Création d'une autorité de certification et des certificats SSL

Freetux

Date de publication : 9 avril 2014
Dernière modification : 27 février 2015

Introduction

Suite à la très récente faille de sécurité, Heartbleed, du logiciel OpenSSL gérant le chiffrement du protocole HTTPS, j'ai été obligé de réactualiser mes propres certificats. Ceci afin d'assurer qu'ils n'aient pas été corrompus par une personne tierce en exploitant la faille sur mon serveur.

Par la même occasion, j'en ai profité pour renouveler mon autorité de certification afin d'y mettre les adresses mail de mon serveur mail que j'ai configuré après la création de ma première autorité.

Ce renouvellement de l'ensemble des certificats a été l'occasion de publier (enfin) un billet expliquant la méthodologie à suivre afin de créer sois même ces certificats et son autorité de certification.

La faille Heartbleed, découverte il y a 2 jours par une équipe de Google, permet à n'importe qui d'avoir accès à la quasi-totalité des informations transitant, en théorie, de façon chiffrée à travers le protocole HTTPS. Les communications peuvent alors être facilement lisibles par des personnes extérieures. Pour faire simple, c'est comme s'il n'existait plus de chiffrement entre votre ordinateur et le serveur. L'ensemble de vos mots de passe, identifiants, discussions, numéros de carte bleu etc sont alors lisibles [1].

Les correctifs (des distributions libres) de Heartbleed sont apparus très rapidement, quelques heures après sa publication, afin d'empêcher le plus rapidement possible son exploitation. Cependant, pendant plusieurs heures, environ un

1. Plus d'informations sur cet article

million [2] de sites web étaient vulnérables (y compris les miens) et à la merci des pirates en tous genres. Encore maintenant, il existe toujours des serveurs et sites web vulnérables. Pour savoir si un site que vous utilisez régulièrement est sûr, je vous invite à vous rendre sur cette page ;-). Pour info, freezetux.net est protégé depuis hier matin : http://filippo.io/Heartbleed/#freezetux.net. ☺

Par conséquent, il est du devoir des administrateurs des serveurs (soucieux de la sécurité des utilisateurs de leurs services... malheureusement certains sont moins touchés que d'autres ou pas aussi rapidement :-/) de renouveler leurs certificats SSL afin d'assurer de nouveau un chiffrement total des communications de leurs utilisateurs.

La suite de cet article présente comment mettre en place sur son serveur une autorité de certification ainsi que des certificats SSL afin de proposer sur ses sites web une communication chiffrée en HTTPS.

Installation de OpenSSL

L'installation de OpenSSL est très simple sur les machines tournant Linux :

- Distributions RedHat/Fedora/Opensuse/CentOS :

```
# yum install openssl
```

- Distributions Debian/Ubuntu :

```
# apt-get install openssl
```

Important : Je vous invite à aller voir ici pour le correctif de la faille pour les distributions Fedora 19 et 20. Pour les autres distributions je vous invite à vous rapprocher de leur communauté respective afin d'obtenir la méthode à suivre pour installer le correctif. ;-)

Remarque : bien que cet article soit adapté pour Fedora, OpenSSL reste identique quelle que soit la distribution utilisée. Il vous est donc possible de lire la suite si vous utilisez une autre distribution.

2. https://www.schneier.com/blog/archives/2014/04/heartbleed.html

Création de l'autorité de certification

Nous allons tout d'abord créer l'autorité de certification qui signera par la suite les certificats générés. Dans un premier temps nous créons les dossiers qui vont accueillir les clé privées/publiques, les certificats et les signatures. Ces dossiers doivent se trouver dans `/etc/pki/CA/` pour les distributions et sous-distributions RedHat (j'ai créé un dossier freezetux pour mettre y mon autorité perso).

Remarque : pour les distributions et sous-distributions Debian, il se trouve dans `/etc/ssl/certs/` Pour cela taper en root :

```
# mkdir -p /etc/pki/CA/freezetux/{conf,private,public} && chmod
  400 /etc/pki/CA/freezetux/private
```

On va maintenant créer le fichier de conf de l'autorité de certification, taper :

```
cd /etc/pki/CA/freezetux/ \&\& vim conf/openssl.cnf
```

et copier le texte ci-dessous.

Remarque : tous les certificats générés dans cet article seront en `4096bit` et les algorithmes de signature en `sha512`. Il vous est bien entendu possible d'adapter selon votre besoin en mettant les clé en `1024`, `2048` ou `4096bit`. Il est tout de même conseillé d'utiliser au minimum `2048bit`. Pour l'algorithme de signature, le `sha1` est clairement déconseillé pour des raisons de sécurité. Vous pouvez utiliser du `sha128`, `sha256` ou encore `sha512`. Cependant, si vous utilisez d'autres valeurs que celles de cet article, assurez-vous de rester cohérent sur l'ensemble des commandes qui vont suivre.

```
[ req ]
default_bits            = 4096
default_keyfile         = ./private/root.pem
default_md              = sha512
prompt                  = no
distinguished_name      = root_ca_distinguished_name
x509_extensions = v3_ca

[ root_ca_distinguished_name ]
countryName             = FR
stateOrProvinceName     = Midi-Pyrenees
```



```
localityName            = Toulouse
0.organizationName      = Nicolas TISSOT
commonName              = Nicolas TISSOT
emailAddress            = nicolas.tissot @ freezetux.net

[ v3_ca ]
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid:always,issuer:always
basicConstraints = CA:true
```

Adapter la section `root_ca_distinguished_name` suivant votre nom de domaine (éviter les accents dans les mots). Enregistrer alors le fichier et fermer le.

On va maintenant créer la clé privée de l'autorité de certification. La durée de validité de cette clé sera de 10 ans (durée normale pour une autorité, mais il est toujours possible de mettre la valeur que l'on souhaite). Cette clé sera dans mon cas en `4096bit` et en `sha512`. Taper alors :

```
# openssl req -nodes -config conf/openssl.cnf -days 3650-x509
  -newkey rsa:4096 -sha512 -out public/root.pem -outform PEM
```

Un fichier `root.pem` est alors créé dans le dossier `private/` et `public/`.

Important : copiez le fichier `root.pem` présent dans le dossier public/ dans un endroit accessible à tous les internautes. Ce fichier permet aux personnes d'ajouter votre autorité dans leur navigateur web.

Maintenant que la clé privée et publique sont créées, retournons dans le fichier openssl.cnf pour continuer à le compléter. Ajoutez :

```
# vim conf/openssl.cnf
```

et ajoutez :

```
[ ca ]
default_ca              = CA_default

[ CA_default ]
dir                     = .
new_certs_dir           = ./signed-keys/
database                = ./conf/index
certificate             = ./public/root.pem
serial                  = ./conf/serial
private_key             = ./private/root.pem
x509_extensions         = usr_cert
```



```
name_opt                = ca_default
cert_opt                = ca_default
default_crl_days        = 30
default_days            = 365
default_md              = sha512
preserve                = no
policy                  = policy_match

[ policy_match ]
countryName             = match
stateOrProvinceName     = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

[ usr_cert ]
basicConstraints=CA:FALSE
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always
nsCaRevocationUrl     = domaine.tld/ca-crl.pem
```

Vous pouvez encore une fois adapter la durée de validité du certificat ou la valeur de la signature (`sha1`, `sha128`, `sha256` ou `sha512`). Donner enfin l'URL de la liste des révocations de l'autorité, `nsCaRevocationUrl` (généralement à la « racine » de votre site).

Enfin, nous allons créer un dossier qui va regrouper l'ensemble des certificats signés par l'autorité. Tapez :

```
# mkdir signed-keys && echo "01" > conf/serial && touch conf/index
```

Il sera alors créé un fichier `.pem` dans le dossier `signed-keys/` pour chaque certificat signé.

Bravo, vous êtes désormais une autorité de certification, non reconnue certe, mais parfaitement fonctionnelle pour signer des certificats SSL. Vous êtes parés pour concurrencer Verisign et Google Inc. :-D

Création des certificats SSL

Notations

Sur votre nom de domaine `bidule.fr`, il est important de différencier 2 certificats afin d'éviter des erreurs lors de la navigation :

- le certificat appelé « nowildcard » désignant `https://bidule.fr` ;
- et le certificat appelé « wildcard » désignant tous les sous domaine `https://<sousdomaine>.bidule.fr`.

Création de la clé privée et du premier certificat

Les certificats doivent se trouver dans le dossier suivants sur les distributions RedHat : `/etc/pki/tls/certs/` J'ai créé un dossier `freezetux` pour mon nom de domaine.

Sur les Debian c'est toujours dans `/etc/ssl/certs/`.

Nous allons dans un premier temps générer la clé privée et le certificat nowildcard simultanément. Pour cela taper la commande ci-dessous. Encore une fois il est possible d'adapter les valeurs de chiffrement et de signature) et évidement, les noms sont ceux que j'ai utilisé, modifiez-les pour votre cas. ;-)

```
# openssl req -new -newkey rsa:4096 -sha512 -nodes -keyout
  freezetux.key -out freezetux_nowildcard.csr
```

Ils vous demande de donner des informations sur votre nom domaine. Ma configuration de mon serveur est la suivante, à vous d'adapter selon votre besoin :

```
Country Name (2 letter code) [XX]:FR
State or Province Name (full name) []:Midi-Pyrenees
Locality Name (eg, city) [Default City]:Toulouse
Organization Name (eg, company) [Default Company Ltd]:Freezetux
Organizational Unit Name (eg, section) []:freezetux.net
Common Name (eg, your name or your server's hostname) []:freezetux.net
Email Address []:contact @ freezetux.net
```

Vous pouvez passer les 2 questions posées après celle de l'Émail.

Une fois terminé, vous aurez dans votre dossier un fichier .key et un fichier .csr correspondant au certificat nowildcard.

Pour vérifier votre certificat taper la commande suivante :


```
# openssl req -in freezetux_nowildcard.csr -noout -text
```

Création du second certificat

Nous allons désormais générer le certificat wildcard pour tous les sous-domaines de votre nom de domaine. Pour cela taper la commande suivante :

```
# openssl req -new -sha512 -key freezetux.key -out
  freezetux_wildcard.csr
```

Il va de nouveau vous reposer les mêmes questions que précédement. La seule différence est d'écrire `*.freezetux.net` à la place de `freezetux.net` afin de désigner l'ensemble des sous domaines possible.

Remarque : il est aussi possible de créer un certificat pour chaque sous-domaine existant. Il suffit d'écrire `truc.freezetux.net` pour générer le certificat pour le sous-domaine truc.

Vous pouvez de nouveau vérifier le certificat généré à l'aide de cette commande :

```
# openssl req -in freezetux_wildcard.csr -noout -text
```

Si tout s'est bien passé vous devriez avoir à cette étape 3 fichiers :

- la clé privée `.key` ;
- le certificat nowildcard au format `.csr` ;
- le certificat wildcard au format `.csr`.

Signature des certificats

Maintenant que nous avons tous nos certificats, nous allons désormais les signer par l'autorité de certification créée au début.

Signature du nowildcard

Retournez dans le dossier de votre autorité de certification. Dans mon cas :

```
etc/pki/CA/freezetux/
```

Taper (modifier la destination et le nom des fichiers `.csr` et `.crt` si besoin) :


```
# openssl ca -batch -config conf/openssl.cnf -in /etc/pki/tls/
  certs/freezetux_nowildcard.csr -out /etc/pki/tls/certs/
  freezetux_nowildcard.crt
```

Si tout se passe bien vous devriez avoir à la fin du retour de la commande ces lignes :

```
Certificate is to be certified until Apr  9 18:29:33 2007 GMT
(365 days)

Write out database with 1 new entries
Data Base Updated
```

Un fichier `freezetux_nowildcard.crt` vient normalement d'être créé. Vous pouvez lire ce qu'il contient avec l'éditeur de texte que vous souhaitez afin de vérifier les informations.

Signature du wildcard

Même principe que pour le précédent :

```
# openssl ca -batch -config conf/openssl.cnf -in /etc/pki/tls/
  certs/freezetux_wildcard.csr -out /etc/pki/tls/certs/
  freezetux_wildcard.crt
```

À cette étape nous avons tous les fichiers utiles présents dans le dossier `own_ca/` créé au début :

- la clé privée au format `.key` ;
- les 2 certificats signés au format `.crt` ;
- le `root.pem` public de l'autorité accessible à tous.

Révocation d'un certificat

Dans le cas où un certificat est dépassé, il est possible de le révoquer. Pour cela, retournons une dernière fois dans le fichier de conf de l'autorité `openssl.cnf` et ajouter à la fin :

```
[ crl_ext ]
authorityKeyIdentifier=keyid:always,issuer:always
```

La commande pour révoquer un certificat est la suivante :

```
# openssl ca -config conf/openssl.cnf -revoke /etc/pki/tls/certs/
  freezetux_wildcard.crt
```


Apache et le HTTPS

La dernière étape est de configurer les vhost de Apache pour le HTTPS. Voilà deux exemples de vhost pour le wildcard et nowildcard. Adapter le pour votre configuration.

- Pour le nowildcard :

```
<VirtualHost *:443>
ServerName freezetux.net
DocumentRoot /var/www/<votresiteprincipal>/
SSLEngine On
SSLCertificateFile /etc/pki/tls/certs/freezetux/freezetux_nowild
card.crt
SSLCertificateKeyFile /etc/pki/tls/certs/freezetux/freezetux.key
</VirtualHost>
```

- Pour le wildcard :

```
<VirtualHost *:443>
ServerName truc.freezetux.net
DocumentRoot /var/www/<votresite>/
SSLEngine On
SSLCertificateFile /etc/pki/tls/certs/freezetux/freezetux_wildca
rd.crt
SSLCertificateKeyFile /etc/pki/tls/certs/freezetux/freezetux.key
</VirtualHost>
```

Le fichier `root.pem` doit être accessible par tous. Par exemple, le mien est présent sur http://freezetux.net/root.pem.

Redémarrez Apache et constater du résultat sur vos sites. ;-)

Important : pour une sécurité maximale de votre clé privée, modifiez ses droits en 400 :

```
chmod 400 freezetux.key
```

Sources et remerciements

J'ai pas tout conservé, mais la conf générale de l'autorité de certification réside dans ces 2 liens :

- http://linux-attitude.fr/post/autorite-de-certification-openssl
- http://jikan.fr/devenir-sa-propre-autorite-de-certification/

Merci aux personnes du salon Fedora sur Jabber pour des gros coups de mains lors de mes débuts en certificats SSL. ☺

