messages. A NOERROR code in that field rather than NXDOMAIN will impact such tools. Implementation of the optional response code restoration scheme will help recover NXDOMAIN visibility for these tools. Note that the DNS header is not cryptographically protected, so the response code field cannot be authenticated. Thus inferring the status of a response from signed data in the body of the DNS message is actually more secure. These tools could be enhanced to recognize the (signed) NXNAME signal.

Because this method does not allow for aggressive negative caching among resolvers, it allows for certain types of denial-of-service attacks to be more effective. This includes so-called Pseudorandom Subdomain Attacks [PRSD-ATTACK], where random names are queried, either directly via botnets or across a wide range of public resolver services, in order to intentionally generate non-existence responses from the authoritative servers for a domain. If the resolver cannot synthesize NXDOMAIN responses from NSEC records, it must pass all queries on to the domain's authority servers, making resource exhaustion more likely at those latter servers, if those servers do not have the capacity to absorb those additional queries.

If the motivating aspects of this specification (minimizing response size and computational costs) are not a concern, DNSSEC deployments can opt for a different method (e.g., traditional online signing or pre-computed signatures), and avoid imposing the challenges of NXDOMAIN visibility.

9. Acknowledgements

The Compact Answers technique (then called "Black Lies") was originally proposed in [BLACK-LIES] by Filippo Valsorda and Olafur Gudmundsson, and implemented by Cloudflare. The Empty Non-Terminal distinguisher RR type was originally proposed in [ENT-SENTINEL] by Shumon Huque, and deployed by NS1. The NXNAME type is based on the FDOM type proposed in [NXDOMAIN-TYPE] by Gudmundsson and Valsorda.

Especially detailed discussions on many technical aspects of this document were conducted with Paul Vixie, Jan Vcelak, Viktor Dukhovni, Ed Lewis, and John Levine. The authors would also like to thank the many other members of the IETF DNS Operations working group for helpful comments and discussions.

10. IANA Considerations

IANA is requested to do the following:

Allocate a new DNS Resource Record type code for NXNAME from the "Resource Record (RR) Types" registry in the "DNS parameters" registry group, from the meta type range. Specifically, the lowest available number (currently 128) in the meta types range is requested to be allocated. A lower number lowers the size of the Type Bit Maps field, which reduces the overall size of the DNS response message.

Type	Value	Meaning
NXNAME	128	NXDOMAIN indicator for Compact Denial of Existence

Allocate the "Compact Answers OK" flag in the "EDNS Header Flags (16 bits)" registry in the "Domain Name System (DNS) Parameters" registry group. Set the Flag field to "CO", as described in Section 5.1.

Allocate a code point for "Invalid Query Type" in the "Extended DNS Error Codes" registry in the "Domain Name System (DNS) Parameters" registry group, as described in Section 3.5.

INFO-CODE	Purpose
30	Invalid Query Type

11. References

11.1. 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/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>>.
- [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>>.
- [RFC4470] Weiler, S. and J. Ihren, "Minimally Covering NSEC Records and DNSSEC On-line Signing", RFC 4470, DOI 10.17487/RFC4470, April 2006, <<https://www.rfc-editor.org/info/rfc4470>>.
- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <<https://www.rfc-editor.org/info/rfc4648>>.
- [RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, DOI 10.17487/RFC5155, March 2008, <<https://www.rfc-editor.org/info/rfc5155>>.
- [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>>.
- [RFC6895] Eastlake 3rd, D., "Domain Name System (DNS) IANA Considerations", BCP 42, RFC 6895, DOI 10.17487/RFC6895, April 2013, <<https://www.rfc-editor.org/info/rfc6895>>.
- [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>>.
- [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>>.
- [RFC9364] Hoffman, P., "DNS Security Extensions (DNSSEC)", BCP 237, RFC 9364, DOI 10.17487/RFC9364, February 2023, <<https://www.rfc-editor.org/info/rfc9364>>.

11.2. Informative References

- [BLACK-LIES] Valsorda, F. and O. Gudmundsson, "Compact DNSSEC Denial of Existence or Black Lies", <<https://tools.ietf.org/html/draft-valsorda-dnsop-black-lies>>.

[ENT-SENTINEL]

Huque, S., "Empty Non-Terminal Sentinel for Black Lies",
<<https://www.ietf.org/archive/id/draft-huque-dnsop-blacklies-ent-01.html>>.

[NXDOMAIN-TYPE]

Gudmundsson, O. and F. Valsorda, "Signaling NSEC record owner name nonexistence", <<https://tools.ietf.org/html/draft-ogud-fake-nxdomain-type/>>.

[PRSD-ATTACK]

Nishida, K., "Water Torture: A Slow Drip DNS DDoS Attack on QTNet", <https://conference.apnic.net/data/39/dnswatertortureonqtnet_1425130417_1425507043.pptx>.

[RFC7129] Gieben, R. and W. Mekking, "Authenticated Denial of Existence in the DNS", RFC 7129, DOI 10.17487/RFC7129, February 2014, <<https://www.rfc-editor.org/info/rfc7129>>.

[RFC8020] Bortzmeyer, S. and S. Huque, "NXDOMAIN: There Really Is Nothing Underneath", RFC 8020, DOI 10.17487/RFC8020, November 2016, <<https://www.rfc-editor.org/info/rfc8020>>.

[RFC8198] Fujiwara, K., Kato, A., and W. Kumari, "Aggressive Use of DNSSEC-Validated Cache", RFC 8198, DOI 10.17487/RFC8198, July 2017, <<https://www.rfc-editor.org/info/rfc8198>>.

[RFC8305] Schinazi, D. and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency", RFC 8305, DOI 10.17487/RFC8305, December 2017, <<https://www.rfc-editor.org/info/rfc8305>>.

[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>>.

Appendix A. Other Approaches

In the past, some implementations of Compact Denial of Existence have used other means to differentiate NXDOMAIN from Empty Non-Terminals.

One method employed by both Cloudflare and Amazon Route53 for a period of time was the following: for responses to Empty Non-Terminals, they synthesized the NSEC Type Bit Maps to include all record types supported except for the queried type. This method has the undesirable effect of no longer being able to reliably determine the existence of ENTs, and of making the Type Bit Maps field larger than it needs to be. It also has the potential to confuse validators and others tools that infer type existence from the NSEC record.

Another way to distinguish NXDOMAIN from ENT is to define the synthetic Resource Record type for ENTs instead, as specified in [ENT-SENTINEL]. This method was successfully deployed in the field by NS1 for a period of time. This typically imposes less work on the server since NXDOMAIN responses are a lot more common than ENTs. At the time it was deployed, it also allowed a common bitmap pattern ("NSEC RRSIG") to identify NXDOMAIN across this and other implementations that returned a broad bitmap pattern for Empty Non-Terminals. However, the advantage of the NXNAME RR type is that it explicitly identifies NXDOMAIN responses, and allows them to be distinguished conclusively from potential ENT responses in other online signing NSEC implementations.

Appendix B. Historical Implementation Notes

At the time of publication, the basic Compact Denial of Existence method using NSEC is implemented by Cloudflare, NS1, Amazon Route53, and Knot DNS's optional online signing module. From early 2021 until November 2023, NS1 had deployed the Empty Non-Terminal distinguisher [ENT-SENTINEL] using the private RR type code 65281. A version of the NXNAME distinguisher using the private RR type code 65238 was deployed by both Cloudflare (from July 2023) and NS1 (from November 2023) until roughly September 2024. Since September 2024 both Cloudflare and NS1 have deployed NXNAME using the officially allocated code point of 128. Oracle Cloud Initiative implemented Compact Denial of Existence using NSEC3 in October 2024.

Authors' Addresses

Shumon Huque
Salesforce
415 Mission Street, 3rd Floor
San Francisco, CA 94105
United States of America
Email: shuque@gmail.com

Christian Elmerot
Cloudflare
101 Townsend St.
San Francisco, CA 94107
United States of America
Email: elmerot@cloudflare.com

Olafur Gudmundsson
Retired / Unaffiliated
Email: ogud@ogud.com