Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 16, 2015

M. Blanchet
Viagenie

October 13, 2014

Finding the Authoritative Registration Data (RDAP) Service
draft-ietf-weirds-bootstrap-08.txt

Abstract

This document specifies a method to find which Registration Data Access Protocol (RDAP) server is authoritative to answer queries for a requested scope, such as domain names, IP addresses or Autonomous System numbers.

Status of This Memo

This Internet-Draft will expire on April 16, 2015.

Copyright Notice

Table of Contents

1. Introduction

Querying and retrieving registration data from registries are defined in the Registration Data Access Protocol (RDAP) [I-D.ietf-weirds-rdap-query][I-D.ietf-weirds-using-http][I-D.ietf-weirds-json-response]. These documents do not specify where to send the queries. This document specifies a method to find which server is authoritative to answer queries for the requested scope.

TLDs, AS numbers, and network blocks are delegated by IANA to registrars that then issue further delegations and maintain information about them. Thus, obviously the bootstrap information needed by RDAP clients is best generated from data and processes already maintained by IANA, whose registries already exist at [ipv4reg], [ipv6reg], [asreg], and [domainreg].

The required new functionality in support of RDAP could be accomplished by augmenting these registries to contain new fields, or creating second parallel registries containing the extra fields whose entries mirror the existing ones. Either approach will satisfy the needs of this document. This document requests IANA to make these registries available for the specific purpose of RDAP use, herein named RDAP Bootstrap Service Registries. An RDAP client fetches the RDAP Bootstrap Service Registries, extracts the data and then does a match with the query data to find the authoritative registration data server and appropriate query base URL.

2. Conventions Used In This Document

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. Structure of RDAP Bootstrap Service Registries

The RDAP Bootstrap Service Registries, as specified in Section 12, will be made available as JSON [RFC7159] objects, to be retrieved via HTTP from a location as specified by IANA. The JSON object for each registry will start with a series of members that contain metadata about the registry such as a version identifier, a timestamp of the publication date of the registry and a description. Following that is a "services" member which contains the registry items themselves, as an array. Each item of the array contains a second-level array, with two elements, each of them being a third-level array.

There is no assumption of sorting except that the two arrays found in each second-level array MUST appear in the correct order: The entries array are followed by the service URL array. An example structure of the JSON output of a RDAP Bootstrap Service Registry is illustrated:

```
{
    "version": "1.0",
    "publication": "YYYY-MM-DDTHH:MM:SSZ",
    "description": "Some text",
    "services": [
      [
        ["entry1", "entry2", "entry3"],
        [
          "https://registry.example.com/myrdap/",
          "http://registry.example.com/myrdap/"
        ]
      ],
      [
        ["entry4"],
        [
          "http://example.org/"
        ]
      ]
    ]
}
```

The formal syntax is described in Section 10.

The "version" corresponds to the format version of the registry. This specification defines "1.0".

The syntax of "publication" value conforms to the Internet date/time format [RFC3339].

The optional "description" string can contain a comment regarding the content of the bootstrap object.

Per [RFC7258], in each array of base RDAP URLs, the secure versions of the transport protocol SHOULD be preferred and tried first. For example, if the base RDAP URLs array contain both https and http URLs, the bootstrap client SHOULD try the https version first.

Base RDAP URLs MUST have a trailing "/" character because they are concatenated to the various segments defined in [I-D.ietf-weirds-rdap-query].

JSON names MUST follow the format recommendations of [I-D.ietf-weirds-using-http]. Any unknown or unspecified JSON object properties or values should be ignored by implementers.

Internationalized Domain Names labels used as entries or base RDAP URLs in the registries defined in this document MUST be only represented using their A-Label form as defined in [RFC5890].

All Domain Names labels used as entries or base RDAP URLs in the registries defined in this document MUST be only represented in lowercase.

4. Domain Name RDAP Bootstrap Service Registry

The JSON output of this registry contains domain labels entries attached to the root, grouped by base RDAP URLs, as shown in this example.

```
{
    "version": "1.0",
    "publication": "YYYY-MM-DDTHH:MM:SSZ",
    "description": "Some text",
    "services": [
      [
        ["net", "com"],
        [
          "https://registry.example.com/myrdap/"
        ]
      ],
      [
        ["org", "mytld"],
        [
          "http://example.org/"
        ]
      ],
      [
        ["xn--zckzah"],
        [
          "https://example.net/rdapxn--zckzah/",
          "http://example.net/rdapxn--zckzah/"
        ]
      ]
    ]
}
```

The domain names authoritative registration data service is found by doing the label-wise longest match of the target domain name with the domain values in the arrays in the IANA Domain Name RDAP Bootstrap Service Registry. The values contained in the second element of the array are the valid base RDAP URLs as described in [I-D.ietf-weirds-rdap-query].

For example, a domain RDAP query for a.b.example.com matches the com entry in one of the arrays of the registry. The base RDAP URL for this query is then taken from the second element of the array, which is an array of base RDAP URLs valid for this entry. The client chooses one of the base URLs from this array; in this example it chooses the only one available, "https://registry.example.com/myrdap/". The segment specified in [I-D.ietf-weirds-rdap-query] is then appended to the base URL to complete the query. The complete query is then "https://registry.example.com/myrdap/domain/a.b.example.com".

If a domain RDAP query for a.b.example.com matches both com and example.com entries in the registry, then the longest match applies and the example.com entry is used by the client.

5. Internet Numbers RDAP Bootstrap Service Registries

This section discusses IPv4 and IPv6 address space and autonomous system numbers.

For IP address space, the authoritative registration data service is found by doing a longest match of the target address with the values of the arrays in the corresponding Address Space RDAP Bootstrap Service registry. The longest match is done the same way as for routing: the addresses are converted in binary form and then the binary strings are compared to find the longest match up to the specified prefix length. The values contained in the second element of the array are the base RDAP URLs as described in [I-D.ietf-weirds-rdap-query]. The longest match method enables covering prefixes of a larger address space pointing to one base RDAP URL while more specific prefixes within the covering prefix being served by another base RDAP URL.

5.1. IPv4 Address Space RDAP Bootstrap Service Registry

The JSON output of this registry contains IPv4 prefix entries, specified in CIDR format and grouped by RDAP URLs, as shown in this example.

```
{
    "version": "1.0",
    "publication": "2024-01-07T10:11:12Z",
    "description": "RDAP Bootstrap file for example registries.",
    "services": [
      [
        ["1.0.0.0/8", "192.0.0.0/8"],
        [
          "https://rir1.example.com/myrdap/"
        ]
      ],
      [
        ["28.2.0.0/16", "192.0.2.0/24"],
        [
          "http://example.org/"
        ]
      ],
      [
        ["28.3.0.0/16"],
        [
          "https://example.net/rdaprir2/",
          "http://example.net/rdaprir2/"
        ]
      ]
    ]
}
```

For example, a query for "192.0.2.1/25" matches the "192.0.0.0/8" entry and the "192.0.2.0/24" entry in the example registry above. The latter is chosen by the client given the longest match. The base RDAP URL for this query is then taken from the second element of the array, which is an array of base RDAP URLs valid for this entry. The client chooses one of the base URLs from this array; in this example it chooses the only one available, "http://example.org/". The {resource} specified in [I-D.ietf-weirds-rdap-query] is then appended to the base URL to complete the query. The complete query is then "https://example.org/ip/192.0.2.1/25".

5.2. IPv6 Address Space RDAP Bootstrap Service Registry

The JSON output of this registry contains IPv6 prefix entries, using [RFC4291] text representation of address prefixes format, grouped by base RDAP URLs, as shown in this example.

```
{
    "version": "1.0",
    "publication": "2024-01-07T10:11:12Z",
    "description": "RDAP Bootstrap file for example registries.",
    "services": [
      [
        ["2001:0200::/23", "2001:db8::/32"],
        [
          "https://rir2.example.com/myrdap/"
        ]
      ],
      [
        ["2600::/16", "2100:ffff::/32"],
        [
          "http://example.org/"
        ]
      ],
      [
        ["2001:0200:1000::/28"],
        [
          "https://example.net/rdaprir2/",
          "http://example.net/rdaprir2/"
        ]
      ]
    ]
}
```

For example, a query for "2001:0200:1000::/48" matches the "2001:0200::/23" entry and the "2001:0200:1000::/28" entry in the example registry above. The latter is chosen by the client given the longest match. The base RDAP URL for this query is then taken from the second element of the array, which is an array of base RDAP URLs valid for this entry. The client chooses one of the base URLs from this array; in this example it chooses "https://example.net/rdaprir2/" because it's the secure version of the protocol. The segment specified in [I-D.ietf-weirds-rdap-query] is then appended to the base URL to complete the query. The complete query is therefore "https://example.net/rdaprir2/ip/2001:0200:1000::/48". If the server does not answer, the client can then use another URL prefix from the array.

5.3. Autonomous Systems RDAP Bootstrap Service Registry

The JSON output of this contains Autonomous Systems Number Ranges entries, grouped by base RDAP URLs, as shown in this example. The first element of each second-level array is an array containing the list of AS number ranges served by the base RDAP URLs found in the second element. The array always contains two AS numbers which

represents the range of AS Numbers between the two elements of the array. When the two AS numbers are identical, then it only refers to that single AS number.

```
{
    "version": "1.0",
    "publication": "2024-01-07T10:11:12Z",
    "description": "RDAP Bootstrap file for example registries.",
    "services": [
      [
        ["2045-2045"],
        [
          "https://rir3.example.com/myrdap/"
        ]
      ],
      [
        ["10000-12000", "300000-400000"],
        [
          "http://example.org/"
        ]
      ],
      [
        ["64512-65534"],
        [
          "http://example.net/rdaprir2/",
          "https://example.net/rdaprir2/"
        ]
      ]
    ]
}
```

For example, a query for AS 65411 matches the 64512-65534 entry in the example registry above. The base RDAP URL for this query is then taken from the second element of the array, which is an array of base RDAP URLs valid for this entry. The client chooses one of the base URLs from this array; in this example it chooses "https://example.net/rdaprir2/". The segment specified in [I-D.ietf-weirds-rdap-query] is then appended to the base URL to complete the query. The complete query is therefore "https://example.net/rdaprir2/autnum/65411". If the server does not answer, the client can then use another URL prefix from the array.

6. Entity

Since there is no global namespace for entities, this document does not describe how to find the authoritative RDAP server for entities. It is possible however that, if the entity identifier was received from a previous query, the same RDAP server could be queried for that

entity or the entity identifier itself is a fully referenced URL that can be queried.

7. Non-existent Entries or RDAP URL Values

The registries may not contain the requested value or the base RDAP URL value may be empty. In these cases, there is no known RDAP server for that requested value and the client SHOULD provide an appropriate error message to the user.

8. Deployment and Implementation Considerations

This method relies on the fact that RDAP clients are fetching the IANA registries to then find the servers locally. Clients SHOULD NOT fetch the registry on every RDAP request. Clients SHOULD cache the registry, but use underlying protocol signalling, such as the HTTP Expires header field [RFC7234], to identify when it is time to refresh the cached registry.

If the query data does not match any entry in the client cached registry, then the client may implement various methods, such as the following:

- In the case of a domain object, the client may first query the DNS to see if the respective entry has been delegated or if it is a mistyped information by the user. The DNS query could be to fetch the NS records for the TLD domain. If the DNS answer is negative, then there is no need to fetch the new version of the registry. However, if the DNS answer is positive, this may mean that the currently cached registry is no longer current. The client could then fetch the registry, parse and then do the normal matching as specified above. This method may not work for all types of RDAP objects.

- If the client knows the existence of an RDAP aggregator or redirector and its associated base URL, and trusts that service, then it could send the query to the redirector, which would redirect the client if it knows the authoritative server that client has not found.

Some authorities of registration data may work together on sharing their information for a common service, including mutual redirection [I-D.ietf-weirds-redirects].

When a new object is allocated, such as a new AS range, a new TLD or a new IP address range, there is no guarantee that this new object will have an entry in the corresponding bootstrap RDAP registry, since the setup of the RDAP server for this new entry may become live

and registered later. Therefore, the clients should expect that even if an object, such as TLD, IP address range or AS range is allocated, the existence of the entry in the corresponding bootstrap registry is not guaranteed.

9. Limitations

This method does not provide a direct way to find authoritative RDAP servers for any other objects than the ones described in this document. In particular, the following objects are not bootstrapped with the method described in this document:

- entities

- queries using search patterns that do not contain a terminating string that matches some entries in the registries

- nameservers

- help

10. Formal Definition

This section is the formal definition of the registries. The structure of JSON objects and arrays using a set of primitive elements is defined in [RFC7159]. Those elements are used to describe the JSON structure of the registries.

10.1. Imported JSON Terms

- OBJECT: a JSON object, defined in Section 2.2 of [RFC7159]

- MEMBER: a member of a JSON object, defined in Section 2.2 of [RFC7159]

- MEMBER-NAME: the name of a MEMBER, defined as a "string" in Section 2.2 of [RFC7159]

- MEMBER-VALUE: the value of a MEMBER, defined as a "value" in Section 2.2 of [RFC7159]

- ARRAY: an array, defined in Section 2.3 of [RFC7159]

- ARRAY-VALUE: an element of an ARRAY, defined in Section 2.3 of [RFC7159]

- STRING: a "string" as defined in Section 2.5 of [RFC7159]

10.2. Registry Syntax

Using the above terms for the JSON structures, the syntax of a registry is defined as follows:

- rdap-bootstrap-registry: an OBJECT containing a MEMBER version and a MEMBER publication and a MEMBER description and a MEMBER services-list

- version: a MEMBER with MEMBER-NAME "version" and MEMBER-VALUE a STRING

- publication: a MEMBER with MEMBER-NAME "publication" and MEMBER-VALUE a STRING

- description: a MEMBER with MEMBER-NAME "description" and MEMBER-VALUE a STRING

- services-list: a MEMBER with MEMBER-NAME "services" and MEMBER-VALUE a services-array

- services-array: an ARRAY, where each ARRAY-VALUE is a service

- service: an ARRAY of 2 elements, where the first ARRAY-VALUE is a an entry-list and the second ARRAY-VALUE is a service-uri-list

- entry-list: an ARRAY, where each ARRAY-VALUE is a entry

- entry: a STRING

- service-uri-list: an ARRAY, where each ARRAY-VALUE is a service-uri

- service-uri: a STRING

11. Security Considerations

By providing a bootstrap method to find RDAP servers, this document helps to ensure that the end-users will get the RDAP data from an authoritative source, instead of from rogue sources. The method has the same security properties as the RDAP protocols themselves. The transport used to access the registries could be more secure by using TLS [RFC5246] if IANA supports it.

12. IANA Considerations

IANA is requested to make the RDAP Bootstrap Services Registries described below available as JSON objects, the syntax of which are described by section 10. The process for adding or updating entries into these registries does not correspond to the registration policies described in [RFC5226]; as stated earlier, these registries are generated from the data, processes, and policies maintained by IANA in their allocation registries ([ipv4reg], [ipv6reg], [asreg], and [domainreg]). IANA is expected to generate the RDAP Bootstrap Services Registries data from these above mentioned registries, according to their own registration policies. This document does not extend or otherwise change those policies.

Each of the RDAP Bootstrap Services Registries needs to be made available for general public on-demand download in the JSON format at a location determined by IANA.

IANA is also advised that the download demand for the RDAP Bootstrap Services Registries may be unusually high compared to other registries that exist already. The technical infrastructure by which registries are published may need to be reviewed.

Multiple entries pointing to the same set of URLs are grouped together in an array. Since multiple entries of non contiguous space may be grouped together, the registry may not be sortable by entries, therefore it is not required or expected that the entries be sorted in a registry.

12.1. IPv4 Address Space RDAP Bootstrap Service Registry

Entries in this registry contain at least the following:

- a CIDR specification of the network block being registered

- one or more URLs that provide the RDAP service regarding this registration.

12.2. IPv6 Address Space RDAP Bootstrap Service Registry

Entries in this registry contain at least the following:

- an IPv6 prefix [RFC4291] specification of the network block being registered

- one or more URLs that provide the RDAP service regarding this registration.

12.3. Autonomous System Number Space RDAP Bootstrap Service Registry

Entries in this registry contain at least the following:

- a range of Autonomous System numbers being registered

- one or more URLs that provide the RDAP service regarding this registration.

12.4. Domain Name Space RDAP Bootstrap Service Registry

Entries in this registry contain at least the following:

- a domain name attached to the root being registered

- one or more URLs that provide the RDAP service regarding this registration.

12.5. Additional Consideration

The HTTP Content-Type returned to clients accessing the JSON output of the registries MUST be "application/json" as defined in [RFC7159].

13. Acknowledgements

The WEIRDS working group had multiple discussions on this topic, including a session during IETF 84, where various methods such as in-DNS and others were debated. The idea of using IANA registries was discovered by the editor during discussions with his colleagues as well as by a comment from Andy Newton. All the people involved in these discussions are herein acknowledged. Linlin Zhou, Jean-Philippe Dionne, John Levine, Kim Davies, Ernie Dainow, Scott Hollenbeck, Arturo Servin, Andy Newton, Murray Kucherawy, Tom Harrison, Naoki Kambe, Alexander Mayrhofer, Edward Lewis, Pete Resnick have provided input and suggestions to this document. Guillaume Leclanche was a co-editor of this document for some revisions; his support is therein acknowledged and greatly appreciated. The section on formal definition was inspired by section 6.2 of [RFC7071].

14. References

14.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

[RFC7159] Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, March 2014.

14.2. Non-Normative References

[I-D.ietf-weirds-json-response]
Newton, A. and S. Hollenbeck, "JSON Responses for the Registration Data Access Protocol (RDAP)", draft-ietf-weirds-json-response-10 (work in progress), October 2014.

[I-D.ietf-weirds-rdap-query]
Newton, A. and S. Hollenbeck, "Registration Data Access Protocol Query Format", draft-ietf-weirds-rdap-query-15 (work in progress), October 2014.

[I-D.ietf-weirds-redirects]
Martinez, C., Zhou, L., and G. Rada, "Redirection Service for Registration Data Access Protocol", draft-ietf-weirds-redirects-04 (work in progress), July 2014.

[I-D.ietf-weirds-using-http]
Newton, A., Ellacott, B., and N. Kong, "HTTP usage in the Registration Data Access Protocol (RDAP)", draft-ietf-weirds-using-http-13 (work in progress), October 2014.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC7071] Borenstein, N. and M. Kucherawy, "A Media Type for Reputation Interchange", RFC 7071, November 2013.

[RFC7234] Fielding, R., Nottingham, M., and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Caching", RFC 7234, June 2014.

[RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, May 2014.

[asreg] Internet Assigned Numbers Authority(IANA), , "Autonomous System (AS) Numbers", <http://www.iana.org/assignments/as-numbers/as-numbers.xml>.

[domainreg] Internet Assigned Numbers Authority(IANA), , "Root Zone Database", <http://www.iana.org/domains/root/db>.

[ipv4reg] Internet Assigned Numbers Authority(IANA), , "IPv4 Address Space", <http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml>.

[ipv6reg] Internet Assigned Numbers Authority(IANA), , "IPv6 Global Unicast Address Assignments", <http://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xml>.

Author's Address

Marc Blanchet
Viagenie
246 Aberdeen
Quebec, QC G1R 2E1
Canada

Email: Marc.Blanchet@viagenie.ca
URI: http://viagenie.ca