DNS Operations Working Group
Internet-Draft
Updates: 8914 (if approved)
Intended status: Standards Track
Expires: 30 August 2026

D. Wing
Citrix
T. Reddy
Nokia
N. Cook
Open-Xchange
M. Boucadair
Orange

26 February 2026

# Structured Error Data for Filtered DNS
# draft-ietf-dnsop-structured-dns-error-17

## Abstract

DNS filtering is widely deployed for various reasons, including network security. However, filtered DNS responses lack structured information for end users to understand the reason for the filtering. Existing mechanisms to provide explanatory details to end users cause harm especially if the blocked DNS response is for HTTPS resources.

This document updates RFC 8914 by signaling client support for structuring the EXTRA-TEXT field of the Extended DNS Error to provide details on the DNS filtering. Such details can be parsed by the client and displayed, logged, or used for other purposes.

## 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://ietf-wg-dnsop.github.io/draft-ietf-dnsop-structured-dns-error/draft-ietf-dnsop-structured-dns-error.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-dnsop-structured-dns-error/.

Discussion of this document takes place on the dnsop Working Group mailing list (mailto:dnsop@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/dnsop/. Subscribe at https://www.ietf.org/mailman/listinfo/dnsop/.

Source for this draft and an issue tracker can be found at https://github.com/ietf-wg-dnsop/draft-ietf-dnsop-structured-dns-error.

## Status of This Memo



## Copyright Notice



## Table of Contents

## 1. Introduction

DNS filters are deployed for a variety of reasons, e.g., endpoint security, parental filtering, and filtering required by law enforcement. Network-based security solutions such as firewalls and Intrusion Prevention Systems (IPS) rely upon network traffic inspection to implement perimeter-based security policies and operate by filtering DNS responses. In a home network, DNS filtering is used for the same reasons as above and additionally for parental control. Internet Service Providers (ISPs) typically block access to some DNS domains due to a requirement imposed by an external entity (e.g., law enforcement agency) also performed using DNS-based content filtering.

End-users or network administrators leveraging DNS services that perform filtering may wish to receive more explanatory information about such a filtering to resolve problems with the filter -- for example to contact the DNS service administrator to allowlist a DNS domain that was erroneously filtered or to understand the reason a particular domain was filtered. With that information, they can choose to use another network, open a trouble ticket with the DNS service administrator to resolve erroneous filtering, log the information, etc.

For the DNS filtering mechanisms described in Section 3, the DNS server can return extended error codes Blocked, Filtered, Censored, or Forged Answer defined in Section 4 of [RFC8914]. However, these codes only explain that filtering occurred but lack detail for the user to diagnose erroneous filtering.

No matter which type of response is generated (forged IP address(es), NXDOMAIN or empty answer, even with an extended error code), the user who triggered the DNS query has little chance to understand which entity filtered the query, how to report a mistake in the filter, or why the entity filtered it at all. This document describes a mechanism to provide such detail.

One of the other benefits of the approach described in this document is to eliminate the need to "spoof" block pages for HTTPS resources. This is achieved since clients implementing this approach would be able to display a meaningful error message, and would not need to connect to such a block page. This approach thus avoids the need to install a local root certificate authority on those IT-managed devices.

This document describes a format for machine-readable data in the EXTRA-TEXT field of [RFC8914]. It updates Section 2 of [RFC8914] which says the information in EXTRA-TEXT field is intended for human consumption (not automated parsing).

This document does not recommend DNS filtering but provides a mechanism for better transparency to explain to the users why some DNS queries are filtered.

## 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.

This document uses terms defined in DNS Terminology [RFC9499].

"Requestor" refers to the side that sends a request. "Responder" refers to an authoritative, recursive resolver, or other DNS component that responds to questions.

"Encrypted DNS" refers to any encrypted scheme to convey DNS messages, for example, DNS-over-HTTPS [RFC8484], DNS-over-TLS [RFC7858], or DNS-over-QUIC [RFC9250].

The document refers to an Extended DNS Error (EDE) using its purpose, not its INFO-CODE as per Table 3 of [RFC8914]. "Forged Answer", "Blocked", "Censored", and "Filtered" are thus used to refer to "Forged Answer (4)", "Blocked (15)", "Censored (16)",and "Filtered (17)".

The term "DNS server" refers to a DNS recursive resolver or a DNS forwarder that generates DNS structured error responses.

In this document, "client security policy evaluation" refers to implementation-defined decision-making performed by the DNS client or consuming application to determine how, or whether, structured error information is used, displayed, or acted upon.

## 3. DNS Filtering Techniques and Their Limitations

DNS responses can be filtered by sending, e.g., a bogus (also called "forged") response, NXDOMAIN error, or empty answer. Also, clients can be informed that filtering occurred by sending an Extended DNS Error code defined in [RFC8914]. Each of these methods have advantages and disadvantages that are discussed below:

1. The DNS response is forged to provide a list of IP addresses that points to an HTTP(S) server alerting the end user about the reason for blocking access to the requested domain (e.g., malware). If the authority component of an HTTP(S) URL is blocked, the network security device (e.g., Customer Premises Equipment (CPE) or firewall) presents a block page instead of the HTTP response from the content provider hosting that domain. If the authority component of an HTTP URL is blocked, the network security device intercepts the HTTP request and returns a block page over HTTP. If the authority component of an HTTPS URL is blocked, the network security device serves the block page over HTTPS. In order to return a block page over HTTPS, the network security device uses a locally generated root certificate and corresponding key pair. The local root certificate is installed on the endpoint while the network security device stores a copy of the private key. During the TLS handshake, the on-path network security device modifies the certificate provided by the server and (re)signs it using the private key from the local root certificate.

    * However, in deployments where DNSSEC is used, this approach becomes ineffective because DNSSEC ensures the integrity and authenticity of DNS responses, preventing forged DNS responses from being accepted.

* The HTTPS server hosted on the network security device will have access to the client's IP address and the hostname being requested. This information will be sensitive, as it will expose the user's identity and the domain name that a user attempted to access.

* Configuring a local root certificate on endpoints is not a viable option in several deployments like home networks, schools, Small Office/Home Office (SOHO), or Small/Medium Enterprise (SME). In these cases, the typical behavior is that the filtered DNS response points to a server that will display the block page. If the client is using HTTPS (via a web browser or another application) this results in a certificate validation error which gives no information to the end-user about the reason for the DNS filtering.

* Enterprise networks do not always assume that all the connected devices are managed by the IT team or Mobile Device Management (MDM) devices, especially in the quite common Bring Your Own Device (BYOD) scenario. In addition, the local root certificate cannot be installed on IoT devices without a device management tool.

* An end user does not know why the connection was prevented and, consequently, may repeatedly try to reach the domain but with no success. Frustrated, the end user may switch to an alternate network that offers no DNS filtering against malware and phishing, potentially compromising both security and privacy. Furthermore, certificate errors train users to click through certificate errors, which is a bad security practice. To eliminate the need for an end user to click through certificate errors, an end user may manually install a local root certificate on a host device. Doing so, however, is also a bad security practice as it creates a security vulnerability that may be exploited by a MITM attack. When a manually installed local root certificate expires, the user has to (again) manually install the new local root certificate.

2. The DNS response is forged to provide an NXDOMAIN answer, causing the DNS lookup to fail. This approach is incompatible with DNSSEC when the client performs validation, as the forged response will fail DNSSEC checks. However, in deployments where the client relies on the DNS server to perform DNSSEC validation, a filtering DNS server can forge an NXDOMAIN response for a valid domain, and the client will trust it. This undermines the integrity guarantees of DNSSEC, as the client has no way to distinguish between a genuine and a forged response. Further, the end user may not understand why a domain cannot be reached

and may repeatedly attempt access without success. Frustrated, the user may resort to using insecure methods to reach the domain, potentially compromising both security and privacy.

3. The extended error codes Blocked and Filtered defined in Section 4 of [RFC8914] can be returned by a DNS server to provide additional information about the cause of a DNS error. These extended error codes do not suffer from the limitations discussed in bullets (1) and (2), but the user still does not know the exact reason nor is aware of the exact entity blocking the access to the domain. For example, a DNS server may block access to a domain based on the content category such as "Malware" to protect the endpoint from malicious software, "Phishing" to prevent the user from revealing sensitive information to the attacker, etc. A user may need to know the contact details of the IT/InfoSec team to raise a complaint.

## 4. I-JSON in EXTRA-TEXT Field

DNS servers that are compliant with this specification and have received an indication that the client also supports this specification as per Section 5.1 send data in the EXTRA-TEXT field [RFC8914] encoded using the Internet JSON (I-JSON) message format [RFC7493].

> Note that [RFC7493] was based on [RFC7159], but [RFC7159] was replaced by [RFC8259].

This document defines the following JSON names:

c: (contact) The contact details of the IT/InfoSec team to report misclassified DNS filtering. This information is important for transparency and also to ease unblocking a legitimate domain name that got blocked due to wrong classification.

The field is a JSON array of contact URIs. When multiple contact details are provided, each contact URI is represented as a separate array element in the JSON array.

Contact URIs conveyed in the "c" field MUST use URI schemes registered in Section 11.3.

This field is optional.

j: (justification) 'UTF-8'-encoded [RFC5198] human-readable explanation for the DNS filtering decision.

This field is particularly useful when no applicable sub-error

code is defined or provided for the returned Extended DNS Error.

The information conveyed in this field MUST NOT be used as input to automated processing that affects security policy enforcement or DNS protocol behavior.

The DNS client determines, according to its client security policy, whether the contents of this field are displayed to the end user, logged, or ignored.

Returning non-UTF-8 data, syntactically invalid content, or deliberately meaningless values (including empty strings) indicates that a DNS server is misbehaving.

This field is optional.

s: (sub-error) An integer representing the sub-error code for this particular DNS filtering case.

The integer values are defined in the IANA-managed registry for DNS Sub-Error Codes in Section 11.4.

This field is optional.

o: (organization) 'UTF-8'-encoded human-friendly name of the organization that filtered this particular DNS query.

This field is optional.

l: (language) The "l" field indicates the language used for the JSON-encoded "j" and "o" fields. The value of this field MUST conform to the language tag syntax specified in Section 2.1 of [RFC5646].

This field is optional but RECOMMENDED to aid in localization.

New JSON names can be defined in the IANA registry introduced in Section 11.2. Such names MUST consist only of lower-case ASCII characters, digits, and hyphen-minus (that is, Unicode characters U+0061 through 007A, U+0030 through U+0039, and U+002D). Also, these names MUST be 63 characters or shorter and it is RECOMMENDED they be as short as possible.

The text in the "j" and "o" names can include international characters. The text will be in natural language, chosen by the DNS administrator to match its expected audience.

If the client supports diagnostic interfaces, it MAY use the "l" field to identify the language of the "j" text and optionally translate it for IT administrators.

The "o" field MAY be displayed to end users, subject to the conditions described in Section 10. If the text is in a language not understood by the end-user, the "l" field can be used to identify the language and support translation into the end-user’s preferred language.

To reduce DNS message size the generated JSON SHOULD be as short as possible: short domain names, concise text in the values for the "j" and "o" names, and minified JSON (that is, without spaces or line breaks between JSON elements).

The JSON data can be parsed to display to the user, logged, or otherwise used to assist troubleshooting and diagnosis of DNS filtering.

The sub-error codes provide a structured way to communicate more detailed and precise description of the cause of an error (e.g., distinguishing between malware-related blocking and phishing-related blocking under the general blocked error).

> An alternate design for conveying the sub-error would be to define new EDE codes for these errors. However, such design is suboptimal because it requires replicating an error code for each EDE code to which the sub-error applies (e.g., "Malware" sub-error in Table 3 would consume three EDE codes).

## 5. Protocol Operation

### 5.1. Client Generating Request

When generating a DNS query, a client that supports this specification MUST include the Structured DNS Error (SDE) option defined in Section 5.4.

The presence of the SDE option indicates that the client desires the DNS server to include an EDE option in the DNS response when DNS filtering is performed, and that any data conveyed in the EXTRA-TEXT field of the EDE option is encoded and processed in accordance with this specification.

### 5.2. Server Generating Response

When the DNS server filters its DNS response to a query (e.g., A or AAAA resource record query), the DNS response MAY contain an empty answer, NXDOMAIN, or (less ideally) forged response, as desired by the DNS server.

If the query contained the SDE EDNS option (Section 5.1), and the DNS server returns an EDE indicating blocking or modification of the response, the DNS server MUST include additional detail in the EXTRA-TEXT field encoded as structured and machine-readable data.

If the SDE option is not present, the DNS server MUST NOT include structured JSON data and MUST convey the EXTRA-TEXT field as human-readable text in accordance with [RFC8914].

Servers MAY decide to return small TTL values in filtered DNS responses (e.g., 10 seconds) to handle domain category and reputation updates. Short TTLs allow for quick adaptation to dynamic changes in domain filtering decisions, but can result in increased query traffic. In cases where updates are less frequent, TTL values of 30 to 60 seconds MAY provide a better balance, reducing server load while still ensuring reasonable flexibility for updates.

Because the DNS client explicitly signals support for structured error information using the SDE option (Section 5.1), and because the EDE option is carried in the non-cached OPT pseudo-RR (Section 6.2.1 of [RFC6891]), the DNS server can tailor its filtered response to the capabilities of the client.

If the query includes the SDE option as per Section 5.1, the server MUST NOT return the "Forged Answer" extended error code because the client can take advantage of EDE’s more sophisticated error reporting (e.g., "Filtered", "Blocked"). Continuing to send "Forged Answer" even to an EDE-supporting client will cause the persistence of the drawbacks described in Section 3.

When the "Censored" extended error code is included in the DNS response, the "c", "j", "o", and "l" fields may be conveyed in the EXTRA-TEXT field. The sub-error codes defined in this specification are not applicable to the "Censored" extended error code and MUST NOT be used in conjunction with it. Future specifications may update this behavior by defining sub-error codes applicable to "Censored".

### 5.3. Client Processing Response

On receipt of a DNS response with an EDE option from a DNS responder, the following ordered actions are performed on the EXTRA-TEXT field:

1. If the DNS response is not received over an encrypted DNS channel, the requestor MUST NOT act upon data in the EXTRA-TEXT field, as there is no mechanism to verify the integrity of such data and it is vulnerable to modification by an on-path attacker. An attacker can inject or modify a structured DNS error response in transit without detection, enabling fabrication of filtering information (e.g., misleading contact information or false resolver identity information) that appears to originate from the resolver. The data MAY be retained for diagnostic or client security policy evaluation purposes.

2. Servers which don’t support this specification might use plain text in the EXTRA-TEXT field. Requestors SHOULD properly handle both plaintext and JSON text in the EXTRA-TEXT field. The requestor verifies that the field contains valid JSON. If not, the requestor MUST consider the server does not support this specification and stop processing the rest of the actions defined in this section, but may instead choose to treat EXTRA-TEXT as per [RFC8914].

3. The EXTRA-TEXT field MUST be an I-JSON message [RFC7493]. If the client fails to parse the field as valid JSON, it MUST treat the data as invalid and MUST NOT process it according to this specification. The client MAY process the EXTRA-TEXT field as unstructured text as specified in [RFC8914].

4. The DNS response MUST also contain an extended error code of "Blocked by Upstream DNS Server", "Blocked", "Censored" or "Filtered" [RFC8914], otherwise the EXTRA-TEXT field is discarded.

5. If the JSON object contains an "s" field and the sub-error code is not defined as applicable to the accompanying Extended DNS Error (EDE) code, the client MUST ignore the value of the "s" field and continue processing the remaining fields in accordance with this specification.

6. If the EXTRA-TEXT field does not contain at least one of the JSON names "c", "j", or "s", or if all of the fields that are present have empty values, the entire JSON object MUST be discarded.

7. If a Contact URI in the "c" field uses a scheme not registered in the Section 11.3 registry, those URIs are discarded. Contact URIs using registered schemes can be processed.

8. If the DNS client has enabled the opportunistic privacy profile for DoT (Section 5 of [RFC8310]) and the identity of the DNS server cannot be verified, the DNS client MUST ignore the "c", "j", and "o" fields, as these fields may influence user behavior and are vulnerable to active attacks in the absence of resolver authentication. If the DNS response was received over an encrypted connection, the client MAY process the "s" field and other parts of the response, as the "s" field is a registry-defined, enumerated value and does not contain free-form text.

9. In opportunistic discovery [RFC9462], where only the IP address of the DNS server is validated and the server identity is not authenticated, the DNS client MUST ignore the "c", "j", and "o" fields. If the DNS response was received over an encrypted connection, the client MAY process the "s" field and other parts of the response.

10. If a DNS client has enabled strict privacy profile (Section 5 of [RFC8310]) for DoT, the DNS client requires an encrypted connection and successful authentication of the DNS server. In doing so, this mitigates both passive eavesdropping and client redirection (at the expense of providing no DNS service if an encrypted, authenticated connection is not available). If the DNS client has enabled strict privacy profile for DoT, the DNS client MAY process the EXTRA-TEXT field of the DNS response.

11. The DNS client MUST ignore any other JSON names that it does not support.

Note that the strict and opportunistic privacy profiles as defined in [RFC8310] only apply to DoT; there has been no such distinction made for DoH.

### 5.4. Structured DNS Error (SDE) EDNS(0) Option Format

The Structured DNS Error (SDE) EDNS(0) option is used by a client to indicate support for I-JSON encoding in the EXTRA-TEXT field of an Extended DNS Error (EDE) option.

The SDE option has no OPTION-DATA. The OPTION-LENGTH field MUST be set to 0. A server receiving an SDE option with a non-zero OPTION-LENGTH MUST ignore the option.

The presence of the SDE option in a query indicates that the client supports processing the EXTRA-TEXT field in accordance with this specification.

## 6. New Sub-Error Codes Definition

The document defines the following new IANA-registered Sub-Error codes.

### 6.1. Reserved

* Number: 0
* Meaning: Reserved. This sub-error code value MUST NOT be sent. If received, it has no meaning.
* Applicability: This code should never be used.
* Reference: This-Document
* Change Controller: IETF

### 6.2. Network Operator Policy

* Number: 5
* Meaning: Network Operator Policy. The code indicates that the request was filtered according to a policy imposed by the operator of the local network (where local network is a relative term, e.g., it may refer to a Local Area Network or to the network of the ISP selected by the user).
* Applicability: Blocked
* Reference: This-Document
* Change Controller: IETF

### 6.3. DNS Operator Policy

* Number: 6
* Meaning: DNS Operator Policy. The code indicates that the request was filtered according to policy determined by the operator of the DNS server. This is different from the "Network Operator Policy" code when a third-party DNS resolver is used.
* Applicability: Blocked
* Reference: This-Document
* Change Controller: IETF

## 7. New Extended DNS Errors

This document defines an addition to the EDE codes defined in [RFC8914].

### 7.1. Extended DNS Error Code TBA1 - Blocked by Upstream DNS Server

The DNS server (e.g., a DNS forwarder) is unable to respond to the request because the domain is on a blocklist due to an internal security policy imposed by an upstream DNS server. This error code is useful in deployments where a network-provided DNS forwarder is configured to use an external resolver that filters malicious domains. When the DNS forwarder receives a Blocked (15) error code from the upstream DNS server, it can replace it with "Blocked by Upstream DNS Server" (TBA1) before forwarding the reply to the DNS client. Additionally, the EXTRA-TEXT field may be forwarded to the DNS client.

## 8. Examples

An example showing the nameserver at ’ns.example.net’ that filtered a DNS "A" record query for ’example.org’ is provided in Figure 1.

```
{
  "c": [
    "tel:+358-555-1234567",
    "sips:bob@bobphone.example.com"
  ],
  "j": "malware present for 23 days",
  "s": 1,
  "o": "example.net Filtering Service",
  "l": "en"
}
```

Figure 1: JSON Returned in EXTRA-TEXT Field of Extended DNS Error Response

In Figure 2 the same content is shown with minified JSON (no whitespace, no blank lines) with ’\’ line wrapping per [RFC8792].

```
{ "c":["tel:+358-555-1234567","sips:bob@bobphone.example.com"],\
"j":"malware present for 23 days",\
"s":1,\
"o":"example.net Filtering Service",\
"l":"en" }
```

Figure 2: Minified Response

## 9. Operational Considerations

When a forwarder receives an EDE option, whether or not (and how) to pass along JSON information in the EXTRA-TEXT field to its client is implementation-dependent [RFC5625] and depends on operator policy. Implementations MAY choose not to forward the JSON information, or they MAY choose to create a new EDE option that conveys the information in the "c", "s", and "j" fields encoded in the JSON object.

The application that triggered the DNS request may have a client security policy to override the contact information (e.g., redirect all complaint calls to a single contact point). In such cases, the content of the "c" attribute MAY be ignored.

## 10. Security Considerations

### 10.1. Authentication and Confidentiality

Security considerations in Section 6 of [RFC8914] apply to this document, except the guard against using EDE content to alter DNS protocol processing. The guard is relaxed in the current specification as it mandates DNS encryption and recommends the use of an authenticated connection to the DNS server, while [RFC8914] assumes that EDE information is unauthenticated and sent over clear text.

To minimize impact of active on-path attacks on the DNS channel, the client validates the response as described in Section 5.3.

### 10.2. Restrictions on Display of "c", "o", and "j" Fields

A client might choose to display the information in the "c" field to the end-user if and only if the encrypted resolver has sufficient reputation, according to some client security policy (e.g., user configuration, administrative configuration, or a built-in list of respectable resolvers). This limits the ability of a malicious encrypted resolver to cause harm. For example, an end user can use the details in the "c" field to contact an attacker to solve the problem of being unable to reach a domain. The attacker can mislead the end user to install malware or spyware to compromise the device security posture or mislead the end user to reveal personal data. If the client decides not to display all of the information in the EXTRA-TEXT field, it can be logged for diagnostics purpose and the client can only display the resolver hostname that blocked the domain, error description for the EDE code and the sub-error description for the "s" field to the end-user.

The same client security policy considerations apply to the display of the "j" field, as it contains free-form, human-readable text that may influence end-user behavior.

When displaying the free-form text of "o", the client MUST NOT make any of those elements into actionable (clickable) links and these fields need to be rendered as text, not as HTML. The contact details of "c" can be made into clickable links to provide a convenient way for users to initiate, e.g., voice calls. The client might choose to display the contact details only when the identity of the DNS server is verified.

Further, clients MUST NOT display the value of the "o" field to the end-user unless one of the following conditions is met:

* The value matches a registered organization name listed in the [IANA-Enterprise] OR

* The value consists solely of an organization name and does not contain any additional free-form content such as instructions, URLs, or messaging intended to influence end-user behavior, as determined by client security policy or heuristics.

If the organization name cannot be verified through registry checks or heuristics, the client MUST NOT display the "o" field to the end-user.

DNS clients MAY keep all fields conveyed in the EXTRA-TEXT field for evaluation according to the client security policy. Such data MUST NOT be automatically trusted, displayed to end users, or used to influence security decisions without appropriate validation.

### 10.3. Security Risks from Legacy DNS Forwarders

An attacker might inject (or modify) the EDE EXTRA-TEXT field with a DNS proxy or DNS forwarder that is unaware of EDE. Such a DNS proxy or DNS forwarder will forward that attacker-controlled EDE option. To prevent such an attack, clients can be configured to process EDE from explicitly configured DNS servers or utilize RESINFO [RFC9606].

## 11. IANA Considerations

This document requests five IANA actions as described in the following subsections.

> Note to the RFC Editor: Please replace RFCXXXX with the RFC number assigned to this document and "TBA1" with the value assigned by IANA.

### 11.1. Structured DNS Error EDNS Option

IANA is requested to register the following new EDNS(0) Option Code in the "DNS EDNS0 Option Codes (OPT)" registry under the "Domain Name System (DNS) Parameters" registry group [IANA-DNS]:

Value: TBD

Name: Structured DNS Error

Status: Standard

Reference: RFC XXXX

### 11.2. New Registry for JSON Names

This document requests IANA to create a new registry, entitled "EXTRA-TEXT JSON Names" under "Extended DNS Error Codes" registry, which is under the "Domain Name System (DNS) Parameters" registry group [IANA-DNS]. The registration request for a new JSON name must include the following fields:

JSON Name: Specifies the name of an attribute that is present in the JSON data enclosed in EXTRA-TEXT field. The name must follow the guidelines in Section 4.

Field Meaning: Provides a brief, human-readable label summarizing the purpose of the JSON attribute.

Short description: Includes a short description of the requested JSON name.

Mandatory (Y/N?): Indicates whether this attribute is mandatory or optional.

Specification: Provides a pointer to the reference document that specifies the attribute.

The registry is initially populated with the following values:

| JSON Name | Field Meaning | Description | Mandatory | Specification |
|---|---|---|---|---|
| c | contact | The contact details of the IT/InfoSec team to report misclassified DNS filtering | N | Section 4 of RFCXXXX |
| j | justification | UTF-8-encoded [RFC5198] textual justification for a particular DNS filtering | N | Section 4 of RFCXXXX |
| s | sub-error | Integer representing the sub-error code for this DNS filtering case | N | Section 4 of RFCXXXX |
| o | organization | UTF-8-encoded human-friendly name of the organization that filtered this particular DNS query | N | Section 4 of RFCXXXX |
| l | language | Indicates the language of the "j" and "o" fields as defined in [RFC5646] | N | Section 4 of RFCXXXX |

Table 1: Initial JSON Names Registry

New JSON names are registered via IETF Review (Section 4.8 of [RFC8126]).

The "Mandatory" column is informational only. This specification does not define any mandatory JSON names. To preserve backward compatibility, any new JSON names registered after publication of this document MUST set the “Mandatory” column to “N”. Future extensions cannot introduce mandatory JSON attributes, as existing implementations are required to ignore unknown JSON names (see Section 5.3).

### 11.3. New Registry for Contact URI Scheme

This document requests IANA to create a new registry, entitled "Contact URI Schemes" under "Extended DNS Error Codes" registry, which is under the "Domain Name System (DNS) Parameters" registry group [IANA-DNS]. The registration request for a new Contact URI scheme has to include the following fields:

* Name: URI scheme name.

* Meaning: Provides a short description of the scheme.

* Reference: Provides a pointer to an IETF-approved specification that defines the URI scheme.

The Contact URI scheme registry is initially populated with the following schemes:

| Name | Meaning | Reference |
|---|---|---|
| sips | SIP Call | [RFC5630] |
| tel | Telephone Number | [RFC3966] |
| mailto | Internet mail | [RFC6068] |

Table 2

The registration procedure for adding new Contact URI schemes to the "Contact URI Schemes" registry is "IETF Review" as defined in Section 4.8 of [RFC8126].

### 11.4. New Registry for DNS Sub-Error Codes

This document requests IANA to create a new registry, entitled "Sub-Error Codes" under "Extended DNS Error Codes" registry, which is under the "Domain Name System (DNS) Parameters" registry group [IANA-DNS]. The registration request for a new sub-error code must include the following fields:

* Number: Is the wire format sub-error code (range 0-255).

* Meaning: Provides a short description of the sub-error.

* EDE Codes Applicability: Indicates which Extended DNS Error (EDE) Codes apply to this sub-error code.

* Reference: Provides a pointer to an IETF-approved specification that registered the code and/or an authoritative specification that describes the meaning of this code.

The Sub-Error Code registry is initially populated with the following values:

| Number | Meaning | EDE Codes Applicability | Reference |
|---|---|---|---|
| 0 | Reserved | Not used | Section 6.1 of this document |
| 1 | Malware | "Blocked", "Blocked by Upstream DNS Server", "Filtered" | Section 5.5 of [RFC5901] |
| 2 | Phishing | "Blocked", "Blocked by Upstream DNS Server", "Filtered" | Section 5.5 of [RFC5901] |
| 3 | Spam | "Blocked", "Blocked by Upstream DNS Server", "Filtered" | Page 289 of [RFC4949] |
| 4 | Spyware | "Blocked", "Blocked by Upstream DNS Server", "Filtered" | Page 291 of [RFC4949] |
| 5 | Network operator policy | "Blocked" | Section 6.2 of this document |
| 6 | DNS operator policy | "Blocked" | Section 6.3 of this document |

Table 3: Initial Sub-Error Code Registry

The registration procedure to add New Sub-Error Codes is IETF Review as defined in Section 4.8 of [RFC8126].

### 11.5. New Extended DNS Error Code

IANA is requested to assign the following Extended DNS Error code from the "Extended DNS Error Codes" registry under the "Domain Name System (DNS) Parameters" registry group [IANA-DNS]:

| INFO-CODE | Purpose | Reference |
|---|---|---|
| TBA1 | Blocked by Upstream DNS Server | RFCXXXX |

Table 4: New DNS Error Code

## 12. References

### 12.1. 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>.

[RFC3966] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, DOI 10.17487/RFC3966, December 2004, <https://www.rfc-editor.org/rfc/rfc3966>.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <https://www.rfc-editor.org/rfc/rfc4949>.

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

[RFC5630] Audet, F., "The Use of the SIPS URI Scheme in the Session Initiation Protocol (SIP)", RFC 5630, DOI 10.17487/RFC5630, October 2009, <https://www.rfc-editor.org/rfc/rfc5630>.

[RFC5646] Phillips, A., Ed. and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 5646, DOI 10.17487/RFC5646, September 2009, <https://www.rfc-editor.org/rfc/rfc5646>.

[RFC5901] Cain, P. and D. Jevans, "Extensions to the IODEF-Document Class for Reporting Phishing", RFC 5901, DOI 10.17487/RFC5901, July 2010, <https://www.rfc-editor.org/rfc/rfc5901>.

[RFC6068] Duerst, M., Masinter, L., and J. Zawinski, "The 'mailto' URI Scheme", RFC 6068, DOI 10.17487/RFC6068, October 2010, <https://www.rfc-editor.org/rfc/rfc6068>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <https://www.rfc-editor.org/rfc/rfc7159>.

[RFC7493] Bray, T., Ed., "The I-JSON Message Format", RFC 7493, DOI 10.17487/RFC7493, March 2015, <https://www.rfc-editor.org/rfc/rfc7493>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/rfc/rfc8126>.

[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>.

[RFC8310] Dickinson, S., Gillmor, D., and T. Reddy, "Usage Profiles for DNS over TLS and DNS over DTLS", RFC 8310, DOI 10.17487/RFC8310, March 2018, <https://www.rfc-editor.org/rfc/rfc8310>.

[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/rfc/rfc8914>.

## 12.2. Informative References

[IANA-DNS] IANA, "Domain Name System (DNS) Parameters, Extended DNS Error Codes", <https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#extended-dns-error-codes>.

[IANA-Enterprise] "Private Enterprise Numbers (PENs)", <https://www.iana.org/assignments/enterprise-numbers/>.

[Impl-1] "Use of DNS Errors To improve Browsing User Experience With network based malware protection", March 2023, <https://datatracker.ietf.org/meeting/116/materials/slides-116-dnsop-dns-errors-implementation-proposal-slides-116-dnsop-update-on-dns-errors-implementation-00>.

[RFC5625] Bellis, R., "DNS Proxy Implementation Guidelines", BCP 152, RFC 5625, DOI 10.17487/RFC5625, August 2009, <https://www.rfc-editor.org/rfc/rfc5625>.

[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>.

[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>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/rfc/rfc8259>.

[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>.

[RFC8792] Watsen, K., Auerswald, E., Farrel, A., and Q. Wu, "Handling Long Lines in Content of Internet-Drafts and RFCs", RFC 8792, DOI 10.17487/RFC8792, June 2020, <https://www.rfc-editor.org/rfc/rfc8792>.

[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>.

[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>.

[RFC9499] Hoffman, P. and K. Fujiwara, "DNS Terminology", BCP 219, RFC 9499, DOI 10.17487/RFC9499, March 2024, <https://www.rfc-editor.org/rfc/rfc9499>.

[RFC9606] Reddy.K, T. and M. Boucadair, "DNS Resolver Information", RFC 9606, DOI 10.17487/RFC9606, June 2024, <https://www.rfc-editor.org/rfc/rfc9606>.

[RPZ] "Response Policy Zone", <https://dnsrpz.info>.

## Appendix A. Interoperation with RPZ Servers

This appendix provides a non-normative guidance for operation with a Response Policy Zones (RPZ) server [RPZ] that indicates filtering with a NXDOMAIN response with the Recursion Available bit cleared (RA=0). This guidance is provided to ease interoperation with RPZ.

When a DNS client supports this specification, it includes the SDE option in its DNS query.

If the server does not support this specification and is performing RPZ filtering, the server ignores the SDE option in the DNS query and replies with NXDOMAIN and RA=0. The DNS client can continue to accept such responses.

If the server does support this specification and is performing RPZ filtering, the server can use the SDE option in the query to identify an SDE-aware client and respond appropriately (that is, by generating a response described in Section 5.2) as NXDOMAIN and RA=0 are not necessary when generating a response to such a client.

## Appendix B. Implementation Status

> Note to the RFC Editor: please remove this appendix prior publication.

At IETF#116, Gianpaolo Scalone (Vodafone) and Ralf Weber (Akamai) presented an implementation of this specification. More details can be found at [Impl-1].

## Acknowledgements

Thanks to Vittorio Bertola, Wes Hardaker, Ben Schwartz, Erid Orth, Viktor Dukhovni, Warren Kumari, Paul Wouters, John Levine, Bob Harold, Mukund Sivaraman, Gianpaolo Angelo Scalone, Mark Nottingham, Stephane Bortzmeyer, Vladimir Cunat, and Daniel Migault for the comments.

Thanks to Ralf Weber and Gianpaolo Scalone for sharing details about their implementation.

Thanks Di Ma and Matt Brown for the DNS directorate reviews, and Joseph Salowey for the Security directorate review.

Thanks Paul Kyzivat for the Art review.

Thanks to ric Vyncke for the AD review.

Authors' Addresses

Dan Wing
Citrix Systems, Inc.
United States of America
Email: danwing@gmail.com

Tirumaleswar Reddy
Nokia
Bangalore
Karnataka
India
Email: kondtir@gmail.com

Neil Cook
Open-Xchange
United Kingdom
Email: neil.cook@noware.co.uk

Mohamed Boucadair
Orange
Rennes
35000
France
Email: mohamed.boucadair@orange.com