            Registration Data Access Protocol (RDAP) Object Tagging

Abstract

   The Registration Data Access Protocol (RDAP) includes a method that
   can be used to identify the authoritative server for processing
   domain name, IP address, and autonomous system number queries. The
   method does not describe how to identify the authoritative server for
   processing other RDAP query types, such as entity queries. This
   limitation exists because the identifiers associated with these query
   types are typically unstructured. This document updates RFC 7484 by
   describing an operational practice that can be used to add structure
   to RDAP identifiers and that makes it possible to identify the
   authoritative server for additional RDAP queries.

Copyright Notice

Table of Contents

1. Introduction

   The Registration Data Access Protocol (RDAP) includes a method
   [RFC7484] that can be used to identify the authoritative server for
   processing domain name, IP address, and Autonomous System Number
   (ASN) queries. This method works because each of these data elements
   is structured in a way that facilitates automated parsing of the
   element and association of the data element with a particular RDAP
   service provider. For example, domain names include labels (such as
   "com", "net", and "org") that are associated with specific service
   providers.

   As noted in Section 9 of RFC 7484 [RFC7484], the method does not
   describe how to identify the authoritative server for processing
   entity queries, name server queries, help queries, or queries using
   certain search patterns. This limitation exists because the
   identifiers bound to these queries are typically not structured in a
   way that makes it easy to associate an identifier with a specific
   service provider. This document describes an operational practice
   that can be used to add structure to RDAP identifiers and makes it
   possible to identify the authoritative server for additional RDAP
   queries.

   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. Object Naming Practice

   Tagging object identifiers with a service provider tag makes it
   possible to identify the authoritative server for processing an RDAP
   query using the method described in RFC 7484 [RFC7484]. A service
   provider tag is constructed by prepending the Unicode HYPHEN-MINUS
   character "-" (U+002D, described as an "unreserved" character in RFC
   3986 [RFC3986]) to an IANA-registered value that represents the
   service provider. For example, a tag for a service provider
   identified by the string value "ARIN" is represented as "-ARIN".

   In combination with the rdapConformance attribute described in
   Section 4, service provider tags are concatenated to the end of RDAP
   query object identifiers to unambiguously identify the authoritative
   server for processing an RDAP query. Building on the example from
   Section 3.1.5 of RFC 7482 [RFC7482], an RDAP entity handle can be
   constructed to allow an RDAP client to bootstrap an entity query.

The following identifier is used to find information for the entity
associated with handle "XXXX" at service provider "ARIN":

    XXXX-ARIN

Clients that wish to bootstrap an entity query can parse this
identifier into distinct handle and service provider identifier
elements. Handles can themselves contain HYPHEN-MINUS characters;
the service provider identifier is found following the last HYPHEN-
MINUS character in the tagged identifier. The service provider
identifier is used to retrieve a base RDAP URL from an IANA registry.
The base URL and entity handle are then used to form a complete RDAP
query path segment. For example, if the base RDAP URL
"https://example.com/rdap/" is associated with service provider
"YYYY" in an IANA registry, an RDAP client will parse a tagged entity
identifier "XXXX-YYYY" into distinct handle ("XXXX") and service
provider ("YYYY") identifiers. The service provider identifier
"YYYY" is used to query an IANA registry to retrieve the base RDAP
URL "https://example.com/rdap/". The RDAP query URL is formed using
the base RDAP URL and entity path segment described in Section 3.1.5
of RFC 7482 [RFC7482] and using "XXXX-YYY" as the value of the handle
identifier. The complete RDAP query URL becomes
"https://example.com/rdap/entity/XXXX-YYYY".

Implementation of this practice requires tagging of unstructured
potential query identifiers in RDAP responses. Consider these elided
examples ("..." is used to note elided response objects) from
Section 5.3 of RFC 7483 [RFC7483] in which the handle identifiers
have been tagged with service provider tags "RIR", "DNR", and "ABC",
respectively:

```
{
  "objectClassName" : "domain",
  "handle" : "XXXX-RIR",
  "ldhName" : "0.2.192.in-addr.arpa",
  "nameservers" :
  [
    ...
  ],
  "secureDNS":
  {
    ...
  },
  "remarks" :
  [
    ...
  ],
  "links" :
```

```
      [
        ...
      ],
      "events" :
      [
        ...
      ],
      "entities" :
      [
        {
          "objectClassName" : "entity",
          "handle" : "XXXX-RIR",
          "vcardArray":
          [
            ...
          ],
          "roles" : [ "registrant" ],
          "remarks" :
          [
            ...
          ],
          "links" :
          [
            ...
          ],
          "events" :
          [
            ...
          ]
        }
      ],
      "network" :
      {
        "objectClassName" : "ip network",
        "handle" : "XXXX-RIR",
        "startAddress" : "192.0.2.0",
        "endAddress" : "192.0.2.255",
        "ipVersion" : "v4",
        "name": "NET-RTR-1",
        "type" : "DIRECT ALLOCATION",
        "country" : "AU",
        "parentHandle" : "YYYY-RIR",
        "status" : [ "active" ]
      }
    }
```

                              Figure 1

```
   {
     "objectClassName" : "domain",
     "handle" : "XXXX-YYY-DNR",
     "ldhName" : "xn--fo-5ja.example",
     "unicodeName" : "foo.example",
     "variants" :
     [
       ...
     ],
     "status" : [ "locked", "transfer prohibited" ],
     "publicIds":
     [
       ...
     ],
     "nameservers" :
     [
       {
         "objectClassName" : "nameserver",
         "handle" : "XXXX-DNR",
         "ldhName" : "ns1.example.com",
         "status" : [ "active" ],
         "ipAddresses" :
         {
           ...
         },
         "remarks" :
         [
           ...
         ],
         "links" :
         [
           ...
         ],
         "events" :
         [
           ...
         ]
       },
       {
         "objectClassName" : "nameserver",
         "handle" : "XXXX-DNR",
         "ldhName" : "ns2.example.com",
         "status" : [ "active" ],
         "ipAddresses" :
         {
           ...
         },
         "remarks" :
```

```
       [
         ...
       ],
       "links" :
       [
         ...
       ],
       "events" :
       [
         ...
       ]
    }
  ],
  "secureDNS":
  {
    ...
  },
  "remarks" :
  [
    ...
  ],
  "links" :
  [
    ...
  ],
  "port43" : "whois.example.net",
  "events" :
  [
    ...
  ],
  "entities" :
  [
    {
      "objectClassName" : "entity",
      "handle" : "XXXX-ABC",
      "vcardArray":
      [
        ...
      ],
      "status" : [ "validated", "locked" ],
      "roles" : [ "registrant" ],
      "remarks" :
      [
        ...
      ],
      "links" :
      [
        ...
```

```
      ],
      "events" :
      [
        ...
      ]
    }
  ]
}
```

                              Figure 2

   As described in Section 5 of RFC 7483 [RFC7483], RDAP responses can
   contain "self" links. Service provider tags and self references
   SHOULD be consistent. If they are inconsistent, the service provider
   tag is processed with higher priority when using these values to
   identify a service provider.

   There is a risk of unpredictable processing behavior if the HYPHEN-
   MINUS character is used for naturally occurring, non-separator
   purposes in an entity handle. This could lead to a client mistakenly
   assuming that a HYPHEN-MINUS character represents a separator and
   that the text that follows HYPHEN-MINUS is a service provider
   identifier. A client that queries the IANA registry for what they
   assume is a valid service provider will likely receive an unexpected,
   invalid result. As a consequence, use of the HYPHEN-MINUS character
   as a service provider tag separator MUST be noted by adding an
   rdapConformance value to query responses as described in Section 4.

   The HYPHEN-MINUS character was chosen as a separator for two reasons:
   1) it is a familiar separator character in operational use, and 2) it
   avoids collision with URI-reserved characters. The list of
   unreserved characters specified in Section 2.3 of RFC 3986 [RFC3986]
   provided multiple options for consideration:

      unreserved = ALPHA / DIGIT / "-" / "." / "_" / "~"

   ALPHA and DIGIT characters were excluded because they are commonly
   used in entity handles for non-separator purposes. HYPHEN-MINUS is
   commonly used as a separator, and recognition of this practice will
   reduce implementation requirements and operational risk. The
   remaining characters were excluded because they are not broadly used
   as separators in entity handles.

3. Bootstrap Service Registry for Provider Object Tags

   The bootstrap service registry for the RDAP service provider space is
   represented using the structure specified in Section 3 of RFC 7484
   [RFC7484]. The JSON output of this registry contains contact
   information for the registered service provider identifiers,
   alphanumeric identifiers that identify RDAP service providers, and
   base RDAP service URLs as shown in this example.

```
{
  "version": "1.0",
  "publication": "YYYY-MM-DDTHH:MM:SSZ",
  "description": "RDAP bootstrap file for service provider object tags",
  "services": [
    [
      ["contact@example.com"],
      ["YYYY"],
      [
        "https://example.com/rdap/"
      ]
    ],
    [
      ["contact@example.org"],
      ["ZZ54"],
      [
        "http://rdap.example.org/"
      ]
    ],
    [
      ["contact@example.net"],
      ["1754"],
      [
        "https://example.net/rdap/",
        "http://example.net/rdap/"
      ]
    ]
  ]
 }
```

                              Figure 3

   Alphanumeric service provider identifiers conform to the suffix
   portion ("\w{1,8}") of the "roidType" syntax specified in Section 4.2
   of RFC 5730 [RFC5730].

3.1. Registration Procedure

   The service provider registry is populated using the "First Come
   First Served" policy defined in RFC 8126 [RFC8126]. Provider
   identifier values can be derived and assigned by IANA on request.
   Registration requests include an email address to be associated with
   the registered service provider identifier, the requested service
   provider identifier (or an indication that IANA should assign an
   identifier), and one or more base RDAP URLs to be associated with the
   service provider identifier.

4. RDAP Conformance

   RDAP responses that contain values described in this document MUST
   indicate conformance with this specification by including an
   rdapConformance [RFC7483] value of "rdap_objectTag_level_0". The
   information needed to register this value in the "RDAP Extensions"
   registry is described in Section 5.2.

   The following is an example rdapConformance structure with the
   extension specified.

```
            "rdapConformance" :
            [
              "rdap_level_0",
              "rdap_objectTag_level_0"
            ]
```

                            Figure 4

5. IANA Considerations

   IANA has created the RDAP "Bootstrap Service Registry for Provider
   Object Tags" listed below and made it available as a JSON object.
   The contents of this registry are described in Section 3; the formal
   syntax is specified in Section 10 of RFC 7484 [RFC7484].

5.1. Bootstrap Service Registry Structure

   Entries in this registry contain the following information:

   o an email address that identifies a contact associated with the
      registered RDAP service provider value.
   o an alphanumeric value that identifies the RDAP service provider
      being registered.
   o one or more URLs that provide the RDAP service regarding this
      registration. The URLs are expected to supply the same data, but
      they can differ in scheme or other components as required by the
      service operator.

5.2. RDAP Extensions Registry

   IANA has registered the following value in the "RDAP Extensions"
   registry:

      Extension identifier: rdap_objectTag
      Registry operator: Any
      Published specification: This document
      Contact: IESG <iesg@ietf.org>

      Intended usage: This extension describes a best practice for
      structuring entity identifiers to enable query bootstrapping.

6. Security Considerations

   This practice uses IANA as a well-known, centrally trusted authority
   to allow users to get RDAP data from an authoritative source, which
   reduces the risk of sending queries to non-authoritative sources and
   divulging query information to unintended parties. Using TLS 1.2
   [RFC5246] or TLS 1.3 [RFC8446], which obsoletes TLS 1.2, to protect
   the connection to IANA allows the server to authenticate itself as
   being operated by IANA and provides integrity protection for the
   resulting referral information, as well as provides privacy
   protection via data confidentiality. The subsequent RDAP connection
   is performed as usual and retains the same security properties of the
   RDAP protocols themselves as described in RFC 7481 [RFC7481].

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>.

   [RFC5730]   Hollenbeck, S., "Extensible Provisioning Protocol (EPP)",
               STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009,
               <https://www.rfc-editor.org/info/rfc5730>.

   [RFC7484]   Blanchet, M., "Finding the Authoritative Registration Data
               (RDAP) Service", RFC 7484, DOI 10.17487/RFC7484, March
               2015, <https://www.rfc-editor.org/info/rfc7484>.

   [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>.

   [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>.

7.2. Informative References

   [RFC3986]   Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform
               Resource Identifier (URI): Generic Syntax", STD 66,
               RFC 3986, DOI 10.17487/RFC3986, January 2005,
               <https://www.rfc-editor.org/info/rfc3986>.

   [RFC5246]   Dierks, T. and E. Rescorla, "The Transport Layer Security
               (TLS) Protocol Version 1.2", RFC 5246,
               DOI 10.17487/RFC5246, August 2008,
               <https://www.rfc-editor.org/info/rfc5246>.

   [RFC7481]   Hollenbeck, S. and N. Kong, "Security Services for the
               Registration Data Access Protocol (RDAP)", RFC 7481,
               DOI 10.17487/RFC7481, March 2015,
               <https://www.rfc-editor.org/info/rfc7481>.

   [RFC7482]   Newton, A. and S. Hollenbeck, "Registration Data Access
               Protocol (RDAP) Query Format", RFC 7482,
               DOI 10.17487/RFC7482, March 2015,
               <https://www.rfc-editor.org/info/rfc7482>.

   [RFC7483]   Newton, A. and S. Hollenbeck, "JSON Responses for the
               Registration Data Access Protocol (RDAP)", RFC 7483,
               DOI 10.17487/RFC7483, March 2015,
               <https://www.rfc-editor.org/info/rfc7483>.

   [RFC8446]   Rescorla, E., "The Transport Layer Security (TLS) Protocol
               Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018,
               <https://www.rfc-editor.org/info/rfc8446>.

Acknowledgements

Authors' Addresses

   Scott Hollenbeck
   Verisign Labs
   12061 Bluemont Way
   Reston, VA 20190
   United States of America

   Email: shollenbeck@verisign.com
   URI: http://www.verisignlabs.com/

   Andrew Lee Newton
   American Registry for Internet Numbers
   PO Box 232290
   Centreville, VA 20120
   United States of America

   Email: andy@arin.net
   URI: http://www.arin.net