Workgroup: DNSOP WG
Internet-Draft:
Updates: 8914 (if approved)
Published: 3 March 2022
Intended Status: Standards Track
Expires: 4 September 2022

Authors:	D. Wing	T. Reddy	N. Cook	M. Boucadair
	Citrix	Akamai	Open-Xchange	Orange

Structured Data for Filtered DNS

Abstract

DNS filtering is widely deployed for network security but filtered DNS responses lack information for the end user to understand the reason for the filtering. Existing mechanisms to provide detail to end users cause harm especially if the blocked DNS response is to an HTTPS website.

This document defines a mechanism to explain the reason for the DNS filtering and provides HTTPS URIs to get more detail. This information can be parsed by the client and displayed, logged, or used for other purposes. This document updates RFC8914 to contain I-JSON in its EXTRA-TEXT field.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

DNS filters are deployed for a variety of reasons including 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, DNS filtering is used for the same reasons as above and additionally for parental control. Internet Service Providers 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.

Users of DNS services which perform filtering may wish to receive more information about such filtering to resolve problems with the filter -- for example to contact the administrator to allowlist a domain that was erroneously filtered or to understand the reason a particular domain was filtered. With that information, the user can choose another network, open a trouble ticket with the DNS administrator to resolve erroneous filtering, log the information, or other uses.

DNS responses can be filtered by sending a bogus (also called, "forged") A or AAAA response, NXDOMAIN error or empty answer, or an extended DNS error (EDE) 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). When an HTTP(S) enabled domain name is blocked, the network security device (e.g., CPE, firewall) presents a block page instead of the HTTP response from the content provider hosting that domain. If an HTTP enabled domain name is blocked, the network security device intercepts the HTTP request and returns a block page over HTTP. If an HTTPS enabled domain is blocked, the block page is also served over HTTPS. In order to return a block page over HTTPS, man in the middle (MITM) is enabled on endpoints by generating a local root certificate and an accompanying (local) public/private key pair. The local root certificate is installed on the endpoint while the network security device(s) stores a copy of the private key. During the TLS handshake, the network security device modifies the certificate provided by the server and (re)signs it using the private key from the local root certificate.

    *However, configuring the local root certificate on endpoints is relatively complex and not viable in several deployments like home networks, schools, and smaller businesess. In these cases, the typical behavior is that the forged DNS response directs the user towards a server hosted to display the block page which breaks the TLS connection. For web-browsing this then results in an HTTPS certificate error message indicating that a secure connection could not be established, which gives no information to the end-user about the reason for the error. The typical errors are "The security certificate presented by this website was not issued by a trusted certificate authority" (Internet Explorer/Edge"), "The site's security certificate is not trusted" (Chrome), "This Connection is Untrusted" (Firefox), "Safari can't verify the identity of the website..." (Safari on MacOS).

    *Enterprise networks do not 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 reset 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-level protection 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 a NXDOMAIN response to cause the DNS lookup to terminate in failure. In this case, an end user does not know why the domain cannot be reached and may repeatedly try to reach the domain but with no success. Frustrated, the end user may use insecure connections to reach the domain, potentially compromising both security and privacy.

3. The extended error codes Blocked, Censored, and Filtered defined in Section 4 of [RFC8914] can be returned by a DNS server to provide additional information about the cause of an DNS error. If the extended error code "Forged Answer" defined in Section 4.5 of [RFC8914] is returned by the DNS server, the client can identify the DNS response is forged together with the reason for HTTPS certificate error.

4. 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 he/she 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 "Adult Content" to enforce parental control, "Violence & Terrorism" due to an external requirement imposed by an external entity (e.g., Law Enforcement Agency), etc. These content categories cannot be standardized because the classification of domains into content categories is vendor specific, typically ranges from 40 to 100 types of categories depending on the vendor and the categories keep evolving. Furthermore, the threat data used to categorize domains may sometimes misclassify domains (e.g., domains wrongly classified as Domain Generation Algorithm (DGA) by deep learning techniques, domain wrongly classified as phishing due to crowd sourcing, new domains not categorized by the threat data). A

user needs to know the contact details of the IT/InfoSec team to raise a complaint.

5. When a resolver or forwarder forwards the received EDE option, the EXTRA-TEXT field only conveys the source of the error (Section 3 of [RFC8914]) and does not provide additional textual information about the cause of the error.

For both DNS filtering mechanisms described above, the DNS server can return extended error codes Blocked, Censored, Filtered, 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 this approach 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 computer-parsable data in the EXTRA-TEXT field of Extended DNS Errors [RFC8914].

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. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

This document uses terms defined in DNS Terminology [RFC8499].

"Requestor" refers to the side that sends a request. "Responder" refers to an authoritative, recursive resolver or other DNS component that responds to questions. Other terminology is used here as defined in the RFCs cited by this document.

"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 [I-D.ietf-dprive-dnsoquic].

3. I-JSON in EXTRA-TEXT field

Servers compliant with this specification will send JSON data in the EXTRA-TEXT field [RFC8914]. Precisely, the JSON object MUST use the I-JSON message format [RFC7493]. Requiring the use of I-JSON instead of more general JSON format improves interoperability.

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

This document defines the following JSON names:

c: (complaint) a partial URI for the user to further diagnose and possibly report mis-classified DNS filtering. The value is converted to an expanded absolute URI. This field is optional, but note its absence still allows a URI to be formed.

d: (domain) Contains the domain-name of the encrypted DNS server. This is used to create the expanded URIs for both the "c" and "r" fields, and also detect undesired forwarding of the EXTRA-TEXT field in the Extended DNS Error option. This field is mandatory.

j: (justification) the textual justification for this particular DNS filtering. This field is mandatory.

o: (organization) human-friendly name of the organization that filtered this particular DNS query. This field is optional.

r: (regulation) a partial URI to retrieve the public or private rule, law, or regulation describing the reason for this DNS filter. This might point at an employment agreement (for an enterprise performing filtering) or a national government regulation (for an ISP performing filtering). This field is optional, but note its absence still allows a URI to be formed.

New JSON names MUST be defined in the IANA registry (Section 7), consist only of lower-case ASCII characters, digits, and hyphens (that is, Unicode characters U+0061 through 007A, U+0030 through U+0039, and U+002D). These names MUST be 63 characters or shorter and it is RECOMMENDED they be as short as possible.

To reduce packet overhead 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 (no spaces, no blank lins).

The JSON data can be parsed to display to the user, logged, or otherwise used to assist the end-user or IT staff with troubleshooting and diagnosing the cause of the DNS filtering.

4. Protocol Operation

4.1. Client Generating Request

When generating a DNS query, the client MUST include the OPT pseudo-RR [RFC6891] to elicit the Extended DNS Error option [RFC8914] in the DNS response.

4.2. Server Generating Response

When the DNS server filters its DNS response to an A or AAAA record query, the DNS response MAY contain an empty answer, NXDOMAIN, or a forged A or AAAA response, as desired by the DNS server. In addition, if the query contained the OPT pseudo-RR the DNS server MAY return more detail in the EXTRA-TEXT field as described in Section 4.3.

Over time a domain name might be filtered, then not filtered, then filtered again. Additionally, the user might take minutes or even days before investigating a filtered DNS query. Thus, the complaint URI is RECOMMENDED to include sufficient detail to determine the filtering state when the DNS filtering occurred. If and how this is encoded into the complaint URI is an implementation decision.

Servers may decide to return small TTL values in filtered DNS responses (e.g., 2 seconds) to handle domain category and reputation updates.

4.3. Client Processing Response

On receipt of a DNS response with an Extended DNS Error option, the following actions are performed if the EXTRA-TEXT field contains valid JSON:

- The Requestor MUST check that the response was received over an encrypted DNS channel. If not, the Requestor MUST discard data in the EXTRA-TEXT field.
- Servers which don't support this specification might use plain text in the EXTRA-TEXT field so that Requestors SHOULD properly handle both plaintext and JSON text in the EXTRA-TEXT field.
- The DNS response MUST also contain an extended error code of "Censored", "Blocked", "Filtered" or "Forged" [RFC8914], otherwise the EXTRA-TEXT field is discarded.

*If either of the mandatory JSON names "d" and "j" are missing or have empty values in the EXTRA-TEXT field, the entire JSON is discarded.

*The Requestor expands the values in "c" and "r" by prefixing the two values with "https://" and the value of the "d" name. Then the Requestor further expands each of the "c" and "r" URIs by appending two URL query parameters: "type" indicating the name of the DNS resource record type queried and "name" indicating the name of the DNS resource record queried.

Note the partial URI value in "c" or "r" will already contain zero or more query parameters so implementations should substitute "?" and "&" accordingly.

*If a DNS client has enabled opportunistic privacy profile (Section 5 of [RFC8310]) for DoT, the DNS client will either fallback to an encrypted connection without authenticating the DNS server provided by the local network or fallback to clear text DNS, and cannot exchange encrypted DNS messages. Both of these fallback mechanisms adversely impacts security and privacy. If the DNS client has enabled opportunistic privacy profile for DoT, the DNS client MUST ignore the EXTRA-TEXT field of the EDE responses, but SHOULD process other parts of the response.

*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; 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 client MAY process the EXTRA-TEXT field of the DNS response. 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.

*If the DNS client determines that the encrypted DNS server does not offer DNS filtering service, it MUST discard the EXTRA-TEXT field of the EDE response. For example, the DNS client can learn whether the encrypted DNS resolver performs DNS-based content filtering or not by retrieving resolver information using the method defined in [I-D.reddy-add-resolver-info].

*When a forwarder receives an EDE option, whether or not (and how) to pass along JSON information in the EXTRA-TEXT on to their client is implementation dependent [RFC5625]. Implementations MAY choose to not forward the JSON information, or they MAY choose to create a new EDE option that conveys the information in the "j" and "d" fields encoded in the JSON object. To check if the EXTRA-

TEXT originated from its server (rather than by something further upstream) the DNS client MUST verify the domain name in the Structured Error "d" value matches the domain name of the encrypted DNS resolver. If this match fails, the DNS client discards the EXTRA-TEXT field.

5. Examples

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

```
{
  "c": "?time=1621902483",
  "d": "ns.example.com",
  "j": "malware present for 23 days",
  "o": "example.net Filtering Service",
  "r": "?country=atlantis"
}
```

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 spaces, no blank lines) with '\' line wrapping per [RFC8792].

```
============== NOTE: '\' line wrapping per RFC 8792 ===============

{"c":"?time=1621902483","d":"ns.example.com","j":"malware present \
for 23 days","o":"example.net Filtering Service","r":\
"?country=atlantis"}
```

Figure 2: Minified response

Upon receipt, the two partial URIs ("c" and "r") are expanded to become fully-formed URIs. The class, type, and name are pulled from the DNS response (that matches the associated query) so that the fully-formed "c" URI becomes "https://ns.example.net?time=1621902483&type=a&name=example.org" and the "r" URI becomes "https://ns.example.net?country=atlantis&type=a&name=example.org".

6. Security Considerations

Security considerations in Section 6 of [RFC8914] apply to this document.

To minimize impact of active on-path attacks on the DNS channel, the client validates the response as described in Section 4.3.

A client might choose to display the information in the EXTRA-TEXT field if and only if the encrypted resolver has sufficient reputation, according to some local 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. If the client decides not to display the 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 and error description for the EDE code to the end-user.

If the browser visits either of the URIs in the response ("c" or "r"), the browser SHOULD reduce the attack surface of the client by using an isolated environment precautions such as clearly labeling the page as untrusted or prevent user interaction with the page. Such isolation should prevent transmitting cookies, block JavaScript, block auto-fill of credentials or personal information, and be isolated from the user's normal environment. The user's preferred language SHOULD be included in the Content-Language header.

The URIs in the response ("c" or "r") may be accessed via a browser. In such a case, the browser SHOULD reduce the attack surface of the client by using an isolated environment precautions such as clearly labeling the page as untrusted or prevent user interaction with the page. Such isolation should prevent transmitting cookies, block JavaScript, block auto-fill of credentials or personal information, and be isolated from the user's normal environment. The Content-Language header can disclose personal information so its inclusion is a local policy choice of the client.

When displaying the free-form text of "o" and "j", the browser SHOULD NOT make any of those elements into actionable (clickable) links.

Although the "d" value is validated, an attacker who is able to inject (or modify) the EXTRA-TEXT field so that a DNS proxy or DNS forwarder, unaware of the option, will forward it and pass the validation checks described in Section 4.3. This means the other JSON fields can be controlled by the attacker. The "j" and "o" fields are, perhaps, the most interesting for an attacker to modify for nefarious purposes, because the "d" field has to match the encrypted DNS server's name and the expanded URIs from the "c" and "r" will point at the DNS resolver not under the attacker's control.

The authors anticipate enhancements to [I-D.reddy-add-resolver-info] will reduce or eliminate the concern described in previous paragraph.

7. IANA Considerations

This document requests IANA to register the "application/json+structured-dns-error" media type in the "Media Types" registry [IANA-MediaTypes]. This registration follows the procedures specified in [RFC6838]:

Type name: application

Subtype name: json+structured-dns-error

Required parameters: N/A

Optional parameters: N/A

Encoding considerations: as defined in Section NN of [RFCXXXX].

Security considerations: See Section NNN of [RFCXXXX].

Interoperability considerations: N/A

Published specification: [RFCXXXX]

Applications that use this media type: Section NNNN of [RFCXXXX].

Fragment identifier considerations: N/A

Additional information: N/A

Person & email address to contact for further information: IETF, iesg@ietf.org

Intended usage: COMMON

Restrictions on usage: none

Author: See Authors' Addresses section.

Change controller: IESG

Provisional registration? No

8. Changes

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

8.1. Changes from 01 to 02

*repurpose Extended DNS Error [RFC8914]'s EXTRA-TEXT field to carry JSON, which causes this document to update RFC8914

- clarified DNS forwarders might either forward EXTRA-TEXT (a) without change or (b) rewrite "j" and "d"

8.2. Changes from 00 to 01

- removed support for multiple responsible parties

- one-character JSON names to minimize JSON length

- partial URI sent in "c" and "r" names, combined with "d" name sent in JSON to minimize attack surface and minimize JSON length

- moved EDNS(0) forgery-mitigation text, some Security Considerations text, and some other text from [I-D.reddy-dnsop-error-page] to this document

9. References

9.1. 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>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <https://www.rfc-editor.org/info/rfc6838>.

[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>.

[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/info/rfc7159>.

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

[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>.

[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/info/rfc8310>.

9.2. Informative References

[I-D.ietf-dprive-dnsoquic] Huitema, C., Dickinson, S., and A. Mankin, "DNS over Dedicated QUIC Connections", Work in Progress, Internet-Draft, draft-ietf-dprive-dnsoquic-10, 28 February 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-dprive-dnsoquic-10>.

[I-D.reddy-add-resolver-info] Reddy, T. and M. Boucadair, "DNS Resolver Information", Work in Progress, Internet-Draft, draft-reddy-add-resolver-info-04, 7 October 2021, <https://datatracker.ietf.org/doc/html/draft-reddy-add-resolver-info-04>.

[I-D.reddy-dnsop-error-page] Reddy, T., Cook, N., Wing, D., and M. Boucadair, "DNS Access Denied Error Page", Work in Progress, Internet-Draft, draft-reddy-dnsop-error-page-08, 4 June 2021, <https://datatracker.ietf.org/doc/html/draft-reddy-dnsop-error-page-08>.

[IANA-MediaTypes] IANA, "Media Types", <https://www.iana.org/assignments/media-types>.

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

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

[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/info/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/info/rfc8484>.

[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>.

[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/info/rfc8792>.

[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>.

Authors' Addresses

Dan Wing
Citrix Systems, Inc.
United States of America

Email: dwing-ietf@fuggles.com

Tirumaleswar Reddy
Akamai
Bangalore
Karnataka
India

Email: kondtir@gmail.com

Neil Cook
Open-Xchange
United Kingdom

Email: neil.cook@noware.co.uk

Mohamed Boucadair
Orange
35000 Rennes
France

Email: mohamed.boucadair@orange.com