DNSOP Working Group R. Bellis
Internet-Draft ISC
Intended status: Standards Track S. Cheshire
Expires: January 7, 2017 Apple Inc.
J. Dickinson
Sinodun
A. Mankin
T. Pusateri
Unaffiliated
July 6, 2016

DNS Session Signaling
draft-bellis-dnsop-session-signal-00

Abstract

The Extension Mechanisms for DNS (EDNS(0)) [RFC6891] is explicitly defined to only have "per-message" semantics. This document defines a new Session Signaling OpCode used to carry persistent "per-session" type-length-values (TLVs), and defines an initial set of TLVs used to handle feature negotiation and to manage session timeouts and termination.

Table of Contents

1. Introduction

The Extension Mechanisms for DNS (EDNS(0)) [RFC6891] is explicitly defined to only have "per-message" semantics. This document defines a new Session Signaling OpCode used to carry persistent "per-session" type-length-values (TLVs), and defines an initial set of TLVs used to handle feature negotiation and to manage session timeouts and termination.

A further issue with EDNS(0) is that there is no standard mechanism for a client to be able to tell whether a server has processed or otherwise acted upon the individual options contained with an OPT RR. The Session Signaling Opcode therefore requires an explicit response to each TLV within a request.

The message format (see Section 3.1) does not completely conform to the standard DNS packet format but is designed such that existing DNS

protocol parsers should be able to read the packet header and then simply ignore the extra data that appears thereafter.

2. Terminology

The terms "initiator" and "responder" correspond respectively to the initial sender and subsequent receiver of a Session Signaling TLV, regardless of which was the "client" and "server" in the usual DNS sense. The term "sender" may apply to either an initiator or responder.

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 [RFC2119].

3. Protocol Details

Session Signaling messages MUST only be carried in protocols and in environments that can guarantee that the same two endpoints are in communication for the entire lifetime of the session.

Session Signaling messages relate only to the specific session in which they are being carried. Where a middle box (e.g. a DNS proxy, forwarder, or session multiplexer) is in the path the message MUST NOT be blindly forwarded in either direction by that middle box. This does not preclude the use of these messages in the presence of a NAT box that rewrites Layer 3 or Layer 4 headers but otherwise maintains the effect of a single session.

<< RB: OSI Layer 5 session analog? This is obviously intended for TCP "sessions" which aren't distinct from Layer 4, but is this also applicable to DNS-o-DTLS, or DNS over UDP with an EDNS cookie - I think probably "yes" for the former, but "no" for the latter. I'm wondering whether "session" is even the right term to be using here >>

3.1. Message Format

A message containing a Session Signaling Opcode does not conform to the usual DNS message format. The 12 octet header format from [RFC1035] is preserved, but the four section count fields (QDCOUNT, ANCOUNT, NSCOUNT and ARCOUNT) MUST all be set to zero.

A list of TLVs are used in place of the usual sections, and MUST appear immediately after the 12 octet header. The total size of the TLVs is calculated from the value of the standard two octet framing word minus the 12 octets of the DNS header.

3.2. Message Handling

Both clients and servers may unilaterally send Session Signaling messages at any point in the lifetime of a session and are therefore considered to be the initiator with respect to that message. The initiator MUST set the value of the QR bit in the DNS header to zero (0), and the responder MUST set it to one (1).

Every Session Signaling request message MUST elicit a response (which MUST have the same ID in the DNS message header as in the request) and every TLV contained within the request requires a corresponding TLV in the response.

In order to preserve the correct sequence of state, Session Signaling requests MUST NOT be processed out of order. Similarly the TLVs in a message MUST be processed in the order in which they are contained in the message, and the order of the TLVs in the response MUST correspond with the order of the TLVs in the request.

<< RB: should the presence of a SS message create a "sequencing point", such that all pending responses must be answered? >>

3.3. TLV Format

```
                                          1   1   1   1   1   1
      0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |SESSION-STATUS |         SESSION-TYPE                          |
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |                        SESSION-LENGTH                         |
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |                                                               |
    /                         SESSION-DATA                          /
    /                                                               /
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

SESSION-STATUS: A 4 bit field used in a response to indicate the success (or otherwise) of an operation, as defined in the DNS Session Signaling Status Codes Registry. It SHOULD contain "NOERROR" (0) in a request message but the responder MUST NOT reject the request if it does not.

SESSION-TYPE: A 12 bit field in network order giving the type of the current Session Signaling TLV per the IANA DNS Session Signaling Type Codes Registry.

SESSION-LENGTH: A 16 bit field in network order giving the size in octets of SESSION-DATA.

SESSION-DATA: Type-code specific. The SESSION-DATA field MUST be NUL padded to an even number of octets such that each Session Signaling TLV is aligned on a two octet boundary relative to the start of the first Session Signaling TLV. Padding octets MUST NOT be included in the calculation of SESSION-LENGTH but MUST be included in the calculation of the overall message length.

<< RB: the padding is specified such that client code can read the type and length fields directly from an aligned uint16_t array (with byte swapping) >>

4. Mandatory TLVs

4.1. Feature Negotiation

4.1.1. TypeCode Support

The TypeCode Support TLV (1) is used to allow a client and server to exchange information about which Session Signaling Type Codes they support.

The SESSION-DATA contains a list of the Session Signaling Type Codes supported by the sender.

```
                                          1   1   1   1   1   1
      0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |                           TYPE CODEs                          |
    /                              ...                              /
    /                                                               /
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

TYPE CODEs: A list of 16 bit words in network order comprising the complete list of Session Signaling Type Codes supported by the sender. Since a Session Signaling Type Code is in reality only a 12 bit value, the four most significant bits of each word MUST be zero. The number of TYPE CODEs can be calculated from the total length of the TLV.

An initiator MAY send its own list of supported Session Signaling Type Codes in a TypeCode Support TLV, and if sent they MUST be complete. Otherwise the SESSION-DATA MUST be empty. In either case the responder MUST respond with its complete list of supported Type Codes.

4.2. Layer 4 Connection Management TLVs

4.2.1. Terminate

The Terminate TLV (64) MAY be sent by a server to request that the client terminate the session, and when sent MUST be the only TLV present. It MUST NOT be requested by a client.

The client SHOULD terminate the session as soon as possible, but MAY wait for any inflight queries to be answered. It MUST NOT initiate any new queries over the existing session, nor send any further TLVs other than its response to the Terminate request.

<< RB: dns-sd push has a "reconnect delay" option but I think it's of questionable value since in an anycast or load-balancing architecture there's no way for the client to know which instance sent the option nor control which server instance the next connection will go to. This would IMHO be better controlled directly at the TCP layer. >>

4.2.2. Idle Timeout

The Idle Timeout TLV (65) has similar semantics to the EDNS TCP Keepalive Option [RFC7828]. It is used by a server to tell the client how long it may leave the current session idle for.

The SESSION-DATA is as follows:

```
                                          1   1   1   1   1   1
      0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |                         IDLE TIMEOUT                          |
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

IDLE TIMEOUT: the idle timeout for the current session, specified as a 16 bit word in network order in units of 100 milliseconds.

It is NOT an error for this TLV and the similar EDNS option to appear within the same session. The client SHOULD pay attention to the most recently received value, regardless of which method was used to send it.

The client SHOULD terminate the current session if it remains idle for longer than the specified timeout (and MAY of course terminate the session earlier). The server MAY unilaterally terminate the connection at any time, but SHOULD allow the client to keep the connection open if further messages are received before the idle timeout expires.

<< RB: this assumes that the EDNS OPT RR is added at the final stage of message processing, and therefore not affected by out-of-order processing - c.f. comment above about sequencing points >>

5. IANA Considerations

5.1. DNS Session Signaling OpCode Registration

IANA are directed to assign the value TBD for the Session Signaling OpCode in the DNS OpCodes Registry.

5.2. DNS Session Signaling Status Codes Registry

IANA are directed to create the DNS Session Signaling Status Codes Registry, with initial values as follows:

Code	Mnemonic	Description	Reference
0	NOERROR	TLV processed successfully	RFC-TBD1
4	NOTIMP	TLV not implemented	RFC-TBD1
5	REFUSED	TLV declined for policy reasons	RFC-TBD1

Registration of additional Session Signaling Status Codes requires Standards Action.

5.3. DNS Session Signaling Type Codes Registry

IANA are directed to create the DNS Session Signaling Type Codes Registry, with initial values as follows:

Type	Name	Status	Reference
0	Reserved		RFC-TBD1
1	TypeCode Support	Standard	RFC-TBD1
2 - 63	Unassigned, reserved for feature negotiation TLVs		
64	Terminate	Standard	RFC-TBD1
65	Idle Timeout	Standard	RFC-TBD1
66 - 127	Unassigned, reserved for session management TLVs		
127 - 3965	Unassigned		
3968 - 4031	Reserved for local / experimental use		
4032 - 4095	Reserved for future expansion		

Registration of additional Session Signaling Type Codes requires Expert Review. << RB: definition of process required? >>

6. Security Considerations

The authors are not aware of any specific security considerations introduced by this specification at this time.

7. Acknowledgements

TBW

8. Normative References

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

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <http://www.rfc-editor.org/info/rfc6891>.

[RFC7828] Wouters, P., Abley, J., Dickinson, S., and R. Bellis, "The edns-tcp-keepalive EDNS0 Option", RFC 7828, DOI 10.17487/RFC7828, April 2016, <http://www.rfc-editor.org/info/rfc7828>.

Authors' Addresses

Ray Bellis
Internet Systems Consortium, Inc.
950 Charter Street
Redwood City CA 94063
USA

Phone: +1 650 423 1200
Email: ray@isc.org

Stuart Cheshire
Apple Inc.
1 Infinite Loop
Cupertino CA 95014
USA

Phone: +1 408 974 3207
Email: cheshire@apple.com

John Dickinson
Sinodun Internet Technologies
Magadalen Centre
Oxford Science Park
Oxford OX4 4GA
United Kingdom

Allison Mankin
Unaffiliated

Email: allison.mankin@gmail.com

Tom Pusateri
Unaffiliated

Phone: +1 843 473 7394
Email: pusateri@bangj.com