Uebung 3

Philip Magnus

October 31, 2024

Documentation for OpenSSL

Abstract

The following documentation is meant to describe the processes for validating certificates with OpenSSL via CRL and OCSP. The document describes also describes the generating and using of RSA private keys to create CSRs and self sign certificates. Additionally the document describes how to generate ECDSA private keys and using them to create CSRs. Finally a process is described to convert certificates in the pem format to the der format.

Validating Certificates

Validating certificates via CRL

To verify a certificate OpenSSL needs all certificates in the certificate chain, OpenSSL is not capable to extract the certificates by itself. For our example you should download the SSL certificates from wikipedia.org as well as the certificate from the CA. The certificate can be found via the certificate for wikipedia.org. Download the "DigiCert TLS Hybrid ECC SHA384 2020 CA1" and save it uinder a name you can remember, in our example *digicert-ca1.pem*.

Now you can use the following OpenSSL command to verify if the wikipedia certificate is listed in the CRL. In our example we expect the certificate to not be listed and thus being a valid certificate.

```
$ openssl verify -crl_check -crl_download -CAfile digicert-ca1.pem wikipedia-org.pem

# Expected output: wikipedia-org.pem: OK
```

Validating certificates via OCSP

To verify a certificate via the Online Certificate Status Protocol (OCSP) we first need to obtain the OCSP-URI from the wikipedia certificate.

For this use the following OpenSSL-Command:

```
$ openssl x509 -noout -ocsp_uri -in wikipedia-org.pem
# Example output: http://ocsp.digicert.com
```

With the now obtained OCSP-URI we can use the following command to validate the certificate:

```
$ openssl ocsp -issuer digicert-ca1.pem -cert wikipedia-org.pem -url "http://ocsp.digicert.com" -text

# Example output:
# OCSP Request Data:
#     Version: 1 (0x0)
#     Requestor List:
# [...]
# Response verify OK
# wikipedia-org.pem: good
#         This Update: Oct 31 00:03:01 2024 GMT
#         Next Update: Nov  6 23:03:01 2024 GMT
```

Generating self signed certificates

Generating RSA private keys

To generate a RSA private-key, with a 3072 Bit length, which is secured by a passphrase use the following command:

```
$ openssl genrsa -aes256 -passout pass:<password> -out <private.key> 3072
```

To decrypt the encrypted RSA key use th efollowing command:

```
$ openssl rsa -in <private.key> -out <private_dec.key>
```

You will need to enter your passphrase and the decrypted key will be written to the file specified by the out argument. Private keys are sensitive data, please keep them secret.

Generating a self signed certificate via RSA key

To generate a self signed certificate we need to work in two parts. First we need to create a Certificate Signing Request (CSR):

```
$ openssl req -new -key <private.key> -out <request.csr>

# Here you will be prompted to fill out the necessary information needed for the CSR
```

With the next command you will generate the certificated itself and sign it with your generated RSA key.

```
$ openssl x509 -req -in <request.csr> -signkey <private.key> -out <certificate.pem> -days 365

# The days option sets how long the generated and signed certificate will be valid.

# Example output:
# Enter pass phrase for private-key.pem:
# Certificate request self-signature ok
# subject=C = AT, ST = Vienna, L = Vienna, O = Test-Org, CN = ServerName, emailAddress = test@test.com
```

The contents of your generated certificated can be viewed with the following command:

```
$ openssl x509 -in certificate.pem -text -noou
```

Generating a ECDSA private key and CSR

The following commands are used to generate a ECDSA private key and the corresponding CSR:

```
# Generate ECDSA private key

$ openssl ecparam -name prime256v1 -genkey -out <private-ecdsa.key>

# Create the corresponding CSR this works like the CSR creation mentioned in the
# self signed certificate section

$ openssl req -new -key <private-ecdsa.key> -out <request_ecdsa.csr>
```

Converting certificates in PEM to DER format

To convert a given certificate in pem format to a certificate in der format you can use the following command:

```
$ openssl x509 -in <certificate.pem> -outform der -out <certificate.der>
```