RFC 9536 : Registration Data Access Protocol (RDAP) Reverse Search

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 28 octobre 2024

Date de publication du RFC : Avril 2024

https://www.bortzmeyer.org/9536.html

Ce RFC normalise une extension au protocole d'accès à l'information RDAP pour permettre des recherches inversées, des recherches par le contenu (≪ quels sont tous les noms de domaine de cette personne ? ≫).

Alors, tout d'abord, un avertissement. L'extension normalisée dans ce RFC est dangereuse pour la vie privée. Je détaille ce point plus loin mais ne réclamez pas tout de suite le déploiement de cette extension : lisez tout d'abord. C'est dès sa section 1 que le RFC met en garde contre un déploiement hâtif !

RDAP (normalisé dans les RFC 9082 [1] et RFC 9083), comme son prédécesseur whois, permet d'obtenir des informations dites sociales (nom, adresse, numéro de téléphone, etc) sur les personnes (morales ou physiques) associées à une ressource Internet réservée, comme un nom de domaine ou un adresse IP. Une limite importante de ces deux protocoles est qu'il faut connaitre l'identifiant (nom de domaine, adresse IP) de la ressource. Or, certaines personnes seraient intéressés à faire l'inverse, découvrir les ressources à partir d'informations sociales. C'est le cas par exemple de juristes cherchant le portefeuille de noms de domaine de quelqu'un qu'ils soupçonnent de menacer leur propriété intellectuelle. Ou de chercheurs en sécurité informatique étudiant toutes les adresses IP utilisées par le C&C d'un botnet et voulant en découvrir d'autres. Ce RFC normalise justement un moyen de faire des recherches inverses. whois n'avait jamais eu une telle normalisation, à cause des risques pour la vie privée, risques qui sont peut-être moins importants avec RDAP.

(Notez quand même que le RFC 9082, section 3.2.1, prévoyait déjà certaines recherches inverses, d'un domaine à partir du nom ou de l'adresse d'un de ses serveurs de noms.)

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc9082.txt

Passons aux détails techniques (section 2). L'URL d'une requête inverse va inclure dans son chemin `/reverse_search` et, bien sûr, des critères de recherche. Par exemple, `/domains/reverse_search/entity?fn=Jean%20Durand&role=registrant` donnera tous les domaines dont le titulaire ("*registrant*") se nomme Jean Durand (`fn` est défini dans la section 6.2.1 du RFC 6350). Le terme après `/reverse_search` indique le type des données auxquelles s'appliquent les critères de recherche (actuellement, c'est forcément `entity`, une personne morale ou physique).

La plupart des recherches inverses porteront sans doute sur quelques champs comme l'adresse de courrier électronique ou le "*handle*" (l'identifiant d'une entité). La section 8 décrit les possibilités typiques mais un serveur RDAP choisit de toute façon ce qu'il permet ou ne permet pas.

La sémantique exacte d'une recherche inverse est décrite en JSONPath (RFC 9535). Vous trouvez le JSONPath dans la réponse (section 5 du RFC), dans le membre `reverse_search_properties_mapping`. Par exemple :

```
"reverse_search_properties_mapping": [
    {
      "property": "fn",
      "propertyPath": "$.entities[*].vcardArray[1][?(@[0]=='fn')][3]"
    }
]
```

L'expression JSONPath `$.entities[*].vcardArray[1][?(@[0]=='fn')][3]` indique que le serveur va faire une recherche sur le membre `fn` du tableau vCard (RFC 7095). Oui, elle est compliquée, mais c'est parce que le format vCard est compliqué. Heureusement, on n'est pas obligé de connaitre JSONPath pour utiliser les recherches inverses, uniquement pour leur normalisation (section 3).

Pour savoir si le serveur RDAP que vous interrogez gère cette extension et quelles requêtes il permet, regardez sa réponse à la question `help` (RFC 9082, section 3.1.6, et RFC 9083, section 7) ; s'il accepte les requêtes inverses, vous y trouverez la valeur `reverse_search` dans le membre `rdapConformance` (section 9), par exemple :

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

Pour les recherches acceptées, regardez le membre `reverse_search_properties`. Par exemple :

```
"reverse_search_properties": [
    {
      "searchableResourceType": "domains",
      "relatedResourceType": "entity",
      "property": "fn"
    }
]
```

Ici, le serveur indique qu'il accepte les requêtes inverses pour trouver des noms de domaine en fonction d'un nom d'entité (ce que nous avons fait dans l'exemple plus haut). Voir la section 4 pour les détails. Si vous tentez une requête inverse sur un serveur, et que le serveur n'accepte pas les requêtes inverses, ou tout simplement n'accepte pas ce type particulier de recherche inverse que vous avez demandé, il répondra avec le code de retour HTTP 501 (section 7). Vous aurez peut-être aussi un 400 si votre requête déplait au serveur pour une raison ou l'autre.

L'extension est désormais placée dans le registre des extensions RDAP <https://www.iana.org/assignments/rdap-extensions/rdap-extensions.xml#rdap-extensions-1>. En outre, deux nouveaux registres sont créés, "*RDAP Reverse Search*" <https://www.iana.org/assignments/rdap-reverse-search/rdap-reverse-search.xml#rdap-reverse-search>, pour les recherches possibles et "*RDAP Reverse Search Mapping*" <https://www.iana.org/assignments/rdap-reverse-search-mapping/rdap-reverse-search-mapping.xml#rdap-reverse-search-mapping> pour les règles JSONPath. Pour y ajouter des valeurs, la politique (RFC 8126) est « spécification nécessaire ».

Revenons maintenant aux questions de vie privée (RFC 6973), que je vous avais promises. La section 12 du RFC détaille le problème. La puissance des recherches inverses les rend dangereuses. Une entreprise concurrente pourrait regarder vos noms de domaine et ainsi se tenir au courant, par exemple, d'un nouveau projet ou d'un nouveau produit. Un malveillant pourrait regarder les noms de domaine d'une personne et identifier ainsi des engagements associatifs que la personne ne souhaitait pas forcément rendre très visibles. Les données stockées par les registres sont souvent des données personnelles et donc protégées par des lois comme le RGPD. Le gestionnaire d'un serveur RDAP doit donc, avant d'activer la recherche inverse, bien étudier la sécurité du serveur (section 13) mais aussi (et ce point n'est hélas pas dans le RFC) se demander si cette activation est vraiment une bonne idée.

En théorie, RDAP pose moins de problème de sécurité que whois pour ce genre de recherches, car il repose sur HTTPS (le chiffrement empêche un tiers de voir questions et réponses) et, surtout, il permet l'authentification ce qui rend possible, par exemple, de réserver les recherches inverses à certains privilégiés, avec un grand choix de contrôle d'accès (cf. annexe A). Évidemment, cela laisse ouvertes d'autres questions comme « qui seront ces privilégiés ? » et « comment s'assurer qu'ils n'abusent pas ? » (là, les journaux sont indispensables pour la traçabilité, cf. l'affaire Haurus <https://www.francetvinfo.fr/faits-divers/l-ex-agent-de-la-dgsi-haurus-condamne-a-5-ans-de-prison-pour-6844196.html>).

Notons qu'outre les problèmes de vie privée, la recherche inverse pose également des problèmes de performance (section 10). Attention avant de la déployer, des requêtes apparemment innocentes pourraient faire ramer sérieusement le serveur RDAP. Si vous programmez un serveur RDAP ayant des recherches inverses, lisez bien les recommandations d'optimisation de la section 10, par exemple en ajoutant des index dans votre SGBD. Et le serveur ne doit pas hésiter, en cas de surcharge, à répondre seulement de manière partielle (RFC 8982 ou RFC 8977).

Je n'ai pas trouvé de code public mettant en œuvre ces recherches inverses. De même, je ne connais pas encore de serveur RDAP déployé qui offre cette possibilité.