Zertifikate

TLS & Co.

- Lokale Zertifizierungsstelle (CA - certificate authority) erstellen
- Serverzertifikat mittels lokaler Zertifizierungsstelle (CA) signieren

Lokale Zertifizierungsstelle (CA - certificate authority) erstellen

Wenn TLS verschlüsselte Webseiten ausschließlich in einer lokalen Netzwerkumgebung (Intranet) betrieben werden, können mit Hilfe einer lokalen Zertifizierungsstelle eigene Zertifikate erstellt werden.

Die Zertifizierungsstelle dient zum Erstellen des Stamm-Zertifikats welches die Gültigkeit der untergeordneten, selbst erstellten Serverzertifikate bestätigt.

Das Root-Zertifikat der Zertifizierungsstelle wird im Zertifikatsspeicher unter „***Vertrauenswürdige Stammzertifizierungsstellen***“ gespeichert.

Alle mit dem privaten Schlüssel dieses Stamm-Zertifikats signierten Zertifikate werden dann von den Browsern als vertrauenswürdig angesehen.

In diesem Beitrag geht es ausschließlich um das Erstellen der lokalen Zertifizierungsstelle unter Linux, des privaten Schlüssels und des Stamm-Zertifikats.

Das Signieren eines Zertifikats, für einen Webserver, mit Hilfe dieser Zertifizierungsstelle wird hier beschrieben.

Erstellen eines privaten Schlüssels

Ein Verzeichnis erstellen wo der private Schlüssel und das Zertifikat ablegt wird:

```
mkdir CA
```

Die Berechtigungen des Verzeichnisses ändern:

```
chmod 700 CA/
```

In das Verzeichnis wechseln:

```
cd CA
```

Den privaten Schlüssel, geschützt mit einem sicheren Passwort, generieren:

```
openssl genrsa -des3 -out ca-testfirma.key 4096
```

> Jeder, der den Schlüssel besitzt kann damit Zertifikate erstellen!

Die Berechtigung es Schlüssels ändern:

```
chmod 500 ca-testfirma.key
```

> Den privaten Schlüssel benötigt man zum Signieren von selbst erstellten Zertifikaten. Es wird daher dringend empfohlen ein externes Backup des privaten Schlüssels aufzubewahren und dieses natürlich vor unbefugtem Zugriff zu schützen!

Erstellen des Stamm-Zertifikats

Als nächstes wird das Stamm-Zertifikat (Root Certificate) erstellt, dass die Gültigkeit aller untergeordneten Serverzertifikate über einen langen Zeitraum bestätigen soll.

Daher wird ein Ablaufdatum weit in der Zukunft gewählt, in diesem Fall 7300 Tage also 20 Jahre.

Zum Erstellen des Zertifikats wird das Passwort des privaten Schlüssels und folgende Informationen für das Zertifikat benötigt:

– County Code – Ländercode nach ISO-3166-1 ALPHA-2
– State or Province Name – Bundesland
– Locality Name – Ortsnamen
– Organization Name – Firmenname / Organisationsbezeichnung
– Organizational Unit Name – Abteilungsname
– Common Name – Zertifikatsname
– Email Address

Der Zertifikatsname (Common Name) ist der Name des Zertifikats welcher später auch im Zertifikatsspeicher angezeigt wird.

Bis auf das Pflichtfeld „Country Code“ könnten alle Felder leer gelassen werden, auf Grund der Überprüfbarkeit des Zertifikats wird jedoch dringend empfohlen sie auszufüllen.

```
openssl req -new -x509 -days 7300 -key ca-testfirma.key -out ca-testfirma.pem
```

> Das Stamm-Zertifikat bestätigt die Gültigkeit aller selbst erstellten Zertifikate. Es wird daher dringend empfohlen ein externes Backup des Stamm-Zertifikates aufzubewahren und dieses natürlich vor unbefugtem Zugriff zu schützen!

Installieren des Stamm-Zertifikats in den Zertifikatspeicher

Mozilla Firefox

Der Mozilla Firefox besitzt einen eigenen Zertifikatsspeicher. Das Stammzertifikat muss unter Einstellungen -> Datenschutz & Sicherheit -> Sicherheit -> Zertifikate anzeigen... -> Zertifizierungsstellen importiert werden.

Chrome, Edge, Opera, Vivaldi, etc...

Auf Chromium basierte Browser verwenden den Zertifikatsspeicher von Windows. Das Stammzertifikat kann über das Snap-In Zertifikate in der Microsoft Management Console (MMC) zu den Vertrauenswürdigen Stammzertifizierungsstellen hinzugefügt werden.

Serverzertifikat mittels lokaler Zertifizierungsstelle (CA) signieren

Im Beitrag Lokale Zertifizierungsstelle (certificate authority – CA) erstellen wurde auf einer Linux-Maschine eine lokale Zertifizierungsstelle erstellt, womit lokale Zertifikate signiert werden können.

Hier wird eine Zertifikatsanfrage (Certificate Signing Request – CSR) erstellt welche mittels Zertifizierungsstelle signiert wird um damit ein gültiges Zertifikat zu bekommen.

Im Beispiel wird ein Zertifikat für einen Website, welche unter https://server.domain.tld erreichbar ist, erstellt.
Für den Server muss der vollqualifizierte Namen (Fully Qualified Domain Name – FQDN) der eigenen Domain angeben werden.

Zertifikatsanfrage erstellen

Da Google Chrome seit der Version 58 das Feld Common Name aus dem Zertifikat ignoriert und nun die Einträge aus den Feld Alternativer Antragstellername (Subject Alternative Name – SAN) zur Überprüfung verwendet, die SAN in OpenSSL jedoch nicht in der Befehlszeile angeben werden können, muss bei der Zertifikatsanfrage und sie Signierung mit Konfigurationsdateien gearbeitet werden.

> Folgende Schritte erfolgen am Server welcher das Zertifikat benötigt.

Wechseln ins Verzeichnis /etc/ssl/private:

```
cd /etc/ssl/private
```

Erstellen eines privaten Schlüssels (private key):

```
openssl genrsa -out myserver.domain.tld.key 4096
```

Erstellen einer Datei namens ***createcsr.conf***:

```
touch createcsr.conf
```

Mit folgendem Inhalt:

```
nano createcsr.conf
```

Die Zeilen 6 bis 11 müssen an den eigenen Bedarf angepasst werden, die Zeilen 19 bis 22 enthalten alle für den Aufruf des Servers benötigten namen und IP-Adressen:

```
[req]
distinguished_name = req_distinguished_name
req_extensions = v3_req

[req_distinguished_name]
countryName = Country Name (z.B.: AT)
stateOrProvinceName = State or Province Name (z.B. Tirol)
localityName = Locality Name (z.B.: Innsbruck)
organizationName = Organization Name (z.B.: Firma XYZ)
organizationalUnitName = Organizational Unit Name (z.B: IT Administration)
commonName = Common Name (FQDN z.B.: myserver.domain.tld)

[v3_req]
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
subjectAltName = @alt_names

[alt_names]
DNS.1 = myserver.domain.tld
DNS.2 = myserver
IP.1 = 192.168.0.100
```

Mit dieser Konfigurationsdatei wird eine Zertifikatsanfrage erstellt:

```
openssl req -new -key /etc/ssl/private/myserver.domain.tld.key -out myserver.domain.tld.csr -
config createcsr.conf
```

Die Zertifikatsanfrage wird wie folgt überprüft:

```
openssl req -noout -text -in myserver.domain.tld.csr
```

Zertifikatsanfrage signieren

> Die Signierung der Zertifikatsanfrage erfolgt auf dem Server, auf dem die lokale Zertifizierungsstelle (CA) eingerichtet wurde.

Erstellen eines neuen Unterverzeichnisses im Verzeichnis CA mit dem Namen ***myserver***:

```
mkdir /CA/Myserver
```

Kopieren der Zertifikatsanfrage ***myserver.domain.tld.csr*** in das neu erstellte Verzeichnis.

Um die Zertifikatsanfrage mit den SAN-Einträgen signieren zu können, wird eine Konfigurationsdatei mit dem Namen ***signcsr.conf*** benötigt:

```
touch signcsr.conf
```

Wir öffnen diese mit unserem Editor und kopieren folgenden Inhalt hinein, wobei die Zeilen 5 und folgende wieder nach den Anforderungen angepasst werden müssen:

```
[SAN]
subjectAltName = @alt_names

[alt_names]
DNS.1 = myserver.domain.tld
DNS.2 = myserver
IP = 192.168.0.100
```

Wir können nun unsere Zertifikatsanfrage signieren, dafür benötigen wir jedoch das Passwort für den privaten Schlüssel der lokalen Zertifizierungsstelle:

```
openssl x509 -req -days 3650 -in myserver.domain.tld.csr -CA ../ca-testfirma.crt -CAkey ../ca-testfirma.key -set_serial 01 -out myserver.domain.tld.pem -extfile signcsr.conf -extensions SAN
```

> Firefox lässt kein Zertifikat zu welches die gleiche Seriennummer hat wie ein bereits bestehendes Zertifikat
> (Error code: sec_error_reused_issuer_and_serial). Für weitere Zertifikate bei -set_serial fortlaufende Nummern verwenden!

Das Zertifikat können wir wie folgt überprüfen:

```
openssl x509 -noout -text -in myserver.domain.tld.pem
```

Das Zertifikat myserver.domain.tld.crt kopieren wir nun ins Verzeichnis /etc/ssl/certs auf unseren Webserver und verwenden es dort für die Einrichtung der SSL-Webseite.