# Création de certificats signés par une autorité de certification pour vos sites de développement

Nous sommes désormais une autorité de certification sur tous nos appareils et nous pouvons signer des certificats pour tous les nouveaux sites de développement qui nécessitent HTTPS. Tout d'abord, nous créons une clé privée pour le site de développement. Notez que nous nommons la clé privée en utilisant l'URL du nom de domaine du site de développement. Ce n'est pas obligatoire, mais cela facilite la gestion si vous avez plusieurs sites :

```
openssl genrsa -out nl.test.key 2048
```

Ensuite, nous créons un CSR :

```
openssl req -new -key nl.test.key -out nl.test.csr
```

> Vous recevrez les mêmes questions que précédemment et, encore une fois, vos réponses n'ont pas d'importance. En fait, elles en ont encore **moins**, car vous ne verrez pas ce certificat dans une liste à côté d'autres.

Enfin, nous allons créer un fichier de configuration d'extension de certificat X509 V3 , qui est utilisé pour définir le nom alternatif du sujet (SAN) du certificat. Dans notre cas, nous allons créer un fichier de configuration appelé `nl.test.ext` contenant le texte suivant :

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

```
[alt_names]
DNS.1 = nl.test
```

Nous allons l'exécuter `openssl x509` car la commande x509 nous permet de modifier les paramètres de confiance du certificat. Dans ce cas, nous l'utilisons pour signer le certificat en conjonction avec le fichier de configuration, ce qui nous permet de définir le nom alternatif du sujet.

Nous exécutons maintenant la commande pour créer le certificat : en utilisant notre CSR, la clé privée de l'autorité de certification, le certificat de l'autorité de certification et le fichier de configuration :

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

Nous disposons désormais de trois fichiers : `nl.test.key` (la clé privée), `nl.test.csr` (la demande de signature de certificat, ou fichier csr) et `nl.test.crt` (le certificat signé). Nous pouvons configurer les serveurs Web locaux pour utiliser HTTPS avec la clé privée et le certificat signé.

Revision #3
Created 24 September 2024 22:10:32 by Admin

Updated 24 September 2024 22:38:08 by Admin