RFC 8063 : Key Relay Mapping for the Extensible Provisioning Protocol

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 15 février 2017

Date de publication du RFC : Février 2017

https://www.bortzmeyer.org/8063.html

Un des problèmes pratiques que pose DNSSEC est le changement d'hébergeur DNS. Si on sous-traite sa gestion des clés cryptographiques, et ses signatures à un tiers, que faire lors du changement de prestataire, pour ne pas casser la chaîne de confiance DNSSEC ? On ne peut évidemment pas demander à l'ancien hébergeur de transmettre la clé privée au nouveau ! Même pour les données non confidentielles, comme la clé publique, la transmission est difficile car les deux hébergeurs n'ont pas de canal de transmission sécurisé commun. Ce nouveau RFC propose d'utiliser comme canal sécurisé le registre de la zone parente et, plus concrètement, de définir une extension au protocole EPP, qui permet un mécanisme de « messagerie » électronique sécurisée, afin de l'utiliser entre deux clients du même registre.

Pour bien comprendre le problème et sa solution, il faut faire un peu de terminologie :

— Bureau d'enregistrement (BE) : l'entité par laquelle il faut passer pour toute opération sur le registre. À noter que tous les registres n'ont pas ce concept et c'est pour cela que notre RFC, comme les autres RFC sur EPP, ne parle pas de BE mais de client EPP.
— Hébergeur DNS ("*DNS operator*" dans le RFC) : l'entité qui gère les serveurs de nom du domaine. C'est souvent le BE mais ce n'est pas du tout obligatoire. Les serveurs DNS peuvent être gérés par une entreprise spécialisée, qui n'est pas BE, ou bien directement par l'utilisateur.
— EPP : protocole d'avitaillement (notamment de noms de domaine), normalisé dans le RFC 5730 [1], entre un serveur (le registre) et un client (le BE, lorsque ce registre utilise des BE).
— DNSSEC : système d'authentification des informations récupérées via le DNS. Il repose sur la cryptographie asymétrique et il y a donc une clé publique (mise dans le DNS) et une clé privée (qui n'est... pas publique). DNSSEC utilise le DNS et doit donc faire attention au temps qui s'écoule ; par exemple, lorsqu'on publie une nouvelle clé (on l'ajoute à l'ensemble DNSKEY), elle ne va pas être visible tout de suite par tous les clients DNS, certains ont en effet mémorisé l'ancien ensemble de clés.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5730.txt

Dans le cas le plus fréquent, l'hébergeur DNS assure la gestion des clés (création, suppression, etc) et connait donc la clé privée, qu'il utilise pour signer les enregistrements. Si le titulaire du domaine veut changer d'hébergeur, pas question bien sûr de transmettre la clé privée. Le nouvel hébergeur (le « gagnant ») va donc créer des nouvelles clés et les utiliser pour signer. Le problème est qu'un résolveur DNS peut avoir des signatures de l'ancien hébergeur (le « perdant ») et des clés du nouveau (ou bien le contraire). Dans ces deux cas, la validation échouera, le domaine sera vu comme invalide.

Une solution à ce problème serait que l'ancien hébergeur mette à l'avance (rappelez-vous, le temps est crucial dès qu'on fait du DNS...) dans les clés qu'il publie la nouvelle clé du nouvel hébergeur. Mais cela suppose qu'il connaisse cette clé. Le titulaire du nom peut servir de relais mais il n'est pas forcément compétent pour cela (« M. Michu, votre nouvel hébergeur a dû vous remettre une clé cryptographique. C'est une série de lettres et de chiffres incompréhensiblles. Pouvez-nous nous la transmettre sans la moindre altération ? »). L'ancien hébergeur ne peut pas non plus utiliser le DNS puisque les serveurs du nouveau ne sont pas encore configurés et, même s'ils le sont, l'ancien hébergeur ne peut pas valider leur contenu avec DNSSEC. L'idée de notre RFC est donc d'utiliser le registre comme intermédiaire de confiance. Si les deux hébergeurs sont également BE, ils ont déjà un canal sécurisé avec le registre (la session EPP). Sinon, on peut espérer que le BE acceptera de servir de relais entre l'hébergeur et le registre.

Avec la solution de ce RFC, le nouvel hébergeur (on va suppose qu'il est également BE, ce sera plus simple) va créer ses clés, les transmettre (la clé publique seulement, bien sûr) au registre via l'extension EPP de notre nouveau RFC, l'ancien hébergeur va les lire (le registre ne sert que de boîte aux lettres sécurisée), les mettre dans la zone DNS. Au bout d'un temps déterminé par le TTL des enregistrements, tous les résolveurs auront l'ancienne et la nouvelle clé publique dans leur mémoire, et pourront valider aussi bien les anciennes que les nouvelles signatures.

Une autre façon de résoudre ce problème serait évidemment que chacun gère sa zone DNS lui-même, et donc ne change jamais d'« hébergeur » mais ce n'est évidemment pas souhaitable pour la plupart des titulaires.

Ce RFC ne spécifie qu'un mécanisme de messagerie, pas une politique, ni une procédure détaillée. La politique est du ressort du registre et de ses BE (via le contrat qui les lie, qui spécifie typiquement des obligations comme « le BE perdant doit coopérer au transfert du domaine, et mettre les nouvelles clés dans la zone qu'il gère encore »). La procédure n'est pas décrite dans un RFC. (Il y a eu une tentative via le document `draft-koch-dnsop-dnssec-operator-change`, mais qui n'a pas abouti. La lecture de ce document est quand même très recommandée.) Le mécanisme de messagerie décrit dans notre RFC est donc « neutre » : il ne suppose pas une politique particulière. Une fois la clé transmise, sa bonne utilisation va dépendre des règles en plus et de si elles sont obéies ou pas. Comme le dit le RFC, « *"The registry SHOULD have certain policies in place that require the losing DNS operator to cooperate with this transaction, however this is beyond this document."* »

Les détails EPP figurent en section 2. Les clés publiques sont dans un élément XML `<keyRelayData>`. Il contient deux éléments, `<keyData>`, avec la clé elle-même (encodée en suivant la section 4 du RFC 5910), et `<expiry>` (optionnel) qui indique combien de temps (en absolu ou bien en relatif) garder cette clé dans la zone. La syntaxe formelle complète figure en section 4, en XML Schema.

Les commandes EPP liées à cette extension figurent en section 3. Certaines commandes EPP ne sont pas modifiées par cette extension, comme `check`, `info`, etc. La commande `create`, elle, est étendue pour permettre d'indiquer la nouvelle clé (un exemple figure dans notre RFC, section 3.2.1). Si le serveur EPP accepte cette demande, il met la clé dans la file d'attente de messages du client EPP qui gère le nom de domaine en question (typiquement le BE « perdant »). Sinon, il répond pas le code de retour 2308.

La nouvelle clé apparaitra dans le système de « messagerie » d'EPP ("*poll queue*"), RFC 5730, section 2.9.2.3. Un exemple de réponse figure dans notre RFC, section 3.1.2.

Quelques points de sécurité pour finir (section 6). Un client EPP méchant pourrait envoyer des clés à plein de gens juste pour faire une attaque par déni de service. C'est au serveur EPP de détecter ces abus et d'y mettre fin. Le serveur EPP peut exiger un `authinfo` correct dans le message de création, pour vérifier que l'action a bien été autorisée par le titulaire. Enfin, cette technique d'envoi des clés ne garantit pas, à elle seule, que tout aura bien été fait de bout en bout. Par exemple, le nouvel hébergeur a tout intérêt à vérifier, par une requête DNS epxlicite, que l'ancien a bien mis la nouvelle clé dans la zone.

Ce RFC a eu une histoire longue et compliquée, malgré une forte demande des utilisateurs. Il y a notamment eu un gros problème avec un brevet futile (comme la plupart des brevets logiciels) de Verisign, qui a fait perdre beaucoup de temps (la déclaration de Verisign est la n° 2393 <https://datatracker.ietf.org/ipr/2393/>, le brevet lui-même est le US.201113078643.A <https://register.epo.org/ipfwretrieve?apn=US.201113078643.A&lng=en>, la décision de l'IETF de continuer malgré ce brevet, et malgré l'absence de promesses de licence, est enregistrée ici <https://mailarchive.ietf.org/arch/msg/regext/5XyA8Z8BG3YP8bmkNAeyq15mH3U>).

Question mise en œuvre, la bibliothèque Net : :DRI <http://search.cpan.org/~pmevzek/Net-DRI/> gère déjà ce RFC. Combien de registres et de BE déploieront ce RFC ? Le coût pour le registre est probablement assez faible, puisqu'il a juste à relayer la demande, utilisant un mécanisme de « messagerie » qui existe déjà dans EPP. Mais, pour les BE, il y a certainement un problème de motivation. Ce RFC aidera le BE « gagnant », et le titulaire du domaine, mais pas le BE « perdant ». Il n'a donc pas de raison de faire des efforts, sauf contrainte explicite imposée par le registre (et l'expérience indique que ce genre de contraintes n'est pas toujours strictement respecté).

Comme bonne explication de ce RFC, vous pouvez aussi lire l'excellent explication de SIDN <https://www.sidn.nl/a/about-sidn/key-relay-keeping-the-dnssec-chain-intact-during-transfers> (avec une jolie image). En parlant de SIDN, vous pouvez noter que leur première mention d'un déploiement d'une version préliminaire de cette solution a eu lieu en 2013 (cf. leur rapport d'activité <http://jaarverslag.sidn.nl/annualreport2013/dot-nl>). Le même SIDN vient de publier un article de premier bilan <https://www.sidn.nl/a/internet-security/key-relay-new-ietf-standard-for-tran
language_id=2> sur ce projet.

Merci à Patrick Mevzek pour les explications, le code et les opinions.