SSL

Sei deine eigene SSL-Zertifizierungsstelle.

Dieses Tutorial basiert auf der Domain `nextcloud.home`. Ändere die Domain daher in deine eigene Domain, wenn du was anderes brauchst.

Es ist auch wichtig, dass die Domain-Adresse von deinem Router umgeleitet wird oder mit AdGuardHome. Dies kann auch in der Datei `/etc/hosts` auf deinem Computer eingestellt werden, aber um die Domain auf jedem Gerät zu erreichen, ist es einfacher, dies direkt im Router oder AdGuardHome zu ändern:

```
nextcloud.domain SERVER-IP
```

mkcert

mkcert ist ein einfaches Werkzeug zur Erstellung von lokal vertrauenswürdigen Entwicklungszertifikaten. Es erfordert keine Konfiguration.

Packete

```
pacman -S nss mkcert
```

Root-Zertifikat erstellen

```
mkcert -install
```

Zertifikate für Ihre Domains erstellen

```
mkcert nextcloud.home
```

Manuell

Generierung des privaten Schlüssels und des Root Zertifikats

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

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

Ändere die folgenden Informationen nach deinen Wünschen. Die Infos werden z.B. angezeigt, wenn

du das Zertifikat über deinen Browser ansiehst.

```
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
```

Erstellung von CA-signierten Zertifikaten für deine Domains

```
openssl genrsa -out nextcloud.home-key.pem 2048
```

```
openssl req -new -key nextcloud.home-key.pem -out nextcloud.home.pem
```

```
nano nextcloud.home.ext
```

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

[alt_names]
DNS.1 = nextcloud.home
```

Script

Erstelle die Datei in `nano /etc/nginx/ssl/ssl.sh`.

```
#!/bin/sh

if [ "$#" -ne 1 ]
then
  echo "Usage: Must supply a domain"
  exit 1
fi

DOMAIN=$1

openssl genrsa -out $DOMAIN-key.pem 2048
openssl req -new -key $DOMAIN-key.pem -out $DOMAIN.pem

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.pem -CA rootCA.pem -CAkey rootCA.key -
CAcreateserial \
-out $DOMAIN.crt -days 825 -sha256 -extfile $DOMAIN.ext
```

```
chmod +x ssl.sh
./ssl.sh nextcloud.home
```

Installation des Root-Zertifikats auf allen Geräten

Du musst auf jedem Gerät eine `rootCA.pem`-Datei erstellen und den Inhalt der Datei `rootCA.pem` dorthin kopieren, wo du sie in Abschnitt generierung_des_privaten_schluessels_und_des_root_zertifikats (manuell) erstellt hast.

Wenn du mkcert benutzt hast, führe einfach den Befehl `cat $(mkcert -CAROOT)/rootCA.pem` aus.

Arch Linux

```
sudo trust anchor --store rootCA.pem
```

Android

User trusted credentials

`Settings` - `Security` - `Encryption and credentials` - `Install a certificate`

Check unter:

`Settings` - `Security` - `Trusted credentials` - `User`

System trusted credentials

Wenn “User trusted credentials” nicht ausreicht und du das Zertifikat im System brauchst, befolge die nächsten Zeilen. Dazu ist allerdings ein gerootetes Gerät erforderlich:

```
hashed_name=`openssl x509 -inform PEM -subject_hash_old -in rootCA.pem |
head -1` && cp rootCA.pem $hashed_name.0
ls $hashed_name.0
```

```
adb root
adb shell mount -o rw,remount /
adb push $hashed_name.0 /system/etc/security/cacerts/
adb shell chmod 644 /system/etc/security/cacerts/$hashed_name.0
adb shell chown root:root /system/etc/security/cacerts/$hashed_name.0
adb shell reboot
```

Du kannst auch die Magisk-Module MagiskTrustUserCerts (Android 13) oder ConscryptTrustUserCerts (Android 14) verwenden, dass das gleiche wie oben macht.

CA-Zertifikate von Drittanbietern für Firefox verwenden

Du kannst auch die Option `Use third party CA certificates` für den Firefox-Browser wählen:

1. Öffne deinen Browser, scrolle nach unten und klicke auf Über firefox/iceraven/mull ...
2. Klick mehrmals auf das Logo und gehe zurück
3. Klicke auf Geheime Einstellungen und aktiviere `Use third party CA certificates`

Nginx

Siehe auch nginx

ssl-params.conf

```
nano /etc/nginx/conf.d/ssl-params.conf
```

```
ssl_protocols TLSv1.3;
ssl_prefer_server_ciphers on;
ssl_ciphers "EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH";
ssl_ecdh_curve secp384r1;
ssl_session_cache shared:SSL:10m;
```

example

```
server {
        listen 80;
        listen [::]:80;
        server_name nextcloud.home;

        # enforce https
        return 301 https://$server_name:443$request_uri;
}

server {
        listen 443 ssl http2;
```

```
        listen [::]:443 ssl http2;
        server_name nextcloud.home;

        ssl_certificate /etc/nginx/ssl/nextcloud.home.pem;
        ssl_certificate_key /etc/nginx/ssl/nextcloud.home-key.pem;
        include conf.d/ssl-params.conf;
        access_log /var/log/nginx/nextcloud.home_access_log;
        error_log /var/log/nginx/nextcloud.home-error_log;

        location / {
                your things;
        }
}
```