Network Working Group P. Sood
Internet-Draft Google
Intended status: Standards Track R. Arends
Expires: December 29, 2019 P. Hoffman
ICANN
June 27, 2019

DNS Resolver Information Self-publication
draft-sah-resolver-information-02

Abstract

This document describes methods for DNS resolvers to self-publish information about themselves, such as whether they perform DNSSEC validation or are available over transports other than what is defined in RFC 1035. The information is returned as a JSON object. The names in this object are defined in an IANA registry that allows for light-weight registration. Applications and operating systems can use the methods defined here to get the information from resolvers in order to make choices about how to send future queries to those resolvers.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Historically, DNS stub resolvers typically communicated with the recursive resolvers in their configuration without needing to know anything about the features of the recursive resolvers. More recently, recursive resolvers have different features that may cause stub resolvers to make choices about which configured resolver from its configuration to use, and also how to communicate with the recursive resolver (such as over different transports). Thus stub resolvers need a way to get information from recursive resolvers about features that might affect the communication.

This document specifies methods for stub resolvers to ask recursive resolvers for such information. In short, a new RRtype is defined for stub resolvers to query using the DNS, and a new well-known URI is defined for stub resolvers to query using HTTP over TLS.

The response from either method is the same: a JSON object. The JSON object MUST use the I-JSON message format defined in [RFC7493]. Note that [RFC7493] was based on RFC 7159, but RFC 7159 was replaced by

[RFC8259]. Requiring the use of I-JSON instead of more general JSON format greatly increases the likelihood of interoperability.

The information that a resolver might want to give to a recursive resolver is not defined in this document; instead other documents will follow that will specify that information and the format that it comes in.

It is important to note that the protocol defined here is only for recursive resolvers, not for authoritative servers. Authoritative servers MUST NOT answer queries that are defined in this protocol. (It is likely that a later protocol will allow authoritative servers to give information in a method similar to the one described in this document.)

1.1. Definitions

In the rest of this document, the term "resolver" without qualification means "recursive resolver" as defined in [RFC8499]. Also, the term "stub" is used to mean "stub resolver".

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Retrieving Resolver Information by DNS

A stub that wants to use the DNS to get information about a resolver can use the DNS query defined here. The query a stub resolver uses is <reverse-ip>.{in-addr,ip6}.arpa/IN/RESINFO. The RRtype "RESINFO" is defined in this document, and the IANA assignment is given in Section 5.1. The contents of the Rdata in the response to this query is defined in Section 4. If the resolver understands the RESINFO RRtype, the RRset in the Answer section MUST have exactly one record.

In this section, "<reverse-ip>.{in-addr,ip6}.arpa" is the domain name associated with the reverse lookup of an IP address of the resolver (resolvers can have multiple addresses). For example, if the resolver is at 192.0.2.1, the query would be 1.2.0.192.in-addr.arpa/IN/RESINFO.

A resolver that receives a query with the RRtype of RESINFO with a QNAME of <reverse-ip>.{in-addr,ip6}.arpa acts as if it is delegated, and responds with its own RESINFO data in the Answer section. The resolver can generate this reply with special code to capture queries for these types of addresses; if the resolver can be configured to

also be authoritative for some zones, it can use that configuration to actually be authoritative for the addresses on which it responds.

A stub that knows a specific type of information it wants MAY ask for that information by prepending a label with the name of the information in its query. For example, if the stub knows that it wants information whose name is "temp-field2", it would send the query temp-field2.<reverse-ip>.{in-addr,ip6}.arpa/IN/RESINFO. As described in Section 4, the JSON object in the response is likely to have name/value pairs in addition to the one requested.

Any query for the RESINFO RRtype that is not in <reverse-ip>.{in-addr,ip6}.arpa/IN or a subdomain of <reverse-ip>.{in-addr,ip6}.arpa/IN is meaningless and MUST result in a NODATA or NXDOMAIN response. Resolvers would not need any special code to meet this requirement; they only need code to handle the RESINFO RRtype that is not in <reverse-ip>.{in-addr,ip6}.arpa/IN or a subdomain of <reverse-ip>.{in-addr,ip6}.arpa/IN .

3. Retrieving Resolver Information by Well-Known URI

A stub that wants to use HTTPS to get information about a resolver can use the well-known URI defined here. Because this uses HTTPS, the stub has the possibility of authenticating the TLS connection. If the connection cannot be authenticated (such as if the stub only knows the IP address of the resolver and the resolver's certificate does not have the IP address, or the correct IP address), the stub MAY still use the results with the same lack of assuredness as it would have with using a DNS request described in Section 2.

The stub MUST use the HTTP GET method. The URI used to get the resolver information is one of:

https://IPADDRESSOFRESOLVER/.well-known/resolver-info/

https://DOMAINNAMEOFRESOLVER/.well-known/resolver-info/

This uses the ".well-known" URI mechanism defined in [RFC8615]. The contents of the response to this query is defined in Section 4.

A resolver that uses this protocol to publish its information SHOULD, if possible, have a TLS certificate whose subject identifiers are any IP address that the resolver is available on, as well as any domain names that the resolver operator uses for the resolver. At the time that this document is published, getting IP addresses in TLS certificates is possible, but there are only a few widely-trusted CAs that issue such certificates. [I-D.ietf-acme-ip] describes a new

protocol that may cause IP address certificates to become more common.

In the future, DHCP and/or DCHPv6 and/or RA may have options that allow the configuration to contain the domain name of a resolver. If so, this can be used for matching the domain name in the TLS certificate.

4. Contents of the Returned I-JSON Object

The JSON object returned by a DNS query or an HTTPS query MUST contain at least one name/value pair: "inventory", described later in this section. The returned object MAY contain any other name/value pairs.

The requirement for the inclusion of the "inventory" name/value pair is so that systems retrieving the information over DNS can create specific queries. Using specific queries can reduce the number of round trips in the case where the answers to queries become large. The "inventory" name/value pair MUST be included in the response even if the query was for a single name.

If the request was over DNS using a subdomain under <reverse-ip>.{in-addr,ip6}.arpa, the resolver SHOULD return an object that contains a name/value pair with that name if the resolver has that information. If the resolver does not have information for that name, it MUST NOT return the name in the object.

If the request was over HTTPS, the resolver SHOULD return an object with all known name/value pairs for which it has information.

All names in the returned object MUST either be defined in the IANA registry or, if for local use only, begin with the substring "temp-". The IANA registry (Section 5.2) will never register names that begin with "temp-".

All names MUST consist only of lower-case ASCII characters, digits, and hyphens (that is, Unicode characters U+0061 through 007A, U+0030 through U+0039, and U+002D), and MUST be 63 characters or shorter. As defined in Section 5.2, the IANA registry will not register names that begin with "temp-", so these names can be used freely by any implementer.

Note that the message returned by the resolver MUST be in I-JSON format. I-JSON requires that the message MUST be encoded in UTF8.

This document only defines one element that can returned: "inventory". All other elements will be defined in other documents.

4.1. The "inventory" name

The "inventory" name lists all of the types of information for which the resolver has data. The value is an array of strings.

4.2. Example

The I-JSON object that a resolver returns might look like the following:

```
{
   "temp-field2": 42,
   "temp-field1": [ "There is", "no \u000B!" ],
   "inventory": [ "inventory", "temp-field1", "temp-field2" ]
}
```

As specified in [RFC7493], the I-JSON object is encoded as UTF8. This example has no un-escaped non-ASCII characters only because they are not currently allowed in Internet Drafts. For example, the exclamation mark in the second name/value pair could instead be the double exclamation mark character, U+203C.

[RFC7493] explicitly allows the returned objects to be in any order.

5. IANA Considerations

5.1. RESINFO RRtype

This document defines a new DNS RR type, RESINFO, whose value TBD will be allocated by IANA from the "Resource Record (RR) TYPEs" sub-registry of the "Domain Name System (DNS) Parameters" registry:

Type: RESINFO

Value: TBD

Meaning: Information self-published by a resolver as an I-JSON (RFC 7493) object

Reference: This document

5.2. Registry for DNS Resolver Information

IANA will create a new registry titled "DNS Resolver Information" that will contain definitions of the names that can be used with the protocols defined in this document. The registration procedure is by Expert Review and Specification Required, as defined in [RFC8126].

The specification that is required for registration can be either an Internet-Draft or an RFC. The reviewer for this registry is instructed to generally be liberal in what they accept into the registry: as long as the specification that comes with the registration request is reasonably understandable, the registration should be accepted.

The registry has the following fields for each element:

Name: The name to be used in the JSON object. This name MUST NOT begin with "temp-". This name MUST conform to the definition of "string" in I-JSON [RFC7493] message format.

Value type: The type of data to be used in the JSON object.

Specification: The name of the specification for the registered element.

5.3. resolver-info Well-known URI

Before this draft is complete, mail will be sent to wellknown-uri-review@ietf.org in order to be registered in the "Well-Known URIs" registry at IANA. The mail will contain the following:

URI suffix: resolver-info

Change controller: IETF

Specification document(s): This document

Status: permanent

6. Security Considerations

Unless a DNS request for <reverse-ip>.{in-addr,ip6}.arpa/IN/RESINFO, or a subdomain, as described in Section 2 is sent over DNS-over-TLS (DoT) [RFC7858] or DNS-over-HTTPS (DoH) [RFC8484], or unless the <reverse-ip>.{in-addr,ip6}.arpa zone is signed with DNSSEC, the response is susceptible to forgery. Stubs and resolvers SHOULD use normal DNS methods for avoiding forgery such as query ID randomization and source port randomization. A stub resolver will know if it is using DoT or DoH, and if it is using DoT it will know if the communication is authenticated (DoH is always authenticated).

An application that is using an operating system API to send queries for <reverse-ip>.{in-addr,ip6}.arpa/IN/RESINFO or a subdomain will only know if query went over authenticated DoT or DoH if the API

supports returning that authentication information. Currently, no common APIs support that type of response.

7. References

7.1. Normative References

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

[RFC7493] Bray, T., Ed., "The I-JSON Message Format", RFC 7493, DOI 10.17487/RFC7493, March 2015, <https://www.rfc-editor.org/info/rfc7493>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

[RFC8499] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <https://www.rfc-editor.org/info/rfc8499>.

7.2. Informative References

[I-D.ietf-acme-ip] Shoemaker, R., "ACME IP Identifier Validation Extension", draft-ietf-acme-ip-06 (work in progress), May 2019.

[RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <https://www.rfc-editor.org/info/rfc7858>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://www.rfc-editor.org/info/rfc8484>.

[RFC8615] Nottingham, M., "Well-Known Uniform Resource Identifiers (URIs)", RFC 8615, DOI 10.17487/RFC8615, May 2019, <https://www.rfc-editor.org/info/rfc8615>.

Acknowledgments

The idea of various types of servers publishing information about themselves has been around for decades. However this idea has not been used in the DNS. This document aims to fix this omission.

Erik Kline suggested using "<reverse-ip>.{in-addr,ip6}.arpa" as the domain name to allow for the possibility of DNSSEC-signed responses.

Authors' Addresses

Puneet Sood
Google

Email: puneets@google.com

Roy Arends
ICANN

Email: roy.arends@icann.org

Paul Hoffman
ICANN

Email: paul.hoffman@icann.org