

# Vertrauenswürdige SSL Zertifikate im Internen Netz

[https://www.youtube.com/embed/bv47DR\\_A0hw](https://www.youtube.com/embed/bv47DR_A0hw)

## DNS Server erstellen

[piHole installieren](#)

## Linux als Zertifikats Server einrichten

```
openssl genrsa -des3 -out myCA.key 2048
```

## Root Zertifikat erstellen

```
openssl req -x509 -new -nodes -key myCA.key -sha256 -days 1825 -out myCA.pem
```

Hier nur bei Organization Name was eingeben, wenn überhaupt. z. B. milecloud

die .pem Datei wird dann auf die Rechner gebracht.

## Zertifikat auf dem mac hinzufügen

```
sudo security add-trusted-cert -d -r trustRoot -k "/Library/Keychains/System.keychain" myCA.pem
```

# Zertifikat auf einem Linux rechner Hinzufügen

```
sudo su

cp myCA.pem /usr/local/share/ca-certificates

update-ca-certificates
```

## Zertifikate für den Clients erstellen

Auf dem Zertrechner

```
openssl genrsa -out proxmox.lan.key 2048

openssl req -new -key proxmox.lan.key -out proxmox.lan.csr
```

## ext Datei erstellen

```
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
subjectAltName = @alt_names

[alt_names]
DNS.1 = proxmox.lan
```

## Gesamte Befehle von Apfelcast

```
##### Create your own Certificate Authority and Certificates #####

1. Create Certificate Authority

1.1 Create central certificate folder
```

```
mkdir ~/certs
```

```
cd ~/certs
```

### 1.2 generate private key for CA

```
openssl genrsa -des3 -out myCA.key 2048
```

### 1.3 create CA root certificate

```
openssl req -x509 -new -nodes -key myCA.key -sha256 -days 1825 -out myCA.pem
```

## 2. Create certificate signed by own CA

### 2.1 generate private key for certificate

```
openssl genrsa -out demo.lan.key 2048
```

### 2.2 create CSR

```
openssl req -new -key demo.lan.key -out demo.lan.csr
```

### 2.3 create an X509 V3 certificate extension config file

```
nano demo.lan.ext
```

```
authorityKeyIdentifier=keyid,issuer
```

```
basicConstraints=CA:FALSE
```

```
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
```

```
subjectAltName = @alt_names
```

```
[alt_names]
```

```
DNS.1 = demo.lan
```

### 2.4 create the certificate: using our CSR, the CA private key, the CA certificate, and the config file

```
openssl x509 -req -in demo.lan.csr -CA myCA.pem -CAkey myCA.key \  
-CAcreateserial -out demo.lan.crt -days 825 -sha256 -extfile demo.lan.ext
```

# .crt und .key werden auf dem Server benötigt, der den Dienst zur Verfügung stellt

## 3. Use Certificate with apache

```
a2enmod ssl
```

```
nano /etc/apache2/sites-available/demo.lan.conf
```

```
<VirtualHost *:443>
```

```
    ServerName demo.lan
```

```
    DocumentRoot /var/www/html
```

```
    SSLEngine on
```

```
    SSLCertificateFile /root/certs/demo.lan.crt
```

```
    SSLCertificateKeyFile /root/certs/demo.lan.key
```

```
</VirtualHost>
```

```
a2ensite demo.lan.conf
```

```
service apache2 restart
```

#### 4. Add CA to client

##### 4.1 Mac OS

```
sudo security add-trusted-cert -d -r trustRoot -k "/Library/Keychains/System.keychain" myCA.pem
```

##### 4.2 Linux

```
sudo cp myCA.pem /usr/local/share/ca-certificates/myCA.crt
```

```
sudo update-ca-certificates
```

## Zertifikat mit Skript erstellen

```
### Skript vorbereiten ###
```

```
nano /root/certs/auto-cert.sh
```

```
...
```

```
chmod +x /root/certs/auto-cert.sh
```

```
bash /root/certs/auto-cert.sh domain.lan
```

```
### Skript vorbereiten ###
```

```
##### Skript #####
```

```
#!/bin/sh
```

```
if [ "$#" -ne 1 ]
```

```
then
```

```
    echo "Usage: Must supply a domain"
```

```
    exit 1
```

```
fi
```

```
DOMAIN=$1
```

```
cd ~/certs
```

```
openssl genrsa -out $DOMAIN.key 2048
```

```
openssl req -new -key $DOMAIN.key -out $DOMAIN.csr
```

```
cat > $DOMAIN.ext << EOF
```

```
authorityKeyIdentifier=keyid,issuer
```

```
basicConstraints=CA:FALSE
```

```
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
```

```
subjectAltName = @alt_names
```

```
[alt_names]
```

```
DNS.1 = $DOMAIN
```

```
EOF
```

```
openssl x509 -req -in $DOMAIN.csr -CA myCA.pem -CAkey myCA.key -CAcreateserial \
```

```
-out $DOMAIN.crt -days 825 -sha256 -extfile $DOMAIN.ext
```

```
##### Skript #####
```

---

Revision #1

Created 21 March 2023 08:13:08 by Hermann

Updated 21 March 2023 08:13:32 by Hermann