Workgroup: ADD
Internet-Draft: draft-homburg-add-codcp-00
Published: 11 July 2022
Intended Status: Standards Track
Expires: 12 January 2023
Authors: P.C. Homburg

Control Options For DNS Client Proxies

Abstract

The introduction of many new transport protocols for DNS in recent years (DoT, DoH, DoQ) significantly increases the complexity of DNS stub resolvers that want to support these protocols. A practical way forward is to have a DNS client proxy in the host operating system. This allows applications to communicate using Do53 and still get the privacy benefit from using more secure protocols over the internet. However, such a setup leaves the application with no control over which transport the proxy uses. This document introduces EDNS(0) options that allow a stub resolver to request certain transport and allow the proxy to report capabilities and actual transports that are available.

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 12 January 2023.

respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Definitions

Do53 The original, plain text DNS transport as described in [RFC1034][RFC1035]. Typically, UDP is used, with the DNS server

listening on port 53. Sometimes, for example, for large responses, TCP is used, also on port 53.

DoH DNS over HTTPS as described in [RFC8484].

DoT DNS over TLS as described in [RFC7858]

DoQ DNS over QUIC ([RFC9000]) as described in [I-D.ietf-dprive-dnsoquic], not to be confused with DNS over HTTP/3 which also uses QUIC

EDNS(0) Option An option as described in [RFC6891]

h2 This TLS ALPN identifies HTTP/2 as described in [RFC7540]

h3 This TLS ALPN identifies HTTP/3, which is HTTP over QUIC and is described in I.D.ietf-quic-http (expired draft)

Interface Name A name that identifies a network interface as described in [RFC3493]. In addition, an interface index converted to a decimal number is also consider an interface name.

PKIX Public-Key Infrastructure using X.509. See [RFC5280]

2. Introduction

The introduction of many new transport protocols for DNS in recent years (DoT, DoH, DoQ) significantly increases the complexity of DNS stub resolvers that want to support these protocols. In addition, for short-lived applications, the overhead of setting a DoH connection is quite high if the application only needs to send a few DNS requests.

A practical way forward is to have a DNS client proxy in the host operating system. A local proxy may provide some benefit to short-lived applications by caching results. In particular if the system uses a so called 'public DNS resolver'. In general we assume that the cache is tagged according to the source of a reply and the transport it is received on.

This allows applications to communicate using Do53 and still get the privacy benefits from using more secure protocols over the internet. However, such a setup leaves the application with no control over which transport the proxy uses. This document introduces EDNS(0) options that allow a stub resolver to request certain transports and allow the proxy to report capabilities and actual transports that are available.

With respect to DNSSEC, we assume that an application that needs DNSSEC validation, for example, for DANE validation or SSHFP, will

perform the DNSSEC validation within the application itself and does not trust the proxy. The proxy can of course do DNSSEC validation as well. Important however, is that an untrusted proxy cannot provide an application with a traditional (unsigned) trust anchor.

For the transport configuration we expect three levels of details. The first is a choice between requiring authenticated encryption, also allowing unauthenticated encryption or doing opportunistic encryption on an best effort basis. The second level is where the application also specifies the names and/or IP addresses of upstream resolvers. The third level is where the application also specifies which transports (Do53, DoT, DoH, DoQ) are allowed to be used. A final transport parameter is the outgoing interface that is to be used.

For authentication we can have a mix of PKIX and DANE. Options are one of the two and not the other, both or one of the two.

In a response, the proxy reports the interface, resolver, and transport used.

As described in Section 3 of [RFC5625], some simple DNS proxies may just forward DNS packets without handling of EDNS(0) options. So what could happen is that an application sends a privacy sensitive request to local proxy, expecting the proxy upstream connection to be encrypted. However, a simple proxy may just forward the request unencrypted to another proxy, for example, one in a CPE that does implement the protocol described in this document. So what could happen is that the request travels unencrypted over a local lan, or if proxies deeper in the network support this protocol, even further without the application noticing that something is wrong.

To handle this case, we introduce an option where the proxy reports whether the connection between the stub resolver and the proxy is host-local, link-local, or site-local or global.

In the ideal case, the host operating system provides applications with a secure way to access a DNSSEC trust anchor that is maintained according to [RFC5011]. However in situations where this is not the case, an application can fall back to [RFC7958]. However, for short lived processes, there is considerable overhead in issuing two HTTP(S) requests to data.iana.org to obtain the trust anchor XML file and the signature over the trust anchor. For this reason, it makes sense to let the proxy cache this information.

3. Key Words

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.

4. Description

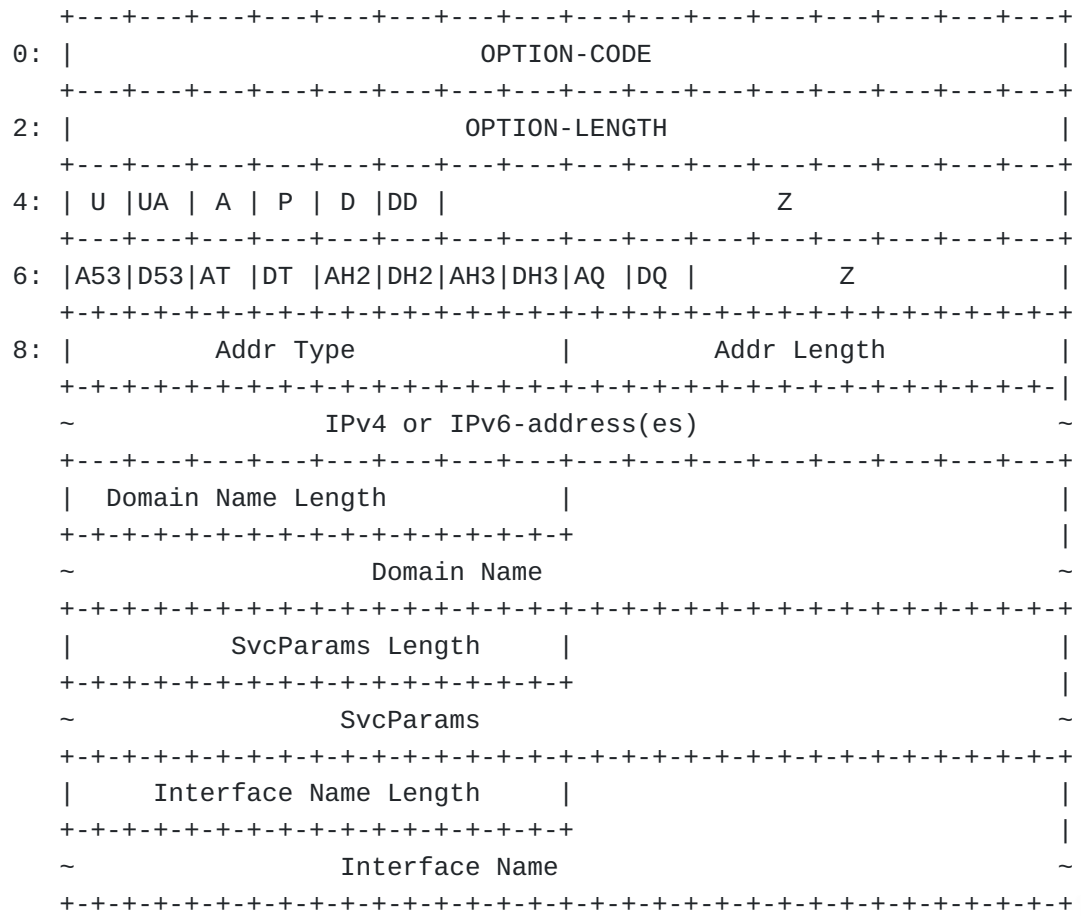
This document introduces three new EDNS(0) options, and one new response code. This first option, called PROXY CONTROL Option, specifies which transports a proxy should use to connect to a recursive resolver.

The second option, called PROXY SCOPE Option, reports the IP address scope of the connection between the application's stub resolver and the proxy.

Finally, the TRUST ANCHOR Option, provides the application with a DNSSEC trust anchor signed by IANA.

The BADPROXYPOLICY error is returned the proxy cannot meet the requirements in a PROXY CONTROL Option or the option is malformed.

5. PROXY CONTROL OPTION



where

OPTION-CODE
To be decided

OPTION-LENGTH
Length of this option excluding the OPTION-CODE and OPTION-LENGTH fields

U
force the use of unencrypted communication (Do53)

UA
require unauthenticated encryption

A
require authenticated encryption

P
authenticate using a PKIX certificate

D
authenticate using DANE

DD
by default disallow other transports (transports that are not explicitly listed)

A53,AT,AH2,AH3,AQ
allow respectively Do53, DoT, DoH H2, DoH H3, DoQ

D53,DT,DH2,DH3,DQ
disallow respectively Do53, DoT, DoH H2, DoH H3, DoQ

Z
reserved, MUST be zero when sending, MUST be ignored when received

Addr Type
Type of addresses, The value 0 if no addresses are included, the value 1 for IPv4, and the value 2 for IPv6.

Addr Length
length of the addresses in octets. Must be a multiple of 4 for IPv4 and a multiple of 16 for IPv6. This field can be zero if no addresses are specified.

IPv4 or IPv6-address(es)
list of IPv4 or IPv6 addresses

Domain Name Length
length of Domain Name. Zero if there is no Domain Name

Domain Name
   domain name for authentication or resolving IP addresses. The domain name is encoded in uncompressed DNS wire format.

SvcParams Length
   length of SvcParams. Zero if there are no service parameters specified.

SvcParams
   Service parameters

Interface Name Length
   length of Interface Name. Zero if no interface is specified.

Interface Name
   name of outgoing interface for transport connections

This option is designed to give control over what level of detail it wants to specify. The first 5 flags (U, UA, A, P, and D) give general requirements for properties of DNS transports that are used by the client proxy. The U, UA, and A flags are mutually exclusive. If more than one flag is set, the proxy SHOULD return a BADPROXYPOLICY error. There are four possibilities:

U = 0, UA = 0, A = 0 An effort is made to reach authenticated encryption, if that fails, unauthenticated encryption is tried. If that also fails, the proxy resorts to an unencrypted transport. It is an error if either or both of the P or D flags is set and the proxy SHOULD return a BADPROXYPOLICY error.

U = 1, UA = 0, A = 0 The proxy only tries only unencrypted transports. It is an error if either or both of the P or D flags is set and the proxy SHOULD return a BADPROXYPOLICY error.

U = 0, UA = 1, A = 0 An effort is made to reach authenticated encryption, if that fails, unauthenticated encryption is tried. It is an error if either or both of the P or D flags is set and the proxy SHOULD return a BADPROXYPOLICY error.

U = 0, UA = 0, A = 1 The proxy only tries authenticated encryption. The P and D flags can be set to control which authentication mechanism has to be used.

The P and D flags allow the application to require a specific authentication mechanism (PKIX or DANE). The meaning of the flags is the following:

P = 0, D = 0 At least one of the two mechanisms has to validate for authenticated encryption to succeed.

P = 1, D = 0
PKIX validation has to succeed, the status of DANE validation is ignored.

P = 0, D = 1 A DANE record has to be present and be DNSSEC valid. A DANE record has a Certificate Usage Field. For some values of this field (the values zero and one), DANE requires PKIX validation. In those cases, PKIX validation is also required according to the DANE specifications. For the values two and three, DANE does not require PKIX and because the P flag is zero, the result of PKIX validation has to be ignored.

P = 1, D = 1 Both PKIX and DANE are required together. For PKIX, this means that PKIX validation has to succeed. For DANE it means that a DANE record has to be present and be DNSSEC valid. Validation using the DANE record has to succeed.

Note that these two flags can only be used in combination with the A flag. The proxy SHOULD return a BADPROXYPOLICY error if either or both of the P or D flags is set and the A flag is clear.

The next flags provide more detailed control over which transports should be used or not. For each of 5 different transports (Do53, DoT, DoH with ALPN h2, DoH with ALPN h3, DoQ) there is a flag to allow (A53,AT,AH2,AH3,AQ) or disallow (D53,DT,DH2,DH3,DQ) the use of the transport. There is space to add more transports later. Note that setting the A flag and the D flag for a protocol (for example, setting both the A53 and the D53 flags) is not allowed and a proxy SHOULD reject such a request.

To future proof applications, there is a single flag DD, that by default disallows transports that are not explicitly listed. With this flag clear, the application allows all transports that are not explicitly disallowed (including future transports). With the flag set, the application has to explicitly list which transports can be used. For example, by setting only DD and AT, the application forces the use of DoT.

When DD = 0:

*all transports are in the pool of potentially usable transports
*D53, DT, DH2, DH3 and DQ remove those transports from the pool.
*The values of A53, AT, AH2, AH3 and AQ are irrelevant

When DD = 1:

*no transports are in the pool of potentially usable transports
*A53, AT, AH2, AH3 and AQ add those transports to the pool
*The values of D53, DT, DH2, DH3 and DQ are irrelevant

Finally, an application can specify its own resolvers or rely on the resolvers that are known to the proxy. If ADN Length and Addr Length are both zero, then the application requests the resolvers known to the proxy. [Note: it is unclear at the moment what to do with any Service Parameters]

If the application specifies only an authentication-domain-name then the proxy is expected to resolve the name to addresses. If only addresses are specified then the proxy assumes that no name is known (though a PKIX certificate may include an address literal in the subjectAltName). If both a name and address are specified then the proxy will use the specified address and use the name for authentication.

To simplify the encoding of the option, an option with addresses will have either IPv4 or IPv6 addresses. If the application wants to specify both IPv4 and IPv6 addresses for a certain authentication-domain-name then it has to include two options.

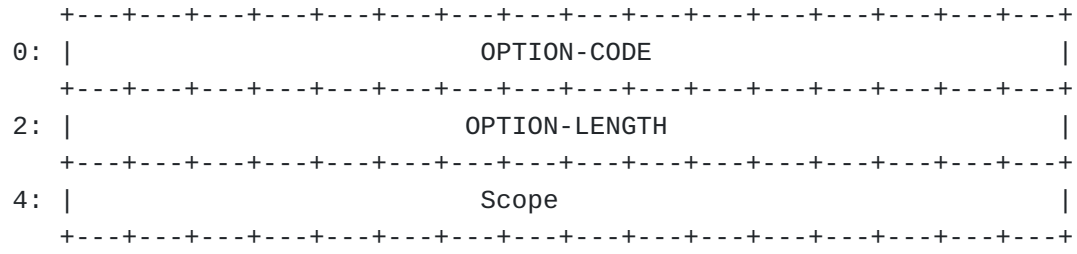
An application may want to specify a DNS resolver that is reachable through an IPv6 link-local address. IPv6 link-local addresses are special in that they require a zone to be specified, either explicitly or implicitly. Typically for a link-local address that appears as a source or destination address, the zone is implicitly the zone of the link the packet travels on. For packets that travel between hosts, there is no goed way to explictly specify the zone of a link-local address because two different hosts do not agree on zone names. However, if the proxy is on the same host as the application, then the zone identifier for the link-local address can be specified in the Interface field. For this purpose an interface name can also be an interface index expressed as a decimal string.

When present, Service Parameters specify how to connect. Otherwise it is up to the proxy to try various possibilities. For Service Parameters, the values of the ipv4hint and ipv6hint fields are ignored. Addresses can only be specified using the addresses field in the PROXY CONTROL Option.

Associated with this option is a new error, BADPROXYPOLICY. When a proxy cannot meet the requirements in a PROXY CONTROL Option or the option is malformed, it returns this error.

If the proxy returns a BADPROXYPOLICY error, the proxy MAY include a PROXY CONTROL Option that lists what the proxy can do. For example, if authenticated encryption is not possible, but unauthenticated is, then the proxy may include an option that has the UA bit set.

6. PROXY SCOPE OPTION



OPTION-CODE
To be decided

OPTION-LENGTH
Length of this option excluding the OPTION-CODE and OPTION-LENGTH fields

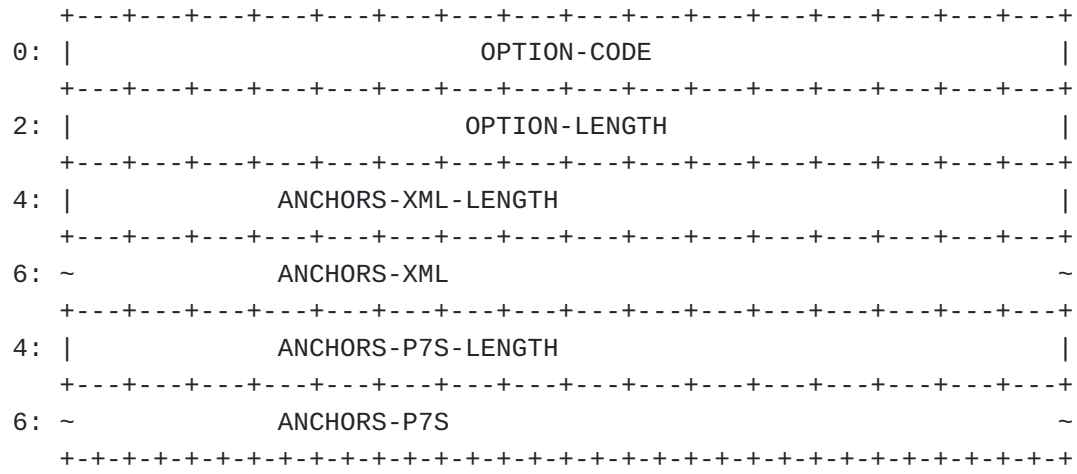
Scope
Scope of the source address of a request. Scope can have the following values:

Value	Scope
0	Undefined
1	Host local
2	Link local
3	Site local
4	Global

Table 1

The purpose of this option is to deal with proxies that forward DNS traffic without first removing any EDNS(0) options. The option requests the DNS proxy that processes the option to report the scope of the source address.

7. TRUST ANCHOR OPTION



where

OPTION-CODE
To be decided

OPTION-LENGTH
Length of this option excluding the OPTION-CODE and OPTION-LENGTH fields

ANCHORS-XML-LENGTH
Length of ANCHORS-XML in network byte order

ANCHORS-XML
Trust anchors in XML format

ANCHORS-P7S-LENGTH
Length of ANCHORS-P7S in network byte order

ANCHORS-P7S
Signature in p7s format

This option provides DNSSEC trust anchors as described in [RFC7958].

8. Protocol Specification

8.1. Client Processing

A stub resolver that wishes to use the PROXY CONTROL Option includes the option in all outgoing DNS requests that require privacy. The option should be initialized according to the needs of the application. In addition the PROXY SCOPE Option can be added. In requests, the Scope field is set to undefined.

If the stub resolver receives a reply without a PROXY CONTROL Option included in the reply, then stub resolver has to assume that traffic will have Do53 levels of privacy. Similarly, a lack of a PROXY SCOPE Option implies a global scope.

If the stub resolver receives a BADPROXYPOLICY error then the proxy was unable to meet the requirements of the PROXY CONTROL Option.

8.1.1. Probing

In cases where the stub resolver expects a local DNS proxy, or where the stub resolver has (a limited) fall back to more private transports, or when the security policy of the application is such that is better to fail than send queries over Do53, the stub resolver first sends a probing query to verify that the proxy supports the PROXY CONTROL and PROXY SCOPE Options.

This request queries "resolver.arpa" for SOA records. The proxy MUST implement this as a Special Use Domain Name. The actual response is not important. The important part is that the proxy returns PROXY CONTROL and PROXY SCOPE Options as described in this document or sets the response code to BADPROXYPOLICY if it cannot meet specified policy.

8.1.2. Trust Anchor

In the ideal case, the host operating system provides applications with a secure way to access a DNSSEC trust anchor that is maintained according to [RFC5011]. However in situations where this is not the case, an application can fall back to [RFC7958]. However, for short lived processes, there is considerable overhead in issuing two HTTP(S) requests to data.iana.org to obtain the trust anchor XML file and the signature over the trust anchor. For this reason, it makes sense to let the proxy cache this information.

If the local operating system does not provide a DNSSEC trust anchor, then the application can ask the proxy. The stub resolver adds the TRUST ANCHOR Option with ANCHORS-XML-LENGTH and ANCHORS-P7S-LENGTH set to zero. If the proxy returns both an ANCHORS-XML and an ANCHORS-P7S, then the application verifies the trust anchor using the trust anchor certificate (which needs to come with the application).

8.2. Server Processing

Proxies are encouraged to cache options that appear in requests under the assumption that a stub resolver will send multiple requests. If a proxy caches DNS responses then the proxy MUST tag cached responses with the properties of the DNS transport. When responding to later requests, the proxy returns a cached entry only if the parameters of the DNS transport match what is specified in the request.

When a proxy receives a new set of requirements, the proxy compiles a list of addresses to connect to and a list of transports to try per address. The proxy SHOULD prefer more private transports over less private ones.

If the proxy cannot obtain a connection to a recursive resolver in a way that matches the provided policy, then the proxy sets the BADPROXYPOLICY response code in the reply.

The proxy MUST implement "resolver.arpa" as a locally served zone. Proxies SHOULD respond to all queries with NODATA unless other behavior is specified in a different document.

If the proxy successfully connects to a recursive resolver and receives a reply, or the query is for a special use domain name that is handled internally in the proxy, then the proxy add a PROXY CONTROL Options dat details the connection to the recursive resolver (i.e., the U, UA, or A flag depending on encryption and authentication, P and or D for authenticated connections, A53, AT, AH2, AH3, or AQ depending on the transport (or none of those for a future transport). Furthermore the proxy includes the address it connected to, the Domain Name if known, any Service Parameters and the outgoing interface name if known.

If the proxy finds a PROXY SCOPE Option, then it calculates the scope from the source address. The proxy adds a PROXY SCOPE Option to a reply and sets the value of Scope to the actual scope of the source address of the request.

If the request contains a TRUST ANCHOR Option, then the proxy tries to fetch the trust anchor XML and p7s files if it does not have them already. If fetching one or both fails then the proxy sets the corresponding length to zero. It is not clear how long the proxy can cache this information. [RFC7958] Does not describe how long these documents can be cache. A simple solution is to take the Expires header in the HTTP reply. The proxy adds a TRUST ANCHOR Option to the reply.

9. Connection Between Stub Resolver And Proxy

Absent other configuration, a stub resolver that implements this standard SHOULD connect to the proxy using Do53 and as remote address either ::1 or 127.0.0.1. In particular, the stub resolver SHOULD avoid using name servers listed in files such as /etc/resolv.conf.

The reason for this is to simplify the integration of local DNS proxies in existing environments. If the stub resolver ignores /etc/resolv.conf then the proxy can use that information to connect to recursive resolvers.

If no DNS server is responding to queries sent using Do53 to ::1 and 127.0.0.1, or if the response indicates that this standard is not supported, then the stub resolver MAY fall back to traditional configuration methods, such as /etc/resolv.conf. However, in that case the stub resolver MUST make sure that doing so does not violate the policy set by the application.

10. Security Considerations

A privacy sensitive application SHOULD first issue a SOA query for resolver.arpa to verify that the local proxy supports the options documented in the document. If the proxy does not support this

document then the application can refrain from sending queries that reveal privacy sensitive names.

By setting the interface name, an application can select an outging interface on the proxy. Proxies should make sure that a query receives from a process that is authorized to do so. By default, a proxy SHOULD allow only process on the same host to use this feature. If an unauthorized process includes an option with the interface name set, then the proxy SHOULD return the BADPROXYPOLICY error.

11. IANA Considerations

IANA has assigned the following DNS EDNS0 option codes:

Value	Name	Status	Reference
TBD	PROXY CONTROL	Standard	RFC xxxx
TBD	PROXY SCOPE	Standard	RFC xxxx
TBD	TRUST ANCHOR	Standard	RFC xxxx

IANA has assigned the following DNS response code as an early allocation per [RFC7120]:

RCODE	Name	Description	Reference
TBD	BADPROXYPOLICY	Unable to conform to policy	RFC xxxx

12. Acknowledgements

Many thanks to Yorgos Thessalonikefs and Willem Toorop for their feedback.

13. 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>.

[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>.

14. Informative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC3493] Gilligan, R., Thomson, S., Bound, J., McCann, J., and W. Stevens, "Basic Socket Interface Extensions for IPv6", RFC 3493, DOI 10.17487/RFC3493, February 2003, <https://www.rfc-editor.org/info/rfc3493>.

[RFC5011] StJohns, M., "Automated Updates of DNS Security (DNSSEC) Trust Anchors", STD 74, RFC 5011, DOI 10.17487/RFC5011, September 2007, <https://www.rfc-editor.org/info/rfc5011>.

[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>.

[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>.

[RFC7120] Cotton, M., "Early IANA Allocation of Standards Track Code Points", BCP 100, RFC 7120, DOI 10.17487/RFC7120, January 2014, <https://www.rfc-editor.org/info/rfc7120>.

[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>.

[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>.

[RFC7958] Abley, J., Schlyter, J., Bailey, G., and P. Hoffman, "DNSSEC Trust Anchor Publication for the Root Zone", RFC

7958, DOI 10.17487/RFC7958, August 2016, <https://www.rfc-editor.org/info/rfc7958>.

[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>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

Author's Address

Philip Homburg

Email: philip@nlnetlabs.nl