

Authentifier des serveurs Internet avec X.509 lorsqu'ils ont la même adresse IP

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 4 décembre 2008. Dernière mise à jour le 16 décembre 2008

<https://www.bortzmeyer.org/auth-x509-plusieurs-noms.html>

On lit souvent qu'il n'est pas possible d'avoir plusieurs serveurs Internet sur la même adresse IP lorsqu'ils sont authentifiés par un certificat X.509. Il faudrait donc donner une adresse IP à chacun. Cette affirmation a des origines réalistes mais est sans doute trop stricte aujourd'hui.

Dans un protocole comme HTTPS, le nom du serveur est transmis une fois la session TLS établie. Il est donc a priori trop tard à ce moment pour choisir un autre certificat. D'où le même « Un seul certificat par adresse IP ». (Un bon résumé du problème figure dans *"Name-based SSL virtual hosts : how to tackle the problem"* <https://www.switch.ch/pki/meetings/2007-01/namebased_ssl_virtualhosts.pdf>.) En fait, il est possible d'avoir une seule adresse IP et néanmoins un certificat différent par *"Virtual Host"* Apache, même si les méthodes existantes ont quelques limites.

Il existe trois méthodes pour HTTPS (toutes ne sont pas forcément applicables aux autres protocoles) :

- La commande HTTP `STARTTLS` (RFC 2817¹) qui permet de transmettre le nom du serveur avant la négociation TLS.
- L'extension X.509 *"Subject Alternative Name"* (RFC 5280) qui permet de mettre plusieurs noms dans un certificat. Le client sera alors content si un des noms correspond.
- L'extension TLS *"Server Name Indication"* (SNI) (RFC 6066) qui permet au client d'indiquer le nom du serveur dans la négociation TLS.

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc2817.txt>

Chacune a ses forces et ses faiblesses et des domaines où elle est meilleure que les autres.

La première, STARTTLS, normalisée dans le RFC 2817, est bien décrite par Pierre Beyssac dans « HTTP et TLS, la RFC méconnue... <<https://signal.eu.org/blog/2007/09/07/http-et-tls-la-rfc-meconnue>> ». Très répandue avec des protocoles comme IMAP ou SMTP, elle a un gros inconvénient pour HTTP : il n'est pas possible d'indiquer dans l'URL que TLS est obligatoire. Un attaquant sur le chemin, ou bien un serveur mal configuré, et on retombe sur du HTTP non protégé.

Mise en œuvre dans Apache depuis la version 2.2, elle n'est présente dans aucun grand navigateur, pour les raisons expliquées par Mozilla dans la bogue #276813 <https://bugzilla.mozilla.org/show_bug.cgi?id=276813>.

Une deuxième méthode est le certificat contenant plusieurs noms. Elle est possible grâce à l'extension X.509 "Subject Alternative Name", normalisée dans le RFC 2459. Elle est décrite en détail dans mon article « Plusieurs noms dans un certificat X.509 <<https://www.bortzmeyer.org/plusieurs-noms-dans-certificat-x509.html>> » ou bien dans celui de Franck Davy, « Apache : Hôtes virtuels et SSL (mod_ssl) <http://www.hsc.fr/ressources/breves/ssl_virtualhosts.html> ». En anglais, on peut lire « "Information about setting up the ApacheServer to serve multiple HTTPS sites using one IP address" <<http://wiki.cacert.org/wiki/CSRGenerator?action=show&redirect=VhostsApache>> ».

Elle fonctionne avec openssl (aussi bien pour générer le certificat que pour le vérifier) mais il faut que la CA qui signe l'accepte et j'ignore si les grosses CA ayant pignon sur rue le font ou pas. C'est la méthode qui semble la plus déployée et donc celle qui apporte le moins de surprises (CAcert a fait des tests détaillés d'interopérabilité <<http://wiki.cacert.org/wiki/VhostTaskForce#InteroperabilityTest>> sur les variantes de cette méthode).

La troisième méthode repose sur une extension du protocole TLS et non pas du format X.509. Cette extension, "Server Name Indication" (SNI), est l'une de celles décrites dans le RFC 6066. Elle envoie le nom du serveur lors de la négociation TLS. Elle est bien expliquée dans l'article de Pierre Beyssac « Adieu RFC 2817, bonjour RFC 3546 <<https://signal.eu.org/blog/2008/11/25/adieu-rfc-2817-bonjour-rfc-3546>> ».

SNI ("Server Name Indication") ne marche pas avec le module Apache mod_ssl, il faut utiliser mod_gnutls. Il y a un certificat par "Virtual Host" et elle ne gère donc pas le cas où un "Virtual Host" a plusieurs noms, avec la directive ServerAlias.

Les serveurs HTTP gérés par le département de recherche & développement de l'AFNIC sont un exemple de déploiement de deux de ces techniques. Le serveur, un Apache, utilise mod_gnutls, avec un certificat par "Virtual Host", et peut donc tirer profit de la troisième méthode, SNI. Au cas où le client ne gère pas l'extension SNI, le certificat par défaut (celui qui est utilisé dans la directive <VirtualHost _default_: 443>) est un certificat avec plusieurs noms.

Voici un test d'un de ces serveurs avec un logiciel en ligne de commande livré avec GnuTLS (bien entendu, TLS étant une norme - RFC 5246 -, on aurait aussi bien pu tester avec openssl) :

```
% gnutls-cli -p 443 --x509cafile /etc/ssl/certs/AFNIC-ReD-CA.pem \
    svn.langtag.net
Processed 1 CA certificate(s).
Resolving 'svn.langtag.net'...
Connecting to '2001:660:3003:3::1:4:443'...
- Certificate type: X.509
```

<https://www.bortzmeyer.org/auth-x509-plusieurs-noms.html>

```
- Got a certificate list of 1 certificates.

- Certificate[0] info:
# The hostname in the certificate matches 'svn.langtag.net'.
# valid since: Thu Dec  4 10:21:30 CET 2008
# expires at: Wed Aug 31 11:21:30 CEST 2011
# fingerprint: F5:78:88:D7:EF:CA:38:92:F3:40:B9:67:D4:B6:48:E6
# Subject's DN: C=FR,ST=Ile-de-France,L=Saint-Quentin-en-Yvelines,O=AFNIC,OU=R&D,CN=www.generic-nic.net,EMAIL=w
# Issuer's DN: C=FR,ST=Ile-de-France,L=Saint-Quentin-en-Yvelines,O=AFNIC,OU=ReD,CN=AFNIC Research and Developme

- Peer's certificate is trusted
...
```

Voilà, tout va bien. Si on est inquiet, on peut vérifier l’empreinte du certificat (ici F5:78:88:D7:EF:CA:38:92:F3:40:B9:67:D4:B6:48:E6). Attention, gnutls-cli affiche l’empreinte MD5 alors que, par défaut, openssl x509 -fingerprint montre l’empreinte SHA-1. Il faut donc, pour regarder le certificat, taper `openssl x509 -fingerprint -md5 -in /ou/est/le/certificat.pem`.

Vishaal Golam me signale qu’il existe une quatrième méthode, avec des restrictions. Mettre un joker dans le certificat, par exemple *.example.net. Ainsi, tout logiciel client qui accepte les jokers (c’est la grande majorité mais attention, avec des sémantiques différentes car ce point n’est pas vraiment normalisé, cf. RFC 2818, section 3.1) acceptera ce certificat pour www.example.net, svn.example.net, etc. Il faut juste que la CA, l’autorité de certification, accepte ces jokers, qui diminuent ses revenus (je ne sais pas quels CA les acceptent mais, par exemple, Thawte le fait <<https://www.thawte.com/ssl-digital-certificates/wildcardssl/>>). Et la méthode ne marche que pour des noms qui ont un domaine en commun (ici example.net).

Merci à Pierre Beyssac, Mario Victor-Oscar, Yves Rutschle, Kim-Minh Kaplan et Benoit Deuffic pour des informations stimulantes et utiles.