Registration Protocols Extensions
Internet-Draft
Intended status: Standards Track
Expires: 16 May 2024

M. Loffredo
M. Martinelli
IIT-CNR/Registro.it

13 November 2023

Registration Data Access Protocol (RDAP) Reverse Search
draft-ietf-regext-rdap-reverse-search-26

Abstract

The Registration Data Access Protocol (RDAP) does not include query capabilities for finding the list of domains related to a set of entities matching a given search pattern. Considering that an RDAP entity can be associated with any defined object class and other relationships between RDAP object classes exist, a reverse search can be applied to other use cases besides the classic domain-entity scenario. This document describes an RDAP extension that allows servers to provide a reverse search feature based on the relationship defined in RDAP between an object class for search and any related object class. The reverse search based on the domain-entity relationship is treated as a particular case.

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 https://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 16 May 2024.

Copyright Notice

Table of Contents

1. Introduction 3
1.1. Background 3
1.2. Conventions Used in This Document 4
2. Reverse Search Path Segment Specification 4
3. Reverse Search Definition 5
4. Reverse Search Properties Discovery 5
5. Reverse Search Properties Mapping 6
6. Reverse Search Response Specification 7
7. Reverse Search Query Processing 7
8. Reverse Searches Based on Entity Details 8
9. RDAP Conformance 10
10. Implementation Considerations 10
11. IANA Considerations 10
11.1. RDAP Extensions Registry 11
11.2. RDAP Reverse Search Registries 11
11.2.1. Creation of the RDAP Reverse Search Registries . . . 11
11.2.2. Submit Request to IANA 11
11.2.3. RDAP Reverse Search Registry 11
11.2.3.1. Template 11
11.2.3.2. Initial Content 12
11.2.4. RDAP Reverse Search Mapping Registry 13
11.2.4.1. Template 13
11.2.4.2. Initial Content 14
12. Privacy Considerations 15
13. Security Considerations 15
14. Acknowledgements 16
15. References 16
15.1. Normative References 16
15.2. Informative References 17
Appendix A. Paradigms to Enforce Access Control on Reverse Search in RDAP 18
Authors' Addresses 20

1. Introduction

The protocol described in this specification aims to extend the RDAP query capabilities and response to enable reverse search based on the relationships defined in RDAP between an object class for search and a related object class. The reverse search based on the domain-entity relationship is treated as a particular case of such a generic model.

RDAP providers willing to implement this specification should carefully consider its implications on the efficiency (see Section 10), the security (see Section 13) and the compliance with privacy regulations (see Section 12) of their RDAP service.

1.1. Background

Reverse Whois is a service provided by many web applications that allows users to find domain names owned by an individual or a company starting from the owner's details, such as name and email. Even if it has been considered useful for some legal purposes (e.g. uncovering trademark infringements, detecting cybercrimes), its availability as a standardized Whois [RFC3912] capability has been objected to for two main reasons, which now don't seem to conflict with an RDAP implementation.

The first objection concerns the potential risks of privacy violation. However, the domain name community is considering a new generation of Registration Directory Services [ICANN-RDS1] [ICANN-RDS2] [ICANN-RA], which provide access to sensitive data under some permissible purposes and in accordance with appropriate policies for requestor accreditation, authentication and authorization. RDAP's reliance on HTTP means that it can make use of common HTTP-based approaches to authentication and authorization, making it more useful than Whois in the context of such directory services. Since RDAP consequently permits a reverse search implementation complying with privacy protection principles, this first objection is not well-founded.

The second objection to the implementation of a reverse search capability has been connected with its impact on server processing. However, the core RDAP specifications already define search queries, with similar processing requirements, so the basis of this objection is not clear.

Reverse searches, such as finding the list of domain names associated with contacts or nameservers, may be useful to registrars as well. Usually, registries adopt out-of-band solutions to provide results to registrars asking for reverse searches on their domains. Possible reasons for such requests are:

* the loss of synchronization between the registrar database and the registry database;
* the need for such data to perform bulk Extensible Provisioning Protocol (EPP) [RFC5730] updates (e.g. changing the contacts of a set of domains, etc.).

Currently, RDAP does not provide any means for a client to search for the collection of domains associated with an entity [RFC9082]. A query (lookup or search) on domains can return the array of entities related to a domain with different roles (registrant, registrar, administrative, technical, reseller, etc.), but the reverse operation is not allowed. Only reverse searches to find the collection of domains related to a nameserver (ldhName or ip) can be requested. Since an entity can be in relationship with any RDAP object [RFC9083], the availability of a reverse search as largely intended can be common to all the object classes allowed for search. Through a further step of generalization, the meaning of reverse search in the RDAP context can be extended to include any query for retrieving all the objects in relationship with another matching a given search pattern.

1.2. Conventions Used in This Document

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. Reverse Search Path Segment Specification

A generic reverse search path is described by the syntax:

{searchable-resource-type}/reverse_search/{related-resource-type}?<search-condition>

The path segments are defined as in the following:

"searchable-resource-type": it MUST be one of the resource types for search defined in Section 3.2 of [RFC9082] (i.e. "domains", "nameservers" and "entities") or a resource type extension;

"related-resource-type": it MUST be one of the resource types for lookup defined in Section 3.1 of [RFC9082] (i.e. "domain", "nameserver", "entity", "ip" and "autnum") or a resource type extension;

"search-condition": a sequence of "property=search pattern" predicates separated by the ampersand character ('&', US-ASCII value 0x0026).

While related-resource-type is defined as having one of a number of different values, the only reverse searches defined in this document are for a related-resource-type of "entity". Reverse searches for the other resource types specified in [RFC9082] and resource type extensions may be defined by future documents.

3. Reverse Search Definition

Based on the content of Section 2, defining a reverse search means to define the triple <searchable resource type, related resource type, property> and the mapping with the corresponding RDAP object member. The mapping is done through the use of a JSONPath expression [I-D.ietf-jsonpath-base]. Reverse searches are registered in the Reverse Search registry (see Section 11.2.3), whereas reverse search mappings are registered in the Reverse Search Mapping registry (see Section 11.2.4). The reason for having two registries is that it may be possible for a single type of reverse search to rely on different members, depending on the server's configuration (see Section 5).

All of the reverse searches defined by this document (see Section 8) have property names that are the same as the name of the RDAP object member that is the subject of the search. For example, the reverse search with the property name "fn" relies on the value of the "fn" member inside the jCard of an entity object. However, it is not necessary that these two names be the same. In particular, remapping of searches as part of the deprecation of an existing member (see Section 5) will typically lead to a member with a different name being used for the search.

Servers MUST NOT provide or implement reverse searches or reverse search mappings that are not registered with IANA.

4. Reverse Search Properties Discovery

Servers complying with this specification MUST extend the help response [RFC9083] with the "reverse_search_properties" member which contains an array of objects with the following mandatory child members:

"searchableResourceType": the searchable resource type of the reverse search query as defined in Section 2;

"relatedResourceType": the related resource type of the reverse search query as defined in Section 2;

"property": the reverse search property used in the predicate of the reverse search query as defined in Section 2;

An example of the help response including the "reverse_search_properties" member is shown in Figure 2.

5. Reverse Search Properties Mapping

To permit clients to determine the member used by the server for a reverse search, servers MUST detail the mapping that is occurring by adding the "reverse_search_properties_mapping" member to the topmost object of a reverse search response. This data is included in the search response, rather than in the help response, because it may differ depending on the query that is sent to the server.

Documents that deprecate or restructure RDAP responses such that a registered reverse search is no longer able to be used MUST either note that the relevant reverse search is no longer available (in the case of deprecation) or describe how to continue supporting the relevant search by adding another mapping for the reverse search property (in the case of restructuring).

The "reverse_search_properties_mapping" member contains an array of objects with the following mandatory child members:

"property": the reverse search property used in the predicate of the current query as defined in Section 2;

"propertyPath": the JSONPath expression of the object member (or members) corresponding to the reverse search property.

The searchable and the related resource types are derived from the query, so there is no need to include them in addition to the property in this member.

This member MUST be included for all properties used in the search, regardless of whether that property has multiple registered mappings as at the time of the search, because new mappings may be registered at any time.

When applied to an object, the JSONPath expression MUST produce a list of values, each of which is a JSON number or string.

An example of a reverse search response including the "reverse_search_properties_mapping" member is shown in Figure 3.

6. Reverse Search Response Specification

Reverse search responses use the formats defined in section 8 of [RFC9083], which correspond to the searchable resource types defined in Section 2.

7. Reverse Search Query Processing

To process a reverse search, the server returns the objects from its data store that are of type searchable-resource-type and that match each of the predicates from the search conditions. To determine whether an object matches a predicate, the server:

* applies the mapping it uses for the reverse search property to the object in order to generate a list of values, each of which MUST be a JSON number or string; and
* checks whether the search pattern matches one or more of those values.

A search pattern matches a value where it equals the string representation of the value, or where it is a match for the value in accordance with the partial string matching behaviour defined in section 4.1 of [RFC9082].

Objects are only included in the search results if they satisfy all included predicates. This includes predicates that are for the same property: it is necessary in such a case for the related object to match against each of those predicates.

Servers MUST return an HTTP 501 (Not Implemented) [RFC9110] response to inform clients of unsupported reverse searches.

Based on their policy, servers MAY restrict how predicates are used to make a valid search condition, by returning a 400 (Bad Request) response when a problematic request is received.

A given reverse search or reverse search mapping MAY define additional or alternative search behaviour past that set out in this section.

8. Reverse Searches Based on Entity Details

Since in RDAP, an entity can be associated with any other object class, the most common kind of reverse search is one based on an entity's details. Such reverse searches arise from the query model by setting the related resource type to "entity".

By selecting a specific searchable resource type, the resulting reverse search aims at retrieving all the objects (e.g. all the domains) that are related to any entity object matching the search conditions.

This section defines the reverse search properties servers SHOULD support for the domain, nameserver, and entity searchable resource types and the entity related resource type:

Reverse search property: role
RDAP member path: $.entities[*].roles
Reference: Section 10.2.4 of [RFC9083]

Reverse search property: handle
RDAP member path: $.entities[*].handle
Reference: Section 5.1 of [RFC9083]

Reverse search property: fn
RDAP member path: $.entities[*].vcardArray[1][?(@[0]=='fn')][3]
Reference: Section 6.2.1 of [RFC6350]

Reverse search property: email
RDAP member path: $.entities[*].vcardArray[1][?(@[0]=='email')][3]
Reference: Section 6.4.2 of [RFC6350]

The presence of a predicate on the reverse search property "role" means that the RDAP response property "roles" MUST contain at least the specified role.

The last two properties are related to jCard elements [RFC7095], but the field references are to vCard [RFC6350], since jCard is the JSON format for vCard.

Examples of reverse search paths based on the domain-entity relationship are presented in Figure 1.

/domains/reverse_search/entity?handle=CID-40*&role=technical

/domains/reverse_search/entity?fn=Bobby*&role=registrant

/domains/reverse_search/entity?handle=RegistrarX&role=registrar

Figure 1: Examples of reverse search queries

An example of the help response including the reverse search properties supported is shown below.

```
{
  "rdapConformance": [
    "rdap_level_0",
    "reverse_search"
  ],
  ...
  "reverse_search_properties": [
    {
      "searchableResourceType": "domains",
      "relatedResourceType": "entity",
      "property": "fn"
    },
    {
      "searchableResourceType": "domains",
      "relatedResourceType": "entity",
      "property": "handle"
    },
    {
      "searchableResourceType": "domains",
      "relatedResourceType": "entity",
      "property": "email"
    },
    {
      "searchableResourceType": "domains",
      "relatedResourceType": "entity",
      "property": "role"
    }
  ],
  ...
}
```

Figure 2: An example of help response including the "reverse_search_properties_mapping" member

An example of a response including the mapping that is occurring for the first reverse search in Figure 1 is shown below.

```
{
  "rdapConformance": [
    "rdap_level_0",
    "reverse_search"
  ],
  ...
  "reverse_search_properties_mapping": [
    {
      "property": "handle",
      "propertyPath": "$.entities[*].handle"
    },
    {
      "property": "role",
      "propertyPath": "$.entities[*].roles"
    }
  ],
  ...
}
```

Figure 3: An example of an RDAP response including the "reverse_search_properties" member

9. RDAP Conformance

Servers complying with this specification MUST include the value "reverse_search" in the rdapConformance property of the help response [RFC9083] and any other response including the "reverse_search_properties_mapping" member. The information needed to register this value in the "RDAP Extensions" registry is described in Section 11.1.

10. Implementation Considerations

To limit the impact of processing the search predicates, servers are RECOMMENDED to make use of techniques to speed up the data retrieval in their underlying data store such as indexes or similar. In addition, risks with respect to performance degradation or result set generation can be mitigated by adopting practices used for standard searches, e.g. restricting the search functionality, limiting the rate of search requests according to the user's authorization, truncating and paging the results [RFC8977], and returning partial responses [RFC8982].

11. IANA Considerations

11.1. RDAP Extensions Registry

IANA is requested to register the following value in the "RDAP Extensions" registry:

* Extension identifier: reverse_search
* Registry operator: Any
* Published specification: This document.
* Contact: IETF <iesg@ietf.org>
* Intended usage: This extension identifier is used for both URI path segments and response extensions related to the reverse search in RDAP.

11.2. RDAP Reverse Search Registries

11.2.1. Creation of the RDAP Reverse Search Registries

IANA is requested to create the "RDAP Reverse Search" and "RDAP Reverse Search Mapping" registries within the group "Registration Data Access Protocol (RDAP)".

These registries follow the Specification Required process as defined in Section 4.5 of [RFC8126].

The designated expert should prevent collisions and confirm that suitable documentation, as described in Section 4.6 of [RFC8126], is available to ensure interoperability.

Creators of either new RDAP reverse searches or new mappings for registered reverse searches SHOULD NOT replicate functionality already available by way of other documents referenced in these registries. Creators MAY register additional reverse search mappings for existing properties, but they SHOULD NOT map a registered reverse search property to a response field with a meaning other than that of the response fields referenced by the mappings already registered for that property. In other words, all the mappings for a reverse search property MUST point to response fields with the same meaning.

11.2.2. Submit Request to IANA

Registration requests can be sent to <iana@iana.org>.

11.2.3. RDAP Reverse Search Registry

11.2.3.1. Template

"Searchable Resource Type": The searchable resource type of the

reverse search query (Section 2) including the reverse search property. Multiple reverse search properties differing only by this field can be grouped together by listing all the searchable resource types separated by comma (see Section 11.2.3.2).

"Related Resource Type": The related resource type of the reverse search query (Section 2) including the reverse search property.

"Property": The name of the reverse search property.

"Description": A brief human-readable text describing the reverse search property.

"Registrant Name": The name of the person registering the reverse search property.

"Registrant Contact Information": An email address, postal address, or some other information to be used to contact the registrant.

"Reference": Document (e.g. the RFC number) and section reference where the reverse search property is specified.

The combination of "Searchable Resource Type", "Related Resource Type" and "Property" MUST be unique across the registry entries.

11.2.3.2. Initial Content

IANA is requested to register the following entries in the "RDAP Reverse Search" registry.

For all entries, the common values are shown in Table 1 whereas the specific values are shown in Table 2.

Registry Property	Value
Searchable Resource Type	domains, nameservers, entities
Related Resource Type	entity
Registrant Name	IETF
Registrant Contact Information	iesg@ietf.org
Reference	This document, Section 8

Table 1: Common values for all entries in the "RDAP Reverse Search" registry

Property	Description
fn	The server supports the domain/nameserver/ entity search based on the full name (a.k.a. formatted name) of an associated entity
handle	The server supports the domain/nameserver/ entity search based on the handle of an associated entity
email	The server supports the domain/nameserver/ entity search based on the email address of an associated entity
role	The server supports the domain/nameserver/ entity search based on the role of an associated entity

Table 2: Specific values for all entries in the "RDAP Reverse Search" registry

11.2.4. RDAP Reverse Search Mapping Registry

11.2.4.1. Template

"Searchable Resource Type": The same as defined in the "Reverse Search Registry".

"Related Resource Type": The same as defined in the "Reverse Search

Registry".

"Property": The same as defined in the "Reverse Search Registry".

"Property Path": The JSONPath of the RDAP property this reverse search property maps to.

"Registrant Name": The name of the person registering this reverse search property mapping.

"Registrant Contact Information": The same as defined in the "Reverse Search Registry".

"Reference": Document (e.g. the RFC number) and section reference where this reverse search property mapping is specified.

The combination of "Searchable Resource Type", "Related Resource Type", "Property" and "Property Path" MUST be unique across the registry entries.

11.2.4.2. Initial Content

IANA is requested to register the following entries in the "RDAP Reverse Search Mapping" registry.

For all entries, the common values are the same as defined in the "RDAP Reverse Search" registry (see Table 1) whereas the specific values are shown in Table 3.

Property	Property Path
fn	$.entities[*].vcardArray[1][?(@[0]=='fn')][3]
handle	$.entities[*].handle
email	$.entities[*].vcardArray[1][?(@[0]=='email')][3]
role	$.entities[*].roles

Table 3: Specific values for all entries in the "RDAP Reverse Search Mapping" registry

12. Privacy Considerations

The search functionality defined in this document may affect the privacy of entities in the registry (and elsewhere) in various ways: see [RFC6973] for a general treatment of privacy in protocol specifications. Registry operators should be aware of the tradeoffs that result from implementation of this functionality.

Many jurisdictions have laws or regulations that restrict the use of "Personal Data", per the definition in [RFC6973]. Given that, registry operators should ascertain whether the regulatory environment in which they operate permits implementation of the functionality defined in this document.

In those cases where this functionality makes use of sensitive information, it MUST only be accessible to authorized users supported by lawful basis.

Since reverse search requests and responses could contain Personally Identifiable Information (PII), reverse search functionality MUST be available over HTTPS only.

Providing reverse search in RDAP carries the following threats as described in [RFC6973]:

* Correlation
* Disclosure
* Misuse of information

Therefore, RDAP providers need to mitigate the risk of those threats by implementing appropriate measures supported by security services (see Section 13).

13. Security Considerations

Security services required to provide controlled access to the operations specified in this document are described in [RFC7481]. A non-exhaustive list of access control paradigms an RDAP provider can implement is presented in Appendix A.

As an additional measure to enforce security by preventing reverse searches to be accessed from unauthorized users, the RDAP providers may consider to physically separate the reverse search endpoints from the other ones by configuring a proxy routing the reverse searches to a dedicated backend server and leveraging further security services offered by other protocol layers such as digital certificates and IP whitelisting.

Finally, the specification of the relationship within the reverse search path allows the RDAP servers to implement different authorization policies on a per-relationship basis.

14. Acknowledgements

The authors would like to acknowledge the following individuals for their contributions to this document: Francesco Donini, Scott Hollenbeck, Francisco Arias, Gustavo Lozano, Eduardo Alvarez, Ulrich Wisser, James Gould and Pawel Kowalik.

Tom Harrison and Jasdip Singh provided relevant feedback and constant support to the implementation of this proposal. Their contributions have been greatly appreciated.

15. References

15.1. Normative References

[I-D.ietf-jsonpath-base]
Gössner, S., Normington, G., and C. Bormann, "JSONPath: Query expressions for JSON", Work in Progress, Internet-Draft, draft-ietf-jsonpath-base-21, 24 September 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-jsonpath-base-21>.

[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>.

[RFC6350] Perreault, S., "vCard Format Specification", RFC 6350, DOI 10.17487/RFC6350, August 2011, <https://www.rfc-editor.org/info/rfc6350>.

[RFC7095] Kewisch, P., "jCard: The JSON Format for vCard", RFC 7095, DOI 10.17487/RFC7095, January 2014, <https://www.rfc-editor.org/info/rfc7095>.

[RFC7481] Hollenbeck, S. and N. Kong, "Security Services for the Registration Data Access Protocol (RDAP)", STD 95, RFC 7481, DOI 10.17487/RFC7481, March 2015, <https://www.rfc-editor.org/info/rfc7481>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <https://www.rfc-editor.org/info/rfc7942>.

[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>.

[RFC9082] Hollenbeck, S. and A. Newton, "Registration Data Access Protocol (RDAP) Query Format", STD 95, RFC 9082, DOI 10.17487/RFC9082, June 2021, <https://www.rfc-editor.org/info/rfc9082>.

[RFC9083] Hollenbeck, S. and A. Newton, "JSON Responses for the Registration Data Access Protocol (RDAP)", STD 95, RFC 9083, DOI 10.17487/RFC9083, June 2021, <https://www.rfc-editor.org/info/rfc9083>.

[RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/info/rfc9110>.

15.2. Informative References

[ICANN-RA] Internet Corporation For Assigned Names and Numbers, "Registry Agreement", July 2017, <https://newgtlds.icann.org/sites/default/files/agreements/agreement-approved-31jul17-en.pdf>.

[ICANN-RDS1] Internet Corporation For Assigned Names and Numbers, "Final Report from the Expert Working Group on gTLD Directory Services: A Next-Generation Registration Directory Service (RDS)", June 2014, <https://www.icann.org/en/system/files/files/final-report-06jun14-en.pdf>.

[ICANN-RDS2] Internet Corporation For Assigned Names and Numbers, "Final Issue Report on a Next-Generation gTLD RDS to Replace WHOIS", October 2015, <http://whois.icann.org/sites/default/files/files/final-issue-report-next-generation-rds-07oct15-en.pdf>.

[OIDCC] OpenID Foundation, "OpenID Connect Core incorporating errata set 1", November 2014, <http://openid.net/specs/openid-connect-core-1_0.html>.

[RFC3912] Daigle, L., "WHOIS Protocol Specification", RFC 3912, DOI 10.17487/RFC3912, September 2004, <https://www.rfc-editor.org/info/rfc3912>.

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

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/info/rfc6973>.

[RFC8977] Loffredo, M., Martinelli, M., and S. Hollenbeck, "Registration Data Access Protocol (RDAP) Query Parameters for Result Sorting and Paging", RFC 8977, DOI 10.17487/RFC8977, January 2021, <https://www.rfc-editor.org/info/rfc8977>.

[RFC8982] Loffredo, M. and M. Martinelli, "Registration Data Access Protocol (RDAP) Partial Response", RFC 8982, DOI 10.17487/RFC8982, February 2021, <https://www.rfc-editor.org/info/rfc8982>.

Appendix A. Paradigms to Enforce Access Control on Reverse Search in RDAP

Access control can be implemented according to different paradigms introducing increasingly stringent rules. The paradigms reported here in the following leverage the capabilities either built-in or provided as extensions by the OpenID Connect [OIDCC]:

* Role-Based Access Control (RBAC): access rights are granted depending on roles. Generally, this is done by grouping users into fixed categories and assigning static grants to each category. A more dynamic approach can be implemented by using the OpenID Connect "scope" claim;
* Purpose-Based Access Control (PBAC): access rules are based on the notion of purpose, being the intended use of some data by a user. It can be implemented by tagging a request with the usage purpose and making the RDAP server check the compliance between the given purpose and the control rules applied to the data to be returned;

* Attribute-Based Access Control (ABAC): rules to manage access rights are evaluated and applied according to specific attributes describing the context within which data are requested. It can be implemented by setting within an out-of-band process additional OpenID Connect claims describing the request context and making the RDAP server check the compliance between the given context and the control rules applied to the data to be returned;
* Time-Based Access Control (TBAC): data access is allowed for a limited time only. It can be implemented by assigning the users with temporary credentials linked to access grants whose scope is limited.

With regard to the privacy threats reported in Section 12, correlation and disclosure can be mitigated by minimizing both the request features and the response data based on user roles (i.e. RBAC). Misuse can be mitigated by checking for the purpose of the request (i.e. PBAC). It can be accomplished according to the following approaches:

* Full Trust: the registry trusts the fairness of an accredited user. The requestor is always legitimized to submit his requests under a lawful basis. Additionally, he can be required to specify the purpose as either a claim of his account or a query parameter. In the former case, the purpose is assumed to be the same for every request. In the latter case, the purpose must be one of those associated to the user;
* Zero Trust: the registry requires documents assessing that the requestor is legitimized to submit a given request. It can be implemented by assigning the requestor with temporary OpenID account linked to the given request (i.e. TBAC) and describing the request through a set of claims (i.e. ABAC). The association between the temporary account and the claims about the request is made by an out-of-band application. In so doing, the RDAP server is able to check that the incoming request is consistent with the request claims linked to the temporary account.

The two approaches can be used together:

* The former is suitable for users carrying out a task in the public interest, or exercising their official authority (e.g. an officer of a cybercrime agency). Similarly, registrars can submit reverse searches on their domains and contacts based on their contractual relationship with the domain holders. In this case, the query results can be restricted to those pertaining a registrar by adding an implicit predicate to the search condition.
* The latter can be taken to allow domain name dispute resolution service providers to request information in defense of the legitimate interests of complainants.

Authors' Addresses

Mario Loffredo
IIT-CNR/Registro.it
Via Moruzzi,1
56124 Pisa
Italy
Email: mario.loffredo@iit.cnr.it
URI: http://www.iit.cnr.it

Maurizio Martinelli
IIT-CNR/Registro.it
Via Moruzzi,1
56124 Pisa
Italy
Email: maurizio.martinelli@iit.cnr.it
URI: http://www.iit.cnr.it