Practical Networking

Practical TLS – Lab Guide

A deep dive into SSL and TLS – the protocols that secure the Internet

Lab 3.4 – File Format Conversions

In this lab you will be given some PEM, DER, and PFX files, and you will be tasked with converting each into other formats. This will give you practice using OpenSSL to do file conversions.

In each case, the full command required will be provided -- but this lab should be more than merely typing practice. Pay close attention to each command argument and try to understand what you are providing to OpenSSL and what you are receiving as output.

Create working directory and acquire lab files

1. Navigate to the folder you created in Lab 0.0 named Practical-TLS
   - `$ cd Practical-TLS`

2. Create a new directory for Lab 3.4 files:
   - `$ mkdir "LAB3.4 - File Format Conversions"`

3. Download one of the following lab files and place it in the newly created folder:
   - **LAB 3.4 Files - File Format Conversions - Practical TLS.tar.gz**
   - **LAB 3.4 Files - File Format Conversions - Practical TLS.zip**

   The lab files are in `.zip` or `.tar.gz` format – the contents within are identical, use whichever file is easier for you

4. Unzip or Untar the file you downloaded into your LAB 3.4 working directory:
   - For the TAR.GZ file, you will use the Linux command: `tar -xvzf "LAB 3.4 ...`
   - For the ZIP file, it should be something like right clicking the file and selecting "Extract All..."

5. When finished, you should have the following 5 files in the LAB 3.4 directory:

```
pippin.com.PEM.cert
pippin.com.PEM.key
merry.com.DER.cert
merry.com.DER.key
sam.com.PFX
```

Converting PEM formatted files

PEM formatted files can be viewed using the Linux `cat` utility, or opened in any text editor. A PEM formatted file is identified by the presence of lines such as:

```
-----BEGIN CERTIFICATE-----            -----BEGIN RSA PRIVATE KEY-----
-----END CERTIFICATE-----              -----END RSA PRIVATE KEY-----
```

1. Examine the PEM formatted files:
   - `cat pippin.com.PEM.cert`
   - `cat pippin.com.PEM.key`
   - `openssl x509 -in pippin.com.PEM.cert -noout -text`
   - `openssl rsa -in pippin.com.PEM.key -noout -text`
2. Convert PEM formatted files into DER files
   - `openssl x509 -in pippin.com.PEM.cert -outform DER -out pippin.com.DER.cert`
   - `openssl rsa -in pippin.com.PEM.key -outform DER -out pippin.com.DER.key`
3. Convert PEM formatted Certificate and Key into a PFX file
   - `openssl pkcs12 -in pippin.com.PEM.cert -inkey pippin.com.PEM.key -export -out pippin.com.PFX`

     *(NOTE: this entire command should be typed on **one line**)*
     - *You will be asked for an Export Password -- press enter to leave it blank*
4. Attempt to view the files you just created using the `cat` utility *(these will all fail)*
   - cat pippin.com.DER.cert
   - cat pippin.com.DER.key
   - cat pippin.com.PFX
5. View the files you created using OpenSSL to interpret the binary encoded DER and PFX files
   - openssl x509 -in pippin.com.DER.cert -inform DER -noout -text
   - openssl rsa -in pippin.com.DER.key -inform DER -noout -text
   - openssl pkcs12 -in pippin.com.PFX -nodes -info
     - *You will be asked for an Import Password -- press enter to leave it blank*
     - Note that the *single* PFX file we created contains *both* the Certificate and Key file.

Converting DER formatted files

DER formatted files are binary encoded, which means you will not be able to open them in a text editor. To identify you are working with a DER formatted file, you can try to interpret the file with `OpenSSl` and the `-inform DER` argument (see step 2 for an example).

1. Attempt to view the DER files using the cat utility *(these will all fail)*
   - `cat merry.com.DER.cert`
   - `cat merry.com.DER.key`
2. View the DER files using OpenSSL:
   - `openssl x509 -in merry.com.DER.cert -inform der -noout -text`
   - `openssl rsa -in merry.com.DER.key -inform der -noout -text`
3. Convert DER formatted files to PEM
   - `openssl x509 -in merry.com.DER.cert -inform der -out merry.com.PEM.cert`
   - `openssl rsa -in merry.com.DER.key -inform der -out merry.com.PEM.key`
4. Convert DER formatted files to PFX
   - `openssl pkcs12 -in merry.com.PEM.cert -inkey merry.com.PEM.key -export -out merry.com.PFX`

     *(NOTE: this entire command should be typed on **one line**)*
     - *Note that you cannot convert directly from DER to PFX, you must first convert to PEM*
5. View the newly created files using `cat` and the appropriate `openssl` commands
   - All necessary commands were shown in the prior tasks.

It should be noted that a DER formatted Private Key is very rare. We provided one in this lab merely to provide additional practice with file format conversions.

Examining the contents of a PFX file

PFX formatted files are binary encoded, which means you will not be able to open them in a text editor. To identify you are working with a PFX formatted file, you can try to interpret the file with `OpenSSL` and the `pkcs12` utility -- this will be shown in step 1 of this task.

A PFX file is a container which can hold many different types of certificates and/or keys. Moreover, PFX files have two password features:

- **Import / Export passwords** -- this sets a password on the entire PFX file. If you set an export password when creating the PFX file, that same password is required when using or reading the file in the future (i.e., when importing the file). This adds a layer of security when transferring PFX files between systems.
- **Key Pass Phrase** -- Anytime openssl pkcs12 displays the Private Key, it attempts to encrypt it using DES; which is a Symmetric Encryption algorithm and therefore requires a secret key. The Key Pass Phrase it requests *is* the encryption key. You can skip this option and display the key in plain text (in PEM format) by specifying the `-nodes` argument (i.e., no DES encryption).

The file `sam.com.PFX` does not have an Import / Export password set -- anytime you are prompted for one you can simply leave it blank and hit enter.

1. Attempt to interpret the PFX and DER file using openssl pkcs12 *(the DER file will fail)*
   - openssl pkcs12 -in sam.com.PFX -noout -info
   - openssl pkcs12 -in merry.com.DER.cert -noout -info
2. Attempt to view the PFX file using the `cat` utility *(this will fail)*
   - cat sam.com.PFX
3. Examine the contents of the PFX file:
   - openssl pkcs12 -in sam.com.PFX -nodes
     - How many Certificates are in the PFX file?
     - Do you see each of their Subject and Issuer DNs?
     - How many Private Keys are in the PFX file?

Converting PFX formatted files

OpenSSL can only convert PFX files into PEM format. If you must convert something into DER format, you can use the PEM to DER conversion commands in the task earlier in this lab.

At first we will export everything contained in the PFX file. Then we will run through a few different options that allow you to extract only specific portions.

For each conversion, use `cat` on the output file to validate what was exported.

1. Extract everything from the PFX file:
   - `openssl pkcs12 -in sam.com.PFX -out sam.com.PEM.ALL -nodes`
2. Extract only the certificates:
   - `openssl pkcs12 -in sam.com.PFX -out sam.com.PEM.CERTS -nokeys`
3. Extract only the CA certificate(s):
   - `openssl pkcs12 -in sam.com.PFX -out sam.com.PEM.CA-CERTS -nokeys -cacerts`
4. Extract only the Client certificate:
   - `openssl pkcs12 -in sam.com.PFX -out sam.com.PEM.EE-CERT -nokeys -clcerts`
5. Extract only the Private Key file:
   - `openssl pkcs12 -in sam.com.PFX -out sam.com.PEM.KEY -nodes -nocerts`
6. Extract the Client Certificate and Private Key file (no CA certs):
   - `openssl pkcs12 -in sam.com.PFX -out sam.com.PEM.EE-CERT-KEY -nodes -clcerts`
7. Use the appropriate `openssl x509` or `openssl rsa` command to examine each file exported
8. Convert a few of the PEM files you created into DER format