

Network Working Group  
Internet-Draft  
Expires: November 23, 2013

M. Andrews  
ISC  
May 22, 2013

A Common Operational Problem in DNS Servers - Failure To Respond.  
draft-andrews-dns-no-response-issue-01.txt

## Abstract

The DNS is a query / response protocol. Failure to respond to queries causes both immediate operational problems and long term problems with protocol development.

This document will identify a number of common classes of queries that some servers fail to respond too. This document will also suggest procedures for TLD and other similar zone operators to apply to reduce / eliminate the problem.

## 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 November 23, 2013.

## Copyright Notice

Copyright (c) 2013 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

1. Introduction . . . . .	3
2. Common queries class that result in non responses. . . . .	3
2.1. EDNS Queries . . . . .	3
2.2. Unknown / Unsupported Type Queries . . . . .	4
2.3. TCP Queries . . . . .	4
3. Remediating . . . . .	4
4. Firewalls and Load Balancers . . . . .	6
5. Normative References . . . . .	6
Author's Address . . . . .	6

## 1. Introduction

The DNS [RFC1034], [RFC1035] is a query / response protocol. Failure to respond to queries causes both immediate operational problems and long term problems with protocol development.

Failure to respond to a query is indistinguishable from a packet loss without doing a analysis of query response patterns and results in unnecessary additional queries being made by DNS clients and unnecessary delays being introduced to the resolution process.

Due to the inability to distinguish between packet loss and nameservers dropping EDNS queries, packet loss is sometimes misclassified as lack of EDNS support which can lead to DNSSEC validation failures.

Allowing servers which fail to respond to queries to remain in the DNS hierarchy for extended periods results in developers being afraid to deploy new type codes. Such servers need to be identified and corrected / replaced.

The DNS has response codes that cover almost any conceivable query response. A nameserver should be able to respond to any conceivable query using them.

Unless a nameserver is under attack, it should respond to all queries directed to it as a result of following delegations. Additionally code should not assume that there isn't a delegation to the server even if it is not configured to serve the zone. Broken delegation are a common occurrence in the DNS and receiving queries for zones that you are not configured for is not a necessarily a indication that you are under attack.

## 2. Common queries class that result in non responses.

There are three common query class that result in non responses today. These are EDNS [RFC2671] queries, queries for unknown (unallocated) or unsupported types and filtering of TCP queries.

### 2.1. EDNS Queries

Identifying servers that fail to respond to EDNS queries can be done by first identifying that the server responds to regular DNS queries then making a series otherwise identical responses using EDNS, then making the original query again. A series of EDNS queries is needed as at least one DNS implementation responds to the first EDNS query with FORMERR but fails to respond to subsequent queries from the same

address for a period until a regular DNS query is made. The EDNS query should specify a UDP buffer size of 512 bytes to avoid false classification of not supporting EDNS due to response packet size.

If the server responds to the first and last queries but fails to respond to most or all of the EDNS queries it is probably faulty. The test should be repeated a number of times to eliminate the likelihood of a false positive due to packet loss.

Firewalls may also block larger EDNS responses but there is no easy way to check authoritative servers to see if the firewall is misconfigured.

## 2.2. Unknown / Unsupported Type Queries

Identifying servers that fail to respond to unknown or unsupported types can be done by making a initial DNS query for a A record, making a number of queries for unallocated type, then making a query for a A record again. IANA maintains a registry of allocated types.

If the server responds to the first and last queries but fails to respond to the queries for the unallocated type it is probably faulty. The test should be repeated a number of times to eliminate the likelihood of a false positive due to packet loss.

## 2.3. TCP Queries

All DNS servers are supposed to respond to queries over TCP [RFC5966]. Firewalls that drop TCP connection attempts rather than resetting the connect attempt or send a ICMP/ICMPv6 administratively prohibited message introduce excessive delays to the resolution process.

Whether a server accepts TCP connections can be tested by first checking that it responds to UDP queries to confirm that it is up and operating then attempting the same query over TCP. A additional query should be made over UDP if the TCP connection attempt fails to confirm that the server under test is still operating.

## 3. Remediating

While the first step in remediating this problem is to get the offending nameserver code corrected, there is a very long tail problem with DNS servers in that it can often take over a decade between the code being corrected and a nameserver being upgraded with corrected code. With that in mind it is requested that TLD, and other similar zone operators, take steps to identify and inform their

customers, directly or indirectly through registrars, that they are running such servers and that the customers need to correct the problem.

TLD operators should construct a list of servers child zones are delegated to along with a delegated zone name. This name shall be the query name used to test the server as it is supposed to exist.

For each server the TLD operator shall make a SOA query the delegated zone name. This should result in the SOA record being returned in the answer section. If the SOA record is not return but some other response is returned this is a indication of a bad delegation and the TLD operator should take whatever steps it normally takes to rectify a bad delegation. If more that one zone is delegated to the server it should choose another zone until it finds a zone which responds correctly or it exhausts the list of zones delegated to the server.

If the server fails to get a response to a SOA query the TLD operator should make a A query as some nameservers fail to respond to SOA queries but respond to A queries. If it gets no response to the A query another delegated zone should be queried for as some nameservers fail to respond to zones they are not configured for. If subsequent queries find a responding zone all delegation to this server need to be checked and rectified using the TLD's normal procedures.

Having identified a working <server, query name> tuple the TLD operator should now check that the server responds to EDNS, Unknown Query Type and TCP tests as described above. If the TLD operator finds that server fails any of the tests, the TLD operator shall take steps to inform the operator of the server that they are running a fault nameserver and that they need to take steps to correct the matter. The TLD operator shall also record the <server, query name> for followup testing.

If repeated attempts to inform and get the customer to correct / replace the fault server are unsuccessful the TLD operator shall remove all delegations to said server from the zone.

It will also be necessary for TLD operators to repeat the scans periodically. It is recommended that this be performed monthly backing off to bi-annually once the numbers of faulty servers found drops off to less than 1 in 100000 servers tested. Follow up tests for faulty servers still need to be performed monthly.

Some operators claim that they can't perform checks at registration time. If a check is not performed at registration time it needs to be performed within a week of registration in order to detect faulty

servers swiftly.

Checking of delegations by TLD operators should be nothing new as they have been required from the very beginnings of DNS to do this [RFC1034]. Checking for compliance of nameserver operations should just be an extension of such testing.

It is recommended that TLD operators setup a test web page which performs the tests the TLD operator performs as part of their regular audits to allow nameserver operators to test that they have correctly fixed their servers. Such tests should be rate limited to avoid these pages being a denial of service vector.

#### 4. Firewalls and Load Balancers

Firewalls and load balancers can affect the externally visible behaviour of a nameserver. Tests for conformance need to be done from outside of any firewall so that the system as a whole is tested.

Firewalls and load balancers should not drop DNS packets that they don't understand. They should either pass through the packets or generate an appropriate error response.

Requests for unknown query types are not attacks and should not be treated as such.

#### 5. Normative References

- [RFC1034] Mockapetris, P., "DOMAIN NAMES - CONCEPTS AND FACILITIES", STD 13, RFC 1034, November 1987.
- [RFC1035] Mockapetris, P., "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION", STD 13, RFC 1035, November 1987.
- [RFC2671] Vixie, P., "Extension Mechanisms for DNS (EDNS0)", RFC 2671, August 1999.
- [RFC5966] Bellis, R., "DNS Transport over TCP - Implementation Requirements", RFC 5966, August 2010.

Author's Address

M. Andrews  
Internet Systems Consortium  
950 Charter Street  
Redwood City, CA 94063  
US

Email: marka@isc.org





Network Working Group  
Internet-Draft  
Expires: April 12, 2014

M. Andrews  
ISC  
October 9, 2013

Add 100.64.0.0/10 prefixes to IPv4 Locally-Served DNS Zones Registry.  
draft-andrews-dnsop-rfc6598-rfc6303-03

## Abstract

RFC6598 specified that: "Reverse DNS queries for Shared Address Space addresses [100.64.0.0/10] MUST NOT be forwarded to the global DNS infrastructure."

This document formally directs IANA to add the associated zones to the "IPv4 Locally-Served DNS Zones Registry" to prevent such queries accidentally leaking to the global DNS infrastructure.

## 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 April 12, 2014.

## Copyright Notice

Copyright (c) 2013 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

1. Introduction . . . . . 3

2. Changes to IPv4 Locally-Served DNS Zones Registry . . . . . 3

    2.1. RFC6598 Zones . . . . . 3

3. IANA Considerations . . . . . 4

4. Security Considerations . . . . . 5

5. Acknowledgements . . . . . 5

6. Normative References . . . . . 5

Author's Address . . . . . 5

## 1. Introduction

[RFC6598] specified that: "Reverse DNS queries for Shared Address Space addresses [100.64.0.0/10] MUST NOT be forwarded to the global DNS infrastructure." [RFC6303] provides guidance on handling such queries.

This document directs the IANA to add the IPv4 reverse zones corresponding to 100.64.0.0/10, a netblock reserved in [RFC6598], to the IPv4 Locally-Served DNS Zone Registry established in [RFC6303].

Unlike [RFC1918] address, which are not expected to be seen by other parties, the addresses from [RFC6598] are expected to be seen by parties other than those deploying the addresses, so it is more crucial that recursive nameservers default to serving these zones locally.

## 2. Changes to IPv4 Locally-Served DNS Zones Registry

To add the following zone listed in RFC6598 Zones (Section 2.1) to the "IPv4 Locally-Served DNS Zone Registry".

### 2.1. RFC6598 Zones

64.100.IN-ADDR.ARPA  
65.100.IN-ADDR.ARPA  
66.100.IN-ADDR.ARPA  
67.100.IN-ADDR.ARPA  
68.100.IN-ADDR.ARPA  
69.100.IN-ADDR.ARPA  
70.100.IN-ADDR.ARPA  
71.100.IN-ADDR.ARPA  
72.100.IN-ADDR.ARPA  
73.100.IN-ADDR.ARPA  
74.100.IN-ADDR.ARPA  
75.100.IN-ADDR.ARPA  
76.100.IN-ADDR.ARPA  
77.100.IN-ADDR.ARPA  
78.100.IN-ADDR.ARPA  
79.100.IN-ADDR.ARPA  
80.100.IN-ADDR.ARPA  
81.100.IN-ADDR.ARPA  
82.100.IN-ADDR.ARPA  
83.100.IN-ADDR.ARPA  
84.100.IN-ADDR.ARPA

85.100.IN-ADDR.ARPA  
86.100.IN-ADDR.ARPA  
87.100.IN-ADDR.ARPA  
88.100.IN-ADDR.ARPA  
89.100.IN-ADDR.ARPA  
90.100.IN-ADDR.ARPA  
91.100.IN-ADDR.ARPA  
92.100.IN-ADDR.ARPA  
93.100.IN-ADDR.ARPA  
94.100.IN-ADDR.ARPA  
95.100.IN-ADDR.ARPA  
96.100.IN-ADDR.ARPA  
97.100.IN-ADDR.ARPA  
98.100.IN-ADDR.ARPA  
99.100.IN-ADDR.ARPA  
100.100.IN-ADDR.ARPA  
101.100.IN-ADDR.ARPA  
102.100.IN-ADDR.ARPA  
103.100.IN-ADDR.ARPA  
104.100.IN-ADDR.ARPA  
105.100.IN-ADDR.ARPA  
106.100.IN-ADDR.ARPA  
107.100.IN-ADDR.ARPA  
108.100.IN-ADDR.ARPA  
109.100.IN-ADDR.ARPA  
110.100.IN-ADDR.ARPA  
111.100.IN-ADDR.ARPA  
112.100.IN-ADDR.ARPA  
113.100.IN-ADDR.ARPA  
114.100.IN-ADDR.ARPA  
115.100.IN-ADDR.ARPA  
116.100.IN-ADDR.ARPA  
117.100.IN-ADDR.ARPA  
118.100.IN-ADDR.ARPA  
119.100.IN-ADDR.ARPA  
120.100.IN-ADDR.ARPA  
121.100.IN-ADDR.ARPA  
122.100.IN-ADDR.ARPA  
123.100.IN-ADDR.ARPA  
124.100.IN-ADDR.ARPA  
125.100.IN-ADDR.ARPA  
126.100.IN-ADDR.ARPA  
127.100.IN-ADDR.ARPA

### 3. IANA Considerations

This document directs IANA to add the zones listed in RFC6598 Zones

(Section 2.1) to the "IPv4 Locally-Served DNS Zone Registry".

#### 4. Security Considerations

This document is thought to present no additional security risks to the Internet.

#### 5. Acknowledgements

I would like to thank Joe Abley for his review comments.

#### 6. Normative References

- [RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", RFC 1918, February 1996.
- [RFC6303] Andrews, M., "Locally Served DNS Zones", BCP 163, RFC 6303, July 2011.
- [RFC6598] Weil, J., Kuarsingh, V., Donley, C., Liljenstolpe, C., and M. Azinger, "IANA-Reserved IPv4 Prefix for Shared Address Space", BCP 153, RFC 6598, April 2012.

#### Author's Address

M. Andrews  
Internet Systems Consortium  
950 Charter Street  
Redwood City, CA 94063  
US

Email: marka@isc.org



Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: December 13, 2014

X. Deng  
M. Boucadair  
France Telecom  
Q. Zhao  
Beijing University of Posts and Telecommunications  
J. Huang  
C. Zhou  
Huawei Technologies  
June 11, 2014

Using Port Control Protocol (PCP) to update dynamic DNS  
draft-deng-pcp-ddns-06

Abstract

This document focuses on the problems encountered when using dynamic DNS in address sharing contexts (e.g., DS-Lite, NAT64) during IPv6 transition. Both issues and possible solutions are documented in this memo.

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 December 13, 2014.

Copyright Notice

Copyright (c) 2014 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

1. Introduction . . . . .	2
1.1. Problem Statement . . . . .	2
1.2. Scope and Goals . . . . .	3
2. Solution Space . . . . .	4
2.1. Locate a Service Port . . . . .	4
2.2. Create Explicit Mappings for Incoming Connections . . . . .	5
2.3. Detect Changes . . . . .	5
3. Some Deployment Solutions . . . . .	6
3.1. Reference Topology . . . . .	6
3.2. For Web Service . . . . .	7
3.3. For Non-web Service . . . . .	8
4. Security Considerations . . . . .	10
5. IANA Considerations . . . . .	10
6. Contributors . . . . .	11
7. Acknowledgements . . . . .	11
8. References . . . . .	11
8.1. Normative References . . . . .	11
8.2. Informative References . . . . .	11
Authors' Addresses . . . . .	12

## 1. Introduction

### 1.1. Problem Statement

Dynamic DNS (DDNS) is a widely deployed service to facilitate hosting servers (e.g., access to a webcam, HTTP server, FTP server, etc.) at customers' premises. There are a number of providers which offer a DDNS service, working in a client and server mode, which mostly use a web-form based communication. DDNS clients are generally implemented in the user's router or computer, which once detects changes to its assigned IP address it automatically sends an update message to the DDNS server. The communication between the DDNS client and the DDNS server is not standardized, varying from one provider to another, although a few standard web-based methods of updating emerged over time.

When the network architecture evolves towards an IPv4 sharing architecture during IPv6 transition, the DDNS client will have to not only inform the IP address updates if any, but also to notify the changes of external port on which the service is listening, because



well known port numbers, e.g., port 80 will no longer be available to every web server. It will also require the ability to configure corresponding port forwarding on CGN (Carrier Grade NAT, [RFC6888]) devices, so that incoming communications initiated from Internet can be routed to the appropriate server behind the CGN.

Issues encountered in address sharing are documented in [RFC6269]. This document focuses on the problems encountered when using dynamic DNS in address sharing contexts (e.g., DS-Lite [RFC6333], NAT64 [RFC6146]). Below are listed the main challenges:

Announce and Discover an alternate service port: The DDNS service must be able to maintain an alternative port number instead of the default port number.

Allow for incoming connections: Appropriate means to instantiate port mappings in the address sharing device must be supported.

Detect changes and trigger DDNS updates: DDNS client must be triggered by the change of the external IP address and the port number. Concretely, upon change of the external IP address (and/or external port number), the DDNS client must refresh the DNS records otherwise the server won't be reachable from outside. This issue is exacerbated in the DS-Lite context because no public IPv4 address is assigned to the CPE.

## 1.2. Scope and Goals

This document describes some candidate solutions to resolve the aforementioned issues with a particular focus on DS-Lite. These solutions may also be valid for other address sharing schemes.

This document sketches deployment considerations based on the PCP (Port Control Protocol, [RFC6887]). Note DDNS may be considered as an implementation of the Rendezvous service mentioned in [RFC6887].

Indeed, after creating an explicit mapping for incoming connections using PCP, it is necessary to inform remote hosts about the IP address, protocol, and port number for the incoming connection to reach the services hosted behind a DS-Lite CGN. This is usually done in an application-specific manner. For example, a machine hosting a game server might use a rendezvous server specific to that game (or specific to that game developer), a SIP phone would use a SIP proxy, and a client using DNS-Based Service Discovery [RFC6763] would use DNS Update [RFC2136][RFC3007], etc. PCP does not provide this rendezvous function.

The rendezvous function may support IPv4, IPv6, or both. Depending on that support and the application's support of IPv4 or IPv6, the PCP client may need an IPv4 mapping, an IPv6 mapping, or both. An example illustrating how the DDNS server may implement such a service notification functionality if necessary is provided in Section 3.

This document does not specify any protocol extension, but instead it focuses on the elaboration of the problem space and illustrate how existing tools can be re-used to solve the problem for some deployment contexts. Particularly, this document requires no changes to PCP or dynamic updates in the standard domain name system [RFC2136], but is rather an operational document to make the current DDNS service providers be aware of the impacts and issues that the IPv6 transitioning and IPv4 address sharing will bring to them, and gives solutions address the forthcoming issues. The current DDNS service providers usually employs a web-based form to maintain DDNS service registration and updates.

Generic deployment considerations for DS-Lite, including B4 remote management and IPv4 connectivity check, can be found in [RFC6908]. This document complements [RFC6908] with deployment considerations related to Rendezvous service maintenance. Additional PCP-related deployment considerations are available at [I-D.boucadair-pcp-deployment-cases].

Solutions relying on DNS-based Service Discovery [RFC6763] or Apple's Back to My Mac (BTMM) Service [RFC6281] are not considered in this document. Moreover, this document does not assume that DDNS service relies on [RFC2136].

IPv4 addresses used in the examples are derived from the IPv4 block reserved for documentation in [RFC6890]. DNS name examples follow [RFC2606].

## 2. Solution Space

### 2.1. Locate a Service Port

As listed below, at least two solutions can be used to associate a port number with a service:

1. Use service URIs (e.g., FTP, SIP, HTTP) which embed an explicit port number. Indeed, Uniform Resource Identifier (URI) defined in [RFC3986] allows to carry port number in the syntax (e.g., mydomain.example:15687).
2. Use SRV records [RFC2782]. Unfortunately, the majority of browsers do not support this record type.

DDNS client and DDNS server are to be updated so that an alternate port number is signaled and stored by the DDNS server. Requesting remote hosts will be then notified with the IP address and port number to reach the server.

## 2.2. Create Explicit Mappings for Incoming Connections

PCP is used to install the appropriate mapping(s) in the CGN so that incoming packets can be delivered to the appropriate server.

## 2.3. Detect Changes

In a network described in Figure 1, DDNS client/ PCP client can either be running on a Customer Premise Equipment (CPE) or be running on the host that is hosting some services itself. There are several possible ways to address the problems stated in Section 1.1:

1. If the DDNS client is enabled, the host issues periodically (e.g., 60 minutes) PCP MAP requests (e.g., messages 1 and 2 in Figure 1) with short lifetime (e.g., 30s) for the purpose of enquiring external IP address and setting. If the purpose is to detect any change of external port, the host must issues a PCP mapping to install a mapping for the internal server. Upon change of the external IP address, the DDNS client updates the records accordingly (e.g., message 3 in Figure 1).
2. If the DDNS client is enabled, it checks the local mapping table maintained by the PCP client. This process is repeated periodically (e.g., 5 minutes, 30 minutes, 60 minutes). If there is no PCP mapping created by PCP client, it issues a PCP MAP request (e.g., messages 1 and 2 in Figure 1) for the purpose of enquiring external IP address and setting up port forwarding mappings for incoming connections. Upon change of the external IP address, the DDNS client updates the records in the DDNS server, e.g., message 3 in Figure 1.

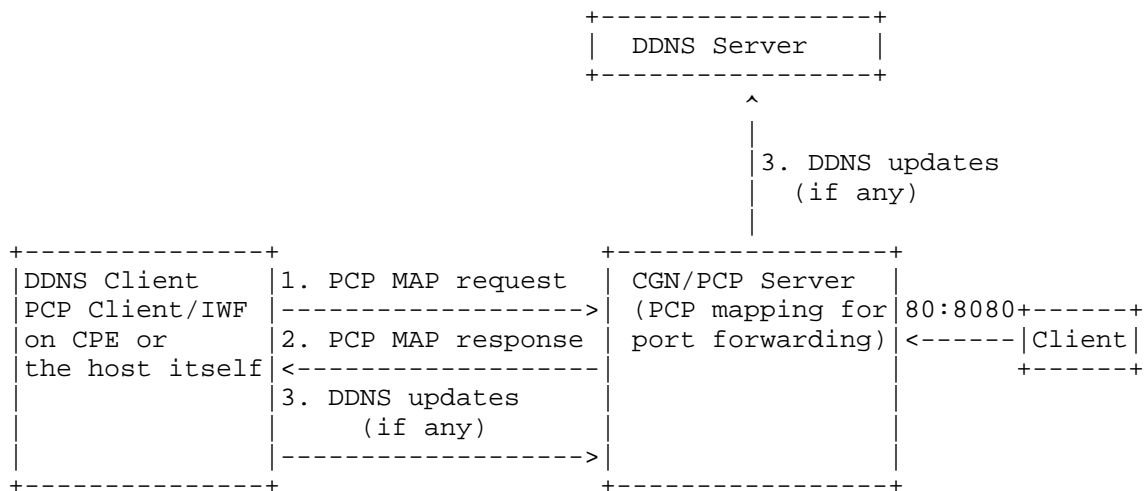


Figure 1: Flow Chart

### 3. Some Deployment Solutions

#### 3.1. Reference Topology

Figure 2 illustrates the topology used for the deployment solutions elaborated in the following sub-sections.

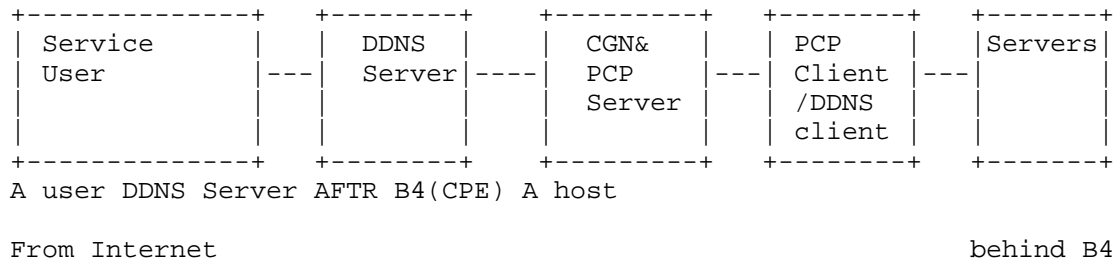


Figure 2: Implementation Topology

Figure 2 involves of the following entities:

- o Servers: refer to the servers that are deployed in the DS-Lite network, or more generally, an IP address sharing environment. They are usually running on a host that has been assigned with a private IPv4 address. Having created a proper mapping via PCP in AFTR, these services have been made available to the Internet users. The services may provide Web, FTP, SIP and other services though these ones may not be able to be seen as using a well

known port from the outside anymore, in the IP address sharing context.

- o B4 (CPE): An endpoint of IPv4-in-v6 tunnel [RFC6333]. A PCP client together with a DDNS client are running on it. After PCP client establishes a mapping on the AFTR, an end user may register its domain name and its external IPv4 address plus port number to its DDNS service provider (DDNS server), manually or automatically by DDNS client. Later, likewise, end users may manually or let DDNS client on behalf of it, to automatically announce IP address and/or port changes to the DDNS server.
- o AFTR: Responsible for maintaining mappings between internal IPv4 Address plus port and external IPv4 address plus port [RFC6333].
- o DDNS server: Maintains a table that associates a registered domain name and a pair of registered host's external IPv4 address plus port number. When being notified IP address and port number changes from DDNS client, DDNS server announces the updates to DNS servers on behalf of end user. [RFC2136] and [RFC3007] may be used by DDNS server to send updates to DNS servers. In many current practices, DDNS server provider usually announce its own IP address as the registered domain names of end users. When HTTP requests reach the DDNS server, they may employ URL Forwarding or HTTP 301 redirection to redirect the request to a proper registered end user by looking up the maintained link table.
- o Service users: refers to users who want to access services behind an IP address sharing network. They issue standard DNS requests to locate the services, which will lead them to a DDNS server, provided that the requested services have been registered to a DDNS service provider. The DDNS server will then handle the rest in the way as described before.

### 3.2. For Web Service

Current DDNS server implementations typically assume that the end servers host web server on the default 80 port. In the DS-Lite context, they will have to take into account that external port assigned by AFTR may be any number other than 80, in order to maintain proper mapping between domain names and external IP plus port. By doing such changes, the HTTP request would be redirected to the AFTR which servers the specific end host that are running servers.

Figure 3 depicts how messages are handled in order to be delivered to the right server.

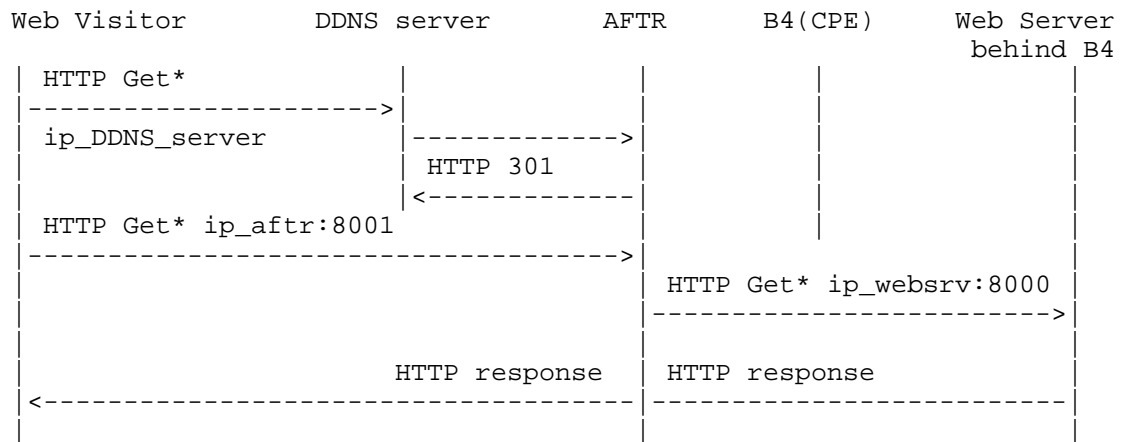


Figure 3: Http Service Messages

When a web user sends out a HTTP GET message to DDNS server after a standard DNS query, DDNS server redirects the request to a registered web server, in this case, by responding with a HTTP 301 message. Then, the HTTP GET message will be sent out to the AFTR, which will in turn find the proper hosts behind it. For simplicity, messages among AFTR, B4 and web server behind B4 are not shown completely; for communications among those nodes, refer to [RFC6333].

### 3.3. For Non-web Service

For non-web services, as mentioned in Section 2, other means will be needed to inform the users about the service information.

[RFC6763] includes an example of DNS-based solution which allows an application running in the end user's device to retrieve service-related information via DNS SRV/TXT records, and list available services. In a scenario where such application is not applicable, following provides another solution for a third party, e.g., DDNS service provider, to disclose services to the Internet users.

A web portal can be used to list available services. DDNS server maintains a web portal for each user FQDN (Fully Qualified Domain Name), which provides users service links. Figure 4 assumes "websrv.example.com" is a user's FQDN provided by a DDNS service provider.

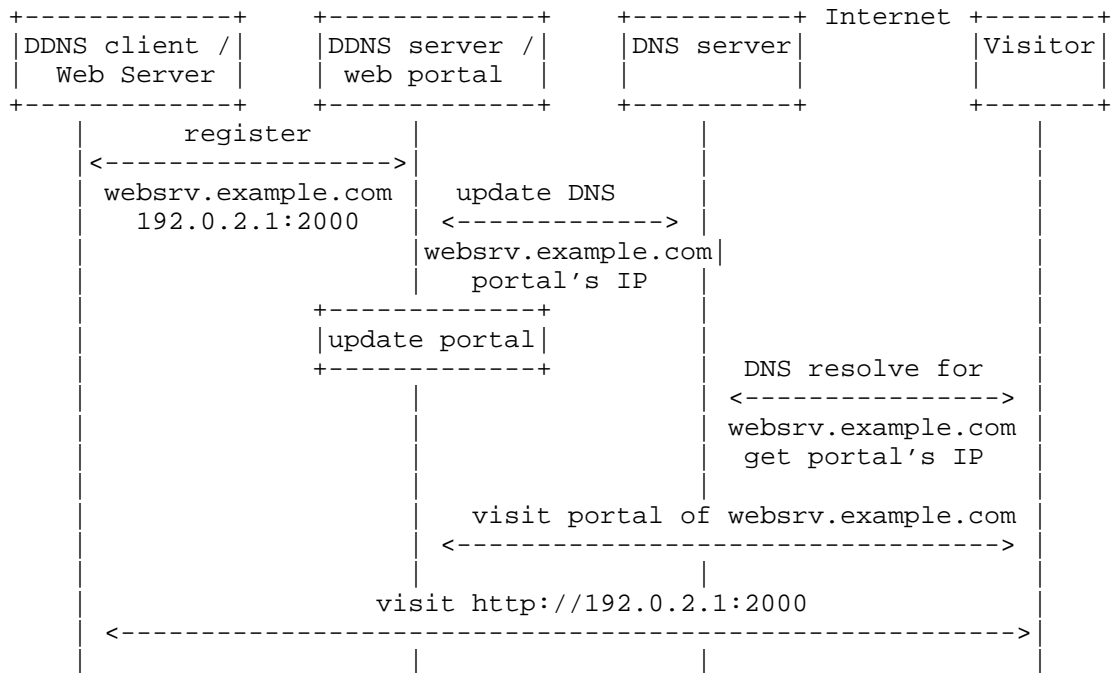


Figure 4: Update Web Portal

The DDNS client registers the servers' information to the DDNS server, including public IP address and port obtained via PCP, user's FQDN and other necessary information. The DDNS server also behaves as portal server, it registers its IP address, port number, and user's FQDN to the DNS system, so that visitors can access the web portal.

DDNS server also maintains a web portal for each user's FQDN, update the portal according to registered information from DDNS client. When a visitor accesses "webserv.example.com", a DNS query will resolve to portal server's address, port number, and the visitor will see the portal and the available services.

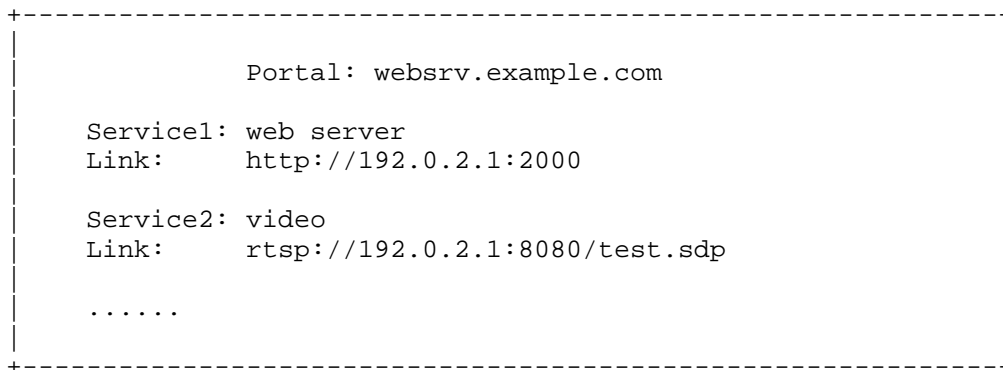


Figure 5: An Example of Web Portal

As shown in Figure 5, the web portal shows the service URLs that are available to be accessed. Multiple services are accessible per user's FQDN.

Some applications which are not HTTP-based can also be delivered using this solution. When a user clicks on a link, the registered application in the client OS will be invoked to handle the link. How this can be achieved is out of the scope of this document.

## 4. Security Considerations

This document does not introduce a new protocol nor specify protocol extensions. Security-related considerations related to PCP [RFC6887] and DS-Lite [RFC6333] should be taken into account.

The protocol between the DDNS client and DDNS server is proprietary in most cases, some extensions may be necessary, which is up to DDNS operators. These operators should enforce security-related policies to avoid that illegitimate users alter records installed by legitimate users or install fake records that would lead to attract illegitimate traffic. Means to protect the DDNS server against DoS (Denial of Service) should be enabled. Note these considerations are not specific to address sharing contexts but are valid for DDNS service in general.

## 5. IANA Considerations

This document does not require any action from IANA.



## 6. Contributors

The following individuals contributed text to the document:

Xiaohong Huang

Beijing University of Posts and Telecommunications, China  
Email: huangxh@bupt.edu.cn

Yan Ma

Beijing University of Posts and Telecommunications, China  
Email: mayan@bupt.edu.cn

## 7. Acknowledgements

Thanks to Stuart Cheshire for bringing up DNS-Based Service Discovery and [RFC6281] where covers DNS-based SD scenario and gives an example of how the application means of solution to address dynamic DNS update, in this case, apple' BTMM, can be achieved.

Many thanks to D. Wing, D. Thaler, and J. Abley for their comments.

## 8. References

### 8.1. Normative References

- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.
- [RFC6887] Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, April 2013.

### 8.2. Informative References

- [I-D.boucadair-pcp-deployment-cases] Boucadair, M., "Port Control Protocol (PCP) Deployment Models", draft-boucadair-pcp-deployment-cases-02 (work in progress), April 2014.

- [RFC2136] Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, April 1997.
- [RFC2606] Eastlake, D. and A. Panitz, "Reserved Top Level DNS Names", BCP 32, RFC 2606, June 1999.
- [RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.
- [RFC3007] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, November 2000.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.
- [RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, June 2011.
- [RFC6281] Cheshire, S., Zhu, Z., Wakikawa, R., and L. Zhang, "Understanding Apple's Back to My Mac (BTMM) Service", RFC 6281, June 2011.
- [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, February 2013.
- [RFC6888] Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", BCP 127, RFC 6888, April 2013.
- [RFC6890] Cotton, M., Vegoda, L., Bonica, R., and B. Haberman, "Special-Purpose IP Address Registries", BCP 153, RFC 6890, April 2013.
- [RFC6908] Lee, Y., Maglione, R., Williams, C., Jacquenet, C., and M. Boucadair, "Deployment Considerations for Dual-Stack Lite", RFC 6908, March 2013.

#### Authors' Addresses

Xiaohong Deng

Email: dxhbupt@gmail.com

Mohamed Boucadair  
France Telecom  
Rennes 35000  
France

Email: mohamed.boucadair@orange.com

Qin Zhao  
Beijing University of Posts and Telecommunications  
China

Email: zhaoqin.bupt@gmail.com

James Huang  
Huawei Technologies  
China

Email: james.huang@huawei.com

Cathy Zhou  
Huawei Technologies  
China

Email: cathy.zhou@huawei.com

Network Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: February 1, 2014

E. Hunt  
ISC  
July 31, 2013

The DNS Extended Server Diagnostics (ESD) Option  
draft-hunt-dns-server-diagnostics-00

Abstract

The widespread adoption of DNSSEC implies more frequent DNSSEC failures. Unfortunately, DNSSEC's failure mode is largely opaque to the client: when validation fails, the only signal that the clients of a validating resolver receive is an empty response with a SERVFAIL response code. This note proposes a protocol extension to allow SERVFAIL responses to include additional diagnostic information, giving the client greater insight into what went wrong and a better chance of delivering useful problem reports to DNS operators.

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 February 1, 2014.

Copyright Notice

Copyright (c) 2013 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

1. Introduction . . . . .	3
1.1. Reserved Words . . . . .	3
2. Protocol . . . . .	3
2.1. Client Behavior . . . . .	4
2.2. Server Behavior . . . . .	4
2.3. The ESD Option . . . . .	4
2.4. ESD Diagnostic Codes . . . . .	5
2.4.1. Internal Server Errors . . . . .	5
2.4.2. General DNS Errors . . . . .	6
2.4.3. Policy-Dependent Security Errors . . . . .	7
2.4.4. Temporary Security Errors . . . . .	8
2.4.5. Fatal Security Errors . . . . .	9
3. Security Considerations . . . . .	11
4. IANA Considerations . . . . .	12
4.1. ESD Option Code . . . . .	12
4.2. Diagnostic Codes . . . . .	12
5. Acknowledgments . . . . .	13
6. References . . . . .	13
6.1. Normative References . . . . .	13
6.2. Informative References . . . . .	14
Author's Address . . . . .	14

## 1. Introduction

DNSSEC's failure mode is largely opaque to the client: when validation fails, the only signal of this that a resolver's clients receive is a SERVFAIL response code.

With no information provided to explain the exact cause of a SERVFAIL response, there is no straightforward way for an end user to determine whether a failure occurred due to a broken local resolver, a zone misconfiguration such as expired signatures, or a spoofing attack. This makes it difficult to address complaints and problem reports to the right people, slowing the correction of DNSSEC errors while increasing the support burden on validating resolver operators.

This note proposes a protocol extension allowing a name server, upon request from a client, to accompany SERVFAIL responses with detailed diagnostic information indicating what specifically caused the failure. In the typical use case for this mechanism, a validating caching name server would be able to convey specific failure information to a non-validating stub resolver or other client.

### 1.1. Reserved Words

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. Protocol

A DNS client such as a non-validating stub resolver may use an EDNS(0) [RFC2671] option, ESD, to solicit extended diagnostic information from a DNS server. The ESD option payload includes a 16-bit "flags" field and a 16-bit diagnostic code; additional payload data may be included in the response.

One bit in the "flags" field is defined as "human-readable": if this bit is set in the solicitation, it indicates a desire for the server to return human-readable text, in addition to machine-readable diagnostic data; this text can be displayed or sent to a logging facility such as syslog [RFC5424]. All other payload data MUST be left empty in the solicitation.

The response payload, in addition to the flags field and the diagnostic code, may also include a supplemental data field whose content and schema are dependent on the diagnostic code being returned. If the "human-readable" flag is set in the response, then the response also includes human-readable text in a counted string,

i.e., a single length octet followed by that number of characters, as in the TXT RDATA format [RFC1035].

## 2.1. Client Behavior

A stub resolver or other DNS client requests extended diagnostic data by sending an ESD option (Section 2.3) in an EDNS(0) OPT pseudo-RR in the query message. The requestor MAY set the "human-readable" bit in the flags field of the request payload. The requestor MUST NOT include any other payload data in the ESD option.

The requestor MUST ignore any ESD option included in a response that does not have response code SERVFAIL.

## 2.2. Server Behavior

A resolver or other name server which encounters a server failure while answering a query that included an ESD option MAY add an ESD option to the SERVFAIL response. If the query did not include an ESD option, then the response MUST NOT include one. The server MUST NOT include an ESD option in any non-SERVFAIL response.

## 2.3. The ESD Option

The OPTION-CODE for the ESD option is [TBD].

The format for the OPTION-DATA portion of an ESD option is shown below:

```

      1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 3 3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+
|          FLAGS          |H|          DIAGCODE          |
+-----+-----+-----+-----+-----+-----+
\          HRTTEXT (optional, resonse only)          \
+-----+-----+-----+-----+-----+-----+
\          SUPPDATA (optional, response only)          \
+-----+-----+-----+-----+-----+-----+

```

in which the fields are defined as follows:

**FLAGS**      A 16-bit field. Bit 15 (H) is defined to mean "human-readable"; when set in the solicitation, this bit signals a desire for human-readable text to be included in the response; when set in the response, it signals that such text has been included. All other bits in the field MUST be set to zero.

- DIAGCODE A 16-bit diagnostic code (Section 2.4) indicating the reason for the SERVFAIL response. In the solicitation payload, this field **MUST** be set to zero.
- HRTEXT An counted string, containing human-readable text suitable for logging. The first octet defines the length of the following text; if the first octet contains 0, the string is empty. This is intended as an opaque string to be passed through to a logging mechanism; content and encoding are outside the scope of the protocol. This field **MUST NOT** be included in a solicitation payload.
- SUPPDATA Optional supplementary data about the cause of the server failure. The presence or absence, content, and schema of this field are entirely dependent on the diagnostic code being returned in the DIAGCODE field (Section 2.4). This field **MUST NOT** be included in a solicitation payload.

## 2.4. ESD Diagnostic Codes

Diagnostic codes are grouped in five general categories: Internal server error conditions (diagnostic codes 1-255), general DNS failures (256-511), policy-dependent security errors (512-767), temporary security errors (768-1023), and fatal security errors (1024-1279). Space has been reserved in the namespace for additional categories and codes. All diagnostic codes are optional; there is no requirement to implement all of them.

The DIAGCODE field **MUST** be set to zero (No Error) in ESD solicitations.

### 2.4.1. Internal Server Errors

These diagnostic codes indicate a system failure in the responding server.

#### 2.4.1.1. Internal Error, Not Otherwise Specified

Diagnostic code 1 indicates an unspecified internal server error unrelated to DNSSEC. A server **MAY** return this code in place of any other internal server error, at the discretion of the implementor or operator, if information about the internal state of the server is regarded as security sensitive. This code has no supplemental data.

#### 2.4.1.2. Out of Memory

Diagnostic code 2 indicates that the server was not able to dynamically allocate memory. This code has no supplemental data.



#### 2.4.1.3. Resource Unavailable

Diagnostic code 3 indicates that a needed resource was not available. This code has no supplemental data.

#### 2.4.1.4. Zone Not Loaded

Diagnostic code 4: The server has been configured to be authoritative for a zone which is an ancestor of the query name, but was unable to load it. The supplemental data contains the name of the zone the server was unable to load, in DNS wire format.

#### 2.4.1.5. Invalid Zone

Diagnostic code 5: The server has been configured to be authoritative for a zone which is an ancestor of the query name, but the zone contents are invalid; for example, there is no SOA RR or NS RRset at the zone apex. The supplemental data contains the name of the zone in DNS wire format.

#### 2.4.1.6. Configuration Error

Diagnostic code 6: The server could not be initialized, e.g., as a result of an error in loading or parsing the configuration file. This code has no supplemental data.

#### 2.4.1.7. Timeout

Diagnostic code 7: Query processing timed out. This code has no supplemental data.

#### 2.4.1.8. Shutting Down

Diagnostic code 8: The server is shutting down. This code has no supplemental data.

#### 2.4.2. General DNS Errors

These codes describe failure conditions due to bad or inconsistent data in the DNS not specifically related to DNSSEC.

##### 2.4.2.1. Lame Delegation

Diagnostic code 256: The name servers which have been delegated responsibility for a zone cannot be reached or are not performing name service for that zone. The supplemental data contains the zone apex name of the faulty zone.

#### 2.4.2.2. Name Expansion Failure

Diagnostic code 257: A CNAME [RFC1034] or DNAME [RFC6672] record fails sanity checks after name expansion. The supplemental data contains the name and RR type (CNAME or DNAME) of the faulty record, in the following format:

```

                                1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 3 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |                                     |
|          RRTYPE                     |          NAME                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

#### 2.4.2.3. Protocol Not Supported

Diagnostic code 258: Processing this query requires a protocol extension that is not supported. This code has no supplemental data.

#### 2.4.3. Policy-Dependent Security Errors

These are errors returned due to locally-configured policy constraints on DNSSEC processing or other security considerations.

##### 2.4.3.1. Key Too Large

Diagnostic code 512: Local policy disallows a DNSKEY being used to validate a record on the grounds that it is too large. The supplemental data specifies the owner name (in DNS wire format) and key tag of the problematic DNSKEY, using the following format:

```

                                1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 3 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |                                     |
|          TAG                     |          NAME                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

##### 2.4.3.2. Key Too Small

Diagnostic code 513: Local policy disallows a DNSKEY being used to validate a record on the grounds that it is too small. The supplemental data contains the nam5 and tag of the problematic key, in the format specified in Section 2.4.3.1.

##### 2.4.3.3. Key Not Authorized

Diagnostic code 514: Local policy does not authorize a key to be used for validation. The supplemental data contains the name and tag of the problematic key, in the format specified in Section 2.4.3.1.

#### 2.4.3.4. Key Algorithm Refused

Diagnostic code 515: Local policy prohibits validation using a given signing algorithm. The supplemental data contains a 16-bit unsigned integer indicating which algorithm has been disallowed.

#### 2.4.3.5. Unauthorized Signer

Diagnostic code 516: Local policy disallows accepting signatures created by this signer. The supplemental data contains the name of the signer that has been disallowed.

#### 2.4.3.6. No Trust Anchor

Diagnostic code 517: There is no trust anchor for the chain of trust needed to validate this data, but local policy requires validation. This code has no supplemental data.

#### 2.4.3.7. Too Many Links

Diagnostic code 518: The chain of trust is longer than local policy permits. This code has no supplemental data.

#### 2.4.3.8. Response Blocked

Diagnostic code 519: The response to this query has been blocked by local policy. This code has no supplemental data.

#### 2.4.4. Temporary Security Errors

These are error conditions occurring from DNSSEC processing which are time-dependent: i.e., problems due to signature validity interval or key expiry.

##### 2.4.4.1. Signature Expired

Diagnostic code 768: An RRSIG is past its useful lifetime. The supplemental data contains the name and covering RR type of the failed RRSIG, in the format specified in Section 2.4.2.2.

##### 2.4.4.2. Signature Not Yet Active

Diagnostic code 769: An RRSIG has not yet begun its useful lifetime. The supplemental data contains the name and covering RR type of the invalid RRSIG, in the format specified in Section 2.4.2.2.

#### 2.4.4.3. Trust Anchor Expired

Diagnostic code 770: A trust anchor can no longer be used. The supplemental data contains the name of the expired trust anchor in wire format.

#### 2.4.5. Fatal Security Errors

These error conditions due to DNSSEC processing are always fatal, regardless of time or local policy.

##### 2.4.5.1. Bogus Data

Diagnostic code 1024: Cryptographic validation failed. The supplemental data contains the name and RR type of data which failed to validate, in the format specified in Section 2.4.2.2.

##### 2.4.5.2. Signature Missing

Diagnostic code 1025: There was no RRSIG found for an RRset, but one should have been present. The supplemental data contains the name and RR type of the data that should have been signed, in the format specified in Section 2.4.2.2.

##### 2.4.5.3. DNSKEY Missing

Diagnostic code 1026: No DNSKEY was found, but one should have been present. The supplemental data contains the zone apex name at which the DNSKEY should have been found, in wire format.

##### 2.4.5.4. Key Tag Mismatch

Diagnostic code 1027: RRSIG records have been found for an RRset which is to be validated, but none of them has a key tag matching a valid DNSKEY. The supplemental data contains the name and covering RR type for the faulty RRSIG, in the format specified in Section 2.4.2.2.

##### 2.4.5.5. DS Missing

Diagnostic code 1028: No DS record was found, but there should have been one present. The supplemental data contains the name at which the DS record should have been found.

##### 2.4.5.6. Next-Secure Record Missing

Diagnostic code 1029: No NSEC [RFC4034] or NSEC3 [RFC5155] record was found, but there should have been one present. The supplemental data

contains the name and RR type (NSEC or NSEC3) of the record that was expected, in the format specified in Section 2.4.2.2.

#### 2.4.5.7. Overreaching Next-Secure Record

Diagnostic code 1030: The "next owner" name in an NSEC or NSEC3 record goes beyond another record which is known to exist. The supplemental data contains the name and RR type (NSEC or NSEC3) of the invalid record, in the format specified in Section 2.4.2.2.

#### 2.4.5.8. Next-Secure Record Pointing Backward

Diagnostic code 1031: The ordering of records in the NSEC or NSEC3 chain does not follow canonical ordering rules. The supplemental data contains the name and RR type (NSEC or NSEC3) of the invalid record, in the format specified in Section 2.4.2.2.

#### 2.4.5.9. Irrelevant Proof

Diagnostic code 1032: A proof of non-existence was provided for a record whose non-existence did not need to be proven. This code has no supplemental data.

#### 2.4.5.10. Incomplete Proof

Diagnostic code 1033: Proof of non-existence is incomplete. The supplemental data contains the name and RR type of the data whose non-existence needed to be proven, in the format specified in Section 2.4.2.2.

#### 2.4.5.11. Wrong RRSIG Owner

Diagnostic code 1034: The RRSIG being used for verification is incorrect for the RR in question. The supplemental data contains the name and covering RR type of the invalid RRSIG, in the format specified in Section 2.4.2.2.

#### 2.4.5.12. Unknown DNSKEY Protocol

Diagnostic code 1035: The DNSKEY protocol value is not set to 3. The supplemental data contains the name and tag of the faulty key, in the format specified in Section 2.4.3.1.

#### 2.4.5.13. DS/DNSKEY Mismatch

Diagnostic code 1036: A mismatch was found between the DNSKEY in a zone and the DS record in the parent. The supplemental data contains the name and tag of the DNSKEY that should have been found, in the

format specified in Section 2.4.3.1.

#### 2.4.5.14. Unknown Algorithm

Diagnostic code 1037: An algorithm specified in a DNSKEY, DS, RRSIG, NSEC3 or NSEC3PARAM record is not recognized by the server. The supplemental data contains the name and RR type of the record that referenced the invalid algorithm.

#### 2.4.5.15. Algorithm Not Supported

Diagnostic code 1038: An algorithm specified in a DNSKEY, DS, RRSIG, NSEC3 or NSEC3PARAM record is recognized by the server but is not implemented. The supplemental data contains the name and RR type of the record that referenced the unsupported algorithm.

#### 2.4.5.16. Not a Zone Key

Diagnostic code 1039: The key that is used to verify signatures on zone data does not have the "Zone Key" flag [RFC4034] set. The supplemental data contains the name and tag of the faulty key, in the format specified in Section 2.4.3.1.

#### 2.4.5.17. Key Revoked

Diagnostic code 1040: A key that is required to complete a chain of trust has its REVOKED bit [RFC5011] set. The supplemental data contains the name and tag of the revoked key, in the format specified in Section 2.4.3.1.

### 3. Security Considerations

An ESD option response contains channel diagnostic data, to be used for logging, troubleshooting, and debugging; it falls outside the scope of DNSSEC per se. Ensuring integrity of the response requires the use of a channel security mechanism such as TSIG [RFC2845] or SIG(0) [RFC2931]. In the absence of any such mechanism, ESD responses MUST NOT be used by clients to make policy decisions with respect to DNSSEC validation, as this could allow an attacker to force a security downgrade or denial of service by forging SERVFAIL messages containing particular ESD responses.

Some of the data in an ESD option response may be security sensitive, particularly those responses which increase the transparency of the current state of a running resolver. In the case of SERVFAIL responses due to authoritative server misconfiguration or other non-local conditions, this is generally not a concern, as the data which

caused the failure are readily available in the DNS and can be obtained by other means. However, information about server failures due to local problems such as out-of-memory conditions may be of tactical use to an attacker. Implementors may wish to provide a mechanism for operators to disable certain types of diagnostic response while allowing others.

#### 4. IANA Considerations

IANA is requested to make the assignments in this section.

##### 4.1. ESD Option Code

This document requests the allocation of an EDNS(0) option code for the ESD option, whose value is [TBD].

##### 4.2. Diagnostic Codes

This document requests the creation of a new registry of ESD diagnostic codes. The registry policy is "Specification Required" [RFC5226]. The initial entries in the registry are:

Value	Description	Reference
0	No Error	
1	Internal Error	[This]
2	Out of Memory	[This]
3	Resource Unavailable	[This]
4	Zone Not Loaded	[This]
5	Invalid Zone	[This]
6	Configuration Error	[This]
7	Timeout	[This]
8	Shutting Down	[This]
9-255	Unassigned	
256	Lame Delegation	[This]
257	Name Expansion Failure	[This]
258	Protocol Not Supported	[This]
259-511	Unassigned	
512	Key Too Large	[This]
513	Key Too Small	[This]
514	Key Not Authorized	[This]
515	Algorithm Refused	[This]
516	Unauthorized Signer	[This]
517	No Trust Anchor	[This]
518	Too Many Links	[This]
519	Response Blocked	[This]

520-767	Unassigned	
768	Signature Expired	[This]
769	Signature Not Yet Active	[This]
770	Trust Anchor Expired	[This]
771-1023	Unassigned	
1024	Bogus Data	[This]
1025	Signature Missing	[This]
1026	DNSKEY Missing	[This]
1027	Key Tag Mismatch	[This]
1028	DS Missing	[This]
1029	Next-Secure Record Missing	[This]
1030	Overreaching Next-Secure Record	[This]
1031	Next-Secure Record Pointing Backward	[This]
1032	Irrelevant Proof	[This]
1033	Incomplete Proof	[This]
1034	Wrong RRSIG Owner	[This]
1035	Unknown DNSKEY Protocol	[This]
1036	DS/DNSKEY Mismatch	[This]
1037	Unknown Algorithm	[This]
1038	Algorithm Not Supported	[This]
1039	Not a Zone Key	[This]
1040	Key Revoked	[This]
1041-65535	Unassigned	

## 5. Acknowledgments

Thanks to Wes Hardaker, Jakob Schlyter, Sam Weiler and Francis Dupont for assistance in categorizing SERVFAIL error types, and Paul Vixie for design input.

## 6. References

### 6.1. Normative References

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2671] Vixie, P., "Extension Mechanisms for DNS (EDNS0)", RFC 2671, August 1999.



- [RFC2845] Vixie, P., Gudmundsson, O., Eastlake, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, May 2000.
- [RFC2931] Eastlake, D., "DNS Request and Transaction Signatures (SIG(0)s)", RFC 2931, September 2000.
- [RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.
- [RFC5011] StJohns, M., "Automated Updates of DNS Security (DNSSEC) Trust Anchors", RFC 5011, September 2007.
- [RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, March 2008.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC6672] Rose, S. and W. Wijngaards, "DNAME Redirection in the DNS", RFC 6672, June 2012.

## 6.2. Informative References

- [RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, March 2009.

## Author's Address

Evan Hunt  
ISC  
950 Charter St  
Redwood City, CA 94063  
USA

Email: [each@isc.org](mailto:each@isc.org)

