Utilisation

Certificats X.509 pour TLS/SSL

Génération d'un certificat

La première étape consiste à générer une clé au moyen d'un algorithme asymétrique.

Génération d'une clé RSA de 2048 bits :

```
openssl genrsa -out /etc/ssl/private/server.key 2048
```

Génération d'une clé ECDSA en utilisant la courbe sect571r1 (NIST/SECG curve over a 571 bit binary field) :

- d'abord on choisit une courbe dans la liste des courbes supportées par OpenSSL :

```
openssl ecparam -list_curves
```

- ensuite on génère la clé :

```
openssl ecparam -out /etc/ssl/private/server.key -name sect571r1 -genkey
```

Une fois la clé en poche on peut générer une CSR (Certificate Signing Request) basée sur cette clé :

```
openssl req -sha256 -new -key /etc/ssl/private/server.key -out
/etc/ssl/private/server.csr
```

S'il nécessaire/utile d'ajouter un SubjectAltName (notamment pour spécifier plusieurs noms d'hôtes valides pour un seul et même certificat), il faut modifier /etc/ssl/openssl.cnf pour qu'il prenne en compte ce réglage (⇒ http://wiki.cacert.org/FAQ/subjectAltName).

Le CSR généré doit être transmis à une autorité de certification qui y répondra en joignant un certificat.

Autorités de certification

Interne (PKI)

Externes

- CACERT est une autorité de certification communautaire (et gratuite), non reconnue par Microsoft Windows et Mozilla Firefox, mais parfois/souvent reconnue par les systèmes UNIX libres (distribution basées sur GNU et Linux, *BSD, etc.) ;
- StartSSL est une startup Israélienne qui distribue des certificats X509 gratuits pour une

utilisation non-commerciale ;
- etc.

autosignature

blah

Paramètres Diffie-Hellman

L'article de l'OpSec de Mozilla dans MISC recommande d'aligner la taille des paramètres Diffie-Hellman sur celle de la clé RSA (si on a choisi une clé RSA).

```
openssl dhparam -2 -out /etc/ssl/private/server.dh 4096
```

Vérifications

⇒ SSL Shopper fournit quelques outils web rapides et simples : http://www.sslshopper.com/ssl-certificate-tools.html ; tous ces outils sont basés sur des commandes OpenSSL simples, détaillées sur le site web.

Décoder un CSR

```
openssl req -in mycsr.csr -noout -text
```

Décoder un certificat X.509

```
openssl x509 -in certificate.crt -text -noout
```

Valider une installation TLS/SSL

```
openssl s_client -connect www.mouet-mouet.net:443
```

PKI

Autorité de certification

Génerer une clé RSA de 2048 bits, et la stocker dans un fichier chiffré en utilisant AES 256 (protégé par une phrase de passe) :

```
openssl genrsa -aes256 -out ca.key 2048
```

Générer un certificat basé sur cette clé (certificat racine) :

```
openssl req -new -x509 -days 3650 -key ca.key -out ca.crt
```

Signature d'un CSR

Une fois un CSR généré (à partir d'une clé elle aussi générée), il doit être signé par une autorité de confiance:

```
openssl x509 -req -sha256 -days 730 -in host.network.tld.csr -CA ca.crt -
CAkey ca.key -set_serial 01 -out host.network.tld.crt
```

Le numéro de série doit être différent pour chaque certificat signé avec cette clé. On peut incrémenter un numéro ou prendre un hachage du CSR...

Distribution du certificat racine

Le certificat racine doit être distribué pour pouvoir être utilisé comme base de validation des certificats signés.

Configuration

blah

Bibliographie

- site web officiel : http://www.openssl.org/ ;
- Wikipedia : http://en.wikipedia.org/wiki/OpenSSL http://fr.wikipedia.org/wiki/OpenSSL ;
- http://www.openbsd.org/faq/faq10.html#HTTPS ;
- http://en.wikipedia.org/wiki/X.509 http://fr.wikipedia.org/wiki/X.509 ;
- http://users.dcc.uchile.cl/~pcamacho/tutorial/crypto/openssl/openssl_intro.html ;
- etc.