Anleitung zur Nutzung von OpenSSL in der DFN-PKI

Kontakt:

Allgemeine Fragen zur DFN-PKI: pki@dfn.de

Technische Fragen zur DFN-PKI: dfnpca@dfn-cert.de

1 OpenSSL

OpenSSL ist eine Open-Source-Implementierung des SSL/TLS-Protokolls und bietet darüber hinaus weitergehende Funktionen zur Zertifikat-Verwaltung und zu unterschiedlichen kryptographischen Funktionen.

Sie können OpenSSL nutzen, um in der DFN-PKI z.B. einen PKSC#10 Zertifikatantrag zu erzeugen.

OpenSSL ist in den meisten Linux Distributionen vorinstalliert, für Microsoft Windows Systeme gibt es OpenSSL z.B.

- als ausführbares Programm von Cygwin unter http://www.cygwin.com/
- als kleineres Installer-Paket von http://www.slproweb.com/products/Win32OpenSSL.html

2 Erzeugen eines PKCS#10 Zertifizikatantrags (Certificate Signing Request, CSR)

Wenn Sie in der DFN-PKI über die Webschnittstelle für Nutzer und Administratoren ein Serverzertifikat beantragen wollen, müssen Sie dort an entsprechender Stelle den Dateinamen eines PKCS#10 Zertifikatantrags (CSR) angeben. Im folgenden wird beschrieben, wie Sie einen RSA Schlüssel (privater und öffentlicher Schlüssel) und den dazugehörigen PKCS#10 Zertifikatrequest (CSR) mit Hilfe von OpenSSL erzeugen können.

2.1 Erzeugen eines RSA Schlüssels

Im Sicherheitsniveau Global der DFN-PKI wird eine Schlüssellänge von 2048 Bit gefordert. Einen 2048 Bit langen RSA Schlüssel können Sie mit folgendem OpenSSL-Kommando erzeugen:

```
openssl genrsa -aes256 -out key.pem 2048
```

In der Datei key.pem werden der private und der öffentliche Schlüssel abgelegt.

Eine Schlüsseldatei ohne Passwortschutz erhalten Sie durch Weglassen der Option `-aes256`. Die ungeschützte Schlüsseldatei muss unbedingt mit anderen Mitteln (z.B. Dateizugriffsrechte) vor unbefugtem Zugriff geschützt werden.

Haben Sie ein spezielles Pseudo-Random-Device (etwa /dev/random oder /dev/urandom) konfiguriert, so können Sie mit

```
openssl genrsa -aes256 -rand /dev/random -out key.pem 2048
```

den Schlüssel mit diesem Pseudozufallszahlengenerator erzeugen.

2.2 Erzeugen eines CSR – RSA Schlüssel liegt vor

Ein RSA Schlüssel (ggf. samt zugehörigem Passwort, wenn dieses gesetzt ist) mit mindestens 2048 Bit Schlüssellänge liegt im PEM-Format in der Datei key.pem vor, die mit der Zeile

'-----BEGIN RSA PRIVATE KEY-----' beginnt und mit der Zeile

'-----END RSA PRIVATE KEY-----' endet

Diese Datei kann mit dem oben beschriebenen OpenSSL Kommando oder mit einem anderen Programm erzeugt worden sein.

Um mit diesem Schlüssel einen Zertifikatrequest zu erzeugen, können Sie folgendes OpenSSL-Kommando verwenden:

```
openssl req -batch -sha256 -new -key key.pem -out request.pem –subj
'/C=DE/ST=<Bundesland/L=<Ort>/O=<Einrichtung>/OU=<Abteilung>/CN=<FQDN>/
emailAddress=<gültige E-Mail-Adresse des Server-Administrators>'
```

Es wird ein Zertifikatrequest für den in der Datei key.pem abgelegten RSA Schlüssel erzeugt.

Die Angaben zum Distinguished Name (DN) des Zertifikats werden im Parameter –subj eingetragen. Dabei sind die Regeln für Zertifikatnamen zu beachten (s. Zertifikatnamen)

Der Zertifikatrequest wird in der Datei request.pem abgelegt.

2.3 Erzeugen eines CSR - RSA Schlüssel wird gleichzeitig erzeugt

Wenn noch kein RSA Schlüssel vorliegt, kann er auch gleichzeitig mit dem Zertifikatrequest erzeugt werden. Dafür können Sie folgendes OpenSSL-Kommando verwenden:

```
openssl req -newkey rsa:2048 -keyout key.pem -out request.pem -subj
/C=DE/O=Test-PKI/CN=testserver.de
```

Die Angaben zum Distinguished Name (DN) des Zertifikats werden im Parameter –subj eingetragen. Dabei sind die Regeln für Zertifikatnamen zu beachten (s. Zertifikatnamen)

Der RSA Schlüssel wird in der Datei key.pem, der Zertifikatrequest in der Datei request.pem abgelegt.

3 Erzeugen einer PKCS#12 Datei

Zur Erzeugung einer PKCS#12 Datei mit dem privaten und öffentlichen Schlüssel, dem zugehörigen Zertifikat und der CA-Kette müssen folgende Eingabedateien vorliegen:

1. Der RSA Schlüssel, den Sie für den Zertifikatantrag (CSR) erzeugt haben (in der Datei key.pem)
2. Das Zertifikat, das Sie von Ihrer Zertifizierungsstelle in der DFN-PKI erhalten haben (in der Datei certificate.pem)
3. Die CA-Zertifikatkette, die Sie sich von https://pki.pca.dfn.de/<Name Ihrer CA>/pub/cacert/chain.txt herunter laden können (in der Datei ca-chain.txt)

Um daraus eine PKCS#12 Datei mit dem Namen pkcs12-file.p12 zu erzeugen, können Sie folgendes OpenSSL-Kommando verwenden:

```
openssl pkcs12 -export -inkey key.pem -in certificate.pem -certfile ca-
chain.txt -out pkcs12-file.p12
```

4 Weitere OpenSSL Kommandos

4.1 Umwandeln einer passwortgeschützten Schlüssel-Datei in eine ungeschützte Datei

Eine Schlüsseldatei ohne Passwort kann nützlich sein, wenn Sie den Schlüssel z.B. direkt in den Anwendungen Apache Web-, OpenLDAP- oder FreeRadius- Server verwenden und diese Dienste beim Booten des Rechners bzw. Starten des Dienstes kein Passwort abfragen.

Mit folgender Befehlszeile können Sie aus einer mit Passwort geschützten Schlüsseldatei (key.pem) eine Schlüsseldatei ohne Passwortschutz (key-no-pw.pem) erzeugen:

```
openssl rsa -in key.pem -out key-no-pw.pem
```

Die ungeschützte Schlüsseldatei muss unbedingt mit anderen Mitteln (z.B. Dateizugriffsrechte) vor unbefugtem Zugriff geschützt werden.

4.2 RSA Schlüssel anzeigen

Mit folgender Befehlszeile zeigt Ihnen OpenSSL den Inhalt der Datei key.pem, die einen RSA Schlüssel (privater und öffentlicher Schlüssel) enthält, in lesbarer Form an:

```
openssl rsa -in key.pem -pubout –text
```

4.3 Zertifikatrequest anzeigen

Mit folgender Befehlszeile zeigt Ihnen OpenSSL den Inhalt der Datei request.pem, die einen Zertifikatrequest enthält, in lesbarer Form an:

```
openssl req -text -verify -in request.pem
```

4.4 Zertifikat anzeigen

Mit folgender Befehlszeile zeigt Ihnen OpenSSL den Inhalt der Datei certificate.pem, die ein Zertifikat enthält, in lesbarer Form an:

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

4.5 Zertifikat und Schlüssel anzeigen

Mit folgender Befehlszeile zeigt Ihnen OpenSSL den Inhalt der Datei cert-and-key.pfx (oder .p12), die ein Zertifikat und den privaten Schlüssel enthält (PKCS#12-Datei), in lesbarer Form an:

```
openssl pkcs12 -info -nokeys -in cert-and-key.pfx
```

Vor der Anzeige wird das Passwort der PKCS#12 Datei abgefragt.

4.6 Fingerprint eines Zertifikats anzeigen

Mit folgenden Befehlszeilen zeigt Ihnen OpenSSL den Fingerprint eines Zertifikats an, das unter dem Dateinamen certificate.pem abgelegt ist:

SHA256: `openssl x509 -noout -sha256 -fingerprint -in certificate.pem`

SHA1: `openssl x509 -noout -sha1 -fingerprint -in certificate.pem`

5 Zertifikatnamen

Bei der Erzeugung eines Zertifikatrequests mit einem OpenSSL – Kommando wird der Zertifikatname (Distinguished Name, DN) im Parameter –subj angegeben.

Der Zertifikatname darf keine Umlaute und andere Sonderzeichen enthalten. Erlaubt sind a-z, A-Z, 0-9, (,), :, ., -, Komma und Leerzeichen. Auf Groß- und Kleinschreibung ist zu achten.

Der Zertifikatname muss den erlaubten bzw. vorgegebenen Angaben entsprechen, die in der Zertifizierungsrichtlinie (CP/CPS) der Zertifizierungsstelle (CA) festgelegt sind. Die Richtlinien der CAs in der DFN-PKI finden Sie unter https://www.pki.dfn.de/policies/ und https://info.pca.dfn.de/

Attribute des Parameters –subj

- ST=<Bundesland> Der deutschsprachige voll ausgeschriebene Name des Bundeslandes, z.B. 'Schleswig-Holstein'.
- L=<ORT> Der deutschsprachige voll ausgeschriebene Name des Ortes, z.B. 'Luebeck'.
- O=<Einrichtung> Der Name der Einrichtung.
- OU=<Abteilung> Der Name der Abteilung.
  OU-Einträge sind meist optional. Es können auch mehrere OU-Einträge in direkter Folge angegeben werden, wenn die Organisationsstruktur z.B. detaillierter abgebildet werden soll.
- CN=<FQDN> Der voll qualifizierte DNS Hostname des Servers, die Domain muss dabei eine offizielle bei einem Domain-Registrator registrierte Domain aus Ihrem Umfeld sein. Fantasie-Domains oder lokale Domains sind nicht erlaubt. Der Hostname im CN muss nicht zwangsläufig im DNS auf eine IP Adresse auflösen.
- emailAddress=<MAIL-ADRESSE DES ADMINISTRATORS> Eine gültige E-Mail-Adresse des Server-Administators, o.ä.