

OpenSSL

- [Neue Zertifikate mit OpenSSL erstellen](#)
- [PFX Zertifikat in .crt und .key Datei exportieren](#)
- [PEM-Zertifikat in CRT-Zertifikat konvertieren](#)

Neue Zertifikate mit OpenSSL erstellen

Einleitung

In dieser Anleitung werden wir mithilfe von **OpenSSL TLS Zertifikate** erstellen, mit dem wir dann unseren **Web-Server** über **HTTPS** erreichbar machen können. **OpenSSL** ist für diesen Zweck eine passende Alternative, da diese keine Verbindung ins Internet benötigt, und **kostenlos** verwendbar ist.

Um diese Zertifikate zu erstellen, müssen wir sicherstellen, dass wir das Paket `openssl` installiert haben. Wenn dies nicht der Fall ist, können wir dieses eben nachinstallieren.

```
apt update && apt install openssl -y
```

Erstellen von PK und CSR

In diesem Abschnitt werden wir einen **PK (Private Key)** und einen **CSR (Certificate Signing Requests)** erstellen. Mit diesen Dateien können wir dann unseren **Web-Server verschlüsseln**.

```
openssl req -newkey rsa:2048 -nodes -keyout domain.key -out domain.csr
```

Wenn wir diesen Befehl abgesetzt haben, müssen wir die Abfragen, die Folgen beantworten. Der erste Parameter `-newkey rsa:2048` gibt an, dass der Schlüssel **2048 bit** lang sein soll, und mit dem **RSA Algorithmus** verschlüsselt werden soll.

Mit dem zweiten Parameter `-nodes` geben wir an, dass wir den **Schlüssel nicht** mit einem **Kennwort verschlüsseln**.

CSR aus einem vorhandenen PK erstellen

Der folgende Befehl kann verwendet werden, wenn wir ein **CSR (Certificate Signing Requests)** erstellen möchten, wir aber schon einen **Private Key** besitzen, bzw. erstellt haben.

```
openssl req -key domain.key -new -out domain.csr
```

Jetzt müssen wir beim ersten Parameter `-key` den Dateinamen unseres Schlüssels angeben. Sobald wir das eingetragen haben, können wir den Befehl abschicken und die **CSR Datei** wird erstellt.

Selbst signiertes SIL-Zertifikat erstellen

Jetzt wollen wir ein **TLS Zertifikat** erstellen, welches nicht von einer externen Zertifizierungsstelle stammt. Dies können wir uns von Gebrauch machen, wenn wir unseren Web-Server verschlüsseln möchten.

```
openssl req -newkey rsa:2048 -nodes -keyout domain.key -x509 -days 365 -out domain.crt
```

Der erste Parameter gibt wieder den **Verschlüsselungsalgorithmus** an. Der Parameter `-x509` gibt an, dass wir unser Zertifikat nicht von einer externen Zertifikatsstelle signieren lassen möchten. Und der folgende Parameter `-days` gibt an, wie lange unser Zertifikat gültig ist.

PFX Zertifikat in .crt und .key Datei exportieren

Einleitung

Um Zertifikate in bestimmte Software einspielen zu können, benötigen wir manchmal die **.crt** und die **.key** Datei. Dazu können wir uns das **OpenSSL-Tool** zu Hilfe nehmen.

Zertifikat exportieren

.crt Datei exportieren

Um die **.crt Datei** zu erhalten, müssen wir nur den folgenden Befehl absetzen. Dabei müssen wir im ersten Schritt den Datei-Namen der **.pfx-Datei** anpassen. Beim Absenden des Befehls geben wir nur noch das Kennwort ein, mit dem das Zertifikat verschlüsselt wurde.

```
openssl pkcs12 -in <PFX-Zertifikat> -clcerts -nokeys -out <CRT-Datei>
```

#Beispiel:

```
openssl pkcs12 -in cert.pfx -clcerts -nokeys -out cert.crt
```

.key Datei exportieren

Um jetzt die **.key Datei** zu erhalten, müssen wir diesmal 2 Befehle absetzen. Einmal um die **.key-Datei** zu erhalten, und im zweiten Schritt um die **.key-Datei** zu entschlüsseln.

```
openssl pkcs12 -in cert.pfx -nocerts -out cert-encrypted.key
```

Im Anschluss folgt jetzt die **Entschlüsselung** der **.key-Datei**.

```
openssl rsa -in cert-encrypted.key -out cert.key
```

Damit haben wir dann beide Dateien erfolgreich exportiert, um diese dann in bestimmten Anwendungen zu verwenden.

PEM-Zertifikat in CRT-Zertifikat konvertieren

In diesem Beitrag geht es kurz darum, wie wir ein **PEM-Zertifikat** in ein **CRT-Zertifikat** konvertieren können. Benötigt wird dafür eine Installation von **OpenSSL**.

```
openssl x509 -outform der -in "C:\<Pfad>\cert.pem" -out "C:\<Pfad>\cert.crt"
```