

Créer sa CA et signer ses certificats

Création de la CA

Pour cela, on va se rendre dans un dossier isolé :

```
# mkdir /root/ca && cd /root/ca
```

puis on va créer l'organisation de dossier suivante:

```
# mkdir certs crl newcerts private csr
```

```
# chmod 700 private
```

Et on créer les fichiers de contrôle :

```
# touch index.txt
```

```
# echo 1000 > serial
```

```
# echo 0000 > crlnumber
```

puis on créer le fichier :

[openssl.cnf](#)

```
[ ca ]
# `man ca`
default_ca = CA_default

[ CA_default ]
# Directory and file locations.
dir                = .
certs              = $dir/certs
crl_dir            = $dir/crl
new_certs_dir     = $dir/newcerts
database          = $dir/index.txt
serial            = $dir/serial
RANDFILE          = $dir/private/.rand

# The root key and root certificate.
private_key       = $dir/private/ca.key
certificate       = $dir/certs/ca.pem

# For certificate revocation lists.
crlnumber        = $dir/crlnumber
crl              = $dir/crl/ca.crl
crl_extensions   = crl_ext
```

```
default_crl_days = 30

# SHA-1 is deprecated, so use SHA-2 instead.
default_md = sha256

name_opt = ca_default
cert_opt = ca_default
default_days = 375
preserve = no
policy = policy_loose

[ req ]
# Options for the `req` tool (`man req`).
default_bits = 4096
distinguished_name = req_distinguished_name
string_mask = utf8only

# SHA-1 is deprecated, so use SHA-2 instead.
default_md = sha256

# Extension to add when the -x509 option is used.
x509_extensions = v3_ca

[ req_distinguished_name ]
countryName = Country Name (2 letter code)
stateOrProvinceName = State or Province Name
localityName = Locality Name
#.organizationName = Organization Name
organizationalUnitName = Organizational Unit Name
commonName = Common Name
emailAddress = Email Address

countryName_default = FR
stateOrProvinceName_default =
localityName_default =
#.organizationName_default = VirtIT
#organizationalUnitName_default =
emailAddress_default =

[ v3_ca ]
# Extensions for a typical CA (`man x509v3_config`).
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical, CA:true
keyUsage = critical, digitalSignature, cRLSign, keyCertSign

[ policy_loose ]
# Allow the intermediate CA to sign a more diverse range of
certificates.
```

```
# See the POLICY FORMAT section of the `ca` man page.
countryName          = optional
stateOrProvinceName = optional
localityName         = optional
organizationName     = optional
organizationalUnitName = optional
commonName           = supplied
emailAddress          = optional

[ server_cert ]
# Extensions for server certificates (`man x509v3_config`).
basicConstraints = CA:FALSE
nsCertType = server
nsComment = "OpenSSL Generated Server Certificate"
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer:always
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth

[ usr_cert ]
# Extensions for client certificates (`man x509v3_config`).
basicConstraints = CA:FALSE
nsCertType = client, email
nsComment = "OpenSSL Generated Client Certificate"
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer
keyUsage = critical, nonRepudiation, digitalSignature, keyEncipherment
extendedKeyUsage = clientAuth, emailProtection

[ crl_ext ]
authorityKeyIdentifier = keyid:always
issuerAltName = issuer:copy
```

On génère ensuite la clé privé du CA root :

```
# openssl genrsa -out private/ca.key 4096
```

et puis le certificat :

```
# openssl req -config openssl.cnf \  
-key private/ca.key \  
-new -x509 -days 7300 -extensions v3_ca \  
-out certs/ca.pem
```

Création d'un certificat

Il faut d'abord créer la clé privée du certificat :

```
# openssl genrsa -out private/server.key 4096
```

puis le CSR :

```
# openssl req -config openssl.cnf -key private/server.key -new -out  
csr/server.csr
```

Et puis on le signe :

```
# openssl ca -config openssl.cnf -extensions server_cert -days 375 -notext -  
in csr/server.csr -out certs/server.pem
```

Révoquer un certificat

On va donc révoquer le certificat :

```
# openssl ca -config openssl.cnf -revoke certs/server.pem
```

puis pour garder un peu de clarté, on va le déplacer dans le dossier crl :

```
# mv certs/server.pem crl/
```

Et puis on génère le fichier CRL:

```
# openssl ca -config openssl.cnf -gencrl -out crl/ca-crl.pem
```

From:

<https://wiki.virtit.fr/> - VirtIT

Permanent link:

https://wiki.virtit.fr/doku.php/kb:crypto:creer_ca

Last update: **2021/03/30 14:01**

