

# Créer un certificat signé par une CA avec openSSH

## Présentation

Avec OpenSSH vous pouvez créer une autorité de certification et signer les certificats.

Pour en savoir plus : <https://sysadmin.cyklodev.com/creer-une-autorite-de-certification-ca-avec-openssl/>

## Démarche

La création d'un certificat auto-signé pour le CA

- Création de la **clé privée** ;
- Création de la **requête de certification de la clé publique** ;
- Création du **certificat** qui va contenir la clé publique et les informations d'identification de l'utilisateur ou du serveur

Dans ce document est présenté la création d'un certificat signé pour un utilisateur

## Création de la clé privée de l'autorité de certification (CA)

Dans l'invite de commandes lancez la commande suivante :

```
$ openssl genrsa -aes256 -out ca-key.pem 2048
```

Indiquez une pass-phrase pour votre nouvelle clé, pass-phrase qui sera demandé lors de la création du certificat.

## Création de la requête de demande de signature du certificat de la CA

Pour générer la requête de demande de signature, il faut utiliser le fichier contenant la clé privée.

Lors de la création de la requête vous avez à saisir vos informations d'identification du CA. Il est important d'indiquer un **Common Name** unique.

- Saisissez la commande suivante :

```
$ openssl req -new -key ca-key.pem -out ca.csr
```

## Génération du certificat auto-signé du CA

Lors de la génération d'un certificat autosigné, vous avez à préciser le format du certificat (x509) et la durée de validation. En général un certificat d'une autorité de certification a une durée plutôt longue (ici 365000 jours)

```
$ openssl x509 -req -days 365000 -in ca.csr -signkey ca-key.pem -out ca.crt
```

Le certificat généré est le fichier **.crt** et la clé privée associée, le fichier **.pem**.

## Configuration du CA

Le fichier de configuration du CA permet notamment de définir :

- le fichier serial qui incrémentera le nombre de certificats signés ;
- le fichier index.txt qui va garder la trace de chaque certificat émis.

Cela ne sera pas abordé dans ce document.

# Signature d'un certificat

Deux cas de figure sont possibles :

- la CA gère à la fois les clés privées et publiques, génère le certificat, le signe et le transmet à l'utilisateur.
- l'utilisateur génère sa bi-clé privée/publique, crée sa demande de certificat (fichier csr) puis demande au CA de signer son certificat. Le CA vérifie alors la validée de la demande (les informations d'identités), signe le certificat et le transmet à l'utilisateur.

La démarche de création d'un certificat signé est la suivante :

- Création de la bi-clé privée et publique (KEY) ;
- Création de la demande de signature (CSR) auprès du CA ;
- Signature du certificat (CRT)

## Création de la clé privée de l'utilisateur ou du serveur

Dans l'invite de commandes lancez la commande suivante :

```
$ openssl genrsa -aes256 -out user-private.pem 2048
```

Indiquez une pass-phrase pour votre nouvelle clé, pass-phrase qui sera demandé lors de la création du certificat.

## Création de la requête de demande de signature du certificat

Pour générer la requête de demande de signature, il faut utiliser le fichier contenant la clé privée.

Lors de la création de la requête vous avez à saisir vos informations d'identification du CA. Il est important d'indiquer un **Common Name** unique.

- Saisissez la commande suivante :

```
$ openssl req -new -key user-private.pem -out user.csr
```

## Génération du certificat avec signature par le CA

Lors de la génération du certificat signé par le CA, vous avez à préciser le format du certificat (x509) et la durée de validation (365 jours).

```
$ openssl x509 -req -days 3650 -CAkey ca-key.pem -in user.csr -signkey ca-key.pem -out user.crt
```

Le certificat généré est le fichier **.crt** et la clé privée associée, le fichier **.pem**.

Pour vérifier que le certificat de l'utilisateur est correct,; utiliser la commande suivante :

```
$ openssl verify -CAfile ca.crt user.pem
```

```
serverMariaDB-cert.pem: OK
```

From:

/ - **Les cours du BTS SIO**

Permanent link:

</doku.php/reseau/certificat/certificatca?rev=1638125126>

Last update: **2021/11/28 19:45**

