OpenSSL y keytool

Preparacion

- crear el directorio *demoCA* (según configuración *openssl.cnf*)
- crear el subdirectorio *newcerts*
- crear el archivo vacío *index.txt*
- crear el archivo *serial* con contenido **01**
- crear el archivo *crlnumber* con contenido **00**

Generación claves CA

openssl req -x509 -newkey rsa:2048 -keyout cakey.pem -days 3650 -out cacert.pem

DN *cn=ca,o=umu, ou=ppc, c=es* (OJO: case-sensitive, campos NO usados en blanco)

Generación claves usuario

keytool -genkey -keystore servidor.ks -keyalg rsa

DN: *cn=localhost(o la dns/ip del sitio web, en caso de poder llamarse con distinto nombre ha de crearse varios certificados, uno por cada dns/ip),ou=ppc, o=umu, c=es* (OJO: case-sensitive, campos NO usados en blanco)

keytool -changealias -keystore servidor.ks -alias mykey -destalias servidor

keytool -list -v -keystore servidor.ks

Firma certificado usuario

keytool -certreq -file servidor.csr -keystore servidor.ks

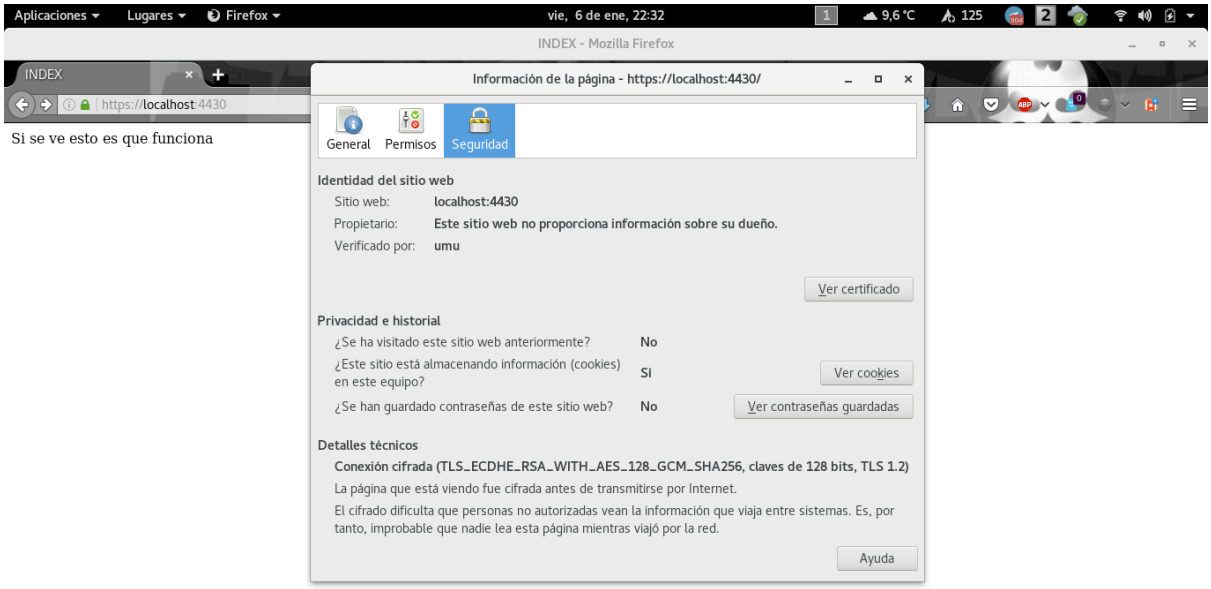
openssl ca -keyfile cakey.pem -in servidor.csr

keytool -import -trustcacerts -alias CA -file cacert.pem -keystore servidor.ks

keytool -import -alias servidor -file servidor.pem -keystore servidor.ks

keytool -list -v -keystore servidor.ks

Por ultimo, añadimos como CA de confianza en nuestro explorador a la CA creada, y ya funciona nuestro certificado, aquí imagen de demostración.



Revocación certificado usuario

openssl ca -keyfile cakey.pem -revoke servidor.pem

openssl ca -keyfile cakey.pem -gencrl -out crl.pem

openssl crl -in crl.pem -outform DER -out crl.der

openssl crl -text -inform DER -in crl.der

Exportación claves usuario (PKCS#12)

- % descargar e instalar la aplicación **portecle-1.4**
- % necesita el JDK 6.0+
- % necesita la instalación del Java *Unlimited Strength Cryptography*
- % abrir el keystore *servidor.ks*, y exportar el certificado servidor (ambas claves, formato PKCS#12)
- % se generará el archivo *servidor.p12*