Despliegue de Aplicaciones Web - DAW Altair 2018/19

Java KeyTool

Juan Antonio León

¿Qué es Java KeyTool?

Java Keytool es un comando que permite generar pares de Private Key/Public Key y almacenarlos en un Java KeyStore.
Java keyStore (JKS) es un almacén de certificados y entidades de certificación usado por las aplicaciones Java para trabajar con SSL. Cada certificado contenido en la keyStore es identificado por un alias único.

Generar un certificado autofirmado

>> keytool -genkey -alias **alias** -keypass **clave** -validity **365** -storepass **claveKeyStore**

- -alias: es el nombre con el que haremos referencia al par de claves creado.
- -keypass: es la clave con la que podremos acceder a la clave privada del par de claves creado.
- -validity: es el tiempo de validez, en días. En nuestro ejemplo, un año.
- -storepass: clave para acceder a nuestro keystore.

Generar el CSR

>> keytool -certreq -alias **alias** -file **certificado.csr** -keypass **clave** -storepass **claveDeKeyStore**

- -file: nombre del fichero de salida, que luego mandaremos a la CA.

Crear nuestro propia CA

OpenSSL nos proporciona un script que nos facilita la tarea de crear la CA. Basta con hacer:

```
>> /usr/lib/ssl/misc/CA.sh -newca
```

- Nos pide el fichero con el certificado de la CA. Si pulsamos “enter” nos creará uno de forma automática (nosotros elegimos esta opción).
- Ahora nos pide la clave para acceder a la clave privada del nuevo certificado que está creando (el certificado de la CA).
- Nos pide el código de país, la provincia, la ciudad, la organización, la unidad organizativa, el nombre y la dirección de correo.

Firmar el CSR

Ya tenemos la CA, ahora firmamos el CSR que habíamos generado en el punto 4. Para esto también usaremos el script CA.sh. Este script trabaja con nombres fijos de ficheros, con lo que espera encontrar el CSR con el nombre “newreq.pem”.

Lo que vamos a hacer es copiar nuestro CSR al directorio padre de “demoCA” con el nombre “newreq.pem”. Es decir, nuestro CSR debe quedar a la misma altura que el directorio “demoCA”, y no dentro de él.

Instalar el certificado de la CA

>> keytool -import -alias **CAcert** -keypass **claveCAcert** -file demoCA/cacert.pem -storepass **claveDeKeyStore**

Si quisiéramos instalar el certificado de la CA en el cacerts keystore podríamos hacer:

>> keytool -import -alias **autentiaCaCert** -keypass **claveDeAutentiaCaCert** -file **demoCA/cacert.pem** -keystore $JAVA_HOME/jre/lib/security/cacerts -storepass **claveDeKeyStore**

Instalar nuestro certificado firmado por la CA

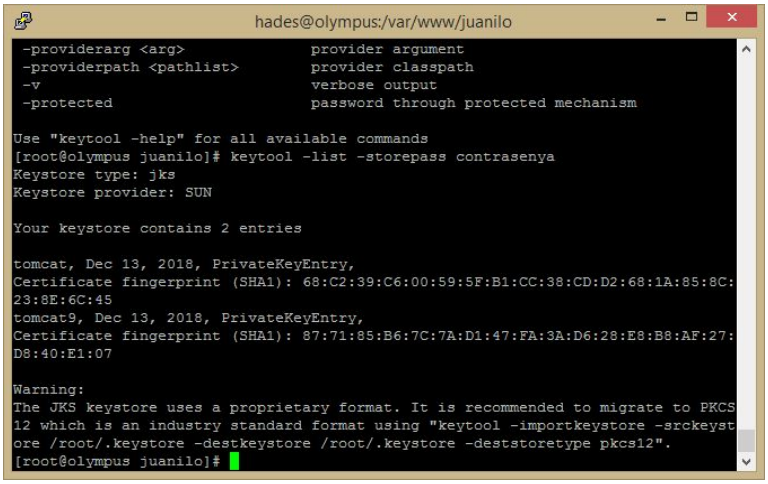
keytool -import -alias **autentiaCert** -keypass **claveDeAutentiaCert** -file **autentiaCertFirmadoPorCA.pem** -storepass **claveDeKeyStore**

Ahora podemos explorar nuestro keystore.

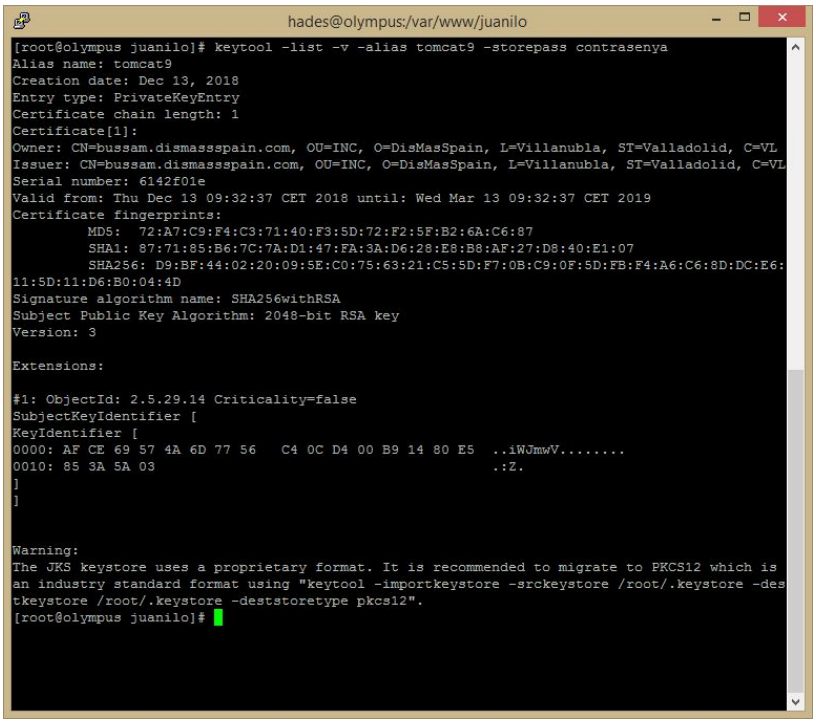
Explorar nuestro keystore

Mostrar el contenido actual del keystore

>> keytool -list -storepass **claveKeyStore**



Mostrar el detalle del certificado elegido



>> keytool -list -v -alias

alias-storepass **claveDeKeyStore**

Gestionar nuestro keystore

Crear la keyStore

```
keytool -genkey -alias alias -keyalg RSA -keysize 2048 -keystore keystore.jks
```

Cambiar contraseña de la keyStore

```
keytool -storepasswd -keystore keystore.jks
```

Importar certificado

```
keytool -import -trustcacerts -alias alias  -keystore keystore.jks
```

Borrar certificado

```
keytool -delete -alias alias -keystore keystore.jks
```