Einen CSR-Requests für ein Serverzertifikat mit openSSL erstellen

Auf Linux-Systemen sollte openSSL grundsätzlich bereits installiert sein.

Für Windows-Systeme können Sie das Programmpaket hier herunterladen:
http://www.slproweb.com/products/Win32OpenSSL.html

Schlüsselpaar generieren

Im vorliegenden Beispiel wollen wir folgendes voraussetzen:

- Die Internet-Adresse des Servers sei server1.uni-freiburg.de
- Die Abteilung ist das Rechenzentum
- Der Schlüsselbund zur Aufbewahrung der eigenen Schlüssel sei /var/lib/.keystore
  (unter Windows üblicherweise C:\Dokumente und Einstellungen\(Benutzername)\.keystore)
- Der Schlüssel soll auf hohe Sicherheit (Länge 4096 Bit) eingestellt sein.

Es soll hier ein Schlüsselpaar inclusive einem selbstsignierten Zertifikat mit folgendem eindeutigen Name erzeugt werden:

```
DN: CN=server1.uni-freiburg.de,OU=Rechenzentrum,O=Albert-Ludwigs-
Universitaet Freiburg,L=Freiburg im Breisgau,ST=Baden-Wuerttemberg,C=DE
```

Für die Uni-FR CA sind die Komponenten O und C obligatorisch: **O=Albert-Ludwigs-Universitaet Freiburg,L=Freiburg im Breisgau,ST=Baden-Wuerttemberg,C=DE**.
Das Element OU kann auch mehrfach auftreten, falls Sie z.B. verschiedene Unterabteilungen im DN nicht nur durch den CN des Servers kenntlich machen wollen.

Das OpenSSL-Kommando zur Herstellung eines Schlüsselpaares lautet folgendermaßen:
(Die Benutzer-Eingabezeilen sind in den Beispielen mit dem System-Prompt '$' gekennzeichnet)

```
ohne Passwortschutz:
$ openssl genrsa  -out  /var/lib/.keystore 4096

mit Passwortschutz:
$ openssl genrsa  -des3 -out  /var/lib/.keystore 4096
enter des-ede3-cbc encryption password: *****
Verifying - enter des-ede3-cbc encryption password: *****
```

Beim Aufruf ohne Passwortschutz liegt Schlüsselpaar jetzt ungeschützt im Schlüsselbund vor.

Falls der künftige Server nicht automatisch mit Passwort auf die Schlüsseldatei zugreifen kann, sollten Sie die Version ohne Passwortschutz verwenden und mit anderen Methoden (Zugriffsrechte des Betriebssystems) dafür sorgen, dass nur berechtigte Personen oder Programme auf die Datei zugreifen können.

Zertifizierungsantrag generieren

Nun soll aus den Angaben im Schlüsselbund ein CSR erstellt werden.

Dazu legen Sie zunächst eine Konfigurationsdatei an, in die Sie die Parameter eintragen, die beim Request wichtig sind:

```
[ req ]
default_bits           = 4096
distinguished_name     = req_distinguished_name
prompt                 = no

[ req_distinguished_name ]
C                      = DE
ST                     = Baden-Wuerttemberg
L                      = Freiburg im Breisgau
O                      = Albert-Ludwigs-Universitaet Freiburg
OU                     = Rechenzentrum
CN                     = server1.uni-freiburg.de
```

Subject Alternative Names

Falls Sie zusätzliche Hostnamen (Subject Alterantive Names, SANs) als Alternativen in das Zertifikat aufgenommen haben wollen, müssen Sie die Konfigurationsdatei in der folgenden Art erweitern:

```
[ req ]
default_bits           = 4096
distinguished_name     = req_distinguished_name
prompt                 = no
req_extensions         = v3_req

[ req_distinguished_name ]
C                      = DE
ST                     = Baden-Wuerttemberg
L                      = Freiburg im Breisgau
O                      = Albert-Ludwigs-Universitaet Freiburg
OU                     = Rechenzentrum
CN                     = server1.uni-freiburg.de

[ v3_req ]
subjectAltName         = @alt_names

[ alt_names ]
DNS.1                  = server1.uni-freiburg.de
DNS.2                  = alt1.uni-freiburg.de
DNS.3                  = alt2.uni-freiburg.de
...
```

Bitte beachten Sie, dass der Common Name (CN) aus dem Abschnitt [req_distinguished_name] nochmals als SAN im Abschnitt [alt_names] aufgeführt wird, da anderenfalls einige Browser mit der Auswertung der entsprechenden Zertifikatsfelder Probleme haben.

Erzeugen, speichern, überprüfen

In unserem Beispiel soll diese Datei den Namen **req_config** erhalten.

Nun lautet das Kommando zum Erzeugen eines CSR:

```
$ openssl req -new -sha256 -key /var/lib/.keystore -out server1.csr -config
req_config
```

Damit erzeugen Sie die **Request-Datei** unter dem Namen **server1.csr** unter Verwendung der zuvor erzeugten Konfigurationsdatei **req_config**. Bei Windows XP funktioniert dieses Verfahren nicht!

Die Request-Datei geben Sie zur Kontrolle als lesbaren Text mit folgendem Kommando aus:

```
$ openssl req -text -in server1.csr
```

Internes