Abstract

This document defines an Extension Mechanisms for DNS (EDNS0) option that allows DNS clients and servers to signal support for DNS protocol capabilities. Clients and servers that support this option can take advantage of new DNS protocol features when completing a transaction, and by caching the set of capabilities advertised by a DNS server, a DNS client can utilize any mutually supported DNS protocol capability in subsequent queries.

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 http://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 January 3, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of
The lack of explicit capability signaling in the DNS protocol [RFC1035] makes it hard to deploy new functionality. For instance, Client Subnet in DNS Queries [RFC7871] defines an EDNS option to be used with a subset of specialized zones on the Internet capable of producing "tailored responses". It describes two strategies for deciding when to originate the Client Subnet option: the use of periodic probes by the resolver, and the use of a safelist of nameservers permitted by the resolver operator to use the option. In practice, few EDNS options have been defined, and EDNS options have not been originated routinely by general purpose resolvers on the Internet. If many EDNS options were to be widely used, it would be unreasonable to expect resolver implementations to perform option-specific probing or resolver operators to perform option-specific safelisting for each newly introduced EDNS option.

EDNS options are not the only aspect of the DNS protocol that can benefit from explicit capability signaling. Extension Mechanisms for DNS (EDNS(0)) [RFC6891] includes a VERSION field in the OPT Pseudo-RR, but encourages clients to set this field to the "lowest implemented level capable of expressing a transaction, to minimize the responder and network load of discovering the greatest common
implementation level between requestor and responder". If new EDNS VERSIONs were to be introduced, capability signaling would permit a DNS transaction initiated with a lower implementation level to be completed with the highest mutually supported implementation level.

Similarly, Q and Meta-TYPEs [RFC6895] (Section 3.1) have been allocated sparingly. Introducing a new general purpose QTYPE is problematic, because by definition the existing installed base of nameservers will not support the new QTYPE.

This document defines an EDNS0 option that allows DNS clients and servers to exchange lists of supported "DNS Capabilities". This new option includes explicit client-side caching semantics that allow future queries to be initiated that take advantage of mutually supported functionality. It also defines two DNS Capabilities. The first, "DNS Features", encodes an array of feature flags that future DNS protocol features may take advantage of. The second, "EDNS0 Option Codes", encodes the set of EDNS0 options supported by the client or server.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Overview

A DNS client that implements this protocol will include the "DNS Capabilities" EDNS0 option described in Section 4 in queries that it initiates (Section 5.1). If a DNS server that implements this protocol receives a query that includes this option, it will generate a corresponding response (Section 5.2) indicating which DNS Capabilities it supports and the length of time that the client may cache the server’s capabilities for (Section 5.3).

4. Option Format

This protocol uses an EDNS0 [RFC6891] option to encode the capabilities supported by the client or server. For each capability, the option contains a TLV element <Capability Type, Capability Length, Capability Value>. Multiple Capability TLVs may be concatenated together, and the ordering of Capability TLVs within the option is not significant. The option is structured as follows:
Internet-Draft              DNS Capabilities                   July 2017

+0 (MSB)                            +1 (LSB)
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
0: |                          OPTION-CODE                          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
2: |                         OPTION-LENGTH                         |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
4: |                      OPTION-TTL-MINUTES                       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
6: |        CAPABILITY TYPE        |       CAPABILITY LENGTH       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
8: | |                                                        |
| / CAPABILITY VALUE...     |
| / /                       |
| / (Additional Capability TLVs...)                |
| / /                                                        |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

| o (Defined in [RFC6891]) OPTION-CODE, 2 octets, for the DNS Capabilities option is TBD-DNS-CAPABILITIES-OPT.
| o (Defined in [RFC6891]) OPTION-LENGTH, 2 octets, contains the length of the payload (everything after OPTION-LENGTH) in octets.
| o OPTION-TTL-MINUTES, 2 octets, unsigned, indicates the number of minutes clients may cache this DNS Capabilities option.
| o CAPABILITY TYPE, 1 octet, identifies the Capability encoded in the Capability TLV, using codes as assigned in TBD-DNS-CAPABILITIES-TYPE-REGISTRY.
| o CAPABILITY LENGTH, 1 octet, unsigned, encodes the number of octets in the CAPABILITY VALUE field.
| o CAPABILITY VALUE, variable number of octets, contains capability-specific data.

The format of the CAPABILITY VALUE field depends on the value of the CAPABILITY TYPE field. This document defines the following CAPABILITY TYPE values:
<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Singleton</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reserved</td>
<td></td>
<td>This document</td>
</tr>
<tr>
<td>1</td>
<td>DNS Features</td>
<td>Y</td>
<td>Section 4.1</td>
</tr>
<tr>
<td>2</td>
<td>EDNS0 Option Codes</td>
<td>Y</td>
<td>Section 4.2</td>
</tr>
<tr>
<td>3-249</td>
<td>Unassigned</td>
<td></td>
<td>This document</td>
</tr>
<tr>
<td>250-254</td>
<td>Reserved for Local/Experimental Use</td>
<td></td>
<td>This document</td>
</tr>
<tr>
<td>255</td>
<td>Reserved</td>
<td></td>
<td>This document</td>
</tr>
</tbody>
</table>

CAPABILITY TYPE values.

The "DNS Features" and "EDNS0 Option Codes" capabilities are further described below.

4.1. The "DNS Features" Capability

The "DNS Features" capability is a variable length array of feature flags encoded as a bitmap with trailing zero octets omitted.

New DNS protocol functionality that requires upgraded semantics may register a flag in this capability. DNS clients and servers that support a new protocol feature with a feature flag in this capability MUST set the corresponding bit in this capability in queries and responses when those messages include a DNS Capabilities option.

Each feature flag is assigned an individual bit in the "DNS Features" bitmap. For example, the first feature flag corresponds to the Most Significant Bit (MSB) of the first octet of the array, the ninth feature flag corresponds to the MSB of the second octet of the array, and the 256th feature flag corresponds to the Least Significant Bit (LSB) of the 32nd octet of the array.

If no "DNS Features" flags are supported by the DNS client or server, this capability MUST be omitted from the DNS Capabilities option. Trailing zero octets in the "DNS Features" bitmap MUST be omitted. The minimum CAPABILITY LENGTH value for this capability is 1, and the maximum CAPABILITY LENGTH value for this capability is 32.
<table>
<thead>
<tr>
<th>Bit</th>
<th>Flag</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-249</td>
<td></td>
<td>Unassigned</td>
<td></td>
</tr>
<tr>
<td>250-254</td>
<td></td>
<td>Reserved for Local/Experimental Use</td>
<td>This document</td>
</tr>
<tr>
<td>255</td>
<td></td>
<td>Reserved</td>
<td>This document</td>
</tr>
</tbody>
</table>

"DNS Features" flags.

The "DNS Features" capability is a singleton capability. It MUST NOT appear more than once in a DNS Capabilities option.

4.2. The "EDNS0 Option Codes" Capability

[ RFC4034 ] ( Section 4.1.2 ) defines the Type Bit Maps field of the NSEC RR using a sparse encoding of the 16-bit code points in the DNS Resource Record (RR) TYPES registry. This document reuses that "window block" bitmap encoding for the "DNS EDNS0 Option Codes (OPT)" registry, which is also a 16-bit code space.

The "EDNS0 Option Codes" capability is a window block encoded bitmap indicating which EDNS0 Option Codes are supported by the DNS client or server. If the "EDNS0 Option Codes" capability is included in the DNS Capability option in a DNS message, the EDNS0 Option Codes supported by the DNS client or server SHOULD be indicated using this capability.

The EDNS0 Option Codes space is split into 256 window blocks, each representing the low-order 8 bits of the 16-bit code space. Each block that has at least one indicated EDNS0 Option Code is encoded using a single octet window number (from 0 to 255), a single octet bitmap length (from 1 to 32) indicating the number of octets used for the window block’s bitmap, and up to 32 octets (256 bits) of bitmap.

Window blocks are present in the "EDNS0 Option Codes" capability in increasing numerical order.

Each bitmap encodes the low-order 8 bits of EDNS0 Option Codes within the window block. The first bit is bit 0. For window block 0, bit 3 corresponds to NSID [ RFC5001 ], bit 8 corresponds to EDNS Client Subnet [ RFC7871 ], and so forth. If a bit is set, it indicates that the corresponding EDNS0 Option Code is supported by the DNS protocol implementation that sent the message containing the "EDNS0 Option Codes" capability.
Window blocks with no EDNS0 Option Codes present MUST NOT be included. Trailing zero octets in the bitmap MUST be omitted. The length of each block’s bitmap is determined by the option code with the largest numerical value, within that block, among the set of option codes indicated as supported. Trailing zero octets not specified MUST be interpreted as zero octets.

The "EDNS0 Option Codes" capability is a singleton capability. It MUST NOT appear more than once in a DNS Capabilities option.

5. Protocol Description

5.1. Originating the Option

A DNS client that implements this protocol SHOULD include the DNS Capabilities option in each EDNS(0) enabled query it sends. If the DNS client supports any DNS Features (Section 4.1), it MUST include a DNS Features capability in the DNS Capabilities option advertising the supported features. If the DNS client supports any EDNS0 Option Codes, it MAY include an EDNS0 Option Codes capability in the DNS Capabilities option advertising the supported option codes.

DNS clients MUST set the OPTION-TTL-MINUTES field to zero.

5.2. Generating a Response

When a query containing the DNS Capabilities option is received, a DNS server supporting DNS Capabilities MAY use the information contained in the client’s option to generate a response that utilizes the functionality that the client has advertised as supported.

A DNS server that implements this protocol and receives a DNS Capabilities option MUST include a DNS Capabilities option in its response. If the DNS server implements any DNS Features (Section 4.1), it MUST include a DNS Features capability in the DNS Capabilities option advertising the supported features. If the DNS server supports any EDNS0 Option Codes, it MUST include an EDNS0 Option Codes capability in the DNS Capabilities option advertising the supported option codes.

If the DNS Capabilities option was not included in a query, a DNS server MUST NOT include one when generating a response.

5.3. Caching the Option

When a DNS client originates a query containing the DNS Capabilities option and receives a response containing the DNS Capabilities option, the data contained in the option SHOULD be cached so that
future queries sent to the same DNS server can utilize functionality that the server has advertised as supported. A query sent to "the same DNS server" means a query sent to a server identified by the same network address as a previous query.

The amount of time that the DNS Capabilities option may be cached for is indicated in the OPTION-TTL-MINUTES field. This allows a maximum cache entry lifetime of 45.5 days. If a DNS Capabilities option for a given DNS server is already cached when a subsequent response containing a DNS Capabilities option is received, the cached option MAY be overwritten with the newer data, and the cache entry’s lifetime MAY be extended or reduced.

If a DNS client receives a response containing the DNS Capabilities option where the OPTION-TTL-MINUTES field is zero, the data contained in the option MUST be discarded, and the response MUST be treated as if it did not contain a DNS Capabilities option.

6. IANA Considerations
   To be written.

7. Implementation Status
   To be written.

8. Security Considerations
   To be written.

9. Acknowledgements
   To be written.

10. TODO

   1. There is a limited amount of space available in a UDP DNS query message (512 octets), and a query message with a maximum size query name (255 octets) and a plausible set of other EDNS0 options could be as much as ~300 octets, leaving ~200 octets for the DNS Capabilities option. What should happen if all of the desired DNS Capabilities data can’t be serialized into a <= 512 octet query message?
11. References

11.1. Normative References


11.2. Informative References


Author’s Address

Robert Edmonds
Fastly
Atlanta, Georgia
United States of America

Email: edmonds@mycre.ws