Stream:	Internet Engineering Task Force (IETF)
RFC:	8914
Category:	Standards Track
Published:	October 2020
ISSN:	2070-1721
Authors:	W. Kumari	E. Hunt	R. Arends	W. Hardaker	D. Lawrence
---	---	---	---	---	---
	Google	*ISC*	*ICANN*	*USC/ISI*	*Salesforce*

RFC 8914
Extended DNS Errors

Abstract

This document defines an extensible method to return additional information about the cause of DNS errors. Though created primarily to extend SERVFAIL to provide additional information about the cause of DNS and DNSSEC failures, the Extended DNS Errors option defined in this document allows all response types to contain extended error information. Extended DNS Error information does not change the processing of RCODEs.

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/rfc8914.

Table of Contents

1. Introduction and Background

There are many reasons that a DNS query may fail -- some of them transient, some permanent; some can be resolved by querying another server, some are likely best handled by stopping resolution. Unfortunately, the error signals that a DNS server can return are very limited and are not very expressive. This means that applications and resolvers often have to "guess" at what the issue is, e.g., was the answer marked REFUSED because of a lame delegation or because the nameserver is still starting up and loading zones? Is a SERVFAIL a DNSSEC validation issue, or is the nameserver experiencing some other failure? What error messages should be presented to the user or logged under these conditions?

A good example of issues that would benefit from additional error information are errors caused by DNSSEC validation issues. When a stub resolver queries a name that is DNSSEC bogus [RFC8499] (using a validating resolver), the stub resolver receives only a SERVFAIL in response. Unfortunately, the SERVFAIL Response Code (RCODE) is used to signal many sorts of DNS errors, and so the stub resolver's only option is to ask the next configured DNS resolver. The result of trying the next resolver is one of two outcomes: either the next resolver also validates and a SERVFAIL is returned again or the next resolver is not a validating resolver and the user is returned a potentially harmful result. With an Extended DNS Error (EDE) option enclosed in the response message, the resolver is able to return a more descriptive reason as to why any failures happened or add additional context to a message containing a NOERROR RCODE.

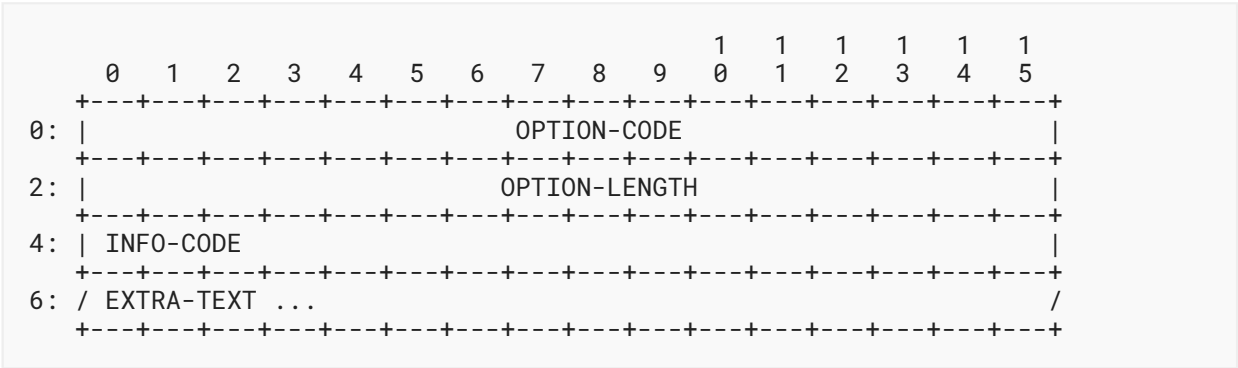
This document specifies a mechanism to extend DNS errors to provide additional information about the cause of an error. The Extended DNS Error codes described in this document can be used by any system that sends DNS queries and receives a response containing an EDE option. Different codes are useful in different circumstances, and thus different systems (stub resolvers, recursive resolvers, and authoritative resolvers) might receive and use them.

1.1. Requirements Notation

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. Extended DNS Error EDNS0 Option Format

This document uses an Extended Mechanism for DNS (EDNS0) [RFC6891] option to include Extended DNS Error (EDE) information in DNS messages. The option is structured as follows:



Field definition details:

OPTION-CODE:
: 2 octets / 16 bits (defined in [RFC6891]) contains the value 15 for EDE.

OPTION-LENGTH:
: 2 octets / 16 bits (defined in [RFC6891]) contains the length of the payload (everything after OPTION-LENGTH) in octets and should be 2 plus the length of the EXTRA-TEXT field (which may be a zero-length string).

INFO-CODE:
: 16 bits, which is the principal contribution of this document. This 16-bit value, encoded in network most significant bit (MSB) byte order, provides the additional context for the RESPONSE-CODE of the DNS message. The INFO-CODE serves as an index into the "Extended DNS Errors" registry, defined and created in Section 5.2.

EXTRA-TEXT:
: a variable-length, UTF-8-encoded [RFC5198] text field that may hold additional textual information. This information is intended for human consumption (not automated parsing). EDE text may be null terminated but **MUST NOT** be assumed to be; the length **MUST** be derived from the OPTION-LENGTH field. The EXTRA-TEXT field may be zero octets in length, indicating that there is no EXTRA-TEXT included. Care should be taken not to include private information in the EXTRA-TEXT field that an observer would not otherwise have access to, such as account numbers.

The Extended DNS Error (EDE) option can be included in any response (SERVFAIL, NXDOMAIN, REFUSED, even NOERROR, etc.) to a query that includes an OPT pseudo-RR [RFC6891]. This document includes a set of initial codepoints but is extensible via the IANA registry defined and created in Section 5.2.

3. Extended DNS Error Processing

When the response grows beyond the requestor's UDP payload size [RFC6891], servers **SHOULD** truncate messages by dropping EDE options before dropping other data from packets. Implementations **SHOULD** set the truncation bit when dropping EDE options. Because long EXTRA-TEXT fields may trigger truncation (which is undesirable given the supplemental nature of EDE), implementers and operators creating EDE options **SHOULD** avoid lengthy EXTRA-TEXT contents.

When a resolver or forwarder receives an EDE option, whether or not (and how) to pass along EDE information on to their original client is implementation dependent. Implementations **MAY** choose to not forward information, or they **MAY** choose to create a new EDE option(s) that conveys the information encoded in the received EDE. When doing so, the source of the error **SHOULD** be attributed in the EXTRA-TEXT field, since an EDNS0 option received by the original client will appear to have come from the resolver or forwarder sending it.

This document does not allow or prohibit any particular extended error codes and information to be matched with any particular RCODEs. Some combinations of extended error codes and RCODEs may seem nonsensical (such as resolver-specific extended error codes received in responses from authoritative servers), so systems interpreting the extended error codes **MUST NOT** assume that a combination will make sense. Receivers **MUST** be able to accept EDE codes and EXTRA-TEXT in all messages, including those with a NOERROR RCODE but need not act on them. Applications **MUST** continue to follow requirements from applicable specifications on how to process RCODEs no matter what EDE values are also received. Senders **MAY** include more than one EDE option and receivers **MUST** be able to accept (but not necessarily process or act on) multiple EDE options in a DNS message.

4. Defined Extended DNS Errors

This document defines some initial EDE codes. The mechanism is intended to be extensible, and additional codepoints can be registered in the "Extended DNS Errors" registry (Section 5.2). The INFO-CODE from the EDE EDNS option is used to serve as an index into the "Extended DNS Error" IANA registry, the initial values for which are defined in the following subsections.

4.1. Extended DNS Error Code 0 - Other

The error in question falls into a category that does not match known extended error codes. Implementations **SHOULD** include an EXTRA-TEXT value to augment this error code with additional information.

4.2. Extended DNS Error Code 1 - Unsupported DNSKEY Algorithm

The resolver attempted to perform DNSSEC validation, but a DNSKEY RRset contained only unsupported DNSSEC algorithms.

4.3. Extended DNS Error Code 2 - Unsupported DS Digest Type

The resolver attempted to perform DNSSEC validation, but a DS RRset contained only unsupported Digest Types.

4.4. Extended DNS Error Code 3 - Stale Answer

The resolver was unable to resolve the answer within its time limits and decided to answer with previously cached data instead of answering with an error. This is typically caused by problems communicating with an authoritative server, possibly as result of a denial of service (DoS) attack against another network. (See also Code 19.)

4.5. Extended DNS Error Code 4 - Forged Answer

For policy reasons (legal obligation or malware filtering, for instance), an answer was forged. Note that this should be used when an answer is still provided, not when failure codes are returned instead. See Blocked (15), Censored (16), and Filtered (17) for use when returning other response codes.

4.6. Extended DNS Error Code 5 - DNSSEC Indeterminate

The resolver attempted to perform DNSSEC validation, but validation ended in the Indeterminate state [RFC4035].

4.7. Extended DNS Error Code 6 - DNSSEC Bogus

The resolver attempted to perform DNSSEC validation, but validation ended in the Bogus state.

4.8. Extended DNS Error Code 7 - Signature Expired

The resolver attempted to perform DNSSEC validation, but no signatures are presently valid and some (often all) are expired.

4.9. Extended DNS Error Code 8 - Signature Not Yet Valid

The resolver attempted to perform DNSSEC validation, but no signatures are presently valid and at least some are not yet valid.

4.10. Extended DNS Error Code 9 - DNSKEY Missing

A DS record existed at a parent, but no supported matching DNSKEY record could be found for the child.

4.11. Extended DNS Error Code 10 - RRSIGs Missing

The resolver attempted to perform DNSSEC validation, but no RRSIGs could be found for at least one RRset where RRSIGs were expected.

4.12. Extended DNS Error Code 11 - No Zone Key Bit Set

The resolver attempted to perform DNSSEC validation, but no Zone Key Bit was set in a DNSKEY.

4.13. Extended DNS Error Code 12 - NSEC Missing

The resolver attempted to perform DNSSEC validation, but the requested data was missing and a covering NSEC or NSEC3 was not provided.

4.14. Extended DNS Error Code 13 - Cached Error

The resolver is returning the SERVFAIL RCODE from its cache.

4.15. Extended DNS Error Code 14 - Not Ready

The server is unable to answer the query, as it was not fully functional when the query was received.

4.16. Extended DNS Error Code 15 - Blocked

The server is unable to respond to the request because the domain is on a blocklist due to an internal security policy imposed by the operator of the server resolving or forwarding the query.

4.17. Extended DNS Error Code 16 - Censored

The server is unable to respond to the request because the domain is on a blocklist due to an external requirement imposed by an entity other than the operator of the server resolving or forwarding the query. Note that how the imposed policy is applied is irrelevant (in-band DNS filtering, court order, etc.).

4.18. Extended DNS Error Code 17 - Filtered

The server is unable to respond to the request because the domain is on a blocklist as requested by the client. Functionally, this amounts to "you requested that we filter domains like this one."

4.19. Extended DNS Error Code 18 - Prohibited

An authoritative server or recursive resolver that receives a query from an "unauthorized" client can annotate its REFUSED message with this code. Examples of "unauthorized" clients are recursive queries from IP addresses outside the network, blocklisted IP addresses, local policy, etc.

4.20. Extended DNS Error Code 19 - Stale NXDOMAIN Answer

The resolver was unable to resolve an answer within its configured time limits and decided to answer with a previously cached NXDOMAIN answer instead of answering with an error. This may be caused, for example, by problems communicating with an authoritative server, possibly as result of a denial of service (DoS) attack against another network. (See also Code 3.)

4.21. Extended DNS Error Code 20 - Not Authoritative

An authoritative server that receives a query with the Recursion Desired (RD) bit clear, or when it is not configured for recursion for a domain for which it is not authoritative, **SHOULD** include this EDE code in the REFUSED response. A resolver that receives a query with the RD bit clear **SHOULD** include this EDE code in the REFUSED response.

4.22. Extended DNS Error Code 21 - Not Supported

The requested operation or query is not supported.

4.23. Extended DNS Error Code 22 - No Reachable Authority

The resolver could not reach any of the authoritative name servers (or they potentially refused to reply).

4.24. Extended DNS Error Code 23 - Network Error

An unrecoverable error occurred while communicating with another server.

4.25. Extended DNS Error Code 24 - Invalid Data

The authoritative server cannot answer with data for a zone it is otherwise configured to support. Examples of this include its most recent zone being too old or having expired.

5. IANA Considerations

5.1. A New Extended DNS Error Code EDNS Option

This document defines a new EDNS(0) option, entitled "Extended DNS Error", with the assigned value of 15 from the "DNS EDNS0 Option Codes (OPT)" registry:

Value	Name	Status	Reference
15	Extended DNS Error	Standard	RFC 8914

Table 1

5.2. New Registry for Extended DNS Error Codes

IANA has created and will maintain a new registry called "Extended DNS Error Codes" on the "Domain Name System (DNS) Parameters" web page as follows:

Range	Registration Procedures
0 - 49151	First Come First Served
49152 - 65535	Private Use

Table 2

The "Extended DNS Error Codes" registry is a table with three columns: INFO-CODE, Purpose, and Reference. The initial content is as below.

INFO-CODE	Purpose	Reference
0	Other Error	Section 4.1
1	Unsupported DNSKEY Algorithm	Section 4.2
2	Unsupported DS Digest Type	Section 4.3
3	Stale Answer	Section 4.4 and [RFC8767]
4	Forged Answer	Section 4.5
5	DNSSEC Indeterminate	Section 4.6

INFO-CODE	Purpose	Reference
6	DNSSEC Bogus	Section 4.7
7	Signature Expired	Section 4.8
8	Signature Not Yet Valid	Section 4.9
9	DNSKEY Missing	Section 4.10
10	RRSIGs Missing	Section 4.11
11	No Zone Key Bit Set	Section 4.12
12	NSEC Missing	Section 4.13
13	Cached Error	Section 4.14
14	Not Ready	Section 4.15
15	Blocked	Section 4.16
16	Censored	Section 4.17
17	Filtered	Section 4.18
18	Prohibited	Section 4.19
19	Stale NXDomain Answer	Section 4.20
20	Not Authoritative	Section 4.21
21	Not Supported	Section 4.22
22	No Reachable Authority	Section 4.23
23	Network Error	Section 4.24
24	Invalid Data	Section 4.25
25-49151	Unassigned	
49152-65535	Reserved for Private Use	Section 5.2

Table 3

6. Security Considerations

Though DNSSEC continues to be deployed, unfortunately a significant number of clients (~11% according to [GeoffValidation]) that receive a SERVFAIL from a validating resolver because of a DNSSEC validation issue will simply ask the next (potentially non-validating) resolver in their list and thus don't get the protections that DNSSEC should provide.

EDE information is unauthenticated information, unless secured by a form of secured DNS transaction, such as [RFC2845], [RFC2931], [RFC8094], or [RFC8484]. An attacker (e.g., a man in the middle (MITM) or malicious recursive server) could insert an extended error response into untrusted data -- although, ideally, clients and resolvers would not trust any unauthenticated information. As such, EDE content should be treated only as diagnostic information and **MUST NOT** alter DNS protocol processing. Until all DNS answers are authenticated via DNSSEC or the other mechanisms mentioned above, there are some trade-offs. As an example, an attacker who is able to insert the DNSSEC Bogus Extended Error into a DNS message could instead simply reply with a fictitious address (A or AAAA) record. Note that DNS RCODEs also contain no authentication and can be just as easily manipulated.

By design, EDE potentially exposes additional information via DNS resolution processes that may leak information. An example of this is the Prohibited EDE code (18), which may leak the fact that the name is on a blocklist.

7. References

7.1. 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>.

[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>.

[RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, DOI 10.17487/RFC5198, March 2008, <https://www.rfc-editor.org/info/rfc5198>.

[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>.

[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>.

[RFC8499] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <https://www.rfc-editor.org/info/rfc8499>.

[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>.

7.2. Informative References

[GeoffValidation] Huston, G., "A quick review of DNSSEC Validation in today's Internet", June 2016, <http://www.potaroo.net/presentations/2016-06-27-dnssec.pdf>.

[RFC2845] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, DOI 10.17487/RFC2845, May 2000, <https://www.rfc-editor.org/info/rfc2845>.

[RFC2931] Eastlake 3rd, D., "DNS Request and Transaction Signatures (SIG(0)s)", RFC 2931, DOI 10.17487/RFC2931, September 2000, <https://www.rfc-editor.org/info/rfc2931>.

[RFC8094] Reddy, T., Wing, D., and P. Patil, "DNS over Datagram Transport Layer Security (DTLS)", RFC 8094, DOI 10.17487/RFC8094, February 2017, <https://www.rfc-editor.org/info/rfc8094>.

[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>.

Acknowledgements

The authors wish to thank Joe Abley, Mark Andrews, Tim April, Vittorio Bertola, Stephane Bortzmeyer, Vladimir Cunat, Ralph Dolmans, Peter DeVries, Peter van Dijk, Mats Dufberg, Donald Eastlake, Bob Harold, Paul Hoffman, Geoff Huston, Shane Kerr, Edward Lewis, Carlos M. Martinez, George Michelson, Eric Orth, Michael Sheldon, Puneet Sood, Petr Spacek, Ondrej Sury, John Todd, Loganaden Velvindron, and Paul Vixie. They also vaguely remember discussing this with a number of people over the years but have forgotten who all of them were. Apologies if we forgot to acknowledge your contributions.

One author also wants to thank the band Infected Mushroom for providing a good background soundtrack. Another author would like to thank the band Mushroom Infectors. This was funny at the time we wrote it, but we cannot remember why...

Authors' Addresses

Warren Kumari
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
United States of America
Email: warren@kumari.net

Evan Hunt
ISC
950 Charter St
Redwood City, CA 94063
United States of America
Email: each@isc.org

Roy Arends
ICANN
Email: roy.arends@icann.org

Wes Hardaker
USC/ISI
P.O. Box 382
Davis, CA 95617
United States of America
Email: ietf@hardakers.net

David C Lawrence
Salesforce
415 Mission St
San Francisco, CA 94105
United States of America
Email: tale@dd.org