Configuration de CAS avec des certificats auto-signés (pour des tests)

On suppose ici que le serveur CAS et uPortal fonctionnement avec la même instance de tomcat. Cette partie explique comment faire fonctionner CAS avec des certificats auto-signés.

Cette partie de documentation n'est utile que pour des tests ; il est déconseillé d'utiliser des certificats auto-signés en production, une PKI devient vite obligatoire.

Création du certificat

On procède d'abord par la génération du bi-clé, dans un keystore (magasin de certificats) java.

On désire ensuite enregistrer juste le certificat dans un keystore de certificats 'trusted' (autorisés).

On peut l'enregistrer dans le keystore de la JVM ; cette méthode n'est pas la meilleure. Il est préférable de l'enregistrer dans un keystore spécifique, qui sera pris en compte lors du lancement de tomcat au lieu du keystore de la JVM.

Génération du couple de clés

On commence par générer un couple de clés (certificat auto-signé) pour le serveur (cas.univ-xxx.fr) et on stocke dans le magasin serveur.keystore :

```
% keytool -genkey -alias tomcat -dname "CN=cas.univ-xxx.fr,OU=X,O=Y,L=Z,S=XY,C=YZ" -keyalg RSA -storepass
yyyyyy -keystore cas.keystore
Enter key password for <tomcat>
(RETURN if same as keystore password):
```

cas.keystore contient à la fois la clé privée du serveur, et le certificat auto-signé qui contient la clé publique.

Exportation du certificat auto-signé

On exporte ensuite le certificat dans un fichier (cas.cert) :

```
% keytool -export -alias tomcat -storepass yyyyyy -file cas.cert -keystore cas.keystore
Certificate stored in file <cas.cert>
```

cas-cert ne contient que le certificat, sans la clé privée.

Ajout du certificat dans le magasin global du JDK

Pas souhaitable. Il est préférable d'enregistrer ce certificat dans un kestore dédié à une instance de tomcat. Voir le paragraphe suivant.

Rappel : le mot de passe par défaut du magasin global de Java est "changeit".

```
% keytool -import -v -trustcacerts -alias tomcat -file cas.cert -keystore $JAVA_HOME/jre/lib/security/cacerts -
keypass xxxxxx -storepass changeit
Owner: CN=cas.ifsic.univ-rennes1.fr, OU=X, O=Y, L=Z, ST=XY, C=YZ
Issuer: CN=cas.ifsic.univ-rennes1.fr,   OU=X, O=Y, L=Z, ST=XY, C=YZ
Serial number: 3eb79610
Valid from: Tue May 06   13:01:36 CEST 2003 until: Mon Aug 04 13:01:36 CEST 2003
Certificate fingerprints:
   MD5: 0B:20:52:BC:7C:EA:28:58:16:FB:3C:2F:C4:D2:E2:35
   SHA1: 00:5A:F5:81:B1:07:1C:EB:C3:1F:C4:89:5E:76:87:43:D7:8F:F0:5F
Trust this certificate? [no]: yes
Certificate was added to keystore
[Saving /usr/java/j2sdk1.4.1_02/jre/lib/security/cacerts]
```

Contrôle :

```
% keytool -list -keystore $JAVA_HOME/jre/lib/security/cacerts -storepass changeit
Keystore type: jks
Keystore provider: SUN

Your keystore contains 16 entries

[...]
tomcat, May 6, 2003, trustedCertEntry,
Certificate fingerprint (MD5): 0B:20:52:BC:7C:EA:28:58:16:FB:3C:2F:C4:D2:E2:35
[...]
```

Copier le fichier serveur.keystore dans un répertoire non listable HTTP (par exemple /etc/x509).

Enregistrement du certificat dans un keystore 'privé'

C'est la méthode préconisée. Au lieu d'enregistrer le certificat (qui ne contient que la clé publique du serveur) dans le keystore global de la JVM, on va l'enregistrer dans un keystore (cas-cacerts) qui sera déclaré lors du lancement de tomcat, et utilisé à la place du keystore global.

```
keytool -import -v -trustcacerts -alias tomcat -file cas.cert -keystore cas-cacerts -keypass xxxxxx -storepass
changeit
```

Maintenant, lors du lancement de tomcat qui doit faire confiance à CAS :

```
CATALINA_OPTS="-Djavax.net.ssl.trustStore=cas-cacerts";export CATALINA_OPTS
```

Configuration de Tomcat

Modifier le fichier server.xml de tomcat (marche sur tomcat 5.5 et 6) :

```
 <Connector port="8443" maxHttpHeaderSize="8192"
                      maxThreads="2000" minSpareThreads="25" maxSpareThreads="75"
                      enableLookups="false" disableUploadTimeout="true"
                      acceptCount="200" scheme="https" secure="true" SSLEnabled="true"
                       clientAuth="false" sslProtocol="TLS"
                       keystoreFile="/etc/x509/cas.keystore" keystorePass="yyyyyyy" />
```