



OpenSSL Certificate Authority

Cycle : Ingénieur 1

Auteur : KANHO Khorem

Année Académique : 2022 -2023





Table des matières

1. Introduction	5
2. AC (Autorité de Certification) racine	5
2.1. Préparer le repertoire	5
2.2. Créer la clé racine	6
2.3 Créer le certificate racine	7
2.4 Vérifier le certificate racine	8
3. AC intermédiaire	9
3.1. Préparer le répertoire	9
3.2. Créer la clé intermédiaire	10
3.3 Créer le certificat intermédiaire	11
3.4 Vérifier le certificat intermédiaire	13
4. Chaîne de certification	14
5. Signature de certificats serveur et client	15
5.1. Créer une clé	15
5.2. Créer un certificat	16
5.3 Vérifier le certificat	17
6. CRL (Certificate revocation lists)	18
6.1. Créer la liste de revocation de certificats	19
6.2. Révoquer un certificat	19
7. OCSP (Online Certificate Status Protocol)	21
7.1. Créer la paire OCSP	21
7.1.1 Créer un certificate de serveur à tester	21
7.1.2	21
7.1.3	22
7.1.4	23
7.2. Révoquer un certificate	24
7.2.2	25
7.2.3	26
7.2.4	26



1. Introduction

1.1. OpenSSL est une bibliothèque cryptographique gratuite et open-source qui fournit plusieurs Outils de ligne de commande pour la gestion des certificats numériques. Certains de ces outils peuvent être utilisé pour agir en tant qu'autorité de certification.

1.2.

1.3. Une autorité de certification (CA) est une entité qui signe des certificats numériques. Beaucoup Les sites Web doivent informer leurs clients que la connexion est sécurisée, afin qu'ils payer une autorité de certification de confiance internationale (par exemple, VeriSign, DigiCert) pour signer un certificat pour leur domaine.

1.4.

Dans certains cas, il peut être plus logique d'agir comme votre propre CA, plutôt que de payer un CA comme DigiCert. Les cas courants incluent la sécurisation d'un site Web intranet, ou pour l'émission de certificats aux clients pour leur permettre de s'authentifier auprès d'un serveur (par exemple, Apache, OpenVPN).

2. AC (Autorité de Certification) racine

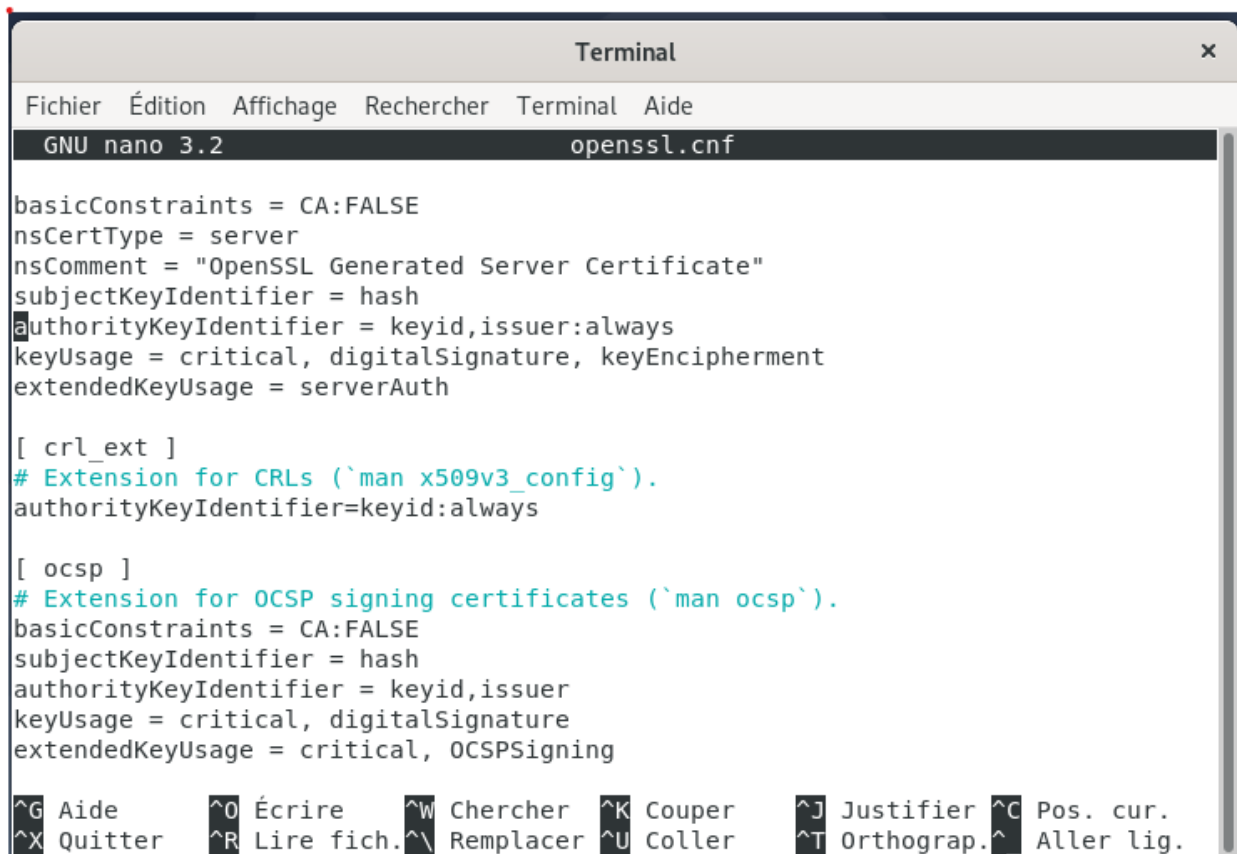
2.1. Préparer le repertoire

Choisissez un répertoire () pour stocker toutes les clés et tous les certificats./root/ca

Créez la structure du répertoire. Le index.txt et serial les fichiers agissent comme une base de données de fichiers plats pour garder une trace des certificats signés.

Vous devez créer un fichier de configuration pour que OpenSSL puisse l'utiliser. Copier la racine CA fichier de configuration du Annexe à /root/ca/openssl.cnf.

La section est obligatoire. Ici, nous demandons à OpenSSL d'utiliser les options de la section. [ca] [CA_default]



```
basicConstraints = CA:FALSE
nsCertType = server
nsComment = "OpenSSL Generated Server Certificate"
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer:always
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth

[ crl_ext ]
# Extension for CRLs (`man x509v3_config`).
authorityKeyIdentifier=keyid:always

[ ocsp ]
# Extension for OCSP signing certificates (`man ocsp`).
basicConstraints = CA:FALSE
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer
keyUsage = critical, digitalSignature
extendedKeyUsage = critical, OCSPSigning
```

2.2. Créer la clé racine

Créez la clé racine (ca.key.pem) et gardez-le absolument sécurisé. N'importe qui dans la possession de la clé racine peut émettre des certificats de confiance. Crypter la clé racine avec cryptage AES 256 bits et un mot de passe fort.

```
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
(base) root@debian:~# mkdir /root/ca
(base) root@debian:~# cd /root/ca
(base) root@debian:~/ca# mkdir certs crt newcerts private
(base) root@debian:~/ca# chmod 700 private
(base) root@debian:~/ca# touch index.txt
(base) root@debian:~/ca# echo 1000 > serial
(base) root@debian:~/ca# touch openssl.cnf
(base) root@debian:~/ca# nano openssl.cnf
(base) root@debian:~/ca# nano openssl.cnf
(base) root@debian:~/ca# cd /root/ca
(base) root@debian:~/ca# openssl genrsa -aes256 -out private/ca.key.pem 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
.....++++
.....
.....
.....
.....
.....++++
e is 65537 (0x010001)
Enter pass phrase for private/ca.key.pem:
Verifying - Enter pass phrase for private/ca.key.pem:
(base) root@debian:~/ca# █
```

2.3 Créer le certificat racine

Utilisez la clé racine (ca.key.pem) pour créer un certificat racine (ca.cert.pem). Donnez au certificat racine une longue date d'expiration, telle que vingt ans. Une fois le le certificat racine expire, tous les certificats signés par l'AC deviennent invalides.

```
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
Verifying - Enter pass phrase for private/ca.key.pem:
(base) root@debian:~/ca# chmod 400 private/ca.key.pem
(base) root@debian:~/ca# cd /root/ca
(base) root@debian:~/ca# openssl req -config openssl.cnf \
> -key private/ca.key.pem \
> -new -x509 -days 7300 -sha256 -extensions v3_ca \
> -out certs/ca.cert.pem
Enter pass phrase for private/ca.key.pem:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:sn
State or Province Name [England]:Dakar
Locality Name []:Sacre Coeur
Organization Name [Alice Ltd]:ECPI
Organizational Unit Name []:Cycle Ingenieur
Common Name []:Khorem KANHO
Email Address []:khoremkanho@gmail.com
(base) root@debian:~/ca# chmod 444 certs/ca.cert.pem
(base) root@debian:~/ca#
```

2.4 Vérifier le certificate racine

La sortie montre:

le utiliséSignature Algorithm

les dates du certificat Validity

le Public-Key longueur du bit

le Issuer, qui est l'entité qui a signé le certificat

le Subject, qui fait référence au certificat lui-même

Le Issuer et Subject sont identiques car le certificat est auto-signé. Notez que tous les certificats racine sont auto-signés.


```
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
Email Address []:khoremkanho@gmail.com
(base) root@debian:~/ca# chmod 444 certs/ca.cert.pem
(base) root@debian:~/ca# openssl x509 -noout -text -in certs/ca.cert.pem
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            4f:ad:8e:b2:f6:2c:9a:13:71:e4:36:c9:22:0c:64:4c:73:f1:5c:8a
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = sn, ST = Dakar, L = Sacre Coeur, O = ECPI, OU = Cycle Ingeni
eur, CN = Khorem KANHO, emailAddress = khoremkanho@gmail.com
        Validity
            Not Before: Jun  5 14:56:45 2023 GMT
            Not After : May 31 14:56:45 2043 GMT
        Subject: C = sn, ST = Dakar, L = Sacre Coeur, O = ECPI, OU = Cycle Ingen
ieur, CN = Khorem KANHO, emailAddress = khoremkanho@gmail.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public-Key: (4096 bit)
            Modulus:
                00:a3:cf:c9:c9:6c:1e:25:ea:77:72:78:1c:94:68:
                52:ae:72:a2:53:54:fb:5e:00:9b:0f:e9:a0:2a:ea:
                eb:6d:51:09:0b:a0:7f:af:5a:26:bf:9d:11:0e:30:
                07:8e:f3:f4:62:5a:de:5e:c7:8a:73:7c:3b:88:6d:
```

3. AC intermédiaire

Une autorité de certification intermédiaire (CA) est une entité qui peut signer certificats au nom de l'AC racine. La racine CA signe l'intermédiaire certificat, formant une chaîne de confiance.

Le but de l'utilisation d'une autorité de certification intermédiaire est principalement de sécurité. La clé racine peut être maintenu hors ligne et utilisé aussi rarement que possible. Si l'intermédiaire la clé est compromise, la racine CA peut révoquer le certificat intermédiaire et créer une nouvelle paire cryptographique intermédiaire.

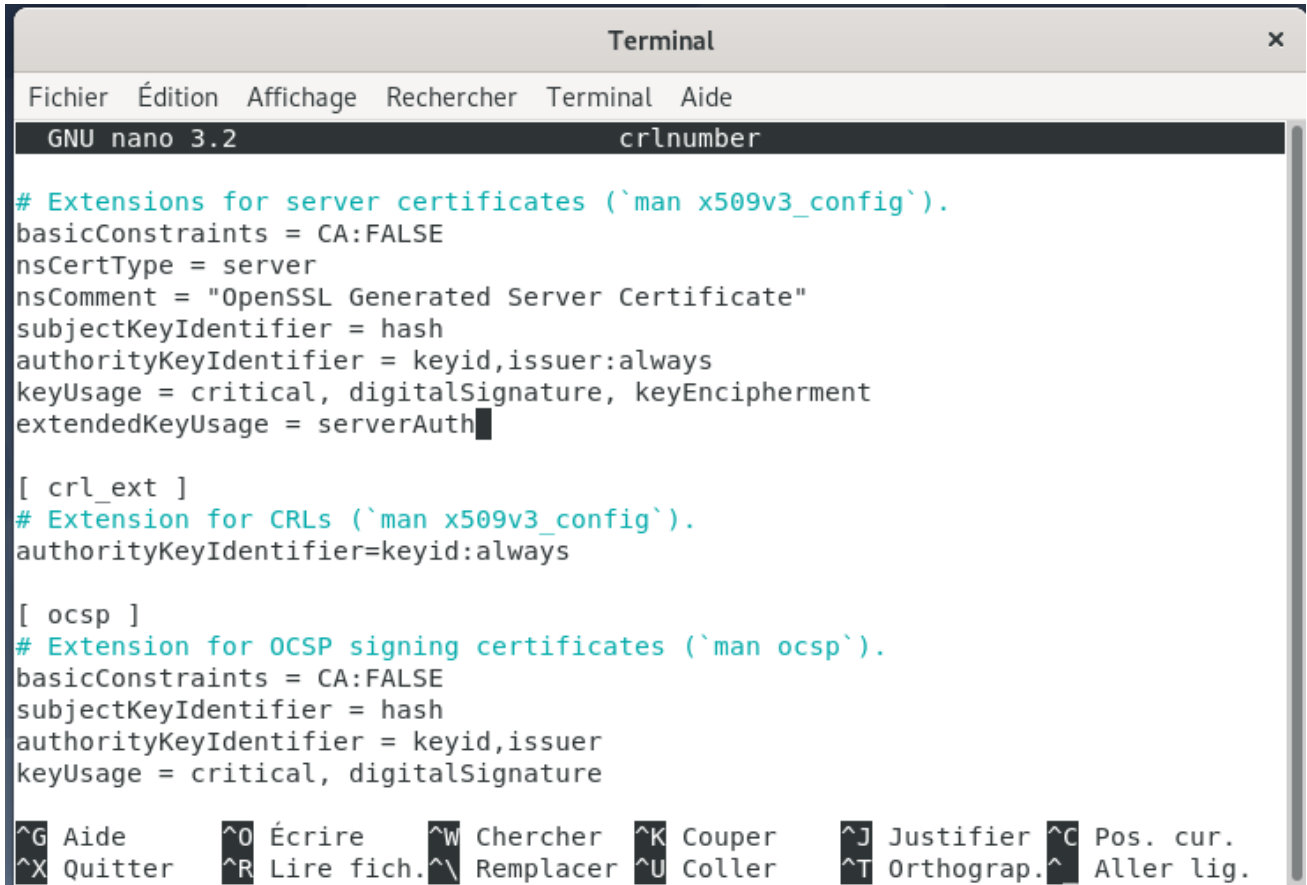
3.1. Préparer le répertoire

Les fichiers CA racine sont conservés dans /root/ca. Choisissez un répertoire différent (/root/ca/intermediate) pour stocker les fichiers CA intermédiaires.

Créez la même structure de répertoire utilisée pour les fichiers CA racine. C'est pratique créer également un csr répertoire pour contenir les demandes de signature de certificat.

Ajouter un crlnumber fichier dans l'arborescence de répertoires CA intermédiaire. crlnumber est utilisé pour garder une trace de listes de révocation de certificats.

Copiez le fichier de configuration CA intermédiaire à partir du Annexe à /root/ca/intermediate/openssl.cnf. Cinq options ont été modifiées par rapport à dans le fichier de configuration de racine CA:



```
# 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

[ crl_ext ]
# Extension for CRLs (`man x509v3_config`).
authorityKeyIdentifier=keyid:always

[ ocsp ]
# Extension for OCSP signing certificates (`man ocsp`).
basicConstraints = CA:FALSE
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer
keyUsage = critical, digitalSignature
```

3.2. Créer la clé intermédiaire

Créez la clé intermédiaire (intermediate.key.pem). Crypter l'intermédiaire clé avec cryptage AES 256 bits et un mot de passe fort.

```
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
85:3b:90:10:24:12:9e:1a:4a:c3:48:1f:1b:f3:9f:c5:6d:1a:
72:97:98:89:42:8b:df:3b:b7:f4:71:10:9a:b6:5b:6f:e1:d5:
9a:98:cf:03:4e:21:e3:7d
(base) root@debian:~/ca# mkdir /root/ca/intermediate
(base) root@debian:~/ca# cd /root/ca/intermediate
(base) root@debian:~/ca/intermediate# mkdir certs crl csr newcerts private
(base) root@debian:~/ca/intermediate# chmod 700 private
(base) root@debian:~/ca/intermediate# touch index.txt
(base) root@debian:~/ca/intermediate# echo 1000 > serial
(base) root@debian:~/ca/intermediate# echo 1000 > /root/ca/intermediate/crlnumbe
r
(base) root@debian:~/ca/intermediate# nano crlnumber
(base) root@debian:~/ca/intermediate# cd /root/ca
(base) root@debian:~/ca# openssl genrsa -aes256 \
> -out intermediate/private/intermediate.key.pem 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
.....++++
.....++++
e is 65537 (0x010001)
Enter pass phrase for intermediate/private/intermediate.key.pem:
Verifying - Enter pass phrase for intermediate/private/intermediate.key.pem:
(base) root@debian:~/ca# chmod 400 intermediate/private/intermediate.key.pem
(base) root@debian:~/ca#
```

3.3 Créer le certificat intermédiaire

Utilisez la clé intermédiaire pour créer une demande de signature de certificat (CSR). Les détails doivent généralement correspondre à la racine CA. Le Commun Nom, cependant, doit être différent

Pour créer un certificat intermédiaire, utilisez la racine CA avec le `v3_intermediate_ca` extension pour signer le CSR intermédiaire. L'intermédiaire le certificat doit être valable pour une période plus courte que le certificat racine. Dix des années seraient raisonnables.

Le `index.txt` fichier est l'endroit où l'OpenSSL ca l'outil stocke le certificat base de données. Ne supprimez ni ne modifiez ce fichier à la main. Il devrait maintenant contenir une ligne qui fait référence au certificat intermédiaire.

```
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
Enter pass phrase for intermediate/private/intermediate.key.pem:
Verifying - Enter pass phrase for intermediate/private/intermediate.key.pem:
(base) root@debian:~/ca# chmod 400 intermediate/private/intermediate.key.pem
(base) root@debian:~/ca# cd /root/ca
(base) root@debian:~/ca# openssl req -config intermediate/openssl.cnf -new -sha2
56 \
> -key intermediate/private/intermediate.key.pem \
> -out intermediate/csr/intermediate.csr.pem
Enter pass phrase for intermediate/private/intermediate.key.pem:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:sn
State or Province Name [England]:Dakar
Locality Name []:Sacre Coeur
Organization Name [Alice Ltd]:ECPI
Organizational Unit Name []:Cycle Ingenieur
Common Name []:Khorem KANHO
Email Address []:khoremkanho@gmail.com
(base) root@debian:~/ca#
```

```
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
stateOrProvinceName      = Dakar
organizationName          = ECPI
organizationalUnitName    = Cycle Ingenieur
commonName                = Khorem KANHO
emailAddress              = khoremkanho@gmail.com
X509v3 extensions:
X509v3 Subject Key Identifier:
    16:98:ED:90:C2:92:59:7B:1B:DD:FB:81:1B:B0:1C:7C:62:B8:69:CF
X509v3 Authority Key Identifier:
    keyid:14:21:31:7F:72:3A:E3:C4:14:1A:C0:58:CB:28:5E:E1:5E:0D:41:F
C
X509v3 Basic Constraints: critical
    CA:TRUE, pathlen:0
X509v3 Key Usage: critical
    Digital Signature, Certificate Sign, CRL Sign
Certificate is to be certified until Jun  2 15:39:33 2033 GMT (3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
(base) root@debian:~/ca#
```

3.4 Vérifier le certificat intermédiaire

Comme nous l'avons fait pour le certificat racine, vérifiez que les détails de l'intermédiaire le certificat est correct.

Vérifiez le certificat intermédiaire par rapport au certificat racine. Un OK indique que la chaîne de confiance est intacte.

```
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
58:cd:65:45:9d:8a:fb:aa:19:19:6c:eb:25:1a:0f:3b:89:13:
a7:da:c8:ec:11:d8:cd:3e:2c:86:36:33:8a:f6:f3:91:96:d8:
51:3d:1a:f8:2e:22:4b:29:bc:c6:ab:de:0c:1f:a2:fb:ae:1c:
fd:21:03:de:61:07:22:31:88:6f:5e:97:f6:6a:60:7a:f8:38:
74:95:8f:fb:81:fa:6a:30:f1:4b:a4:7a:9e:2b:f2:53:d3:6c:
24:d0:84:7f:6b:95:32:d0:ad:ee:e4:7c:39:d5:c3:7d:cc:58:
db:51:a6:ff:94:f8:a5:3f:4a:96:83:c1:bf:d9:30:9c:56:b4:
d4:12:4f:f2:83:99:b1:12:a5:0c:1e:da:cc:cb:ba:49:a5:34:
d1:78:5c:65:17:22:fb:2b:95:ad:88:28:23:d9:d2:4d:9b:8f:
08:d1:56:0f:82:e2:12:d7:d1:70:3a:68:5b:dc:93:1d:74:1d:
43:55:77:cb:ee:87:c7:ff:f4:20:0b:7a:6c:76:e8:be:1e:72:
db:32:8a:d4:6e:31:4f:c1:d7:bb:b2:82:3c:d3:46:14:a0:0d:
02:c8:91:ec:b1:9d:6e:8a:7b:14:93:30:ef:f7:ca:44:20:30:
74:c3:d8:6c:4e:60:b8:06:ef:b3:83:a9:a3:97:d6:f7:cd:e7:
03:02:cc:e8:2e:10:43:dd:5b:51:df:61:f6:87:31:75:6b:7c:
f3:b9:fa:60:ab:cd:2f:fe:d5:ba:f2:c7:f6:49:e9:1d:b7:7b:
19:42:45:65:f9:c4:64:0f:cc:b1:7d:98:1a:94:76:af:1c:01:
bb:ef:8c:df:9a:74:24:58:bf:e4:5d:2c:e2:25:be:f2:06:53:
92:69:d6:57:fe:a5:0d:e2:dc:67:c1:d5:af:b2:7e:8d:df:61:
70:25:03:a3:5a:55:9d:cc
(base) root@debian:~/ca# openssl verify -CAfile certs/ca.cert.pem \
> intermediate/certs/intermediate.cert.pem
intermediate/certs/intermediate.cert.pem: OK
(base) root@debian:~/ca#
```

4. Chaîne de certification

Lorsqu'une application (par exemple, un navigateur Web) essaie de vérifier un certificat signé par l'AC intermédiaire, il doit également vérifier le certificat intermédiaire par rapport à le certificat racine. Pour compléter la chaîne de confiance, créez un certificat CA chaîne à présenter à l'application.

Pour créer la chaîne de certificats CA, concaténer l'intermédiaire et la racine certificats ensemble. Nous utiliserons ce fichier plus tard pour vérifier les certificats signés par l'AC intermédiaire.


```
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
fd:21:03:de:61:07:22:31:88:6f:5e:97:f6:6a:60:7a:f8:38:
74:95:8f:fb:81:fa:6a:30:f1:4b:a4:7a:9e:2b:f2:53:d3:6c:
24:d0:84:7f:6b:95:32:d0:ad:ee:e4:7c:39:d5:c3:7d:cc:58:
db:51:a6:ff:94:f8:a5:3f:4a:96:83:c1:bf:d9:30:9c:56:b4:
d4:12:4f:f2:83:99:b1:12:a5:0c:1e:da:cc:cb:ba:49:a5:34:
d1:78:5c:65:17:22:fb:2b:95:ad:88:28:23:d9:d2:4d:9b:8f:
08:d1:56:0f:82:e2:12:d7:d1:70:3a:68:5b:dc:93:1d:74:1d:
43:55:77:cb:ee:87:c7:ff:f4:20:0b:7a:6c:76:e8:be:1e:72:
db:32:8a:d4:6e:31:4f:c1:d7:bb:b2:82:3c:d3:46:14:a0:0d:
02:c8:91:ec:b1:9d:6e:8a:7b:14:93:30:ef:f7:ca:44:20:30:
74:c3:d8:6c:4e:60:b8:06:ef:b3:83:a9:a3:97:d6:f7:cd:e7:
03:02:cc:e8:2e:10:43:dd:5b:51:df:61:f6:87:31:75:6b:7c:
f3:b9:fa:60:ab:cd:2f:fe:d5:ba:f2:c7:f6:49:e9:1d:b7:7b:
19:42:45:65:f9:c4:64:0f:cc:b1:7d:98:1a:94:76:af:1c:01:
bb:ef:8c:df:9a:74:24:58:bf:e4:5d:2c:e2:25:be:f2:06:53:
92:69:d6:57:fe:a5:0d:e2:dc:67:c1:d5:af:b2:7e:8d:df:61:
70:25:03:a3:5a:55:9d:cc
(base) root@debian:~/ca# openssl verify -CAfile certs/ca.cert.pem \
> intermediate/certs/intermediate.cert.pem
intermediate/certs/intermediate.cert.pem: OK
(base) root@debian:~/ca# cat intermediate/certs/intermediate.cert.pem \
> certs/ca.cert.pem > intermediate/certs/ca-chain.cert.pem
(base) root@debian:~/ca# chmod 444 intermediate/certs/ca-chain.cert.pem
(base) root@debian:~/ca#
```

5. Signature de certificats serveur et client

Nous signerons des certificats en utilisant notre CA intermédiaire. Vous pouvez les utiliser certificats signés dans diverses situations, telles que la sécurisation des connexions un serveur Web ou pour authentifier les clients se connectant à un service.

5.1. Créer une clé

Nos paires racine et intermédiaire sont de 4096 bits. Certificats de serveur et de client expirent normalement après un an, nous pouvons donc utiliser en toute sécurité 2048 bits à la place.

Si vous créez une paire cryptographique à utiliser avec un serveur Web (par exemple, Apache), vous devrez saisir ce mot de passe chaque fois que vous redémarrez le Web serveur. Vous voudrez peut-être omettre le -aes256 option pour créer une clé sans mot de passe.

```
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
74:c3:d8:6c:4e:60:b8:06:ef:b3:83:a9:a3:97:d6:f7:cd:e7:
03:02:cc:e8:2e:10:43:dd:5b:51:df:61:f6:87:31:75:6b:7c:
f3:b9:fa:60:ab:cd:2f:fe:d5:ba:f2:c7:f6:49:e9:1d:b7:7b:
19:42:45:65:f9:c4:64:0f:cc:b1:7d:98:1a:94:76:af:1c:01:
bb:ef:8c:df:9a:74:24:58:bf:e4:5d:2c:e2:25:be:f2:06:53:
92:69:d6:57:fe:a5:0d:e2:dc:67:c1:d5:af:b2:7e:8d:df:61:
70:25:03:a3:5a:55:9d:cc
(base) root@debian:~/ca# openssl verify -CAfile certs/ca.cert.pem \
> intermediate/certs/intermediate.cert.pem
intermediate/certs/intermediate.cert.pem: OK
(base) root@debian:~/ca# cat intermediate/certs/intermediate.cert.pem \
> certs/ca.cert.pem > intermediate/certs/ca-chain.cert.pem
(base) root@debian:~/ca# chmod 444 intermediate/certs/ca-chain.cert.pem
(base) root@debian:~/ca# cd /root/ca
(base) root@debian:~/ca# openssl genrsa -aes256 \
> -out intermediate/private/www.example.com.key.pem 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
Enter pass phrase for intermediate/private/www.example.com.key.pem:
Verifying - Enter pass phrase for intermediate/private/www.example.com.key.pem:
(base) root@debian:~/ca# chmod 400 intermediate/private/www.example.com.key.pem
(base) root@debian:~/ca#
```

5.2. Créer un certificat

Utilisez la clé privée pour créer une demande de signature de certificat (CSR). La RSE les détails n'ont pas besoin de correspondre à l'AC intermédiaire. Pour les certificats de serveur, le Nom commun doit être un nom de domaine pleinement qualifié (, par exemple, `www.example.com`), alors que pour les certificats clients, il peut s'agir de tout identifiant unique (, par exemple, un e-mail adresse). Notez que le Nom commun ne peut pas être le même que votre racine ou certificat intermédiaire.

Pour créer un certificat, utilisez l'AC intermédiaire pour signer le CSR. Si le le certificat va être utilisé sur un serveur, utilisez le `server_cert` extension. Si le certificat doit être utilisé pour l'authentification de l'utilisateur, utilisez le `usr_cert` extension. Les certificats ont généralement une validité d'un an, bien qu'un CA donne généralement quelques jours supplémentaires pour plus de commodité.


```
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Aide

SSL Server
Netscape Comment:
OpenSSL Generated Server Certificate
X509v3 Subject Key Identifier:
F9:01:E2:B6:C1:30:17:12:8B:D9:60:8F:24:4B:F9:83:FD:79:FC:F0
X509v3 Authority Key Identifier:
keyid:16:98:ED:90:C2:92:59:7B:1B:DD:FB:81:1B:B0:1C:7C:62:B8:69:C
F
DirName:/C=sn/ST=Dakar/L=Sacre Coeur/O=ECPI/OU=Cycle Ingenieur/C
N=Khorem KANH0/emailAddress=khoremkanho@gmail.com
serial:10:00

X509v3 Key Usage: critical
Digital Signature, Key Encipherment
X509v3 Extended Key Usage:
TLS Web Server Authentication
Certificate is to be certified until Jun 14 16:00:48 2024 GMT (375 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
(base) root@debian:~/ca#
```

5.3 Vérifier le certificate

Le Émetteur est le CA intermédiaire. Le Objet fait référence au certificat lui-même.

La sortie affichera également la Extensions X509v3. Lors de la création du certificat, vous avez utilisé soit le server_cert ou usr_cert extension. Le les options de la section de configuration correspondante seront reflétées dans le sortie.

Utilisez le fichier de chaîne de certificats CA que nous avons créé plus tôt (ca-chain.cert.pem) à vérifier que le nouveau certificat a une chaîne de confiance valide.

```
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
X509v3 Extended Key Usage:
    TLS Web Server Authentication
Certificate is to be certified until Jun 14 16:00:48 2024 GMT (375 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
(base) root@debian:~/ca# chmod 444 intermediate/certs/www.example.com.cert.pem
(base) root@debian:~/ca# openssl x509 -noout -text \
> -in intermediate/certs/www.example.com.cert.pem
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 4096 (0x1000)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = sn, ST = Dakar, O = ECPI, OU = Cycle Ingenieur, CN = Khorem
KANHO, emailAddress = khoremkanho@gmail.com
        Validity
            Not Before: Jun  5 16:00:48 2023 GMT
            Not After : Jun 14 16:00:48 2024 GMT
        Subject: C = sn, ST = Dakar, L = Sacre Coeur, O = ECPI, OU = Cycle Ingen
ieur, CN = Khorem KANHO, emailAddress = khoremkanho@gmail.com
```

6. CRL (Certificate revocation lists)

Une liste de révocation de certificats (CRL) fournit une liste de certificats qui ont a été révoqué. Une application client, telle qu'un navigateur Web, peut utiliser un CRL pour vérifier l'authenticité d'un serveur. Une application serveur, telle qu'Apache ou OpenVPN, peut utiliser un CRL pour refuser l'accès aux clients qui ne sont plus fiables.

Publiez le CRL dans un emplacement accessible au public (par exemple, <http://example.com/intermediate.crl.pem>). Les tiers peuvent aller chercher le CRL à partir de cet emplacement pour vérifier si des certificats sur lesquels ils s'appuient ont été révoqué.

6.1. Créer la liste de revocation de certificats

Lorsqu'une autorité de certification signe un certificat, elle encodera normalement le Emplacement CRL dans le certificat. Ajouter crlDistributionPoints au sections appropriées. Dans notre cas, ajoutez-le au section.

```
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
48:07:80:a5:00:91:c8:5b:19:1e:60:36:72:d2:3d:7b:cb:3a:
42:6b:2f:bb:10:d6:51:ad:31:2b:07:bc:db:28:b4:ce:a9:c3:
54:49:65:96:2f:eb:bc:5d:28:e7:7c:a3:f7:d1:b8:03:e3:6f:
dc:e7:37:a3:ba:97:5b:28
(base) root@debian:~/ca# openssl verify -CAfile intermediate/certs/ca-chain.cert
.pem \
> intermediate/certs/www.example.com.cert.pem
intermediate/certs/www.example.com.cert.pem: OK
(base) root@debian:~/ca# cd /root/ca
(base) root@debian:~/ca# openssl ca -config intermediate/openssl.cnf \
> -gencrl -out intermediate/crl/intermediate.crl.pem
Using configuration from intermediate/openssl.cnf
Enter pass phrase for /root/ca/intermediate/private/intermediate.key.pem:
(base) root@debian:~/ca# openssl crl -in intermediate/crl/intermediate.crl.pem -
noout -text
Certificate Revocation List (CRL):
  Version 2 (0x1)
  Signature Algorithm: sha256WithRSAEncryption
  Issuer: C = sn, ST = Dakar, O = ECPI, OU = Cycle Ingenieur, CN = Khorem
KANHO, emailAddress = khoremkanho@gmail.com
  Last Update: Jun  5 16:08:35 2023 GMT
  Next Update: Jul  5 16:08:35 2023 GMT
  CRL extensions:
    X509v3 Authority Key Identifier:
```

6.2. Révoquer un certificate

Passons à travers un exemple. Alice exécute le serveur Web Apache et possède un dossier privé d'images de chaton mignonnes. Alice veut lui accorder ami, Bob, accès à cette collection.

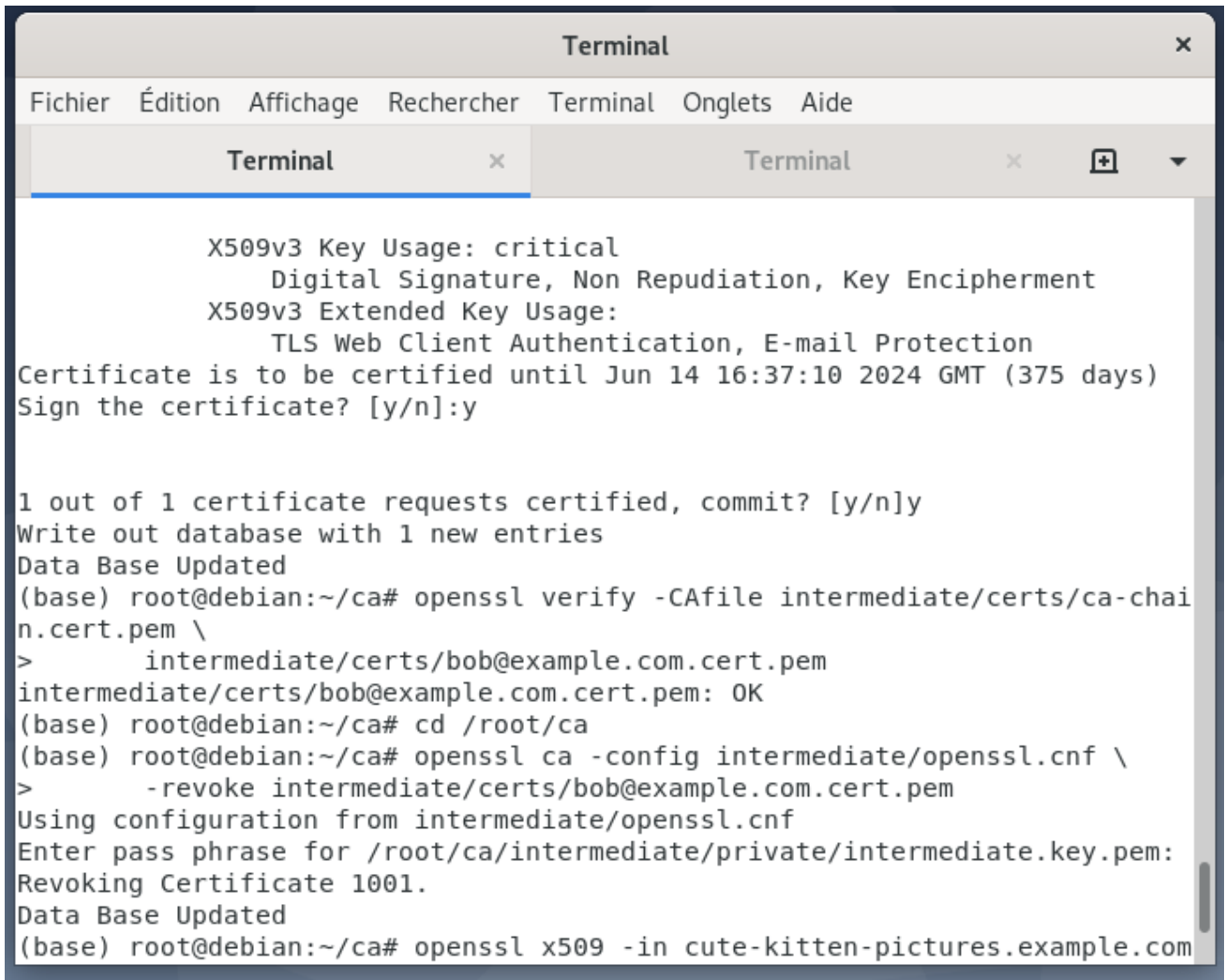
Bob crée une clé privée et une demande de signature de certificat (CSR).

Bob envoie sa RSE à Alice, qui la signe ensuite.

Alice vérifie que le certificat est valide:

Alice envoie à Bob le certificat signé. Bob installe le certificat dans sa toile navigateur et est maintenant capable d'accéder aux photos du chaton d'Alice. Vive!

Malheureusement, il s'avère que Bob se conduit mal. Bob a posté le chaton d'Alice des photos de Hacker News, affirmant qu'elles sont les siennes et gagnant énormément de popularité. Alice le découvre et doit révoquer immédiatement son accès.



```
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Onglets  Aide

X509v3 Key Usage: critical
    Digital Signature, Non Repudiation, Key Encipherment
X509v3 Extended Key Usage:
    TLS Web Client Authentication, E-mail Protection
Certificate is to be certified until Jun 14 16:37:10 2024 GMT (375 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
(base) root@debian:~/ca# openssl verify -CAfile intermediate/certs/ca-chain.cert.pem \
>      intermediate/certs/bob@example.com.cert.pem
intermediate/certs/bob@example.com.cert.pem: OK
(base) root@debian:~/ca# cd /root/ca
(base) root@debian:~/ca# openssl ca -config intermediate/openssl.cnf \
>      -revoke intermediate/certs/bob@example.com.cert.pem
Using configuration from intermediate/openssl.cnf
Enter pass phrase for /root/ca/intermediate/private/intermediate.key.pem:
Revoking Certificate 1001.
Data Base Updated
(base) root@debian:~/ca# openssl x509 -in cute-kitten-pictures.example.com
```

6.3 Utilisation côté client du CRL

Pour les certificats de serveur, il s'agit généralement d'une application côté client (, par exemple, un Web navigateur) qui effectue la vérification. Cette application doit avoir une télécommande accès au CRL.

Si un certificat a été signé avec une extension qui comprend `crldistributionPoints`, une application côté client peut lire ces informations et récupérer le CRL à partir de l'emplacement spécifié.



Les points de distribution CRL sont visibles dans le certificat X509v3 détails.

7. OCSP (Online Certificate Status Protocol)

7.1. Créer la paire OCSP

L'outil OpenSSL peut agir en tant que répondeur OCSP, mais il n'est destiné qu'à pour les tests. Il existe des intervenants OCSP prêts pour la production, mais ceux-ci vont au-delà de la Portée du présent guide.

7.1.1 Créer un certificat de serveur à tester

Le répondeur OCSP a besoin d'une paire cryptographique pour signer la réponse qui il envoie à la partie requérante. La paire cryptographique OCSP doit être signée par la même autorité de certification qui a signé le certificat en cours de vérification.

Créez une clé privée et chiffrez-la avec le chiffrement AES-256.

```
(base) root@debian:~/ca# cd /root/ca
(base) root@debian:~/ca# openssl genrsa -aes256 \
> -out intermediate/private/ocsp.example.com.key.pem 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
Enter pass phrase for intermediate/private/ocsp.example.com.key.pem:
Verifying - Enter pass phrase for intermediate/private/ocsp.example.com.ke
y.pem:
(base) root@debian:~/ca#
```

7.1.2

Créez une demande de signature de certificat (CSR). Les détails doivent généralement correspondre celles de l'autorité de certification signataire. Le nom usuel, cependant, doit être un nom de domaine.



```
(base) root@debian:~/ca# openssl req -config intermediate/openssl.cnf -new
-sha256 -key intermediate/private/ocsp.example.com.key.pem -o
ut intermediate/csr/ocsp.example.com.csr.pem
Enter pass phrase for intermediate/private/ocsp.example.com.key.pem:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN
.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:sn
State or Province Name [England]:Dakar
Locality Name []:Sacre Coeur
Organization Name [Alice Ltd]:ECPI
Organizational Unit Name []:Cycle Ingenieur
Common Name []:Khorem KANHO
Email Address []:khoremkanho@gmail.com
(base) root@debian:~/ca# █
```

7.1.3

Signez la CSR avec l'autorité de certification intermédiaire.

```
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Onglets  Aide
Terminal x Terminal x + ▾
commonName = Khorem KANHO
emailAddress = khorem.kanho@ecpi.edu.sn
X509v3 extensions:
X509v3 Basic Constraints:
CA:FALSE
X509v3 Subject Key Identifier:
93:2D:31:00:7E:1E:8C:C8:30:0A:1A:02:26:2B:47:41:16:D8:8E:6
C
X509v3 Authority Key Identifier:
keyid:16:98:ED:90:C2:92:59:7B:1B:DD:FB:81:1B:B0:1C:7C:62:B
8:69:CF
X509v3 Key Usage: critical
Digital Signature
X509v3 Extended Key Usage: critical
OCSP Signing
Certificate is to be certified until Jun 14 17:16:11 2024 GMT (375 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
(base) root@debian:~/ca#
```

7.1.4

Vérifiez que le certificat possède les extensions X509v3 correctes.


```
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Onglets  Aide
Terminal  x  Terminal  x  +  v
(base) root@debian:~/ca# openssl x509 -noout -text \
> -in intermediate/certs/ocsp.example.com.cert.pem
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 4099 (0x1003)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = sn, ST = Dakar, O = ECPI, OU = Cycle Ingenieur, CN = Khorem KANHO, emailAddress = khoremkanho@gmail.com
        Validity
            Not Before: Jun  5 17:16:11 2023 GMT
            Not After : Jun 14 17:16:11 2024 GMT
        Subject: C = sn, ST = Dakar, L = Sacre Coeur, O = ECPI, OU = Cycle Ingenieur, CN = Khorem KANHO, emailAddress = khorem.kanho@ecpi.edu.sn
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public-Key: (4096 bit)
            Modulus:
                00:ba:1e:b1:2d:60:c4:67:9d:41:a9:4f:19:9c:18:
                0d:0e:34:08:cb:a3:bd:b4:c8:c6:58:46:70:af:55:
                2c:6e:22:68:d9:d7:70:ec:2b:5e:e4:d1:93:dc:e9:
                ae:61:8a:28:f6:cc:24:28:a6:a6:bf:3a:7b:b0:61:
                37:a2:0d:7f:5a:56:ac:81:ec:69:0b:ac:d1:78:70:
                29:1c:02:8d:a4:77:11:56:77:e0:ae:00:3b:81:41:
```

7.2. Révoquer un certificate

L'outil OpenSSL peut agir en tant que répondeur OCSP, mais il n'est destiné qu'à pour les tests. Il existe des intervenants OCSP prêts pour la production, mais ceux-ci vont au-delà de la Portée du présent guide.ocsp

Créez un certificat de serveur à tester.


```

Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Onglets  Aide

Terminal x Terminal x + ▾

Netscape Comment:
  OpenSSL Generated Server Certificate
X509v3 Subject Key Identifier:
  9E:CD:DE:53:7F:BC:C2:63:0F:3D:E2:D7:D7:42:B9:9A:75:9F:E6:2
D
X509v3 Authority Key Identifier:
  keyid:16:98:ED:90:C2:92:59:7B:1B:DD:FB:81:1B:B0:1C:7C:62:B
8:69:CF
  DirName:/C=sn/ST=Dakar/L=Sacre Coeur/O=ECPI/OU=Cycle Ingen
ieur/CN=Khorem KANHO/emailAddress=khoremkanho@gmail.com
  serial:10:00

X509v3 Key Usage: critical
  Digital Signature, Key Encipherment
X509v3 Extended Key Usage:
  TLS Web Server Authentication
Certificate is to be certified until Jun 14 17:31:09 2024 GMT (375 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]
Write out database with 1 new entries
Data Base Updated
(base) root@debian:~/ca#
  
```

7.2.2

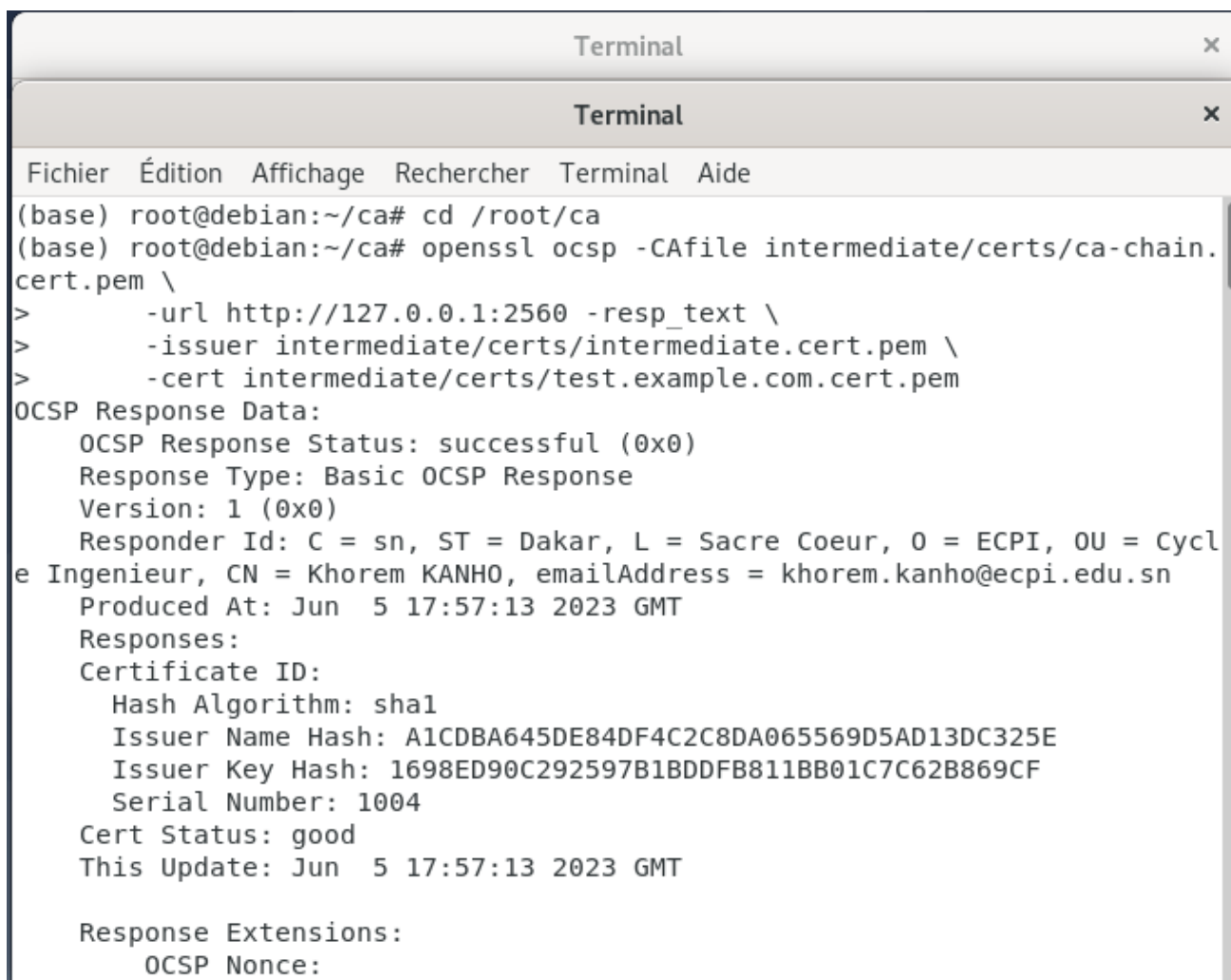
Exécutez le répondeur OCSP sur . Plutôt que de stocker l'état de révocation dans un fichier CRL séparé, le répondeur OCSP lit directement. Le La réponse est signée avec la paire cryptographique OCSP (à l'aide des options et).

```

(base) root@debian:~/ca# openssl ocsp -url http://127.0.0.1:2560 \
> -index intermediate/index.txt \
> -CA intermediate/certs/ca-chain.cert.pem \
> -rkey intermediate/private/ocsp.example.com.key.pem \
> -rsigner intermediate/certs/ocsp.example.com.cert.pem \
> -nrequest 1
Enter pass phrase for intermediate/private/ocsp.example.com.key.pem:
ocsp: waiting for OCSP client connections...
  
```

7.2.3

Dans un autre terminal, envoyez une requête au répondeur OCSP. L'option Spécifie le certificat à interroger.



```
Terminal
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
(base) root@debian:~/ca# cd /root/ca
(base) root@debian:~/ca# openssl ocsp -CAfile intermediate/certs/ca-chain.
cert.pem \
> -url http://127.0.0.1:2560 -resp_text \
> -issuer intermediate/certs/intermediate.cert.pem \
> -cert intermediate/certs/test.example.com.cert.pem
OCSP Response Data:
  OCSP Response Status: successful (0x0)
  Response Type: Basic OCSP Response
  Version: 1 (0x0)
  Responder Id: C = sn, ST = Dakar, L = Sacre Coeur, O = ECPI, OU = Cycl
e Ingenieur, CN = Khorem KANH0, emailAddress = khorem.kanho@ecpi.edu.sn
  Produced At: Jun  5 17:57:13 2023 GMT
  Responses:
  Certificate ID:
    Hash Algorithm: sha1
    Issuer Name Hash: A1CDBA645DE84DF4C2C8DA065569D5AD13DC325E
    Issuer Key Hash: 1698ED90C292597B1BDDFB811BB01C7C62B869CF
    Serial Number: 1004
  Cert Status: good
  This Update: Jun  5 17:57:13 2023 GMT

  Response Extensions:
    OCSP Nonce:
```

7.2.4

Révoquez le certificat.

```
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Onglets  Aide
Terminal x Terminal x + ▾
> -rkey intermediate/private/ocsp.example.com.key.pem \
> -rsigner intermediate/certs/ocsp.example.com.cert.pem \
> -nrequest 1
Enter pass phrase for intermediate/private/ocsp.example.com.key.pem:
unable to load responder private key
140296029492416:error:06065064:digital envelope routines:EVP_DecryptFinal_
ex:bad decrypt:crypto/evp/evp_enc.c:612:
140296029492416:error:0906A065:PEM routines:PEM_do_header:bad decrypt:cryp
to/pem/pem_lib.c:461:
(base) root@debian:~/ca# openssl ocsp -url http://127.0.0.1:2560 \
> -index intermediate/index.txt \
> -CA intermediate/certs/ca-chain.cert.pem \
> -rkey intermediate/private/ocsp.example.com.key.pem \
> -rsigner intermediate/certs/ocsp.example.com.cert.pem \
> -nrequest 1
Enter pass phrase for intermediate/private/ocsp.example.com.key.pem:
ocsp: waiting for OCSP client connections...
(base) root@debian:~/ca# openssl ca -config intermediate/openssl.cnf \
> -revoke intermediate/certs/test.example.com.cert.pem
Using configuration from intermediate/openssl.cnf
Enter pass phrase for /root/ca/intermediate/private/intermediate.key.pem:
Revoking Certificate 1004.
Data Base Updated
(base) root@debian:~/ca#
```