RFC 7481 : Security Services for the Registration Data Access Protocol

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 4 avril 2015

Date de publication du RFC : Mars 2015

https://www.bortzmeyer.org/7481.html

Une des faiblesses les plus souvent citées de l'ancien protocole whois est son manque de sécurité : pas d'authentification du client (et donc pas de moyen simple de servir des contenus plus ou moins complets selon le client), et pas de confidentialité. Le candidat à sa succession, RDAP <https://www.bortzmeyer.org/weirds-rdap.html>, va-t-il faire mieux ? En tout cas, il propose de nombreux services de sécurité, détaillés dans ce RFC. Comment et au profit de qui les utiliser, voici qui va sans doute déclencher de nouvelles disputes de gouvernance.

Le nouveau protocole RDAP <https://www.bortzmeyer.org/weirds-rdap.html> est normalisé dans plusieurs RFC comme le RFC 7480 [1] qui décrit comment faire tourner RDAP sur HTTP avec une architecture REST. La grande majorité des fonctions de sécurité de RDAP découlent de cette utilisation de HTTP. Ces fonctions étaient réclamés dès le cahier des charges pour un successeur de whois, le RFC 3707.

La section 3 de notre RFC liste les services de sécurité attendus, et la manière dont RDAP les fournit, en général en s'appuyant sur les services équivalents fournis par HTTP et HTTPS. Ainsi, la premier service demandé est le contrôle d'accès et la section 10.2.2 du RFC 7483, RFC consacré au format des réponses RDAP (en JSON) prévoit un mécanisme standard pour indiquer qu'une partie de la réponse a été délibérement tronquée ou omise.

Ensuite, l'authentification (section 3.1.4.2 du RFC 3707). Pas d'autorisation fine sans authentification préalable. Avec whois, on doit servir les mêmes données à tout le monde, car il n'y a pas d'authentification disponible (parfois, une authentification sommaire est faite via l'adresse IP du client, un registre peut servir davantage d'informations à ses employés, en vérifiant que l'adresse IP source est celle du

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc7480.txt

réseau local du registre). Quand RDAP tourne sur HTTP, il doit donc utiliser les mécanismes d'authentification de HTTP, décrits dans le RFC 7235. Si on utilise le mécanisme "*basic*", qui envoie en clair le mot de passe, il faut en plus utiliser TLS (RFC 2818). On peut aussi tout faire avec TLS, en authentifiant le client par un certificat (section 7.4.6 du RFC 5246) mais c'est optionnel dans RDAP, contrairement aux mécanismes du RFC 7235, qui doivent être présents. Ces mécanismes ne permettent pas d'authentifier le serveur, mais on peut, là encore, se servir de TLS pour cela (en vérifiant le certificat du serveur).

Les mécanismes classiques d'authentification via HTTP nécessitent que le client gère des informations de créance pour tous les serveurs. Or, un même client RDAP peut parler à de nombreux serveurs. Il est donc intéressant de regarder du côté des mécanismes d'authentification fédérés, qui peuvent permettre de n'avoir qu'une seule information de créance, utilisée pour tous les serveurs. Il existe pour cela plusieurs techniques utilisable en HTTP, OAuth (RFC 6749), OpenID, des assertions SAML, ou bien des certificats client avec des AC reconnues par tous les serveurs. À noter que RDAP ne dispose pas (pas encore ?) d'une solution pour découvrir quel mécanisme d'authentification est utilisé par quel serveur.

Une fois qu'on a l'authentification, on peut bâtir l'autorisation et donner des accès aux clients en fonction de leur identité (section 3.1.4.2 du RFC 3707). On peut par exemple :

— Permettre un accès à tout client (mode « anonyme » comme avec le whois actuel) mais en ne donnant accès qu'à peu d'informations,
— Ne donner un accès complet qu'aux informations ayant un rapport avec le client (par exemple permettre à un titulaire de tout voir sur ses propres noms),
— Donner un accès complet aux clients authentifiés comme travaillant pour la NSA.

Au bout du compte, il s'agit de décisions politiques, à prendre dans chaque registre, et ce RFC n'indique donc pas de politique, il indique juste les bases sur lesquelles mettre en œuvre une telle politique.

Parmi les services de sécurité, il y a bien sûr la disponibilité : un serveur RDAP super-protégé mais qui serait tout le temps en panne n'aurait guère d'utilité. Notre RFC rappelle donc qu'il existe un RFC sur les attaques par déni de service (le RFC 4732). Un serveur RDAP est autorisé à se prémunir contre les attaques et les excès, par exemple en limitant d'autorité le trafic. Si un client est refusé en raison de cette limitation, le code de retour HTTP recommandé est le 429 (RFC 6585). Si le client est simplement trop enthousiaste, il ralentira alors ses accès. Évidemment, si c'est un attaquant, il continuera.

Autre service de sécurité essentiel, la confidentialité. Whois n'avait aucun mécanisme de protection contre un tiers qui écoute le réseau (d'un autre côté, comme tout était public...) RDAP permet d'utiliser HTTP sur TLS (RFC 2818), permettant ainsi au chiffrement de protéger la confidentialité des données. (On note que les serveurs RDAP expérimentaux actuellement existants sont loin de tous permettre un accès en HTTPS.) À noter que TLS protège le canal, pas les données : il n'y a pas de chiffrement de bout en bout dans RDAP, pour l'instant.

Et le dernier service de sécurité dont on veut disposer est l'intégrité, la garantie que les données ne soient pas modifiées en route. Il n'existe pas non plus de mécanisme de signature des données dans RDAP, mécanisme qui assurerait une protection de bout en bout. Il faut donc utiliser TLS, dont les paquets comportent un MAC qui permet de détecter les modifications.

On a parlé de confidentialité mais la section 4 de notre RFC pose le problème dans une perspective plus générale, celle de la protection de la vie privée. Ce problème est la plaie de whois (« notre conflit du Moyen-Orient », disait Fadi Chehadé en faisant allusion à la durée du problème et à l'absence de perspectives de solution) et a fait l'objet d'innombrables débats, à l'ICANN ou ailleurs. Comme whois ne fournit aucun accès différencié (tout est public ou pas distribué du tout), les données stockées dans les registres, qui sont souvent sensibles, sont envoyées à tous. Résultat, la seule protection est souvent de mentir, dissimulant ses données personnelles. Parfois, il existe des services intermédiaires qui masquent les données, relaient le courrier, etc. RDAP ne change pas la question des données récoltées et stockées

par le registre (si le registre se fait pirater, ou bien s'il donne toutes ses données à des services de surveillance étatiques ou non, la vie privée sera compromise, RDAP ou pas RDAP). En revanche, il permet de ne pas envoyer les données à tout le monde, réservant par exemple les données personnelles à certains clients. RDAP permet par exemple de transmettre davantage de données à des membres des forces de l'ordre, dûment authentifiés. Notez donc que RDAP va soulever des problèmes à lui, que n'avait pas whois. Whois ne permettait techniquement pas de donner plus d'informations aux forces de l'ordre, ou aux ayant-droits, ou à d'autres catégories. RDAP le permet techniquement et certains acteurs le demanderont sans aucun doute.

Comme toujours en protection des données personnelles, le mieux, pour éviter les ennuis, est de ne pas récolter les données du tout. Cette solution simple est la plus efficace, protégeant aussi bien du piratage que d'un opérateur indélicat. Ainsi, les réponses RDAP utilisent jCard (RFC 7095) pour les données personnelles. jCard a énormément de champs possibles et tous ne sont pas pertinents pour le registre. Il est donc possible de ne pas les collecter (et donc de ne pas les envoyer).

À noter qu'il existe aussi un risque lié à la vie privée pour le client : les requêtes qu'il fait peuvent donner des informations sur son activité. Le chiffrement protège contre l'accès d'un tiers à ces informations, mais pas contre le registre lui-même, qui doit donc rester discret sur les requêtes qu'il reçoit.

La section 5 du RFC couvre quelques risques de sécurité résiduels. Par exemple, on voit souvent dans les réponses whois des tentatives d'injection, notamment de code HTML. Notre RFC rappelle donc que les données renvoyées par RDAP ne sont pas forcément sûres et qu'un client RDAP ne doit pas les afficher aveuglément, ni les mettre dans une base de données sans faire attention <https://www.bortzmeyer.org/sql-injection.html>.

Et, enfin, la section 5 de notre RFC rappelle aussi que le protocole peut assurer des choses comme l'intégrité des données (ce qui a été reçu par le client est bien ce qui avait été envoyé par le serveur) mais que cela ne garantit pas l'exactitude des données : celles-ci étaient peut-être fausses dès le début, soit par négligence de la part de celui qui les a déclarées, soit parce que mentir est, à l'heure actuelle, souvent le seul moyen de protéger ses données personnelles.