Serverzertifikat mit openSSL bearbeiten

Auf dieser Seite zeigen wir Ihnen, wie Sie mit openSSL ein Schlüsselpaar generieren, den CSR (Certificate Signing Request) erzeugen und schließlich das von der Uni-FR CA gelieferte Zertifikat in das Server- System einbauen.

Hintergrundinformationen dazu finden Sie in dem Dokument: **Serverzertifikat beantragen**.

Die Anleitung der DFN PKI: **Anleitung zur Nutzung von OpenSSL in der DFN-PKI (PDF-Datei)**

Die Anleitung des DFN CERT: **OpenSSL-Kurzreferenz**

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.
  - Bis 30. September 2019 kann auch ein neues Zertifikat mit 2084 Bit Schlüssellänge erstellt werden. Die Güligkeitsdauer des Zertifikats darf dann allerdings, nach BSI-Richtlinien, das Datum 31.12.2022 nicht überschreiten.
  - **Spätestens ab 30. September 2019 muss bei einem neuen Zertifikat eine Schlüssellänge von 4096 Bit verwendet werden.**
- Der Gültigkeitszeitraum wird auf die maximale von der Uni-FR CA akzeptierten Zeitdauer von 27 Monate konfiguriert

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
```

Die Datei server1.csr können Sie im Web-Interface der Uni-FR CA

Link zur DFN PKI zur Beantragung des Zertifikats

direkt über die Schaltfläche **Durchsuchen** hochladen lassen.

Das Antragsverfahren mit Hilfe des Web-Interface der Uni-FR CA ist beschrieben in dem Dokument

- **Serverzertifikat beantragen**

Zertifikat und privaten Schlüssel installieren

Sobald Sie das Serverzertifikat von der Uni-FR CA per Mail erhalten haben, speichern Sie die PEM-formatierte Datei des Attachements ab, z.B. unter dem Namen **server1.pem**.

Außerdem speichern Sie das **Zertifikat der Wurzelzertifizierungsstelle** der Deutschen Telekom, das **Zwischenzertifikat der DFN-PKI** sowie das **Zertifikat der Uni-FR CA** als PEM-Datei ab. Klicken Sie dazu mit der rechten Maustaste auf die Links und wählen Sie die Funktion „Ziel speichern unter…“. Die heruntergeladenen Dateien werden gespeichert unter den Namen **g_deutsche-telekom-root-ca-2.crt**, **g_dfn_intermediatecert.crt** bzw. **g_unifrcacert.crt**.

Sie finden eine aktuellere Version der Zertifikatskette hier

Die Installation des Zertifikates hängt von den Anforderungen des Dienstes ab, für den es beantragt wurde.

Im vorliegenden **Beispiel** soll gezeigt werden, wie das Zertifikat für **Jakarta Tomcat** zugänglich gemacht wird.

Als erstes importieren Sie die drei Zertifikate der Zertifikatskette in die keystore-Datei:

```
ohne Passwortschutz:
$ cat g_deutsche-telekom-root-ca-2.crt >>/var/lib/.keystore
$ cat g_dfn_intermediatecert.crt >>/var/lib/.keystore
$ cat g_unifrcacert.crt >>/var/lib/.keystore
$ cat server1.pem >>/var/lib/.keystore

mit Passwortschutz (bei bisher ungeschütztem .keystore):
$ cat g_deutsche-telekom-root-ca-2.crt >>/var/lib/.keystore
$ cat g_dfn_intermediatecert.crt >>/var/lib/.keystore
$ cat g_unifrcacert.crt >>/var/lib/.keystore
$ cat server1.pem >>/var/lib/.keystore
$ cat /var/lib/.keystore | openssl enc -e -des3 -out /var/lib/.keystore
enter des-ede3-cbc encryption password: *****
Verifying - enter des-ede3-cbc encryption password: *****

mit Passwortschutz (bei bisher geschütztem .keystore):
$ openssl enc -d -des3 -in /var/lib/.keystore -out tempstore
enter des-ede3-cbc decryption password: *****
$ cat g_deutsche-telekom-root-ca-2.crt >>/var/lib/.keystore
$ cat g_dfn_intermediatecert.crt >>/var/lib/.keystore
$ cat g_unifrcacert.crt >>/var/lib/.keystore
$ cat server1.pem >>tempstore
$ openssl enc -e -des3 -in tempstore -out /var/lib/.keystore
enter des-ede3-cbc encryption password: *****
Verifying - enter des-ede3-cbc encryption password: *****
$ rm tempstore
```

Die Zertifikate können in einem nicht korrektem Encoding vorliege. Dies führt im

Apachen zum AH02561-Fehler und einem nicht aufstarten, in diesen Fall sollten Sie die .crts umkodieren

```
openssl x509 -in g_deutsche-telekom-root-ca-2.crt -inform DER -out root.crt
```

Falls das Zertifikat unter Android nicht funktioniert, kann es an der Reihenfolge des cat-Befehles liegen. Versuchen Sie

```
cat server1.pem g_unifrcacert.crt g_dfn_intermediatecert.crt g_deutsche-telekom-root-ca-2.crt >full.crt
```

Nun teilen Sie **tomcat** in der Datei **server.xml** im **conf**-Verzeichnis mit, wo das Server-Zertifikat zu finden ist. Falls Sie den Zertifikatsspeicher mit einem Passwort geschützt haben, müssen Sie dieses hier im Klartext (!) eingeben.

```
...
    <!-- Define a SSL HTTP/1.1 Connector on port 443 -->
    <Connector port="443" maxHttpHeaderSize="8192"
               maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
               enableLookups="false" disableUploadTimeout="true"
               acceptCount="100" scheme="https" secure="true"
               keystoreFile="/var/lib/.keystore"
               keystorePassword="******"
               clientAuth="false" sslProtocol="TLS" />
...
```

Das keystore-Password wurde mit '*' unkenntlich gemacht.
In diesem Fall ist also die Schlüsseldatei mit einem Passwort geschützt.
Sie müssen aber nun dafür sorgen, dass die Datei server.xml mit dem Klartextpasswort nicht von unbefugten gelesen werden kann.
Falls Sie den **Apache Webserver** mit mod_ssl einsetzen, tragen Sie die Keystore-Datei wie sie im obigen Beispiel hergestellt wurde, nach folgendem Schema in die Konfigurationsdatei **httpd-ssl.conf** ein:

```
...
#   Server Certificate:
#   Point SSLCertificateFile at a PEM encoded certificate.  If
#   the certificate is encrypted, then you will be prompted for a
#   pass phrase.  Note that a kill -HUP will prompt again.  Keep
#   in mind that if you have both an RSA and a DSA certificate you
#   can configure both in parallel (to also allow the use of DSA
#   ciphers, etc.)
SSLCertificateFile "/var/lib/cert-from-ca.pem"

#   Server Private Key:
#   If the key is not combined with the certificate, use this
```

```
#   directive to point at the key file.  Keep in mind that if
#   you've both a RSA and a DSA private key you can configure
#   both in parallel (to also allow the use of DSA ciphers, etc.)
SSLCertificateKeyFile "/var/lib/.keystore"
...
```

Zertifikatskette bereitstellen

Wenn der Server die Zertifikatskette (SSLCertificateChainFile) nicht zum Client überträgt, ist man gezwungen, dort die Zwischenzertifikate manuell zu importieren. Es ist also zu empfehlen, seinen „Kunden“ diesen Service zu bieten.

In der SSL-Konfigurationsdatei von Apache findet man dazu den Parameter „SSLCertificateChainFile“.

Speichern Sie die Zertifikatskette, die aus drei aneinander gehängten PEM-Dateien besteht an einen Platz Ihrer Wahl und lassen Sie den Parameter darauf zeigen. Erhältlich ist das **ChainFile** auf den DFN-Webseiten der Uni FR CA.

```
#   Server Certificate Chain:
#   Point SSLCertificateChainFile at a file containing the
#   concatenation of PEM encoded CA certificates which form the
#   certificate chain for the server certificate. Alternatively
#   the referenced file can be the same as SSLCertificateFile
#   when the CA certificates are directly appended to the server
#   certificate for convinience.
SSLCertificateChainFile "/var/lib/chain.pem"
...
```

Normalerweise müssen Sie die Anwendungen anschließend neu starten, damit die Konfiguration wirksam wird und das Zertifikat verwendet werden kann.

Literatur

- http://www.dfn-cert.de/informationen/themen/verschluesselung_und_pki/openssl-kurzreferenz.html
- http://wiki.cacert.org/FAQ/subjectAltName
- http://apetec.com/support/GenerateSAN-CSR.htm
- http://wiki.gwdg.de/index.php/Erzeugung_von_Zertifikatantraegen_mit_Subject_Alternative_Name_fuer_virtuelle_Webserver_auf_Basis_von_OpenSSL_fuer_die_DFN-PKI
  (An die lokalen Gegebenheiten anpassen!)
- http://apetec.com/support/GenerateCSR.htm
  (Für IIS 7 Microsoft Windows Server 2008)

Zertifikate - Artikelübersicht, Sicherheit - Artikelübersicht, Artikel zum tag: openssl