RFC 8739 : Support for Short-Term, Automatically-Renewed (STAR) Certificates in Automated Certificate Management Environment (ACME)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 12 mars 2020

Date de publication du RFC : Mars 2020

https://www.bortzmeyer.org/8739.html

Quand une Autorité de Certification (AC) émet un certificat numérique, une question de sécurité se pose : que se passe-t-il si un attaquant met la main sur la clé privée associée à ce certificat, et peut donc usurper l'identité du titulaire légitime ? La réponse traditionnelle était la **révocation** du certificat par l'AC dès qu'elle est prévenue. Pour diverses raisons, ce processus de révocation est peu fiable, ce qui laisse comme seule ligne de défense l'**expiration** du certificat. C'est le rôle du champ « *"Not After"* » dans un certificat. Pour la sécurité, on voudrait que la date d'expiration soit proche, pour ne pas laisser un éventuel attaquant profiter de son forfait trop longtemps. Mais ce n'est pas très pratique pour le titulaire que de renouveller son certificat très souvent, même avec un protocole comme ACME qui permet l'automatisation. Ce nouveau RFC propose une extension à ACME, qui autorise des certificats de **très courte durée de vie** (quelques jours seulement) mais renouvellés encore plus facilement qu'avec le ACME classique.

Petit rappel sur ACME : ce protocole, normalisé dans le RFC 8555 [1], permet d'obtenir de manière automatique un certificat correspondant à une identité qui est, la plupart du temps, un nom de domaine. Comme ACME permet l'automatisation, il résout le problème de la révocation en utilisant des certificats dont la durée de vie se compte en mois et plus en années. Ainsi, l'AC Let's Encrypt émet des certificats qui durent trois mois. Mais même trois mois, ça peut être long, si quelqu'un a piqué votre clé privée et se sert de ce certificat. Si on souhaite des certificats durant quelques **jours**, peut-on utiliser ACME ? En théorie, oui, mais, en pratique, l'AC pourrait ne pas aimer cette charge supplémentaire, et puis que ferait le titulaire si l'AC était indisponible pendant 48 h et qu'on ne puisse pas renouveller le certificat ?

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc8555.txt

D'où l'idée des certificats STAR ("*Short-Term, Automatically-Renewed*"), initialement décrits dans l'article « "*Towards Short-Lived Certificates*" <https://www.ieee-security.org/TC/W2SP/2012/papers/w2sp12-final9.pdf> », de Topalovic, E., Saeta, B., Huang, L., Jackson, C., et D. Boneh, puis dans l'"*Internet-Draft*" draft-nir-saag-star. Les certificats seront de très courte durée de vie, et publiés un peu à l'avance par l'AC, sans demande explicite du client. Celui-ci pourra par contre demander l'interruption de la série de certificats, si sa clé privée a été compromise.

La section 2 de notre RFC explique le déroulement des opérations. Le client (IdO, pour "*Identifier Owner*") demande à l'AC une série de certificats STAR, l'AC, aux intervalles indiqués, crée et publie les certificats, à tout moment, l'IdO peut arrêter la série. Commençons par le commencement, le démarrage de la série. C'est du ACME classique (RFC 8555), avec ses défis (par exemple, l'IdO doit prouver qu'il contrôle bien le nom de domaine qui sert d'identité). L'IdO doit envoyer l'extension ACME nommée auto-renewal. L'AC indique au client où seront publiés les certificats de la série.

Ensuite, la publication de la série. Tous les certificats de la série utilisent la même clé privée. (Par défaut, les clients ACME classiques créent une nouvelle clé pour chaque renouvellement du certificat.) Ces certificats sont publiés à l'URL annoncé à la fin de la phase précédente.

Lorsqu'il le souhaite, l'IdO peut demander à l'AC d'interrompre la publication de la série de certificats. Notez qu'on ne révoque jamais ces certificats STAR, puisque de toute façon ils expirent très vite.

Les détails du protocole figurent en section 3 du RFC. Ainsi, l'objet auto-renewal (désormais dans le registre des champs de l'objet Order <https://www.iana.org/assignments/acme/acme.xml#acme-order-object-fields>) a plusieurs champs intéressants, comme start-date (début de la série), end-date (fin de la série, mais elle pourra se terminer plus tôt, en cas d'annulation explicite), lifetime (durée de vie des certificats, notez que la valeur réelle dépendra de la politique de l'AC, cf. section 6.2). Voici un exemple de cet objet, à ajouter aux requêtes de demande de certificat :

```
"auto-renewal": {
          "start-date": "2019-01-10T00:00:00Z",
          "end-date": "2019-01-20T00:00:00Z",
          "lifetime": 345600,          // 4 days
          "lifetime-adjust": 259200    // 3 days
        }
```

Les champs possibles dans un auto-renewal sont listés dans un registre IANA <https://www.iana.org/assignments/acme/acme.xml#acme-order-auto-renewal-fields>. D'autres champs pourront être ajoutés dans le futur, en suivant la politique « Spécification nécessaire » (RFC 8126.)

L'objet Order (section 7.1.6 du RFC 8555 sera en état ready tant que la série des certificats continuera.

L'AC annoncera sa capacité à faire du STAR (ici à la fin de son annonce) :

```
{
  "new-nonce": "https://example.com/acme/new-nonce",
  "new-account": "https://example.com/acme/new-account",
  "new-order": "https://example.com/acme/new-order",
  ...
  "meta": {
    "terms-of-service": "https://example.com/acme/terms/2017-5-30",
```

```
        ...
        "auto-renewal": {
          "min-lifetime": 86400,
          "max-duration":  31536000,
          "allow-certificate-get": true
        }
      }
    }
```

Pour arrêter la série avant `end-date`, le client ACME mettra cet état à `canceled` :

```
POST /acme/order/ogfr8EcolOT HTTP/1.1
Host: example.org
Content-Type: application/jose+json

{
       "protected": base64url({
         "alg": "ES256",
         "kid": "https://example.com/acme/acct/gw06UNhKfOve",
         "nonce": "Alc00Ap6Rt7GMkEl3L1JX5",
         "url": "https://example.com/acme/order/ogfr8EcolOT"
       }),
       "payload": base64url({
         "status": "canceled"
       }),
       "signature": "g454e3hdBlkT4AEw...nKePnUyZTjGtXZ6H"
}
```

Le serveur ACME répondra alors 403 à toutes les requêtes de récupération d'un certificat de la série annulée, de préférence en ajoutant (RFC 7807) `urn:ietf:params:acme:error:autoRenewalCanceled`. (Cette erreur, et quelques autres, ont été ajoutées au registre des erreurs ACME <`https://www.iana.org/assignments/acme/acme.xml#acme-error-types`>.)

Comme vous avez vu, la théorie est simple. Maintenant, il y a un certain nombre de détails opérationnels sur lesquels il faut se pencher, détaillés en section 4. D'abord, le problème des horloges. Les certificats X.509 utilisent partout des temps (la date limite de validité, par exemple) et le respect de ces temps dépend de l'horloge de la machine. Si votre ordinateur a deux mois d'avance, il considérera les certificats comme expirés alors qu'ils ne devraient pas l'être. C'est un problème général de la cryptographie, comme montré par l'article « "*Where the Wild Warnings Are : Root Causes of Chrome HTTPS Certificate Errors*" <`https://acmccs.github.io/papers/p1407-acerA.pdf`> », qui signale que des déviations de plusieurs jours chez les clients ne sont pas rares. Mais c'est évidemment plus grave avec des certificats à très courte durée de vie. Si on a des certificats Let's Encrypt classiques, qui durent trois mois et qu'on renouvelle une semaine avant leur expiration, même si l'horloge du client déconne de plusieurs jours, ça passera. En revanche, avec les certificats STAR, la désynchronisation des horloges aura des conséquences dans bien plus de cas.

La décision d'utiliser STAR ou pas, et le choix de la durée de vie des certificats, va dépendre de la population d'utilisateurs qu'on attend. Le RFC note que les problèmes d'horloge sont bien plus fréquents sur Windows que sur Android, par exemple.

Autre risque avec STAR, la charge supplémentaire pour les journaux "*Certificate Transparency*" (RFC 9162). Si STAR devenait le principal mode d'émission de certificats (c'est peu probable), leur trafic serait multiplié par cent. Avant la publication de ce RFC, de nombreuses discussions avec le groupe de travail

IETF trans <https://datatracker.ietf.org/wg/trans/> et avec les opérateurs des principaux journaux ont montré qu'il n'y avait a priori pas de risque, ces journaux peuvent encaisser la charge supplémentaire.

Questions mises en œuvre de STAR, il y a eu une scission (non publique ?) de Boulder, le serveur de Let's Encrypt et du client certbot pour y ajouter STAR. Il y a également un client et serveur avec STAR dans Lurk <https://github.com/mami-project/lurk>.

La section 6 de notre RFC revient sur les questions de sécurité liées à STAR. Ainsi, comme l'expiration remplace la révocation, on ne peut plus exiger la suppression immédiate d'un certificat. (Mais, on l'a dit, la révocation marche tellement mal en pratique que ce n'est pas une grande perte.) En cas de compromission de la clé privée, on peut demander l'arrêt de l'émission des certificats mais (et cela ne semble pas mentionné par le RFC), si on perd son compte ACME, ou simplement le numnique <https://www.bortzmeyer.org/nonce.html> ACME, on ne peut plus annuler cette émission, et on doit attendre l'expiration de la séquence (indiquée par end-date.)