```
DNSOP WG                                                       T. Reddy
Internet-Draft                                                   McAfee
Intended status: Standards Track                                N. Cook
Expires: March 3, 2021                                    Open-Xchange
                                                                D. Wing
                                                                 Citrix
                                                           M. Boucadair
                                                                 Orange
                                                        August 30, 2020
```

DNS Access Denied Error page
draft-reddy-dnsop-error-page-04

Abstract

   When a DNS server filters a query the response conveys no detailed
   explanation of why that query was blocked, leading to end-user
   confusion. This document defines a method to return an URI that
   explains the reason a DNS query was filtered.

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 March 3, 2021.

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. These are discussed in more detail below:

   o Various network security services are provided by Enterprise
      networks to protect endpoints (e.g., Hosts including IoT devices).
      Network-based security solutions such as firewalls and Intrusion
      Prevention Systems (IPS) rely on network traffic inspection to
      implement perimeter-based security policies. The network security
      services may, for example, prevent malware download, block known
      malicious domains, block phishing sites, etc. These network
      security services act on DNS queries originating from endpoints.
      For example, DNS firewalls, a method of expressing DNS response
      policy information inside specially constructed DNS zones, known
      as Response Policy Zones (RPZs) allows DNS servers to modify DNS
      responses in real time to stop access to malware and phishing
      domains. Note that some of the commonly known types of malware
      are viruses, worms, trojans, bots, ransomware, backdoors, spyware,
      and adware.

   o Network devices in a home network offer network security to
      protect the devices connected to the home network by performing
      DNS-based content filtering. The network security service may,

for example, block access to specific domains to enforce parental
control, block access to malware sites, etc.

o ISPs typically block access to some domains due to a requirement
   imposed by an external entity (e.g., Law Enforcement Agency) by
   performing DNS-based content filtering.

DNS responses can be filtered by sending a bogus ("forged") A or AAAA
response, NXDOMAIN error or empty answer, or an extended error code
defined in [I-D.ietf-dnsop-extended-error]. Each of these have
advantages and disadvantages, discussed below:

1. The DNS response is forged providing IP addresses that point to a
    HTTP(S) server alerting the end user of the reason for blocking
    access to the domain (e.g., malware). When a HTTP(S) enabled
    domain name is blocked, the network security device presents a
    block page instead of the HTTP response from the content
    provider. 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, and the network security device(s) store 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 with the private key from the local root
    certificate.

    * However, configuring the local root certificate on endpoints
       is not viable option in several deployments like Home
       networks, Schools, Small Office/Home Office (SOHO), and Small/
       Medium Enterprise (SME). 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 devices
      connected to their network are managed by the IT team or
      Mobile Device Management (MDM) devices, especially in the
      quite common BYOD ("Bring Your Own Device") 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 unsuccessfully reach the
      domain. Frustrated, the end user may use insecure interfaces
      to reach the domain, potentially compromising both security
      and privacy. Furthermore, certificate errors train users to
      click through certificate errors, which is poor 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 [Chrome-Install-Cert] on a host device.
      Doing so, however, is also poor security practice as it
      creates a security vulnerability that may be exploited by a
      MITM attack. When the 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 unsuccessfully reach the domain. Frustrated,
       the end user may use insecure interfaces to reach the domain,
       potentially compromising both security and privacy.

   3. The extended error codes Blocked, Censored, and Filtered defined
       in [I-D.ietf-dnsop-extended-error] can be returned by the DNS
       server to provide additional information about the cause of an
       DNS error. If the extended error code "Forged answer" defined in
       [I-D.ietf-dnsop-extended-error] is returned by the DNS server,
       the client can identify the DNS response is forged and the reason
       for HTTPS certificate error. These extended error codes do not
       suffer from the limitations discussed in (1) and (2) but the user
       still does not know the exact reason nor the user is aware of the
       exact entity blocking the access to the domain. For example, a
       DNS server may block access domain based on the content category
       like "Adult Content" to enforce parental control, "Violence &
       Terrorism" due to an external requirement imposed by an external
       entity (e.g., Law Enforcement Agency), etc. The content
       categories for domains 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.
       Further, the threat data used to categorize domains may sometimes

mis-classify domains (e.g., Domains wrongly classified as DGA (Domain Generation Algorithm) by deep learning techniques, domain wrongly classified as phishing due to crowd sourcing, new domains not categorized by the threat data). The end user needs to know the contact details of the IT/InfoSec team to raise a complaint.

No matter which type of response is generated (forged IP address, NXDOMAIN or empty answer, or an extended error code), the user who generated the 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 a URI which, when accessed, provides such information to the user.

One of the other benefits of this approach is to eliminate the need to "spoof" block pages for HTTPS resources, as the block page no longer needs to create a signed certificate when blocking a destination. This avoids the need to install a local root certificate authority on those IT-managed devices.
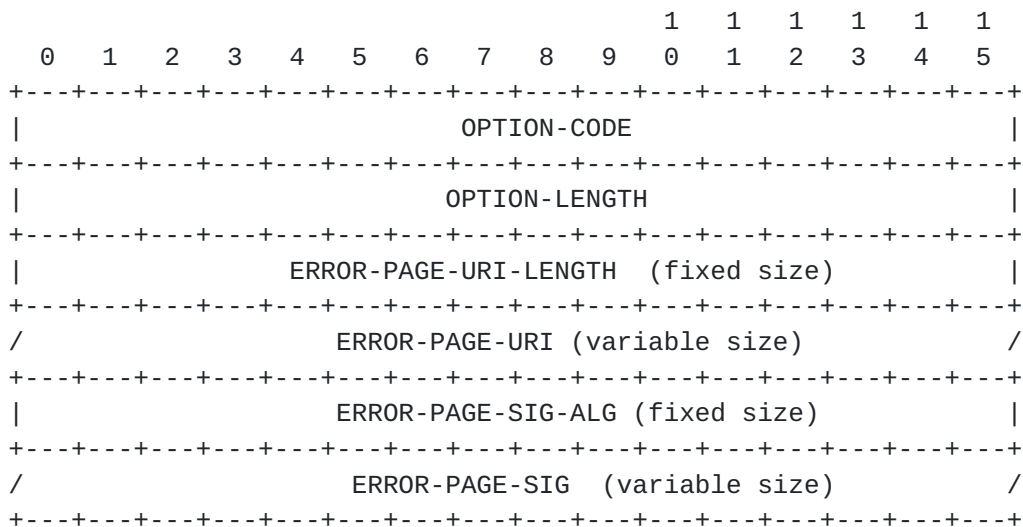
2. Terminology

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 makes use of the terms defined in [RFC8499] and [I-D.ietf-dnsop-terminology-ter].

'Encrypted DNS' refers to DNS-over-HTTPS [RFC8484], DNS-over-TLS [RFC7858], or DNS-over-QUIC [I-D.ietf-dprive-dnsoquic].

3. Error page URI EDNS0 option format

This document uses an EDNS0 [RFC6891] option to include the URI that gives additional information in a DNS response about the cause of blocking access to a domain. The option is structured as follows:

```
                                     1   1   1   1   1   1
      0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |                        OPTION-CODE                           |
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |                        OPTION-LENGTH                         |
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |              ERROR-PAGE-URI-LENGTH  (fixed size)             |
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    /                  ERROR-PAGE-URI (variable size)              /
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |              ERROR-PAGE-SIG-ALG (fixed size)                 |
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    /                  ERROR-PAGE-SIG  (variable size)             /
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

The description of the fields is as follows:

o OPTION-CODE: TBD, indicates the code assigned for Error page URI
   (Section 6.1.2 of [RFC6891]). [RFC Editor: change TBD to the
   proper code once assigned by IANA.]

o OPTION-LENGTH: See Section 6.1.2 of [RFC6891]. This fields
   contains the length of the payload (everything after OPTION-
   LENGTH) in octets. The variability of the option length stems
   from the variable-length ERROR-PAGE-URI and ERROR-PAGE-SIG fields.

o ERROR-PAGE-URI-LENGTH: This 16-bit field indicates the length of
   ERROR-PAGE-URI. It MUST NOT be set to 0.

o ERROR-PAGE-URI: a variable length UTF-8 encoded [RFC5198] text
   field containing the URI Template [RFC6570] that gives additional
   information about the cause of blocking access to a domain. The
   ERROR-PAGE-URI field MUST NOT be zero octets in length.

o ERROR-PAGE-SIG-ALG: A 16-bits field that contains the algorithm
   used to generate the signature for the Error page URI Template.
   The values are defined in the TLS SignatureScheme [TLS-SIG-SCHEME]
   with limitations described in Section 5.

o ERROR-PAGE-SIG, a variable length field containing the signature
   of the Error page URI Template. The signature generation process
   is discussed in Section 5.

The Error page URI option MUST only be included in error responses
SERVFAIL, NXDOMAIN, or REFUSED to a query that included the OPT
Pseudo-RR [RFC6891].

The URI Template defined in ERROR-PAGE-URI describes how to construct
the URL to fetch the error page. The agent acting as HTTPS client on
the endpoint encodes a FQDN to which access is denied into an HTTP
GET request to retrieve the error page. The HTTPS server returning
the error page defines the URI used by the HTTP GET request through
the use of a URI Template. The URI Template is processed with a
defined variable "target-domain" whose value is set to the FQDN to
which access is denied. The FQDN is encoded using base64url
[RFC4648] and then provided as the variable value for "target-domain"
to expand the URI Template into a URI reference in the HTTP GET
request. Padding characters for base64url MUST NOT be included.

An example is illustrated below:

If the URI Template is "https://block.example.net/block-page{?target-
domain}" for the HTTPS server returning the error page and access to
the target domain "example.com" is blocked by the encrypted DNS
server, the variable "target-domain" has the value "example.com"
base64url encoded into an HTTP GET request. In the above example,
the expansion of the above URI Template is
"https://block.example.net/block-page?target-domain=ZXhhbXBsZS5jb20".

HTTP/2 [RFC7540] is the minimum RECOMMENDED version of HTTP to use to
retrieve the error page. The HTTPS client retrieving the error page
MUST verify the entire certification path per [RFC5280]. The HTTPS
client additionally uses validation techniques as described in
[RFC6125] to compare the domain name in the error page URI to the
server certificate provided in TLS handshake. See [RFC7525] for
additional TLS recommendations.

4. Error page URI Processing

The DNS client MUST follow the following rules to process the Error
page URI EDNS0 option:

o The Error page URI EDNS0 option is susceptible to forgery. In
   order to defend against this attack the DNS client MUST NOT
   process the DNS response with Error page URI EDNS0 option unless
   DNS messages exchanged are cryptographically protected using
   encrypted DNS.

o If an 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. The fallback
   adversely impacts security and privacy. If the DNS client has

enabled opportunistic privacy profile for DoT, the client MUST NOT
process the DNS response with Error page URI EDNS0 option.

o   If an 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 can process the
    DNS response with Error page URI EDNS0 option. Note that the
    strict and opportunistic privacy profiles as defined in [RFC8310]
    only applies to DoT protocol, there has been no such distinction
    made for DoH protocol.

o   If the DNS response contains more than one Error page URI EDNS0
    option, the DNS client MUST discard multiple Error page URI EDNS0
    options in the DNS response.

o   The Error page URI EDNS0 option MUST be processed by the DNS
    client for a "Censored", "Blocked" or "Filtered" extended error
    codes and MUST be ignored for any other type of extended DNS error
    code. When "Censored", "Blocked" or "Filtered" extended error
    code is returned in conjunction with an Error page URI EDNS0
    option, any other resource records in the answer MUST be ignored
    by clients supporting this specification.

o   If the encrypted DNS server does not offer DNS filtering service
    (e.g., prevent access to malware sites or objectionable content
    (e.g., legal obligation)), the DNS client MUST reject the Error
    page URI EDNS0 option. The mechanism discussed in
    [I-D.reddy-add-server-policy-selection] can be used by the DNS
    client to assess whether the encrypted DNS server performs DNS-
    based content filtering.

o   If the Error page URI EDNS0 option is included in a NOERROR RCODE
    message, the DNS client MUST reject the Error page URI EDNS0
    option.

o   The DNS client MUST reject the error page URI if the scheme is not
    "https".

o   The DNS client verifies the signature in the ERROR-PAGE-SIG field
    following the mechanism discussed in Section 5. If the signature
    is valid, the client can positively identify that the Error page
    URI EDNS0 option has been generated by the encrypted DNS server
    and the encrypted DNS server did not forward the Error Page URI

EDNS0 option from an upstream resolver. If signature validation
fails, the DNS client MUST reject the Error page URI EDNS0 option.

Because EDNS is a hop-by-hop extension to DNS and EDNS responses are
not cached (Section 6.2.1 of [RFC6891]), a DNS resolver MUST NOT
propagate a received Error Page URI EDNS0 option, because an attacker
could insert a bogus URI; instead, if access to a domain is denied,
the DNS resolver MUST generate its own Error Page URI. Because DNS
forwarders (or DNS proxies) are supposed to propagate unknown EDNS0
options (Section 4.1 and Section 4.4.1 of [RFC5625]), the Error Page
URI EDNS0 option may get propagated. To detect both of these
scenarios, the Error Page URI Template is protected with an object
signature as described in Section 5.

5. Sign and Verify

The algorithms for generating signature for DNS resource record sets
(RRsets) are defined in [DNSKEY-IANA]. The "mandatory-to-implement"
algorithms are RSA, Elliptic Curve Digital Signature Algorithm
(ECDSA), and Edwards-curve Digital Security Algorithm (EdDSA)
[RFC8624]. Along similar lines, the encrypted DNS server's end-
entity certificate's public key and the signature algorithm with
which the key can be used are RSA, ECDSA, and EdDSA [RFC8446]. If
ECDSA is used, it is RECOMMENDED to use the deterministic digital
signature generation procedure of the ECDSA, specified in [RFC6979].

The signature is generated by the encrypted DNS server using the
Error page URI Template, private key of the encrypted DNS server's
end-entity certificate as inputs to the signature algorithm. The
signature algorithm in the ERROR-PAGE-SIG-ALG field MUST be
compatible with the key in the DNS server's end-entity certificate.
The implementation MUST support the same set of algorithms in the TLS
client for validating the signature in the CertificateVerify message
from the server in the TLS handshake and in the DNS client to
validate the signature for the Error page URI Template. As a
reminder, the server's end-entity certificate's public key will be
compatible with the selected authentication algorithm from the
client's "signature_algorithms" TLS extension (Section 4.4.2.2 of
[RFC8624]).

If the signature algorithm in the ERROR-PAGE-SIG-ALG field is not
compatible with the key in the DNS server's end-entity certificate,
the DNS client MUST reject the Error page URI EDNS0 option. The DNS
client verifies the signature using the signature in the ERROR-PAGE-
SIG field, error page URI Template and DNS server's end-entity
certificate's public key as inputs to the signature algorithm. For
example, if Ed25519 is used, Ed25519 signature algorithm and

verification of the Ed25519 signature are described in Sections 5.1.6
and 5.1.7 of [RFC8032], respectively.

6. ERROR Page

The following text outlines the RECOMMENDED contents of an error page
to assist the operator developing the error page.

o The exact reason for blocking access to the domain. If the domain
   is blocked based on some threat data, the threat type associated
   with the blocked domain can be provided/displayed to the end user.
   For example, the reason can indicate the type of malware blocked
   like spyware and the damage it can do the security and privacy of
   the user.

o The domain name blocked.

o If query was blocked by regulation, a pointer to a regulatory text
   that mandates this query block.

o The entity (or organization) blocking the access to the domain and
   contact details of the IT/InfoSec team to raise a complaint.

o The blocked error page to not include Ads and dynamic content.

The content of the error page discussed above is non-normative, the
above text only provides the guidelines and template for the error
page and.

o Does not attempt to offer an exhaustive list for the contents of
   an error page.

o It is not intended to form the basis of any legal/compliance for
   developing the error page.

7. Usability Considerations

The error page SHOULD be returned in the user's preferred language as
expressed by the Accept-Language header. If the error page is
displayed in a language not known to the end user and assuming
Internationalization features failed, browser extensions to translate
to user's native language can be used. For example, "Google
Translate" extension [Chrome-Translate] provided by Google on Chrome
can be used by the user to translate the error page. The "Google
Translate" extension automatically detects whether the language of a
page is different from the language the user has selected. If it is
in a different language, a banner appears at the top of the page.

The user can click on the Translate button in the banner to have all
the text on the page appear in the language selected by the user.

8. **Security Considerations**

Security considerations in [I-D.ietf-dnsop-extended-error] and
[RFC8624] need to be taken into consideration.

The Error page URI EDNS0 option causes an HTTPS retrieval by the
client. To prevent forgery of the Error page URI EDNS0 option, this
specification requires it only be sent only over an encrypted DNS
channel with an authorized DNS server.

The DNS client can learn the filtering capability of an encrypted DNS
server using [I-D.reddy-add-server-policy-selection].
[I-D.reddy-add-server-policy-selection] also discusses how a DNS
client can authenticate it is connecting to an encrypted DNS server
hosted by a specific organization (e.g., ISP). This information is
cryptographically signed to attest its authenticity. It is
particularly useful when the encrypted DNS server is insecurely
discovered and prevents the client from connecting to an attacker's
encrypted DNS server.

To reduce threat surface when retrieving the Error page URL over
HTTPS, the client SHOULD perform the retrieval using an isolated
environment. 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.
Browsers perform some of these limitations today when accessing
captive portals [Safari-Cookie], during private browsing, or using
containerization [Facebook-Container]).

The encrypted DNS session provides transport security for the
interaction between the DNS client and server, but DNSSEC signing and
validation is not possible for the Error Page URI EDNS0 option
returning the error page URI template. For example, if access to a
doman is blocked, the encrypted DNS resolver can rewrite the response
to send NXDOMAIN error and Error page URI EDNS0 option in the DNS
response, it will omit DNSSEC RRsets, because the modified responses
cannot be verified by DNSSEC signatures. However, the signature in
the Error Page URI EDNS0 option provides data origin authentication
of the Error Page URI EDNS0 option. Nevertheless, the Error Page URI
EDNS0 option returning the error page URI Template should be treated
only as diagnostic information and MUST NOT alter DNS protocol
processing.

By design, the object referenced by the error page URL potentially
exposes additional information about the DNS resolution process that

may leak information. An example of this is the reason for blocking
the access to the domain name and the entity blocking access to the
domain.

9. IANA Considerations

9.1. A New Error Page URI EDNS Option

This document defines a new EDNS(0) option, entitled "Error Page
URI", assigned a value of TBD from the "DNS EDNS0 Option Codes (OPT)"
registry [to be removed upon publication:
[http://www.iana.org/assignments/dns-parameters/dns-
parameters.xhtml#dns-parameters-11]

```
Value  Name                  Status    Reference
-----  ---------------       ------    -----------------
 TBD   Error Page URI        Standard      [ This document ]
```

10. Acknowledgements

Thanks to Vittorio Bertola, Wes Hardaker, Ben Schwartz, Erid Orth,
Viktor Dukhovni and Bob Harold for the comments.

11. References

11.1. Normative References

[I-D.ietf-dnsop-extended-error]
          Kumari, W., Hunt, E., Arends, R., Hardaker, W., and D.
          Lawrence, "Extended DNS Errors", draft-ietf-dnsop-
          extended-error-16 (work in progress), May 2020.

[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>.

[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>.

[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>.

   [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S.,
              Housley, R., and W. Polk, "Internet X.509 Public Key
              Infrastructure Certificate and Certificate Revocation List
              (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008,
              <https://www.rfc-editor.org/info/rfc5280>.

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

   [RFC6125] Saint-Andre, P. and J. Hodges, "Representation and
              Verification of Domain-Based Application Service Identity
              within Internet Public Key Infrastructure Using X.509
              (PKIX) Certificates in the Context of Transport Layer
              Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March
              2011, <https://www.rfc-editor.org/info/rfc6125>.

   [RFC6570] Gregorio, J., Fielding, R., Hadley, M., Nottingham, M.,
              and D. Orchard, "URI Template", RFC 6570,
              DOI 10.17487/RFC6570, March 2012,
              <https://www.rfc-editor.org/info/rfc6570>.

   [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>.

   [RFC6979] Pornin, T., "Deterministic Usage of the Digital Signature
              Algorithm (DSA) and Elliptic Curve Digital Signature
              Algorithm (ECDSA)", RFC 6979, DOI 10.17487/RFC6979, August
              2013, <https://www.rfc-editor.org/info/rfc6979>.

   [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre,
              "Recommendations for Secure Use of Transport Layer
              Security (TLS) and Datagram Transport Layer Security
              (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May
              2015, <https://www.rfc-editor.org/info/rfc7525>.

   [RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext
              Transfer Protocol Version 2 (HTTP/2)", RFC 7540,
              DOI 10.17487/RFC7540, May 2015,
              <https://www.rfc-editor.org/info/rfc7540>.

   [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>.

   [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol
              Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018,
              <https://www.rfc-editor.org/info/rfc8446>.

   [RFC8624] Wouters, P. and O. Sury, "Algorithm Implementation
              Requirements and Usage Guidance for DNSSEC", RFC 8624,
              DOI 10.17487/RFC8624, June 2019,
              <https://www.rfc-editor.org/info/rfc8624>.

   [TLS-SIG-SCHEME]
              "IANA, TLS SignatureScheme",
              <https://www.iana.org/assignments/tls-parameters/tls-
              parameters.xhtml#tls-signaturescheme>.

11.2. Informative References

   [Chrome-Install-Cert]
              "How to manually install the Securly SSL certificate in
              Chrome", <support.securly.com/hc/en-us/articles/206081828-
              How-to-manually-install-the-Securly-SSL-certificate-in-
              Chrome>.

   [Chrome-Translate]
              "Google Translate",
              <https://chrome.google.com/webstore/detail/google-
              translate/aapbdbdomjkkjkaonfhkkikfgjllcleb/RK%3D2/
              RS%3DBBFW_pnWkPY0xPMYsAZI5xOgQEE->.

   [DNSKEY-IANA]
              "IANA, Domain Name System Security (DNSSEC) Algorithm
              Numbers",
              <http://www.iana.org/assignments/dns-sec-alg-numbers>.

   [Facebook-Container]
              "Facebook container for Firefox",
              <https://www.mozilla.org/en-US/firefox/
              facebookcontainer/>.

   [I-D.ietf-dnsop-terminology-ter]
              Hoffman, P., "Terminology for DNS Transports and
              Location", draft-ietf-dnsop-terminology-ter-02 (work in
              progress), August 2020.

[I-D.ietf-dprive-dnsoquic]
          Huitema, C., Mankin, A., and S. Dickinson, "Specification
          of DNS over Dedicated QUIC Connections", draft-ietf-
          dprive-dnsoquic-00 (work in progress), April 2020.

[I-D.reddy-add-server-policy-selection]
          Reddy.K, T., Wing, D., Richardson, M., and M. Boucadair,
          "DNS Server Selection: DNS Server Information with
          Assertion Token", draft-reddy-add-server-policy-
          selection-04 (work in progress), July 2020.

[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>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital
          Signature Algorithm (EdDSA)", RFC 8032,
          DOI 10.17487/RFC8032, January 2017,
          <https://www.rfc-editor.org/info/rfc8032>.

[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>.

[Safari-Cookie]
          "Isolated cookie store (CVE-2016-1730)",
          <https://support.apple.com/en-us/HT205732>.

Authors' Addresses

Tirumaleswar Reddy
McAfee, Inc.
Embassy Golf Link Business Park
Bangalore, Karnataka 560071
India

Email: kondtir@gmail.com

Neil Cook
Open-Xchange
UK

Email: neil.cook@noware.co.uk

Dan Wing
Citrix Systems, Inc.
USA

Email: dwing-ietf@fuggles.com

Mohamed Boucadair
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com