Create Certificates

> All subsequent entries, commands, etc. are required in the terminal of the PI OS!
>
> Depending on which instructions you have already read on the Internet, different directories are always suggested or used for storing the certificates. I would suggest that you create a directory that you always use to create and edit your own certificates and specify as the storage location in the browser 'VirtualHost' blocks.
>
> For our example we use:
>
> TLD = lan
> SLD = home
> subdomain = nas

1. Create a trusted root certification authority

1.1 Create a private key for the root certification authority

command: "openssl genrsa -des3 -out <SLD, ohne TLD>.key 2048"

- e.g. your domain name = zuhause.lan, then only "zuhause"!
- you will then be asked for a "pass phrase" **8-tung**! please assign a password that you will not forget, otherwise massive problems may arise later, as you will need this password from time to time, depending on where and how you want to use your certificate!
- > one more tip:
  > - The longer the password, the better the encryption. Although I wouldn't overdo it either, as everything only takes place in your home network.
  > - By the way: if you realize in the course of your work that you want to use a different TLD or SLD, you will have to start all over again. Certificates cannot be changed!

1.2 Create the root certificate

- command: "openssl req -x509 -new -nodes -key <SLD>.key -sha256 -days 3650 -out >SLD>.pem"

- The following additional information is now requested:
- > Country Name (2 letter code): <>
  > State or Province Name (full name): <>
  > Locality Name (eg, city): <>
  > Organization Name (eg, company): <>
  > Organizational Unit Name (eg, section): <>
  > Common Name (e.g. server FQDN or YOUR name): <SLD.TLD>
  > Email Address: <>

<> = you can fill in, but you don't have to!

2. Create certificates based on the root certificate

> You must repeat this procedure for each subdomain (mostly devices, e.g. NAS)!

2.1 Create a private key for the certificate

command: "openssl genrsa -out <subdomain.SLD.TLD>.key 2048"

- our example = nas.zuhause.lan

2.2 Create a CSR (**C**ode **S**igning **R**equest)

command: "openssl req -new -key <subdomain.SLD.TLD>.key -out <subdomain.SLD.TLD>.csr"

- our example = nas.zuhause.lan

2.3 Create an X509 V3 certificate extension configuration file

command: "nano <subdomain.SLD.TLD>.ext"

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

[alt_names]
DNS.1 = <subdomain.SLD.TLD>
```

- our example = nas.zuhause.lan

2.4 create the certificate

(with our CSR, the private key of the CA, the CA certificate and the configuration file)

command: "openssl x509 -req -in <subdomain.SLD.TLD>.csr -CA <SLD>.pem -CAkey <SLD>.key -CAcreateserial -out <subdomain.SLD.TLD>.crt -days 3650 -sha256 -extfile <subdomain.SLD.TLD>.ext"

- our example = SLD: zuhause - subdomain.sld.TLD: nas.zuhause.lan

> hey, you "perseverers...", you actually made it... well, yes - almost! □□ The root certificate still has to be distributed so that the browsers can successfully validate your certificate and the web server still needs the information about which certificate it should use, but then, I promise, "you'll get along with your neighbor" □□

Revision #3
Created 18 June 2025 08:55:22 by tomek
Updated 18 June 2025 09:47:41 by tomek