CSR unter Ubuntu

Diese Seite befindet sich aktuell noch im Aufbau ⚠️

Wie Sie einen CSR (Certificate Signing Request, deutsch: Zertifikatsignierungsanforderung) unter Linux (hier: Ubuntu) erstellen, erfahren Sie in den 4 Schritten der Schritt-für-Schritt-Anleitung.

Um ein Nutzerzertifikat von einer CA (Certificate Authority) anzufordern, muss ein **"Certificate Signing Request"** (CSR) erstellt werden. Ein CSR ist ein digitaler Antrag, aus einem öffentlichen Schlüssel ein digitales Zertifikat zu erstellen. Voraussetzung hierfür ist wiederum ein privater Schlüssel.

Schritt-für-Schritt-Anleitung

Privaten Schlüssel erzeugen

Hinweis

Der **private Schlüssel** sollte immer auf eigens verwalteten Systemen erzeugt und gespeichert werden und **niemals Dritten** in die Hände **gegeben** werden - auch nicht der signierenden CA.

1. Öffnen Sie ein **„Terminal“**-Fenster. Im folgenden Beispiel soll der private Schlüssel als **/etc/ssl/private/private.pem** erzeugt werden. Da normale Benutzer auf diesen Ordner keinen Zugriff haben, sollten Sie sich zunächst Root-Zugriffsrechte mittels

Terminal

```
sudo -i
```

verschaffen.

2. Der **private Schlüssel** wird mit dem folgenden Kommando erzeugt:

Terminal

```
openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:4096 -out /etc/ssl/private/private.pem
```

Hinweis

Mit *rsa_keygen_bits:4096* wird der Verschlüsselungs-Algorithmus festgelegt.

Empfehlung:

- RSA-4096

Grund: Nutzerzertifikate mit den ECC-Schlüsseltypen 384 *und* 265 können *nur für Signatur und Authentisierung, aber nicht für Verschlüsselung verwendet werden.*

Alternative: Den privaten Schlüssel zusätzlich mit einem Passwort versehen

Das o.g. Kommando erzeugt einen privaten Schlüssel, der nicht durch ein Passwort geschützt ist.

Das ist notwendig, da z.B. Apache-Server sonst bei jedem Neustart die manuelle Eingabe des Passworts verlangen würde oder diese in der Konfiguration im Klartext hinterlegt werden müsste.

Wenn Sie Ihren privaten Schlüssel dennoch zusätzlich mit einem Passwort versehen möchten, geben Sie stattdessen folgendes Kommando und anschließend das zu vergebende Passwort ein:

Terminal

```
openssl genpkey -aes256 -algorithm RSA -pkeyopt rsa_keygen_bits:4096 -out /etc/ssl/private/private.pem
```

Das Passwort muss mindestens 4 Zeichen lang sein.

3. Beenden Sie den Root-Zugriff indem Sie

Terminal

```
exit
```

eingeben.

CSR erzeugen

Als nächstes kann nun der **CSR erzeugt** werden. Dieser beinhaltet keine sensiblen Daten und kann deshalb an einer beliebigen Stelle erstellt werden, wo man ihn später wieder findet. Im folgenden Beispiel wird der CSR im eigenen Heimatverzeichnis abgelegt.

Terminal

```
sudo openssl req -new -key /etc/ssl/private/private.pem -out ~/public.csr
```

Nach Bestätigen des Befehls mit der Eingabetaste werden Details zum Inhalt des Zertifikates abgefragt. **Füllen** Sie das **Formular** wie im Beispiel unten **aus**
.

Manche Einträge sind //optional und können mit der Eingabetaste übersprungen werden.

Denken Sie unbedingt daran, die richtige E-Mail-Adresse (***vorname.nachname@hhu.de***) anzugeben!!

Wenn Sie ein Serverzertifikat beantragen

`Common Name (e.g. server FQDN or YOUR name) []:` Hier muss der Domainname stehen, für welchen das Zertifikat eingesetzt werden soll (bspw. testseite.hhu.de).

Terminal

```
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:Nordrhein-Westfalen
Locality Name (eg, city) []:Duesseldorf
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Heinrich-Heine-Universitaet Duesseldorf
Organizational Unit Name (eg, section) []:ZIM
Common Name (e.g. server FQDN or YOUR name) []: //optional, z.B. testseite.hhu.de (der Domainname, wenn Sie ein
Serverzertifikat beantragen)
Email Address []:vorname.nachname@hhu.de //(Ihre Mailadresse)

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []: //optional
An optional company name []: //optional
```

Die Zertifikatsdatei ist **fertig** und besteht - wie auch der private Schlüssel - aus einer einfachen Text-Datei mit kryptischem Inhalt.

Beispiel:

public.csr

```
-----BEGIN CERTIFICATE REQUEST-----
MIIBhzCB6gIBADBFMQswCQYDVQQGEwJBVTETMBEGA1UECAwKU29tZS1TdGF0ZTEh
MB8GA1UECgwYSW50ZXJuZXQgV2lkZ2l0cyBQdHkgTHRkMIGbMBAGByqGSM49AgEG
BSuBBAAjA4GGAAQBpbiPH0XWzTXy7WP3QGxP5x4yTp5KqV9SQnV5qRkWZqp3fXIe
YnN39/MmUYxUNNG/ly970hHYxAeig1k6sFljaVUAlYPVMzMnWbs8GhNioXuA3GnV
5JtiizJ35ABZ5lNGOI1fm8h+DInMsrlGw+Eo21nqSfYV2m5cifMG4tvI/9PZoAWg
ADAKBggqhkjOPQQDAgOBiwAwgYcCQU2Pbmg6FKQdtgLUzspUZBKOu3ccxBSvCQlK
UDGkguCG9oQF61xSrUg+6z/qRyyiMVuQ/OkAgOHm5Z47lgyRARjBAkIA/VfcpPtR
0WvhsFvTrD8nvgblJGT+kk4jj42gf7n+q71OmtrNh9jTAuzz+fC1F+Taq56KX1Ku
2SKzOn2OSUJBAuY=
-----END CERTIFICATE REQUEST-----
```

Die erste und letzte Zeile sind Teil der Zertifikatsdatei und dürfen nicht entfernt werden.