Report on Certificate Tools

Oliver Bodnár

August 2016

Contents

0.1 Overview

0.2 OpenSSL

0.2.1 type

OpenSSL uses PKCS12 to store keys and or certificates. Certificates and keys made in OpenSSL however are being made into PEM or DER encoded file. Said keys/certificates can then be stored inside single .pfx file made with

0.2.2 Viewing stored certificates

PEM encoded certificates (.pem—.cer—.crt): openssl x509 -in sample$_{c}ert.extention-text-noout$

DER encoded certificates (.der): openssl x509 -in certificate.der -inform der -text -noout

Importing PEM or DER encoded keys or certificates into PKCS12 file:

openssl pkcs12 -export -in file.pem -out file.p12 -name "My Certificate" -certfile othercerts.pem

-certfile othercerts.pem option is used only if importing more certificates into 1 PKCS12 file is wanted.

0.2.3 License

OpenSSL is free to use commercialy, however creating CA is not advised - OPENSSL O'REILY page 59 - but also hinted in manual pages - The ca command is quirky and at times downright unfriendly.

The ca utility was originally meant as an example of how to do things in a CA . It was not supposed to be used as a full blown CA itself: nevertheless some people are using it for this purpose.

The ca command is effectively a single user command: no locking is done on the various files and attempts to run more than one ca command on the same database can have unpredictable results.

0.2.4 Generation of / signing a certificate

Generate keys and self-signed certificate

It is possible to generate private key and self signed certificate with command openssl req -x509 -sha256 -nodes -days 365 -newkey rsa:2048 -keyout privateKey.key -out certificate$_{n}ame.crt$

Generate keys and Certificate Signing Request

It is possible to generate private key and CSR with command openssl req -out CSR.csr -new -newkey rsa:2048 -nodes -keyout privateKey.key

Certificate Signing Request signing

CSR signing can be done by CA created in openSSL by command openssl ca -config ca/openssl.cnf -extensions server$_{c}ert - days375 - notext - mdsha256 - inca/csr/www.example.com.csr.pem - outca/certs/www.example.com.cert.pem$

Create combination of private key and signed chain

First creation of the request is needed, after that request must be signed. To combine them together the creation of pkcs12 file is needed with commands: openssl pkcs12 -export -out outfilename.p12 -in signed$_{c}ertificate.crt - inkeyprivateKey.key - chain - CAfileca - all.crt - passwordpass : PASSWORD$

0.2.5 Specification of key length and algorithm

Key length and algorithm is specified while generating key, currently OpenSSL supports Public-key cryptography algorithms: RSA, DSA, Diffie-Hellman key exchange, Elliptic Curve

In the past also support for GOST R 34.10-2001 but as of January 2016 deprecated (https://mta.openssl.org/pipermail/openssl-commits/2016-January/003023.html)

0.2.6 Specification of validity

Validity is specified by CA at the moment of signing CSR.

0.2.7 Basic Constraints

Specification of constraints is made by changing basicConstraints part in [v3$_{r}eq]partofopenssl.cnfofCA[v3_{r}eq]$

basicConstraints=critical,CA:BOOL$_{V}AL, pathlen :< maxChainLengthInteger >$

0.2.8 Seting Subject Alternative Name for end certificates

To set SAN the change of subjectAltName in openssl.cnf is needed

[v3$_{r}eq]subjectAltName = @alt_{n}ames$

[alt$_{n}ames]DNS.1 = example1.comDNS.2 = example2.comDNS. < next_{n}umber >= dns_{w}ebaddress.com$

0.2.9 Using Cryptographic Service Provider

//TO DO

0.2.10 Conversions

PKCS12 and JKS conversion in OpenSSL //AFAIK cannot be done using only OpenSSL needed to be tested/researched more

0.2.11 Export certificate only from PKCS12 file

It can be done with command: openssl pkcs12 -in [yourfile.pfx] -clcerts -nokeys -out [certificate.crt]

0.2.12 Export private key only from PKCS12 file

To export private key use command: openssl pkcs12 -nocerts -in [filename.pfx] -out [site.key]

0.2.13 Importing certificates and private keys into PKCS12 file

//should be possible with openssl pkcs12 -export -in file.pem -out file.p12 -name "My Certificate" -certfile othercerts.pem but need to test it.

0.3 Keygen