mod17_ex06: Managing x509 Format

The purpose of this exercise is to develop administrative skills for the openssl command. The openssl command is used to create and to manage certificates. An important function of the openssl command is converting certificates from one protocol to another.

Reference Information

The following documents provide information related to this exercise.

- TBD

1. Understanding the x509 Standard

1.1 Describing x509 Standard.

One of the confusions around Public Key certificates is the number of formats used for these keys. The International Telecommunications Union defines all of these formats in the X.509 standard. The X.509 public key certificates are used in many Internet protocols; TLS is just one of many. They are also used for electronic signatures in a wide range of devices. The basic description is a X.509 certificate must have a public key which includes an identity (a hostname, or an organization, or an individual), and is either signed by a certificate authority or is self-signed. When a public key certificate is signed by a trusted certificate authority someone holding the certificate can rely upon the public key to establish secure communications with another party.

1.2 X.509 Certificate Encoding Formats and Extensions

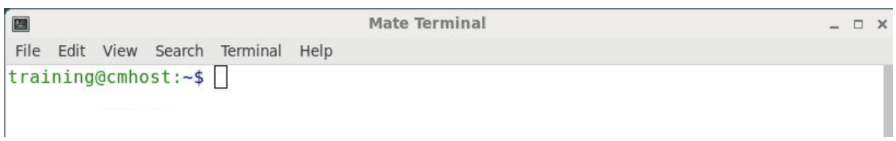
X.509 certificates can be in text or binary. The openssl tool is used to convert from one format to another. This is an important function for the openssl tool.

- **Base64 (ASCII)**
- PEM
- .pem Privacy Enhanced Mail (all of these keys are pem files)
- .key private key
- .csr certificate signing request
- .crt certificate = signed public key
- .ca-bundle
- PKCS#7
- .p7b
- .p7s

- **Binary**
- DER
- .der
- .cer
- PKCS#12
- .pfx
- .p12

*.pem, *.crt, *.ca-bundle, *.cer, *.p7b, *.p7s files contain one or more X.509 digital certificate files that use base64 (ASCII) encoding. You get one of these in a zip file downloaded from your user account or receive such file from the Certificate Authority.

2. Open a Mate terminal as the user training.



3. Converting x509 formats.

3.1 Change directory to client.

```
Mate Terminal
File  Edit  View  Search  Terminal  Help
training@cmhost:~/tls/local$ cd ~/tls/client
training@cmhost:~/tls/client$
```

```
$ cd ~/tls/client
```

3.2 Issue the openssl help command.

```
training@cmhost:~/tls/client$ openssl pkcs12 -help
Usage: pkcs12 [options]
where options are
-export       output PKCS12 file
-chain        add certificate chain
-inkey file   private key if not infile
-certfile f   add all certs in f
-CApath arg   - PEM format directory of CA's
-CAfile arg   - PEM format file of CA's
-name "name"  use name as friendly name
-caname "nm"  use nm as CA friendly name (can be used more than once).
-in  infile   input filename
-out outfile  output filename
-noout        don't output anything, just verify.
-nomacver     don't verify MAC.
```

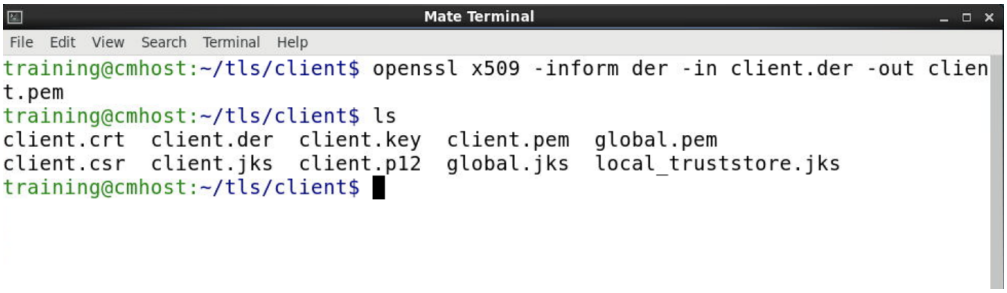
```
$ openssl pkcs12 -help
```

3.3 Convert a PEM certificate to DER format. DER is one of the binary formats for certificates.

```
Mate Terminal
File  Edit  View  Search  Terminal  Help
training@cmhost:~/tls/client$ ls
client.crt  client.jks  client.p12  global.pem
client.csr  client.key  global.jks  local_truststore.jks
training@cmhost:~/tls/client$ openssl x509 -in client.crt -outform der -out clie
nt.der
training@cmhost:~/tls/client$ ls
client.crt  client.der  client.key  global.jks  local_truststore.jks
client.csr  client.jks  client.p12  global.pem
training@cmhost:~/tls/client$
```

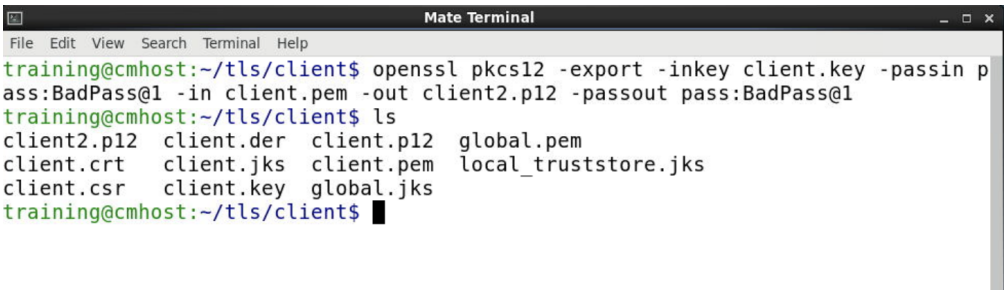
```
$ openssl x509 -in client.crt -outform der -out client.der
```

3.4 Convert the DER certificate to a PEM format.




```
$ openssl x509 -inform der -in client.der -out client.pem
```

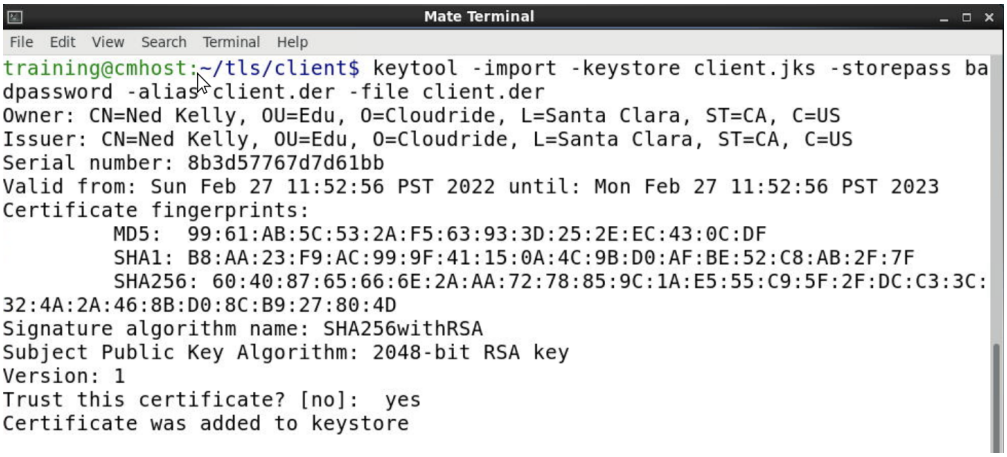
3.5 Convert the PEM certificate to P12 format. This is the most common binary format.




```
$ openssl pkcs12 -export -inkey client.key -passin pass:<password> -in
client.pem -out client2.p12 -passout pass:<password>
```

4. Importing x509 formats into Java keystore.

4.1 Import the der format key into local.jks. Password is badpassword. Add it.




```
$ keytool -import -keystore client.jks -storepass badpassword -file
client.der -alias client.der
```