

dnsop  
Internet-Draft  
Intended status: Standards Track  
Expires: January 4, 2018

W. Kumari  
Google  
Z. Yan  
CNNIC  
W. Hardaker  
USC/ISI  
D. Lawrence  
Akamai Technologies  
July 3, 2017

Returning extra answers in DNS responses.  
draft-wkumari-dnsop-multiple-responses-05

## Abstract

This document (re)introduces the ability to provide multiple answers in a DNS response. This is especially useful as, in many cases, the entity making the request has no a priori knowledge of what other questions it will need to ask.

## 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 4, 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 Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">2</a>
<a href="#">1.1.</a>	Requirements notation . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Background . . . . .	<a href="#">3</a>
<a href="#">3.</a>	Terminology . . . . .	<a href="#">4</a>
<a href="#">4.</a>	Returning multiple answers . . . . .	<a href="#">4</a>
<a href="#">5.</a>	The EXTRA Resource Record . . . . .	<a href="#">5</a>
<a href="#">5.1.</a>	File Format . . . . .	<a href="#">5</a>
<a href="#">5.2.</a>	Wire Format . . . . .	<a href="#">5</a>
<a href="#">6.</a>	Signaling support . . . . .	<a href="#">6</a>
<a href="#">7.</a>	Stub-Resolver Considerations . . . . .	<a href="#">6</a>
<a href="#">8.</a>	Use of Additional information . . . . .	<a href="#">6</a>
<a href="#">9.</a>	IANA Considerations . . . . .	<a href="#">6</a>
<a href="#">10.</a>	Security Considerations . . . . .	<a href="#">7</a>
<a href="#">11.</a>	Acknowledgements . . . . .	<a href="#">7</a>
<a href="#">12.</a>	Normative References . . . . .	<a href="#">7</a>
<a href="#">Appendix A.</a>	Changes / Author Notes. . . . .	<a href="#">8</a>
	Authors' Addresses . . . . .	<a href="#">9</a>

## [1.](#) Introduction

In many cases a name being resolved in the DNS provides the reason behind why the name is being resolved. This may allow the authoritative nameserver to predict what other answers a recursive resolver will soon query for. By providing multiple answers in the response, the authoritative name server operator can assist a caching recursive resolver in pre-populating its cache before a stub resolver or other client asks for the subsequent queries. Apart from decreasing the latency for end users [[RFC6555](#)], this also decreases the total number of queries that the recursive resolver needs to send and the authoritative server needs to answer.

For example, the domain name administrator of Example Widgets, Inc (example.com) knows that the web page at [www.example.com](#) contains various other resources, including some images (served from [images.example.com](#)), some Cascading Style Sheets (served from [css.example.com](#)) and some JavaScript (served from [data.example.com](#)).

An application attempting to resolve `www.example.com` is very likely to be a web browser rendering the page and will likely also need to resolve all of these additional names as well. Providing all of these answers in response to a query for `www.example.com` allows the recursive resolver to pre-populate its cache and have these answers

available immediately when a stub resolver or other DNS client asks for them. What is important to notice here is that the stub resolver does not know what other questions it will need to make until after it has already made the request for `www.exmaple.com`, received the reply, made the HTTP connection and parsed the HTML.

Other examples where this technique may be useful include SMTP (by including mail server addresses, SPF and DKIM records when serving the MX record), SRV (by providing the target information in addition to the SRV response) and TLSA (by providing any TLSA records associated with a name). This same technique can also be used to include both the IPv4 (A) and IPv6 (AAAA) addresses for any singular address query.

This technique, described in this document, is purely an optimization and enables a zone publisher to distribute other related answers that the client is likely to need along with an answer to the original request. Users get a better experience, recursive resolvers need to send less queries, authoritative servers have to answer fewer queries, etc.

### 1.1. Requirements notation

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](#)].

## 2. Background

The existing DNS specifications [[RFC1034](#)] allow for supplemental information to be included in the "additional" section of the DNS response, but in order to defeat cache poisoning attacks most implementations either ignore or don't trust additional records they didn't ask for. For more background, see [[Ref.Bellovin](#)] and [[RFC2181](#)].

Not trusting the information in the additional section was necessary since there was no way to authenticate it. If a resolver queries for `www.example.com` and received answers for `www.invalid.com` as well, it is impossible for a non-validating resolver to tell if these were actually from `invalid.com` or if an attacker was trying to push bad information for `invalid.com` into the resolver's cache. In a world of ubiquitous DNSSEC deployment [Ed note: By the time this document is published, there *will* be ubiquitous DNSSEC :-) ], a validating resolver can validate, authenticate and trust the records in the additional information.

### [3.](#) Terminology

**Additional records** Additional records are records that the authoritative nameserver has included in the Additional section.

**EXTRA Resource Record** The EXTRA resource record (defined below) carries a list of additional records to send.

**Primary query** A Primary query (or primary question) is a QNAME that the name server operator would like to return additional answers for.

**Supporting DNSSEC information** Supporting DNSSEC information is the DNSSEC RRSIGs that prove the authenticity of the Additional records.

**Stub Resolver** The term "Stub Resolver" is used in this document to refer to the most common instance of a DNS client sending DNS requests to a Recursive Resolver. However, other DNS clients are not excluded from these usages and where we write "Stub Resolver" you may read it as "Stub Resolver or other DNS client".

### [4.](#) Returning multiple answers

The authoritative nameserver should include as many of the instructed additional records identified by the Extra Resource Record and Supporting DNSSEC information as will fit in the response packet. These additional records (and Supporting DNSSEC information) are appended to the additional section of the response.

In order to include additional records in a response, these conditions need to be met:

1. Additional records **MUST** only be included when the Name Server is authoritative for the zone, and the records to be returned are DNSSEC signed.
2. The supporting DNSSEC information necessary to perform validation on the records **MUST** be included.
3. The Authoritative Name Server **SHOULD** include as many of the additional records as will fit in the response. Additional records **SHOULD** be inserted in the order specified in the Additional records list.
4. Zone administrators **SHOULD** only include records identified in the EXTRA Resource Records that they expect a client to need.

## [5.](#) The EXTRA Resource Record

To allow a zone content administrator to instruct the name server which additional records to serve when it receives a query to a label, we introduce the EXTRA Resource Record (RR). These additional records are appended to the additional section (note that the EXTRA RR itself is not appended). The EXTRA resource record **MAY** still be queried for directly (e.g for debugging), in which case the record itself is returned.

### [5.1.](#) File Format

The format of the EXTRA RR is:

```
label EXTRA "label,type; label,type; label,type; ..."
```

For example, if the operator of example.com would like to also return A record answers for images.example.com, css.html.example.com and both an A and AAAA for data.example.com when queried for www.example.com, they would create the following record:

```
www.example.com.  EXTRA "images,A; css,A; data,A; data,AAAA;"
```

The entries in the EXTRA list are ordered. An authoritative nameserver SHOULD insert the records in the order listed when filling the response packet. This is to allow the operator to express a preference in case all the records will not fit in the response. The TTL of the records added to the Additional section are MUST be the same as if queried directly.

In some cases a zone content administrator might not know what all additional records clients need. For example, the owner of `www.example.com` may have outsourced his DNS operations to a third party, and / or the DNS operator might not interact with the web development team. DNS server operators may use tools to mine their query logs for records to include. For example, if, in a large majority of cases, a recursive server asks for `foo.example.com` and then very soon after asks for `bar.example.com`, it may make sense to optimize this by opportunistically returning `bar` when queried for `foo`. This functionality could also be included in the authoritative name server software itself. The exact mechanisms and heuristics used for this are not discussed in this document.

## 5.2. Wire Format

The wire format of the EXTRA RR is the same as the wire format for a TXT RR:

```
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                               TXT-DATA                               /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Where TXT-DATA is one or more <character-string>s.

The EXTRA RR has RR type TBD [RFC Editor: insert the IANA assigned value and delete this note]

## 6. Signaling support

Recursive Resolvers (or other DNS clients) that support EXTRA records MAY signal this by setting the OPT record's EXTRA bit (bit NN [TBD: assigned by IANA] in the EDNS0 extension header to 1).

## [7.](#) Stub-Resolver Considerations

No modifications need to be made to stub-resolvers to get the predominate benefit of this protocol, since the majority of the speed gain will take place between the validating recursive resolver and the authoritative name server. However, stub resolvers may choose to support this technique, and / or may query directly for the EXTRA RR if it wants to pre-query for data that will likely be needed in the process of supporting applications.

## [8.](#) Use of Additional information

When deciding to use additional records in the additional section, a resolver must follow certain rules:

1. Additional records **MUST** be validated before being used.
2. Additional records **SHOULD** have lower priority in the cache than answers received because they were requested. This is to help evict Additional records from the cache first (to help prevent cache filling attacks).
3. Recursive resolvers **MAY** choose to ignore Additional records for any reason, including CPU or cache space concerns, phase of the moon, etc. It may choose to accept all, some or none of the Additional records.

## [9.](#) IANA Considerations

This document contains the following IANA assignment requirements:

1. The EXTRA bit discussed in [Section 6](#) needs to be allocated. [ Ed: This section to be completed later ]

## [10.](#) Security Considerations

Additional records will make DNS responses even larger than they are currently, leading to larger records that can be used in DNS reflection attacks. One could mitigate this by only serving responses to EXTRA requests over TCP or when using Cookies [[RFC5395](#)], although there is no easy way to signal this to a client other than through the use of the truncate bit.

A malicious authoritative server could include a large number of extra records (and associated DNSSEC information) and attempt to DoS the recursive by making it do lots of DNSSEC validation. However, this is not considered a realistic threat; CPU for validation is cheap compared to bandwidth. This can be mitigated by allowing the recursive resolver to ignore Additional records whenever it considers itself under attack or its CPU resources are otherwise over-committed.

This specification requires that the all of the Additional records are signed, and all necessary DNSSEC information for validation be included to avoid cache poisoning attacks.

## 11. Acknowledgements

The authors to thank Mark Andrews, John Dickinson, Kazunori Fujiwara, Bob Harold, John Heidemann, and Tony Finch. The authors apologize in advance for others who contributed, but who we managed to forget.

## 12. Normative References

[Ref.Bellovin]

Bellovin, S., "Using the Domain Name System for System Break-Ins", 1995,  
<<https://www.cs.columbia.edu/~smb/papers/dnshack.pdf>>.

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

[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>>.

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", [RFC 2181](#), DOI 10.17487/RFC2181, July 1997,  
<<http://www.rfc-editor.org/info/rfc2181>>.



Considerations", [RFC 5395](#), DOI 10.17487/RFC5395, November 2008, <<http://www.rfc-editor.org/info/rfc5395>>.

[RFC6555] Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", [RFC 6555](#), DOI 10.17487/RFC6555, April 2012, <<http://www.rfc-editor.org/info/rfc6555>>.

[RFC7873] Eastlake 3rd, D. and M. Andrews, "Domain Name System (DNS) Cookies", [RFC 7873](#), DOI 10.17487/RFC7873, May 2016, <<http://www.rfc-editor.org/info/rfc7873>>.

## [Appendix A](#). Changes / Author Notes.

[RFC Editor: Please remove this section before publication ]

From -04 to -05:

- o In the deadline rush, Warren forgot to add Tale. Fixed.
- o Some more text fixups and clarifications.

From -03 to -04:

- o Some additional text explaining how this differs from solutions which include multiple queries (you don't know what to ask until you have received some answers).

From -02 to -03:

- o Sat down and rewrote and cleaned up large sections of text.
- o Changed name of RR from Additional to EXTRA (the term "Additional" is overloaded in general)
- o Clarified that stub resolvers and other clients MAY use this specification.
- o Attempted to clarify that the individual RRs are added to the response, not the EXTRA record itself. The EXTRA RR can be queried directly.

From -00 to -01:

- o Nothing change in the template.

Authors' Addresses

Warren Kumari  
Google  
1600 Amphitheatre Parkway  
Mountain View, CA 94043  
US

Email: warren@kumari.net

Zhiwei Yan  
CNNIC  
No.4 South 4th Street, Zhongguancun  
Beijing 100190  
P. R. China

Email: yanzhiwei@cnnic.cn

Wes Hardaker  
USC/ISI  
P.O. Box 382  
Davis, CA 95617  
US

Email: ietf@hardakers.net

David C Lawrence  
Akamai Technologies  
150 Broadway  
Cambridge, MA 02142-1054  
US

Email: tale@akamai.com

