RFC 7072 : A Reputation Query Protocol

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 22 novembre 2013

Date de publication du RFC : Novembre 2013

https://www.bortzmeyer.org/7072.html

Comment obtenir de l'information sur la **réputation** d'une entité (un nom de domaine, une adresse IP, etc) sur l'Internet ? Il existe actuellement tout un tas de méthodes "*ad hoc*" et le but du groupe de travail IETF repute <http://tools.ietf.org/wg/repute> est de fournir une alternative standard. Le RFC 7070 [1] décrit le fonctionnement général de ce système. Ce cadre général autorise plusieurs protocoles concrets de récupération de l'information et ce RFC décrit un de ces protocoles, fondé sur HTTP.

Le mécanisme est assez simple et le RFC très court. Le client qui veut se renseigner auprès d'un serveur de réputation travaille en deux temps :

— Il récupère un gabarit auprès du serveur de réputation, et l'utilise pour fabriquer un URI,
— Il récupère ensuite le **reputon**, exprimé en JSON (cf. RFC 7071), via cet URI.

Ce mécanisme en deux étapes permet de la souplesse : les reputons peuvent se trouver derrière des URI très différents, qui n'ont pas besoin d'être « en dur » dans le client.

Ainsi, le service de réputation d'OpenDKIM <http://www.opendkim.org/>, accessible en repute.opendkim.org a le gabarit http://{service}/repute.php{?subject,application,assertion,service,reporter} ce qui s'expanse en http://repute.opendkim.org/repute.php?subject=nom-de-domaine&application=email (reporter est optionnel). Rappel de syntaxe des gabarits : le point d'interrogation ne s'applique pas qu'à la variable suivante mais à toutes les variables entre crochets.

Il n'y a pas de mécanisme de découverte d'un serveur de réputation : typiquement, cela se fait manuellement. En pratique, beaucoup de ces services nécessiteront une inscription ou un abonnement, de toute façon. Notez que le RFC ne spécifie pas de mécanisme d'authentification du client : on se sert par exemple des solutions classiques de HTTP. De même, si on veut de la confidentialité, on doit utiliser les mécanismes de chiffrement habituels, donc HTTPS (voir section 5 de ce RFC).

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc7070.txt

Le client doit aussi connaître le nom de l'application et le nom de l'assertion à tester. Il récupére ensuite le gabarit, en utilisant un URI « bien connu » (RFC 8615), `/.well-known/repute-template` (désormais dans le registre IANA des URI bien connus <https://www.iana.org/assignments/well-known-uris/well-known-uris.xhtml>). Ce gabarit suit la syntaxe du RFC 6570. Le client remplace alors les variables. Par exemple, pour le gabarit ci-dessus, la variable `application` va être remplacée par `email-id` (le serveur de réputation peut gérer plusieurs applications). Pour reprendre l'exemple du RFC, si le gabarit du service `example.com` avait été `http://{service}/{application}/{subject` et qu'on cherchait des informations sur la réputation de `example.org` dans le cadre de l'application `email-id` (normalisée dans le RFC 7073), sur l'assertion `spam`, l'expansion du gabarit donnerait `http://example.com/email-id/example.org/spam`. Quelles sont les variables possibles dans un gabarit ?

- `application` est le nom de l'application, pris dans le registre IANA <https://www.iana.org/assignments/reputation-parameters/reputation-parameters.xhtml#applications>,
- `service` est le nom (ou l'adresse IP) du service de réputation demandé (celui d'où on a obtenu le gabarit),
- `subject` est l'entité dont on veut connaître la réputation,
- `assertion` est l'assertion qui nous intéresse, les valeurs possibles dépendant de l'application,
- Sans compter des variables spécifiques à l'application.

Si un paramètre est optionnel mais que la syntaxe du gabarit ne permet pas de l'omettre, le client met une chaîne de caractères vide. L'auteur du gabarit doit veiller à ce que cela ne mène pas à des URI incorrects. Ainsi, le gabarit `http://{service}/{application}/{subject}/{assertion}/{a}/{b}`, si `a` est facultatif, pourrrait mener à une expansion où on aurait deux barres obliques de suite, ce que bien des serveurs HTTP réduisent à une seule, faussant ainsi la lecture de l'URI. Une bonne solution serait d'utiliser le mécanisme des gabarits du RFC 6570 pour les paramètres optionnels et donc de réécrire le gabarit vers `http://{service}/{application}/{subject}/{assertion}/{?a,b}`.

La réponse générée par le serveur de réputation sera alors un reputon (RFC 7071), étiqueté `application/reputo`

Il existe une implémentation (due à Murray S. Kucherawy, un des auteurs du RFC) dans OpenDKIM <http://www.opendkim.org/> à partir de la 2.9 (actuellement en version beta), le serveur est écrit en PHP (le serveur produit du JSON avec `printf`...) et le client en C. Testons-là avec le service public `repute.opendkim.org`. On récupère le gabarit :

```
% curl http://repute.opendkim.org/.well-known/repute-template
http://{service}/repute.php{?subject,application,assertion,service,reporter,format}
```

On fabrique ensuite une requête curl à la main (un vrai client du service de réputation le ferait automatiquement à partir du gabarit) :

```
% curl 'http://repute.opendkim.org/repute.php?subject=gmail.com&assertion=spam&application=email-id&service=
Content-Type: application/reputon+json

{
  "application": "email-id",
  "reputons": [
    {
"rater": "repute.opendkim.org",
"assertion": "spam",
"rated": "gmail.com",
"rating": 0.0113348,
"identity": "dkim",
"rate": 1735,
"sample-size": 181,
"generated": 1383463475
    }
  ]
}
```

On voit que Gmail émet apparemment peu de spam (classement à 0,0113348). Ce service est mis en œuvre avec les messages reçus par le domaine `opendkim.org` et seulement s'ils sont signés par DKIM (donc, n'essayez pas avec un domaine non signé comme `laposte.net`, vous aurez un "*No data available*").