Utiliser l'Autorité de Certification CAcert

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 13 novembre 2012. Dernière mise à jour le 14 novembre 2012

http://www.bortzmeyer.org/cacert.html

Comme vous le savez peut-être déjà, les sites Web auxquels on accède avec un URL commençant par https sont sécurisés avec le protocole TLS (décrit dans le RFC 6347 [1]). Pour s'assurer de l'authenticité du site, avant de commencer le chiffrement de la session, TLS se repose sur des **certificats**, qui sont simplement une la partie publique d'une clé cryptographique plus la signature d'un organisme auquel on doit faire confiance. Dans le format de certificats le plus répandu, X.509, cet organisme se nomme une Autorité de Certification. Elles sont typiquement chères et n'apportent pas forcément une sécurité géniale (les forces du marché tirent les procédures de vérification vers le bas...) Une alternative est donc d'employer une Autorité de Certification gratuite, fonctionnant de manière simple et totalement automatisée, CAcert.

Écartons tout de suite un faux débat : est-ce que CAcert est une « vraie » Autorité de Certification (AC), une « officielle » ? C'est un faux débat car il n'existe pas d'AC officielle : chaque éditeur de logiciels (par exemple Microsoft pour Internet Explorer et Mozilla pour Firefox) décide selon ses propres critères de quels AC sont mises ou pas dans son magasin d'AC (et ce ne sont pas forcément les mêmes). Donc, CAcert, comme toute AC, est reconnue par les gens qui lui font confiance, point. (En France, l'ANSSI publie une liste des AC « qualifiées » <https://www.ssi.gouv.fr/fr/produits-et-prestataires/prestataires-de-services-de-confiance-qualifies/> mais cette liste n'a rien à voir avec ce qui se trouve dans les navigateurs, la seule chose importante pour les utilisateurs.)

Mais est-ce que CAcert fournit le même niveau de sécurité que les grosses AC commerciales comme VeriSign (l'activité d'AC sous ce nom appartient désormais à Symantec) ou Comodo ? D'abord, ces AC ne sont pas forcément parfaites comme l'ont montré le piratage de Comodo ou l'Opération Tulipe Noire contre DigiNotar. Ensuite, tout dépend des menaces envisagées. S'il s'agit simplement de protéger la page d'administration du blog de votre association, payer Symantec ou Comodo est franchement exagéré.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6347.txt

Par contre, comme CAcert n'est pas inclus par défaut dans les magasins de tous les navigateurs, vos visiteurs utilisant HTTPS recevront peut-être le fameux avertissement comme quoi le site n'est pas authentifié par une autorité reconnue (l'intéressante étude « "*Crying wolf*" <http://static.usenix.org/events/sec09/tech/full_papers/sunshine.pdf> » étudie cette question des avertissements de sécurité TLS.) En pratique, le moyen le plus simple est alors d'ajouter CAcert à son navigateur. C'est simple mais, évidemment, si vous gérez un site avec beaucoup de visiteurs pas du tout informaticiens, cela peut devenir très gênant. Si vous avez un public limité, ou bien si celui-ci est plutôt "*geek*" (ceux-ci auront souvent déjà installé CAcert), c'est moins grave.

Dans votre cas, testez en allant en <https://www.cacert.org/>. S'il n'y a aucun avertissement de sécurité et que vous voyez le fameux cadenas, c'est bon, CAcert est connu de votre navigateur. Sinon, si vous n'avez pas encore mis CAcert dans les AC reconnues, rendez-vous en <http://www.cacert.org/index.php?id=3> (je n'ai pas mis `https` puisque, à ce stade, vous ne connaissez pas l'AC CAcert). Du point de vue sécurité, il y a à ce moment un problème de "*bootstrap*" : pour authentifier le site de CAcert vous avez besoin du certificat... qui se trouve sur ce site. Si vous êtes prudent, vous allez valider le certificat de l'AC CAcert avant de l'installer, par exemple en vérifiant son empreinte (`13:5C:EC:36:F4:9C:B8:E9:3B:1A:B2:70:CD:80:88:46:76:CE:8F:33` en SHA-1) auprès d'un ami qui l'a déjà installé. Par exemple, avec OpenSSL, si vous avez téléchargé le certificat en `root.crt` :

```
% openssl x509 -fingerprint -in root.crt
SHA1 Fingerprint=13:5C:EC:36:F4:9C:B8:E9:3B:1A:B2:70:CD:80:88:46:76:CE:8F:33
...
```

Ensuite, installer le certificat de l'AC nécessite simplement de suivre le lien « Root Certificate (PEM Format) ». À partir de là, les certificats émis par CAcert seront acceptés par votre navigateur. Si la pensée que CAcert pourrait devenir méchant et émettre un vrai-faux certificat pour le site Web de votre banque, rassurez-vous en pensant que n'importe laquelle des centaines d'AC installées par défaut dans votre navigateur (regardez par exemple tout ce qu'il y a par défaut, de CNNIC à "*TÜRKTRUST Elektronik Sertifika Hizmet Sa[Caractère Unicode non montré [2]]lay[Caractère Unicode non montré]c[Caractère Unicode non montré]s[Caractère Unicode non montré]*") peut en faire autant.

C'est bon ? Vous pouvez visiter <https://www.cacert.org/> sans avertissement de sécurité ? Parfait, maintenant, allons demander à CAcert des certificats.

Comment fonctionne CAcert ? On se rend sur le site Web <https://www.cacert.org/> et on crée un compte, comme sur tant d'autres sites Web. On choisit évidemment un mot de passe sérieux. Puis on se connecte. Commençons par l'option "*Domains*". Normalement, votre liste de domaines est vide à ce stade. CAcert a besoin de connaître les domaines que vous gérez, pour signer ensuite des certificats. Choisissez "*Add*" pour ajouter un domaine. Vous pouvez indiquer n'importe lequel ? Non, heureusement pas. CAcert vérifie que vous êtes bien un administrateur du domaine en envoyant un secret qu'il a choisi à une des adresses de courrier du domaine. Pour cela, il analyse le résultat de whois et il se sert aussi des adresses normalisées dans le RFC 2142. Pour le domaine `langtag.net`, cela donne :

```
Please choose an authority email address
a4edcf75406c24a90c3b19d3711c0de1-18767@contact.gandi.net
root@langtag.net
hostmaster@langtag.net
postmaster@langtag.net
admin@langtag.net
webmaster@langtag.net
```

2. Car trop difficile à faire afficher par LaTeX

La première adresse venant de whois et les suivantes du RFC 2142. Les fanas de sécurité noteront deux choses : l'utilisateur de CAcert choisit quelle adresse servira, donc c'est la sécurité de l'adresse la moins protégée qui détermine la sécurité de l'ensemble. Et, deuxième point, la sécurité de CAcert dépend de celle du courrier, et des protocoles qu'il utilise, comme le DNS (CAcert n'exige pas, aujourd'hui, que les noms de domaines dans les adresses de courrier soient protégés par DNSSEC). Notez que pas mal d'AC commerciales ne font pas mieux, question vérifications !

Vous allez ensuite recevoir dans votre boîte aux lettres un message vous demandant de visiter une page Web dont l'URL est secret, pour confirmer que vous avez bien reçu le message (et que vous êtes donc bien administrateur du domaine) :

```
From: support@cacert.org
To: a4edcf75406c24a90c3b19d3711c0de1-18767@contact.gandi.net
Subject: [CAcert.org] Email Probe
X-Mailer: CAcert.org Website

Below is the link you need to open to verify your domain 'langtag.net'. Once your
address is verified you will be able to start issuing certificates to your heart's
content!

http://www.cacert.org/verify.php?type=domain&domainid=227678&hash=ddefe7cdfdea431278b86216bf833b8f
...
```

(Le lien est à usage unique et je n'ai donc pas de crainte à le donner ici.)

Une fois la visite faite, le nom de domaine est associé à votre compte et vous le voyez dans la liste du menu "*Domains*". Vous pouvez alors demander la signature de certificats. En effet, CAcert ne fait pas le certificat pour vous (car, alors, il connaîtrait la partie privée de la clé). Vous devez créer un certificat, le transmettre à CAcert, qui le signe. Avec les outils d'OpenSSL, cela se passe ainsi :

```
[On fabrique le CSR, Certificat Signing Request]
% openssl req -new -nodes -newkey rsa:2048 -keyout server.key -out server.csr -days 1000
...
Common Name (eg, YOUR name) []: www.langtag.net
```

Le `-newkey` qui demande une clé RSA de 2 048 bits est obligatoire. CAcert n'accepte plus les clés de 1 024 bits, trop vulnérables. Notez que la durée de validité demandée, ici mille jours, n'a aucune importance, CAcert mettra sa propre durée, six mois. Ensuite, on copie/colle le CSR (fichier d'extension `.csr`) dans le formulaire sur le site de CAcert. Ce dernier affiche, pour vérification, les informations du certificat qu'il incluera dans le certificat final. Les informations exclues sont celles que CAcert ne peut vérifier (donc, presque tout sauf le nom de domaine.) Le message est « "*No additional information will be included on certificates because it can not be automatically checked by the system.*" ». CAcert fournit alors un certificat qu'on copie/colle dans un fichier d'extension `.crt`. On peut le vérifier avec son OpenSSL local :

```
% openssl x509 -text -in server.crt
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 818193 (0xc7c11)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: O=Root CA, OU=http://www.cacert.org, CN=CA Cert Signing Authority/emailAddress=support@cacert.or
        Validity
            Not Before: Nov 13 20:55:41 2012 GMT
            Not After : May 12 20:55:41 2013 GMT
        Subject: CN=www.langtag.net
...
```

Naturellement, on n'est pas obligé d'utiliser `openssl` directement, on peut aussi se servir d'un script plus accessible comme CSRGenerator <http://svn.cacert.org/CAcert/Software/CSRGenerator/csr>.

Et voilà, on a un beau certificat qui marche, il n'y a plus qu'à le copier là où l'attend le serveur HTTP, Apache, Nginx ou un autre.

CAcert est capable de reconnaître pas mal d'extensions de X.509 comme les `subjectAltName`, si pratiques lorsqu'il s'agit de mettre plusieurs sites Web derrière une seule adresse IP <http://www.bortzmeyer.org/auth-x509-plusieurs-noms.html>. Il prévient alors :

```
Please make sure the following details are correct before proceeding any further.

CommonName: www.generic-nic.net
subjectAltName: DNS:www.generic-nic.net
subjectAltName: DNS:svn.generic-nic.net
subjectAltName: DNS:viewvc.generic-nic.net
No additional information will be included on certificates because it can not be automatically checked by th

The following hostnames were rejected because the system couldn't link them to your account, if they are val
Rejected: svn.rd.nic.fr
```

Le nom rejeté, à la fin, est parce que le compte CAcert utilisé avait vérifié le domaine `generic-nic.net` mais pas le domaine `nic.fr`.

Voilà, vous pouvez maintenant vous détendre en sécurité, grâce à vos nouveaux certificats. CAcert vous préviendra lorsqu'ils seront proches de l'expiration et qu'il faudra les renouveler (naturellement, si vous êtes prudent, vous superviserez cela vous-même <http://www.bortzmeyer.org/tester-expiration-certifs.html>.)

Notez que CAcert propose d'autres services, par exemple des certificats plus vérifiés, en utilisant la communauté <https://wiki.cacert.org/FAQ/Privileges>. Une alternative souvent citée pour avoir des certificats partiellement gratuits est StartSSL mais je n'ai pas testé.

Merci à John Doe (ah, ah) pour ses remarques et à Florian Maury pour sa relecture et son appréciation que cet article est « de loin, le plus mauvais conseil que j'ai pu lire sur ton blog ». Sur la valeur de CAcert, vous pouvez aussi lire une discussion chez OpenBSD <http://marc.info/?l=openbsd-bugs&m=138437411216098> et une chez Debian <http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=718434> (deux systèmes qui incluent le certificat de CAcert par défaut).