RFC 8738 : Automated Certificate Management Environment (ACME) IP Identifier Validation Extension

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 1 mars 2020

Date de publication du RFC : Février 2020

https://www.bortzmeyer.org/8738.html

Le protocole ACME, surtout connu via son utilisation par l'AC Let's Encrypt, permet de prouver la « possession » d'un nom de domaine, pour avoir un certificat comprenant ce nom. Ce court RFC spécifie une extension à ACME qui permet de prouver la « possession » d'une adresse IP, ce qui permettra d'obtenir via ACME des certificats utilisant une adresse.

Le protocole ACME est normalisé dans le RFC 8555[1]. Son principe est qu'on demande un certificat pour un identificateur (à l'heure actuelle, forcément un nom de domaine) et que le serveur ACME va alors vous défier de prouver que vous contrôlez bien ce nom, par exemple en publiant une chaîne de caractères choisie par le serveur dans un serveur HTTP accessible via ce nom de domaine. Or, les identificateurs dans les certificats PKIX ne sont pas forcément des noms de domaine. Les adresses IP, par exemple, sont prévues. Examinons les certificats du résolveur DNS public Quad9 <https://www.bortzmeyer.org/quad9.html> :

```
% openssl s_client -connect 9.9.9.9:853 -showcerts | openssl x509 -text
...
X509v3 Subject Alternative Name:
    DNS:*.quad9.net, DNS:quad9.net, IP Address:9.9.9.9, IP Address:9.9.9.10, IP Address:9.9.9.11, IP Address:9.9
...
```

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

On voit qu'outre des noms comme quad9.net, ce certificat inclut aussi des adresses IP comme 9.9.9.9 et 2620:fe::9. Mais un tel certificat ne pouvait pas s'obtenir automatiquement via ACME.

Notre RFC résout ce problème en ajoutant un nouveau type d'identificateur ACME, ip (section 3 du RFC). Les types d'identificateurs ACME sont décrits dans la section 9.7.7 du RFC 8555. Le nouveau type ip a été placé dans le registre IANA des types d'identificateur <https://www.iana.org/assignments/acme/acme.xml#acme-identifier-types>. La valeur doit être une adresse IP sous forme texte (normalisée très sommairement dans la section 2.1 du RFC 1123 pour IPv4, et dans la section 4 du RFC 5952 pour IPv6.)

Comme il s'agit d'authentifier des adresses IP, le défi ACME de type dns-01 n'est pas pertinent et ne doit pas être utilisé (section 7). Par contre, on peut (section 4 du RFC) utiliser les défis http-01 (RFC 8555, section 8.3) et le récent tls-alpn-01 (RFC 8737.)

Pour le défi HTTP, le serveur ACME va se connecter en HTTP à l'adresse IP indiquée, en mettant cette adresse dans le champ Host:. Pour le défi TLS avec ALPN, le certificat doit contenir un subjectAltName de type iPAddress. Un piège : contrairement au champ Host: de HTTP, l'adresse IP nue ne peut pas être utilisée dans le SNI (RFC 6066, « "*Currently, the only server names supported are DNS hostnames*" »). Il faut donc utiliser un nom dérivé de l'adresse, en in-addr.arpa ou ip6.arpa. Par exemple, si on veut un certificat pour 2001:db8::1, il faudra mettre 1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa dans le SNI.

Un défi utilisant la « résolution inverse » (via une requête DNS dans in-addr.arpa ou ip6.arpa) avait été envisagé mais n'a pas été retenu (les domaines de la « résolution inverse » sont en général mal maintenus et il est difficile d'obtenir des entrées dans ces domaines.)

La section 9 de notre RFC étudie les conséquences de cette extension pour la sécurité. Le principal point à noter est que ce RFC ne spécifie qu'un mécanisme. L'AC a toute liberté pour définir une politique, elle peut par exemple refuser par principe les adresses IP dans les certificats, comme elle peut les accepter avec des restrictions ou des contrôles supplémentaires. Par exemple, il ne serait pas raisonnable d'allouer de tels certificats pour des adresses IP appartenant à des plages très dynamiques, pouvant changer d'utilisateur très souvent.

Côté mise en œuvre, pour le serveur Boulder <https://github.com/letsencrypt/boulder> (celui utilisé par Let's Encrypt), la discussion est ici <https://github.com/letsencrypt/boulder/issues/2706>.