Creació i gestió d'una CA privada amb OpenSSL

Índex

1. Introducció **1**

2. Conceptes previs **1**
Jerarquia de certificats 1
Terminologia 2
Extensions X.509 més importants 2

3. Instal·lació d'OpenSSL **2**

4. Estructura de directoris de la CA **3**
Descripció dels directoris 4

5. Configuració del fitxer openssl.cnf **4**
5.1. Configuració de la CA arrel 4
5.2. Opcions de configuració clau explicades 7
Paràmetres de `[ CA_default ]` 7
Paràmetres de `[ req ]` 7
Valors de `keyUsage` 8
Valors de `extendedKeyUsage` 8

6. Creació de la CA arrel (Root CA) **8**
6.1. Generació de la clau privada de la CA arrel 8
6.2. Creació del certificat autosignat de la CA arrel 10
6.3. Verificació del certificat de la CA arrel 10

7. Creació d'una CA intermèdia (Intermediate CA) **12**
7.1. Fitxer de configuració de la CA intermèdia 12
7.2. Generació de la clau privada de la CA intermèdia 13
7.3. Creació del CSR de la CA intermèdia 13
7.4. Signatura del certificat de la CA intermèdia per la CA arrel 14
7.5. Creació de la cadena de certificats (chain) 15
7.6. Verificació de la cadena 15

8. Emissió de certificats **16**
8.1. Certificat de servidor web (TLS) 16
Pas 1: Genera la clau privada del servidor 16
Pas 2: Crea el CSR amb Subject Alternative Names (SAN) 16
Pas 3: Signa el certificat amb la CA intermèdia 18
Pas 4: Verificar el certificat del servidor 19
8.2. Certificat de client (autenticació) 21
8.3. Exportació a format PKCS#12 (per a navegadors i Windows) 23
8.4. Conversió entre formats 23

9. Revocació de certificats i llista CRL **24**
9.1. Revoca un certificat 24
9.2. Genera la CRL 24
9.3. Publicació de la CRL via HTTP 25
9.4. Verificació de revocació amb CRL 25

10. Verificació i inspecció de certificats **26**
10.1. Ordres d'inspecció essencials . . . 26
10.2. Verificació de connexions TLS . . . 26
10.3. Inspecció del fitxer index.txt . . . 27

11. Opcions de configuració avançades **27**
11.1. Certificats wildcard . . . 27
11.2. Configuració OCSP (Online Certificate Status Protocol) . . . 28
11.3. Configuració de períodes de validesa . . . 29
11.4. Automatització amb scripts . . . 29
Script per emetre certificats de servidor . . . 29
11.5. Instal·lació de la CA al sistema Ubuntu . . . 31
11.6. Integració amb serveis del sistema . . . 32
Apache2 . . . 32
Nginx . . . 32

12. Casos d'ús pràctics **33**
12.1. CA per a un laboratori ASIX . . . 33
12.2. Certificats per a Samba AD DC . . . 33
12.3. Certificats per a Kea DHCP + ISC Stork . . . 34

13. Bones pràctiques de seguretat **34**
Protecció de claus privades . . . 34
Períodes de validesa . . . 34
Algoritmes criptogràfics . . . 34
Auditoria i registre . . . 35
Còpies de seguretat . . . 35

14. Resum d'ordres **35**
Gestió de claus . . . 35
Gestió de CSR . . . 36
Gestió de certificats . . . 36
Gestió de CRL i OCSP . . . 36
Conversió de formats . . . 37

Cicle formatiu: CFGS Administració de sistemes informàtics en xarxa (ASIX)

Mòdul: 0375 - Serveis de xarxa i internet / 0378 - Seguretat i alta disponibilitat

Sistema operatiu: Ubuntu Server 26.04 LTS

---

1. Introducció

Una **Autoritat de Certificació (CA, Certificate Authority)** és una entitat de confiança encarregada d'emetre, signar, revocar i gestionar certificats digitals. En entorns corporatius, laboratoris o xarxes privades, és habitual crear una CA pròpia per garantir la seguretat de les comunicacions sense dependre de CA públiques.

Una CA privada permet:

- Emetre certificats TLS/SSL per a servidors web interns (Apache, Nginx, etc.)
- Autenticar clients i servidors en connexions VPN
- Signar correus electrònics (S/MIME)
- Garantir la integritat de fitxers i codi
- Gestionar autenticació de xarxa (802.1X, LDAP, etc.)

NOTA

OpenSSL és l'eina estàndard a GNU/Linux per gestionar infraestructures de clau pública (PKI). Ubuntu 26.04 inclou OpenSSL 3.x per defecte.

2. Conceptes previs

Abans de continuar, cal tenir clars aquests conceptes fonamentals:

Jerarquia de certificats

```
Root CA (CA Arrel)
    └── Intermediate CA (CA Intermèdia)
            ├── Certificat servidor web
            ├── Certificat client VPN
            └── Certificat servidor LDAP
```

La CA arrel té la màxima confiança. En producció, **mai s'utilitza directament** per signar certificats finals: s'usa una CA intermèdia per protegir la clau privada de la CA arrel.

Terminologia

Terme	Descripció
Clau privada	Fitxer secret que no s'ha de compartir mai
CSR	Certificate Signing Request --- sol·licitud de signatura
Certificat (.crt / .pem)	Document públic signat per la CA
CRL	Certificate Revocation List --- llista de certificats revocats
OCSP	Online Certificate Status Protocol --- alternativa a CRL
PEM	Format de codificació Base64 (el més habitual a GNU/Linux)
DER	Format binari (habitual a Windows i Java)
PKCS#12 (.p12/.pfx)	Contenidor que inclou clau privada + certificat

Extensions X.509 més importants

Extensió	Funció
`basicConstraints`	Indica si el certificat pot signar altres certificats
`keyUsage`	Usos de la clau (signatura digital, xifratge, etc.)
`extendedKeyUsage`	Usos estesos (servidor TLS, client TLS, correu, etc.)
`subjectAltName`	Noms alternatius (SAN): IP, DNS, correu
`crlDistributionPoints`	URL on trobar la CRL
`authorityInfoAccess`	URL del servei OCSP i de la CA emissora

3. Instal·lació d'OpenSSL

Ubuntu 26.04 inclou OpenSSL 3.x per defecte. Comprova la versió i instal·la les eines necessàries:

Comprova la versió d'OpenSSL

```
openssl version -a
```

Resposta esperada:

```
OpenSSL 3.5.5 27 Jan 2026 (Library: OpenSSL 3.5.5 27 Jan 2026)
built on: Tue Jun  2 17:21:36 2026 UTC
platform: debian-amd64
options:  bn(64,64)
```

```
compiler: gcc -fPIC -pthread -m64 -Wa,--noexecstack -Wall
 ↪  -fzero-call-used-regs=used-gpr -Wa,--noexecstack -g -O2
 ↪  -Werror=implicit-function-declaration -fno-omit-frame-pointer
 ↪  -mno-omit-leaf-frame-pointer
 ↪  -ffile-prefix-map=/build/openssl-UIHF8N/openssl-3.5.5=.
 ↪  -fstack-protector-strong -fstack-clash-protection -Wformat
 ↪  -Werror=format-security -fcf-protection -fdebug-prefix-map=/buil
 ↪  d/openssl-UIHF8N/openssl-3.5.5=/usr/src/openssl-3.5.5-1ubuntu3.2
 ↪  -DOPENSSL_USE_NODELETE -DL_ENDIAN -DOPENSSL_PIC
 ↪  -DOPENSSL_BUILDING_OPENSSL -DZLIB -DZSTD -DNDEBUG -Wdate-time
 ↪  -D_FORTIFY_SOURCE=3
OPENSSLDIR: "/usr/lib/ssl"
ENGINESDIR: "/usr/lib/x86_64-linux-gnu/engines-3"
MODULESDIR: "/usr/lib/x86_64-linux-gnu/ossl-modules"
Seeding source: os-specific JITTER (3060300)
CPUINFO: OPENSSL_ia32cap=0xfefa3203478bffff:0x00000000208c2569:0x000
 ↪  0000030000400:0x0000000000000000:0x0000000000000000
```

Actualitza el sistema

```
sudo apt update && sudo apt upgrade
```

Instal·la OpenSSL i eines relacionades

```
sudo apt install openssl ca-certificates
```

Verifica la instal·lació

```
openssl version
```

Sortida esperada:

```
OpenSSL 3.5.5 27 Jan 2026 (Library: OpenSSL 3.5.5 27 Jan 2026)
```

4. Estructura de directoris de la CA

Crea una estructura ben organitzada per a la nostra PKI. En aquest exemple, el directori arrel de la CA serà `/opt/ca`.

Per què `/opt`?

L'estàndard FHS (Filesystem Hierarchy Standard) defineix on va cada cosa a GNU/Linux. `/opt` és el punt de muntatge per a programari i dades autònoms que no formen part del sistema base, és a dir, aplicacions o infraestructures que gestiones tu manualment. Una CA privada és exactament això: una infraestructura independent, amb el seu propi conjunt de dades, claus, certificats i configuració, que no pertany a cap paquet del sistema.

Crea l'estructura de directoris per a la CA arrel

```
sudo mkdir -p /opt/ca/root-ca/{certs,crl,csr,newcerts,private}
sudo chmod 700 /opt/ca/root-ca/private
sudo touch /opt/ca/root-ca/index.txt
echo 1000 | sudo tee /opt/ca/root-ca/serial
echo 1000 | sudo tee /opt/ca/root-ca/crlnumber
```

Crea l'estructura per a la CA intermèdia

```
sudo mkdir -p
 ↪  /opt/ca/intermediate-ca/{certs,crl,csr,newcerts,private}
sudo chmod 700 /opt/ca/intermediate-ca/private
sudo touch /opt/ca/intermediate-ca/index.txt
echo 1000 | sudo tee /opt/ca/intermediate-ca/serial
echo 1000 | sudo tee /opt/ca/intermediate-ca/crlnumber
```

Descripció dels directoris

Directori	Contingut
`certs/`	Certificats emesos per la CA
`crl/`	Llistes de revocació (CRL)
`csr/`	Sol·licituds de signatura pendents
`newcerts/`	Còpies dels certificats emesos (per número de sèrie)
`private/`	Claus privades (permís 700, accés restringit)
`index.txt`	Base de dades de certificats emesos
`serial`	Número de sèrie del següent certificat
`crlnumber`	Número de sèrie de la següent CRL

5. Configuració del fitxer openssl.cnf

El fitxer de configuració `openssl.cnf` defineix el comportament de la CA. Crea fitxers de configuració separats per a la CA arrel i la intermèdia.

5.1. Configuració de la CA arrel

```
sudo nano /opt/ca/root-ca/openssl.cnf
```

```
# ====================================================================
# Configuració de la CA Arrel
# CA Privada - ASIX - Thos
# ====================================================================

[ ca ]
default_ca = CA_default
```

```
[ CA_default ]
# Directoris principals
dir               = /opt/ca/root-ca
certs             = $dir/certs
crl_dir           = $dir/crl
new_certs_dir     = $dir/newcerts
database          = $dir/index.txt
serial            = $dir/serial
RANDFILE          = $dir/private/.rand

# Clau i certificat de la CA
private_key       = $dir/private/ca.key.pem
certificate       = $dir/certs/ca.cert.pem

# CRL
crlnumber         = $dir/crlnumber
crl               = $dir/crl/ca.crl.pem
crl_extensions    = crl_ext
default_crl_days  = 30

# Algoritme de hash per a les signatures
default_md        = sha256

# Control de noms (no permet noms duplicats)
name_opt          = ca_default
cert_opt          = ca_default

# Validesa per defecte dels certificats emesos
default_days      = 375

# Preservar el DN del sol·licitant
preserve          = no

# Política de comprovació de camps del DN
policy            = policy_strict

[ policy_strict ]
# La CA arrel només signa CA intermèdies amb DN coincident
countryName             = match
stateOrProvinceName     = match
organizationName        = match
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

[ policy_loose ]
# Política permissiva per a la CA intermèdia
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
```

```
emailAddress            = optional

[ req ]
# Opcions per a la generació de CSRs
default_bits        = 4096
distinguished_name  = req_distinguished_name
string_mask         = utf8only
default_md          = sha256
x509_extensions     = v3_ca

[ req_distinguished_name ]
# Camps del Distinguished Name (DN)
countryName                     = País (codi de 2 lletres)
stateOrProvinceName             = Comunitat Autònoma
localityName                    = Localitat
0.organizationName              = Organització
organizationalUnitName          = Unitat Organitzativa
commonName                      = Nom Comú
emailAddress                    = Adreça de correu

# Valors per defecte
countryName_default             = ES
stateOrProvinceName_default     = Catalunya
localityName_default            = Mataro
0.organizationName_default      = IES Thos i Codina
organizationalUnitName_default  = Departament Informatica
commonName_default              = CA Arrel - Thos
emailAddress_default            = admin@thosicodina.local

[ v3_ca ]
# Extensions per al certificat de la CA arrel (autosignat)
subjectKeyIdentifier   = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints       = critical, CA:true
keyUsage               = critical, digitalSignature, cRLSign,
 ↪  keyCertSign

[ v3_intermediate_ca ]
# Extensions per al certificat de la CA intermèdia
subjectKeyIdentifier   = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints       = critical, CA:true, pathlen:0
keyUsage               = critical, digitalSignature, cRLSign,
 ↪  keyCertSign

[ usr_cert ]
# Extensions per a certificats d'usuari/client
basicConstraints       = CA:FALSE
nsCertType             = client, email
nsComment              = "Certificat de client generat per OpenSSL"
subjectKeyIdentifier   = hash
authorityKeyIdentifier = keyid,issuer
```

```
keyUsage               = critical, nonRepudiation, digitalSignature,
 ↪  keyEncipherment
extendedKeyUsage       = clientAuth, emailProtection

[ server_cert ]
# Extensions per a certificats de servidor
basicConstraints       = CA:FALSE
nsCertType             = server
nsComment              = "Certificat de servidor generat per OpenSSL"
subjectKeyIdentifier   = hash
authorityKeyIdentifier = keyid,issuer:always
keyUsage               = critical, digitalSignature, keyEncipherment
extendedKeyUsage       = serverAuth

[ crl_ext ]
# Extensions per a les CRL
authorityKeyIdentifier = keyid:always

[ ocsp ]
# Extensions per al servei OCSP
basicConstraints       = CA:FALSE
subjectKeyIdentifier   = hash
authorityKeyIdentifier = keyid,issuer
keyUsage               = critical, digitalSignature
extendedKeyUsage       = critical, OCSPSigning
```

5.2. Opcions de configuració clau explicades

Paràmetres de `[ CA_default ]`

Paràmetre	Valor exemple	Descripció
`default_md`	`sha256`	Hash per defecte per a les signatures. Mai usar MD5 o SHA1
`default_days`	`375`	Dies de validesa dels certificats emesos
`default_crl_days`	`30`	Dies de validesa de la CRL
`preserve`	`no`	Si `yes`, preserva l'ordre dels camps del DN del CSR
`policy`	`policy_strict`	Política de validació de camps del DN

Paràmetres de `[ req ]`

Paràmetre	Valor exemple	Descripció
`default_bits`	`4096`	Mida de la clau RSA en bits (mínim 2048, recomanat 4096)
`default_md`	`sha256`	Hash per defecte
`string_mask`	`utf8only`	Codificació dels camps de text
`prompt`	`no`	Si `no`, no demana dades interactivament

Valors de `keyUsage`

Valor	Ús
`digitalSignature`	Signatura digital
`nonRepudiation`	No repudi (prova d'autoria)
`keyEncipherment`	Xifratge de claus (RSA)
`dataEncipherment`	Xifratge de dades
`keyAgreement`	Intercanvi de claus (Diffie-Hellman)
`keyCertSign`	Signatura de certificats (només CA)
`cRLSign`	Signatura de llistes CRL (només CA)

Valors de `extendedKeyUsage`

Valor	Ús
`serverAuth`	Autenticació de servidor TLS
`clientAuth`	Autenticació de client TLS
`codeSigning`	Signatura de codi
`emailProtection`	Protecció de correu (S/MIME)
`timeStamping`	Segells de temps
`OCSPSigning`	Respondedor OCSP

6. Creació de la CA arrel (Root CA)

6.1. Generació de la clau privada de la CA arrel

IMPORTANT

La clau privada de la CA arrel és el secret més crític de tota la PKI. Protegeix-la amb una contrasenya forta i guarda-la en un lloc segur (preferiblement fora de línia).

```
cd /opt/ca
```

Genera clau RSA de 4096 bits amb xifratge AES-256

```
sudo openssl genrsa -aes256 -out root-ca/private/ca.key.pem 4096
```

Introdueix la **pass phrase** i repeteix-la.

```
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

NOTA

Una pass phrase (o frase de pas) és una contrasenya utilitzada per xifrar una clau privada quan es guarda en disc.

Una contrasenya sol ser una paraula curta. Una pass phrase és idealment una frase sencera, cosa que la fa molt més difícil de trencar.

Estableix permisos restrictius

```
sudo chmod 400 root-ca/private/ca.key.pem
```

Verifica la clau generada

```
sudo openssl rsa -in root-ca/private/ca.key.pem -check
```

```
Enter pass phrase for root-ca/private/ca.key.pem:
RSA key ok
writing RSA key
-----BEGIN PRIVATE KEY-----
MIIJQQIBADANBgkqhkiG9w0BAQEFAASCCSswggknAgEAAoICAQCXFZEttJVNuqWC
TRCa9yCB1EGOMDz1xvEKMRSvU+ye+CcZkCdT3cXnxweElHR37AII6QBlzJL4ObG1
VNvWa9zktgTxBUBVnFnsX600TuSnkVyMgzmZYDauI1Csxc9keWFdeOlAp498e+WT
tCUfTHTlTih6hI7kbDKekRqhNdW7cDOcHUfM/IH0yJzikcOfs84APLYMOwDSCTdJ
18h6ekCSnssWK9hiIWczcls/e8qLOzcLEOsjGkQQ/nDz6/vJi0KTz9ZE4EvD6tB0
...
4QKCAQBgeXYMyS6rIBj+maI6VPW775ExXhb+Y9u+NV8vb3BvsOOiqR5TUDrqqJB4
2atjFRlIkJCLWvU1T1DzqSN1kGEfxRrcG6/86lK1O5/vsORg1rahsb3Lhxh40m/2
H7p4U3DZUuUL5PuvSPJU1x8uE8ZjfKwsSejJae22oAWC2dBYjCTlS4RMtNPtKVEm
82MMmoFzxPpgTpowaOWdTPfzYjh1RhCCSfZJn3/S4eTXWUp4ssteF7J3DrlNhtlt
BjPM6Y3P49vPaGl7EW13GbpxzDy2tac3PFNiUJ6BT/PXc7wHZKFmIcSW3zD10rh4
vmkhFw4cV6S0YvL05Fvi2U/YXP4U
-----END PRIVATE KEY-----
```

Alternativa amb claus de corba el·líptica (ECDSA):

Generar clau ECDSA (més ràpida i segura que RSA equivalent)

```
sudo openssl ecparam \
    -name prime256v1 \
    -genkey \
    -noout \
    -out root-ca/private/ca.key.pem
```

Xifrar la clau amb AES-256

```
sudo openssl ec \
    -in root-ca/private/ca.key.pem \
    -aes256 \
    -out root-ca/private/ca.key.enc.pem
```

6.2. Creació del certificat autosignat de la CA arrel

Crea el certificat autosignat (vàlid 20 anys = 7300 dies)

```
sudo openssl req \
    -config root-ca/openssl.cnf \
    -key root-ca/private/ca.key.pem \
    -new \
    -x509 \
    -days 7300 \
    -sha256 \
    -extensions v3_ca \
    -out root-ca/certs/ca.cert.pem
```

Escriu la pass phrase i omple els camps del Distinguished Name (DN) quan se't demanin:

```
Enter pass phrase for root-ca/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.
-----
País (codi de 2 lletres) [ES]:
Comunitat Autònoma [Catalunya]:
Localitat [Mataro]:
Organització [IES Thos i Codina]:
Unitat Organitzativa [Departament Informatica]:
Nom Comú [CA Arrel - Thos]:
Adreça de correu [admin@thosicodina.local]:
```

Estableix permisos de lectura

```
sudo chmod 444 root-ca/certs/ca.cert.pem
```

6.3. Verificació del certificat de la CA arrel

Mostra la informació del certificat

```
sudo openssl x509 -noout -text -in root-ca/certs/ca.cert.pem
```

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
           69:21:54:90:33:a3:1d:5c:31:d3:ba:c9:86:61:13:31:d1:42:52:3e
```

```
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=ES, ST=Catalunya, L=Mataro, O=IES Thos i Codina,
↪   OU=Departament Informatica, CN=CA Arrel - Thos,
↪   emailAddress=admin@thosicodina.local
        Validity
            Not Before: Jun 29 14:59:33 2026 GMT
            Not After : Jun 24 14:59:33 2046 GMT
        Subject: C=ES, ST=Catalunya, L=Mataro, O=IES Thos i Codina,
↪   OU=Departament Informatica, CN=CA Arrel - Thos,
↪   emailAddress=admin@thosicodina.local
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (4096 bit)
                Modulus:
                    00:97:15:91:2d:b4:95:4d:ba:a5:82:4d:10:9a:f7:
                    20:81:d4:41:8e:30:3c:f5:c6:f1:0a:31:14:af:53:
                    ec:9e:f8:27:19:90:27:53:dd:c5:e7:c7:07:84:94:
                    af:17:e7:81:f6:e3:d4:69:d2:12:ac:c1:5a:81:c5:
                    aa:77:70:11:57:e2:ac:ca:e0:61:ee:3e:d7:50:40:
                    ...
                    08:f0:74:58:10:ab:80:e6:4b:c3:e7:57:d0:f2:9b:
                    ce:53:aa:16:85:e0:76:48:55:ad:f9:64:ba:cb:02:
                    45:3c:d5:8f:87:c5:bb:17:75:ff:91:63:d2:72:9e:
                    fc:cd:a2:34:b0:21:5d:dc:9e:0d:59:cf:09:ea:ba:
                    e9:df:22:86:38:21:dd:d1:12:f3:74:a4:3f:bf:fc:
                    b5:d2:55
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:

↪   C1:7F:E6:A7:37:51:FE:9C:CE:C0:BA:9A:BB:5B:27:84:40:45:9B:CE
            X509v3 Authority Key Identifier:

↪   C1:7F:E6:A7:37:51:FE:9C:CE:C0:BA:9A:BB:5B:27:84:40:45:9B:CE
            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Key Usage: critical
                Digital Signature, Certificate Sign, CRL Sign
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        20:a1:6b:04:f6:66:a8:77:f9:bd:74:7f:33:a0:8e:cc:03:1b:
        6f:85:77:53:dc:91:be:af:f2:ae:f0:e3:f9:42:2d:80:2c:df:
        b4:0e:af:ac:0c:b4:d5:b4:d6:c2:95:f8:b0:ab:ff:63:3f:32:
        0e:fa:41:4d:90:89:77:49:26:3f:6d:9f:90:7b:26:46:65:f7:
        ...
        ee:fe:3f:10:fd:c5:fb:8e:fd:f4:54:6c:f2:bc:97:30:8c:01:
        2f:ed:39:4b:99:04:f8:8b:2b:f8:38:3e:c0:e0:43:51:fb:78:
        fc:e0:fc:8f:cd:18:78:fa:5e:4e:19:2f:51:65:02:b6:7a:fb:
        57:20:8d:65:c5:f9:f8:c3:5c:f3:fa:6d:e3:8a:f4:f5:4c:23:
        47:e4:b3:50:ca:de:85:c1:e2:3e:9c:fc:e9:2c:67:9d:63:96:
        c4:80:90:e1:b1:b5:ef:f8
```

Comprova dates de validesa

```
sudo openssl x509 -noout -dates -in root-ca/certs/ca.cert.pem
```

```
notBefore=Jun 29 14:59:33 2026 GMT
notAfter=Jun 24 14:59:33 2046 GMT
```

Verifica la signatura (autosignada)

```
sudo openssl verify \
    -CAfile root-ca/certs/ca.cert.pem \
    root-ca/certs/ca.cert.pem
```

Resposta esperada:

```
root-ca/certs/ca.cert.pem: OK
```

7. Creació d'una CA intermèdia (Intermediate CA)

7.1. Fitxer de configuració de la CA intermèdia

Copia la configuració de l'arrel a la carpeta `intermediate-ca`

```
sudo cp root-ca/openssl.cnf intermediate-ca/openssl.cnf
```

Edita el fitxer de configuració

```
sudo nano /opt/ca/intermediate-ca/openssl.cnf
```

Modifica el bloc `[ CA_default ]` per apuntar als directoris de la CA intermèdia:

```
[ CA_default ]
dir               = /opt/ca/intermediate-ca
private_key       = $dir/private/intermediate.key.pem
certificate       = $dir/certs/intermediate.cert.pem
crl               = $dir/crl/intermediate.crl.pem
# ... (resta igual que la CA arrel però amb política loose)
policy            = policy_loose
```

I modifica el `commonName_default`:

```
commonName_default = CA Intermedia - Thos
```

7.2. Generació de la clau privada de la CA intermèdia

Genera clau privada de la CA intermèdia

```
sudo openssl genrsa \
    -aes256 \
    -out intermediate-ca/private/intermediate.key.pem \
    4096
```

Introdueix la **pass phrase** i repeteix-la.

```
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

Modifica els permisos

```
sudo chmod 400 intermediate-ca/private/intermediate.key.pem
```

7.3. Creació del CSR de la CA intermèdia

Crea la sol·licitud de signatura (CSR)

```
sudo openssl req \
    -config intermediate-ca/openssl.cnf \
    -new \
    -sha256 \
    -key intermediate-ca/private/intermediate.key.pem \
    -out intermediate-ca/csr/intermediate.csr.pem
```

```
Enter pass phrase for intermediate-ca/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.
-----
País (codi de 2 lletres) [ES]:
Comunitat Autònoma [Catalunya]:
Localitat [Mataro]:
Organització [IES Thos i Codina]:
Unitat Organitzativa [Departament Informatica]:
Nom Comú [CA Intermedia - Thos]:
Adreça de correu [admin@thosicodina.local]:
```

7.4. Signatura del certificat de la CA intermèdia per la CA arrel

La CA arrel signa el certificat de la CA intermèdia (10 anys)

```
sudo openssl ca \
    -config root-ca/openssl.cnf \
    -extensions v3_intermediate_ca \
    -days 3650 \
    -notext \
    -md sha256 \
    -in intermediate-ca/csr/intermediate.csr.pem \
    -out intermediate-ca/certs/intermediate.cert.pem
```

```
Using configuration from root-ca/openssl.cnf
Enter pass phrase for /opt/ca/root-ca/private/ca.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4096 (0x1000)
        Validity
            Not Before: Jun 29 15:28:43 2026 GMT
            Not After : Jun 26 15:28:43 2036 GMT
        Subject:
            countryName               = ES
            stateOrProvinceName       = Catalunya
            organizationName          = IES Thos i Codina
            organizationalUnitName    = Departament Informatica
            commonName                = CA Intermedia - Thos
            emailAddress              = admin@thosicodina.local
        X509v3 extensions:
            X509v3 Subject Key Identifier:

 ↪  0D:E5:1D:1A:99:55:C3:D8:71:BC:02:27:10:F8:2E:34:E7:02:6C:AC
            X509v3 Authority Key Identifier:

 ↪  C1:7F:E6:A7:37:51:FE:9C:CE:C0:BA:9A:BB:5B:27:84:40:45:9B:CE
            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 26 15:28:43 2036 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
Database updated
```

Modifica els permisos

```
sudo chmod 444 intermediate-ca/certs/intermediate.cert.pem
```

7.5. Creació de la cadena de certificats (chain)

Concatena el certificat intermedi i el de la CA arrel

```
sudo cat intermediate-ca/certs/intermediate.cert.pem \
    root-ca/certs/ca.cert.pem \
    | sudo tee intermediate-ca/certs/ca-chain.cert.pem > /dev/null
```

Modifica els permisos

```
sudo chmod 444 intermediate-ca/certs/ca-chain.cert.pem
```

7.6. Verificació de la cadena

Verifica el certificat intermedi contra la CA arrel

```
sudo openssl verify \
    -CAfile root-ca/certs/ca.cert.pem \
    intermediate-ca/certs/intermediate.cert.pem
```

```
intermediate-ca/certs/intermediate.cert.pem: OK
```

Verifica la cadena completa

```
sudo openssl verify \
    -CAfile intermediate-ca/certs/ca-chain.cert.pem \
    intermediate-ca/certs/intermediate.cert.pem
```

```
intermediate-ca/certs/intermediate.cert.pem: OK
```

8. Emissió de certificats

8.1. Certificat de servidor web (TLS)

Pas 1: Genera la clau privada del servidor

Clau sense contrasenya (necessari per a servidors web que arrenquen automàticament)

```
sudo openssl genrsa \
    -out intermediate-ca/private/www.thos.local.key.pem \
    2048
```

Modifica els permisos

```
sudo chmod 400 intermediate-ca/private/www.thos.local.key.pem
```

Pas 2: Crea el CSR amb Subject Alternative Names (SAN)

Per a certificats de servidor moderns cal especificar els SAN. Crea un fitxer de configuració per al CSR:

```
cat > /tmp/www.thos.local.cnf << 'EOF'
[ req ]
default_bits       = 2048
distinguished_name = req_distinguished_name
req_extensions     = req_ext
prompt             = no

[ req_distinguished_name ]
C  = ES
ST = Catalunya
L  = Mataro
O  = IES Thos i Codina
OU = Departament Informatica
CN = www.thos.local

[ req_ext ]
subjectAltName = @alt_names

[ alt_names ]
DNS.1 = www.thos.local
DNS.2 = thos.local
DNS.3 = intranet.thos.local
IP.1  = 192.168.1.10
EOF
```

Genera el CSR

```
sudo openssl req \
    -new \
```

```
    -sha256 \
    -key intermediate-ca/private/www.thos.local.key.pem \
    -config /tmp/www.thos.local.cnf \
    -out intermediate-ca/csr/www.thos.local.csr.pem
```

Verifica el CSR

```
sudo openssl req -text -noout -verify \
    -in intermediate-ca/csr/www.thos.local.csr.pem
```

```
Certificate request self-signature verify OK
Certificate Request:
    Data:
        Version: 1 (0x0)
        Subject: C=ES, ST=Catalunya, L=Mataro, O=IES Thos i Codina,
 ↪  OU=Departament Informatica, CN=www.thos.local
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:bf:a7:9f:44:20:b7:72:fc:76:a1:96:43:d9:14:
                    5e:94:d8:75:b9:fb:b1:fb:a8:cb:7b:dd:1d:f9:69:
                    25:82:8f:b5:24:12:ab:47:9b:2d:be:ba:29:b5:94:
                    0f:03:e4:27:a8:7a:15:e7:8d:01:ef:d9:ca:40:4e:
                    ...
                    d6:ff:74:b0:04:bd:07:e0:b8:d4:f9:fa:57:ed:a5:
                    91:3e:ff:0d:7d:4b:d4:16:72:b3:c2:c1:13:73:b8:
                    9f:44:c1:17:b0:25:58:be:b1:6a:eb:a6:a1:19:20:
                    29:41
                Exponent: 65537 (0x10001)
        Attributes:
            Requested Extensions:
                X509v3 Subject Alternative Name:
                    DNS:www.thos.local, DNS:thos.local,
 ↪  DNS:intranet.thos.local, IP Address:192.168.1.10
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        15:12:53:14:68:55:a2:35:e5:57:bb:c2:4f:c5:cf:24:7a:b8:
        74:8f:9e:66:bc:4e:38:29:20:1b:4d:c4:ff:5f:f8:41:c9:d1:
        76:cb:3e:f7:a2:3e:f9:70:35:32:92:75:f2:96:e1:ad:b6:06:
        ...
        9b:a1:01:d8:0e:ca:81:02:2f:32:78:63:50:80:f3:1b:a1:e3:
        b4:54:d0:be:1f:8d:62:98:f6:59:d0:86:0e:dc:fb:62:9a:91:
        72:fa:1f:bf:e0:4f:a2:18:70:cd:d8:06:07:c4:81:c9:b5:37:
        ba:66:e4:79
```

Pas 3: Signa el certificat amb la CA intermèdia

Crea un fitxer d'extensions per preservar els SAN:

```
cat > /tmp/server_ext.cnf << 'EOF'
[ server_cert ]
basicConstraints       = CA:FALSE
nsCertType             = server
subjectKeyIdentifier   = hash
authorityKeyIdentifier = keyid,issuer:always
keyUsage               = critical, digitalSignature, keyEncipherment
extendedKeyUsage       = serverAuth
subjectAltName         = @alt_names

[ alt_names ]
DNS.1 = www.thos.local
DNS.2 = thos.local
DNS.3 = intranet.thos.local
IP.1  = 192.168.1.10
EOF
```

Signa el certificat (1 any)

```
sudo openssl ca \
    -config intermediate-ca/openssl.cnf \
    -extfile /tmp/server_ext.cnf \
    -extensions server_cert \
    -days 365 \
    -notext \
    -md sha256 \
    -in intermediate-ca/csr/www.thos.local.csr.pem \
    -out intermediate-ca/certs/www.thos.local.cert.pem
```

```
Using configuration from intermediate-ca/openssl.cnf
Enter pass phrase for
 ↪  /opt/ca/intermediate-ca/private/intermediate.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4096 (0x1000)
        Validity
            Not Before: Jun 29 16:16:57 2026 GMT
            Not After : Jun 29 16:16:57 2027 GMT
        Subject:
            countryName               = ES
            stateOrProvinceName       = Catalunya
            localityName              = Mataro
            organizationName          = IES Thos i Codina
            organizationalUnitName    = Departament Informatica
            commonName                = www.thos.local
        X509v3 extensions:
            X509v3 Basic Constraints:
```

```
                CA:FALSE
            Netscape Cert Type:
                SSL Server
            X509v3 Subject Key Identifier:

 ↪  9D:E4:68:D3:55:66:D1:1A:31:66:5B:27:36:89:E2:97:A0:8A:56:C9
            X509v3 Authority Key Identifier:

 ↪  keyid:35:3B:74:47:F5:7C:34:FC:30:DA:CB:65:F4:1C:3E:4C:DC:F7:3A:8E
                DirName:/C=ES/ST=Catalunya/L=Mataro/O=IES Thos i
 ↪  Codina/OU=Departament Informatica/CN=CA Arrel -
 ↪  Thos/emailAddress=admin@thosicodina.local
                serial:10:00
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
            X509v3 Subject Alternative Name:
                DNS:www.thos.local, DNS:thos.local,
 ↪  DNS:intranet.thos.local, IP Address:192.168.1.10
Certificate is to be certified until Jun 29 16:16:57 2027 GMT (365
 ↪  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
Database updated
```

Modifica els permisos

```
sudo chmod 444 intermediate-ca/certs/www.thos.local.cert.pem
```

Pas 4: Verificar el certificat del servidor

Mostra informació completa

```
sudo openssl x509 -text -noout \
    -in intermediate-ca/certs/www.thos.local.cert.pem
```

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 4096 (0x1000)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=ES, ST=Catalunya, O=IES Thos i Codina,
 ↪  OU=Departament Informatica, CN=CA Intermedia - Thos,
 ↪  emailAddress=admin@thosicodina.local
        Validity
```

```
            Not Before: Jun 29 16:16:57 2026 GMT
            Not After : Jun 29 16:16:57 2027 GMT
        Subject: C=ES, ST=Catalunya, L=Mataro, O=IES Thos i Codina,
↪  OU=Departament Informatica, CN=www.thos.local
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:b9:13:d3:d5:85:fe:b0:65:8e:f8:90:04:ca:14:
                    7d:9d:aa:73:e2:4a:3e:63:c0:1c:e9:0d:8d:06:23:
                    3e:95:0e:11:7a:4a:16:cb:54:38:64:be:a8:78:4c:
                    ...
                    ed:29:b6:b3:30:ed:20:66:6f:66:ed:e6:88:bb:59:
                    d0:2a:e0:f0:1e:11:f1:ab:24:bf:6a:6b:55:71:ca:
                    86:c5:48:2f:69:06:ef:0f:02:0c:3b:0e:5b:93:59:
                    b2:af
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Cert Type:
                SSL Server
            X509v3 Subject Key Identifier:

↪  9D:E4:68:D3:55:66:D1:1A:31:66:5B:27:36:89:E2:97:A0:8A:56:C9
            X509v3 Authority Key Identifier:

↪  keyid:35:3B:74:47:F5:7C:34:FC:30:DA:CB:65:F4:1C:3E:4C:DC:F7:3A:8E
                DirName:/C=ES/ST=Catalunya/L=Mataro/O=IES Thos i
↪  Codina/OU=Departament Informatica/CN=CA Arrel -
↪  Thos/emailAddress=admin@thosicodina.local
                serial:10:00
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
            X509v3 Subject Alternative Name:
                DNS:www.thos.local, DNS:thos.local,
↪  DNS:intranet.thos.local, IP Address:192.168.1.10
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        0e:a4:d8:70:5e:7c:d5:05:00:2d:1d:92:41:f0:5f:ad:da:fa:
        89:3a:55:14:76:71:ec:8e:31:21:d8:fb:2c:a2:f1:0a:7d:b6:
        ce:41:9e:f5:c2:e9:d9:5f:50:6f:b4:39:22:ff:11:fc:8f:49:
        ...
        ae:63:55:f8:c6:7c:ac:d5:67:68:ec:4e:c7:d6:52:7f:c6:98:
        fd:7c:2c:67:d8:f0:5e:ca:8d:5f:7d:cc:f6:a8:61:d4:f4:31:
        53:46:4b:fd:94:37:5a:22
```

Verifica contra la cadena de CA

```
sudo openssl verify \
    -CAfile intermediate-ca/certs/ca-chain.cert.pem \
    intermediate-ca/certs/www.thos.local.cert.pem
```

```
intermediate-ca/certs/www.thos.local.cert.pem: OK
```

8.2. Certificat de client (autenticació)

Clau del client

```
sudo openssl genrsa \
    -aes256 \
    -out intermediate-ca/private/client.usuari.key.pem \
    2048
```

```
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

CSR del client

```
sudo openssl req \
    -config intermediate-ca/openssl.cnf \
    -new \
    -sha256 \
    -key intermediate-ca/private/client.usuari.key.pem \
    -out intermediate-ca/csr/client.usuari.csr.pem
```

```
Enter pass phrase for intermediate-ca/private/client.usuari.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.
-----
País (codi de 2 lletres) [ES]:
Comunitat Autònoma [Catalunya]:
Localitat [Mataro]:
Organització [IES Thos i Codina]:
Unitat Organitzativa [Departament Informatica]:
Nom Comú [CA Intermedia - Thos]:
Adreça de correu [admin@thosicodina.local]:
```

Signa com a certificat de client

```
sudo openssl ca \
    -config intermediate-ca/openssl.cnf \
    -extensions usr_cert \
    -days 365 \
    -notext \
    -md sha256 \
    -in intermediate-ca/csr/client.usuari.csr.pem \
    -out intermediate-ca/certs/client.usuari.cert.pem
```

```
Using configuration from intermediate-ca/openssl.cnf
Enter pass phrase for
 ↪  /opt/ca/intermediate-ca/private/intermediate.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4097 (0x1001)
        Validity
            Not Before: Jun 29 16:22:08 2026 GMT
            Not After : Jun 29 16:22:08 2027 GMT
        Subject:
            countryName               = ES
            stateOrProvinceName       = Catalunya
            localityName              = Mataro
            organizationName          = IES Thos i Codina
            organizationalUnitName    = Departament Informatica
            commonName                = CA Intermedia - Thos
            emailAddress              = admin@thosicodina.local
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Cert Type:
                SSL Client, S/MIME
            Netscape Comment:
                Certificat de client generat per OpenSSL
            X509v3 Subject Key Identifier:

 ↪  3E:17:8E:06:9D:4B:B7:73:40:9C:56:43:4D:CD:F5:52:D6:9C:E0:4D
            X509v3 Authority Key Identifier:

 ↪  35:3B:74:47:F5:7C:34:FC:30:DA:CB:65:F4:1C:3E:4C:DC:F7:3A:8E
            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 29 16:22:08 2027 GMT (365
 ↪  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
Database updated
```

8.3. Exportació a format PKCS#12 (per a navegadors i Windows)

Exporta la clau i el certificat en un únic fitxer .p12

```
sudo openssl pkcs12 \
    -export \
    -out intermediate-ca/certs/client.usuari.p12 \
    -inkey intermediate-ca/private/client.usuari.key.pem \
    -in intermediate-ca/certs/client.usuari.cert.pem \
    -certfile intermediate-ca/certs/ca-chain.cert.pem \
    -name "Certificat Usuari - IES Thos i Codina"
```

```
Enter pass phrase for intermediate-ca/private/client.usuari.key.pem:
Enter Export Password:
Verifying - Enter Export Password:
```

8.4. Conversió entre formats

PEM → DER (format binari per a Java / Windows)

```
sudo openssl x509 \
    -in intermediate-ca/certs/www.thos.local.cert.pem \
    -outform DER \
    -out intermediate-ca/certs/www.thos.local.cert.der
```

DER → PEM

```
sudo openssl x509 \
    -in intermediate-ca/certs/www.thos.local.cert.der \
    -inform DER \
    -out intermediate-ca/certs/www.thos.local.cert.pem
```

PKCS#12 → PEM (extreure clau i certificat)

```
sudo openssl pkcs12 \
    -in client.usuari.p12 \
    -out client.usuari.pem \
    -nodes
```

9. Revocació de certificats i llista CRL

9.1. Revoca un certificat

Revoca un certificat. Motius: unspecified, keyCompromise, CACompromise, affiliationChanged, superseded, cessationOfOperation, certificateHold.

```
sudo openssl ca \
    -config intermediate-ca/openssl.cnf \
    -revoke intermediate-ca/certs/www.thos.local.cert.pem \
    -crl_reason keyCompromise
```

9.2. Genera la CRL

Genera la CRL actualitzada

```
sudo openssl ca \
    -config intermediate-ca/openssl.cnf \
    -gencrl \
    -out intermediate-ca/crl/intermediate.crl.pem
```

```
Using configuration from intermediate-ca/openssl.cnf
Enter pass phrase for
 ↪  /opt/ca/intermediate-ca/private/intermediate.key.pem:
```

Verifica la CRL

```
sudo openssl crl \
    -in intermediate-ca/crl/intermediate.crl.pem \
    -noout \
    -text
```

```
Certificate Revocation List (CRL):
        Version 2 (0x1)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=ES, ST=Catalunya, O=IES Thos i Codina,
 ↪  OU=Departament Informatica, CN=CA Intermedia - Thos,
 ↪  emailAddress=admin@thosicodina.local
        Last Update: Jun 29 16:26:25 2026 GMT
        Next Update: Jul 29 16:26:25 2026 GMT
        CRL extensions:
            X509v3 Authority Key Identifier:

 ↪  35:3B:74:47:F5:7C:34:FC:30:DA:CB:65:F4:1C:3E:4C:DC:F7:3A:8E
            X509v3 CRL Number:
                4096
No Revoked Certificates.
    Signature Algorithm: sha256WithRSAEncryption
```

```
    Signature Value:
        74:81:1e:63:08:62:03:f2:f5:9b:f3:27:b4:54:be:92:6e:88:
        20:c7:04:d4:03:fd:54:eb:23:81:0d:69:c8:ce:2f:a2:3b:cb:
        89:13:57:96:0c:79:bb:44:18:ac:79:cf:0a:1b:02:e0:27:f4:
        ...
        98:4d:80:71:d5:3a:0c:98:5a:05:84:b6:4e:75:6d:e1:8a:6f:
        1c:8b:4f:9e:33:1d:4d:7b:1d:9c:b6:4f:6a:7d:73:ec:42:a4:
        2d:c3:8d:c5:5f:96:b8:0c
```

Converteix CRL a format DER (necessari per a alguns servidors)

```
sudo openssl crl \
    -in intermediate-ca/crl/intermediate.crl.pem \
    -outform DER \
    -out intermediate-ca/crl/intermediate.crl
```

9.3. Publicació de la CRL via HTTP

Per publicar la CRL i que els clients la puguin consultar, cal servir-la via HTTP:

Instal·la un servidor web simple

```
sudo apt install nginx
```

Copia la CRL al directori web

```
sudo mkdir /var/www/html/crl
sudo cp intermediate-ca/crl/intermediate.crl.pem /var/www/html/crl/
```

En el openssl.cnf, configura la distribució de CRL:

```
# crlDistributionPoints =
 ↪  URI:http://crl.thos.local/crl/intermediate.crl.pem
```

9.4. Verificació de revocació amb CRL

Verifica un certificat comprovant la CRL

```
sudo openssl verify \
    -crl_check \
    -CAfile intermediate-ca/certs/ca-chain.cert.pem \
    -CRLfile intermediate-ca/crl/intermediate.crl.pem \
    intermediate-ca/certs/www.thos.local.cert.pem
```

10. Verificació i inspecció de certificats

10.1. Ordres d'inspecció essencials

```
# Mostra tota la informació d'un certificat
openssl x509 -text -noout -in certificat.pem

# Mostra el subjecte
openssl x509 -noout -subject -in certificat.pem

# Mostra l'emissor
openssl x509 -noout -issuer -in certificat.pem

# Mostra les dates de validesa
openssl x509 -noout -dates -in certificat.pem

# Mostra l'empremta digital (fingerprint)
openssl x509 -noout -fingerprint -sha256 -in certificat.pem

# Mostra els SAN
openssl x509 -noout -ext subjectAltName -in certificat.pem

# Verifica la coherència entre clau i certificat
openssl x509 -noout -modulus -in certificat.pem | openssl md5
openssl rsa  -noout -modulus -in clau.pem      | openssl md5
# Els dos valors han de ser iguals
```

10.2. Verificació de connexions TLS

```
# Comprova el certificat d'un servidor en producció
openssl s_client \
    -connect www.thos.local:443 \
    -CAfile intermediate-ca/certs/ca-chain.cert.pem

# Mostra la cadena de certificats d'un servidor
openssl s_client \
    -connect www.thos.local:443 \
    -showcerts

# Verifica un servidor LDAPS
openssl s_client \
    -connect ldap.thos.local:636 \
    -CAfile intermediate-ca/certs/ca-chain.cert.pem
```

10.3. Inspecció del fitxer index.txt

```
# Mostra la base de dades de certificats emesos
sudo cat intermediate-ca/index.txt

# Format de cada línia:
# [estat] [data_expiració] [data_revocació] [núm_sèrie] [DN_subjecte]
# V = Vàlid, R = Revocat, E = Expirat
```

11. Opcions de configuració avançades

11.1. Certificats wildcard

```
cat > /tmp/wildcard.cnf << 'EOF'
[ req ]
default_bits       = 2048
distinguished_name = req_distinguished_name
req_extensions     = req_ext
prompt             = no

[ req_distinguished_name ]
C  = ES
ST = Catalunya
O  = IES Thos i Codina
CN = *.thos.local

[ req_ext ]
subjectAltName = @alt_names

[ alt_names ]
DNS.1 = *.thos.local
DNS.2 = thos.local
EOF
```

```
openssl req -new -sha256 \
    -key clau.key.pem \
    -config /tmp/wildcard.cnf \
    -out wildcard.csr.pem
```

11.2. Configuració OCSP (Online Certificate Status Protocol)

L'OCSP és una alternativa moderna a la CRL per comprovar la validesa dels certificats en temps real.

```
# Genera clau i certificat per al respondedor OCSP
sudo openssl genrsa \
    -out intermediate-ca/private/ocsp.key.pem 4096

sudo openssl req \
    -config intermediate-ca/openssl.cnf \
    -new \
    -sha256 \
    -key intermediate-ca/private/ocsp.key.pem \
    -out intermediate-ca/csr/ocsp.csr.pem

sudo openssl ca \
    -config intermediate-ca/openssl.cnf \
    -extensions ocsp \
    -days 375 \
    -notext \
    -md sha256 \
    -in intermediate-ca/csr/ocsp.csr.pem \
    -out intermediate-ca/certs/ocsp.cert.pem

# Inicia el respondedor OCSP
sudo openssl ocsp \
    -port 2560 \
    -text \
    -sha256 \
    -index intermediate-ca/index.txt \
    -CA intermediate-ca/certs/ca-chain.cert.pem \
    -rkey intermediate-ca/private/ocsp.key.pem \
    -rsigner intermediate-ca/certs/ocsp.cert.pem \
    -out /var/log/ocsp.log &

# Consulta l'estat d'un certificat via OCSP
openssl ocsp \
    -CAfile intermediate-ca/certs/ca-chain.cert.pem \
    -url http://ocsp.thos.local:2560 \
    -issuer intermediate-ca/certs/intermediate.cert.pem \
    -cert intermediate-ca/certs/www.thos.local.cert.pem
```

11.3. Configuració de períodes de validesa

Les bones pràctiques actuals recomanen els períodes de validesa següents:

Tipus de certificat	Validesa recomanada
CA arrel	20 anys (7300 dies)
CA intermèdia	10 anys (3650 dies)
Servidor TLS	1 any (365 dies) --- màx. 398 dies per a navegadors
Client VPN	1-2 anys
OCSP	1 any
Codi (code signing)	3 anys

NOTA

Des del 2020, els navegadors web (Chrome, Firefox, Safari) **no accepten certificats TLS de servidor amb una validesa superior a 398 dies**.

11.4. Automatització amb scripts

Script per emetre certificats de servidor

```
#!/bin/bash
# emet-cert.sh --- Emissió automàtica de certificats de servidor
# Ús: ./emet-cert.sh <FQDN> [IP]

set -e

FQDN="${1:?Especifica el FQDN del servidor}"
IP="${2:-}"
CA_DIR="/opt/ca/intermediate-ca"
DAYS=365

# Genera clau
openssl genrsa -out "${CA_DIR}/private/${FQDN}.key.pem" 2048
chmod 400 "${CA_DIR}/private/${FQDN}.key.pem"

# Crea fitxer de configuració temporal
CNF=$(mktemp)
cat > "$CNF" << EOF
[ req ]
default_bits       = 2048
distinguished_name = req_distinguished_name
req_extensions     = req_ext
prompt             = no

[ req_distinguished_name ]
C  = ES
ST = Catalunya
O  = IES Thos i Codina
CN = ${FQDN}
```

```
[ req_ext ]
subjectAltName = @alt_names

[ alt_names ]
DNS.1 = ${FQDN}
$([ -n "$IP" ] && echo "IP.1 = $IP")
EOF

# Genera CSR
openssl req -new -sha256 \
    -key "${CA_DIR}/private/${FQDN}.key.pem" \
    -config "$CNF" \
    -out "${CA_DIR}/csr/${FQDN}.csr.pem"

# Signa
openssl ca \
    -config "${CA_DIR}/openssl.cnf" \
    -extfile "$CNF" \
    -extensions req_ext \
    -days "$DAYS" \
    -notext -md sha256 \
    -in "${CA_DIR}/csr/${FQDN}.csr.pem" \
    -out "${CA_DIR}/certs/${FQDN}.cert.pem"

rm -f "$CNF"

echo "Certificat emès correctament: ${CA_DIR}/certs/${FQDN}.cert.pem"
```

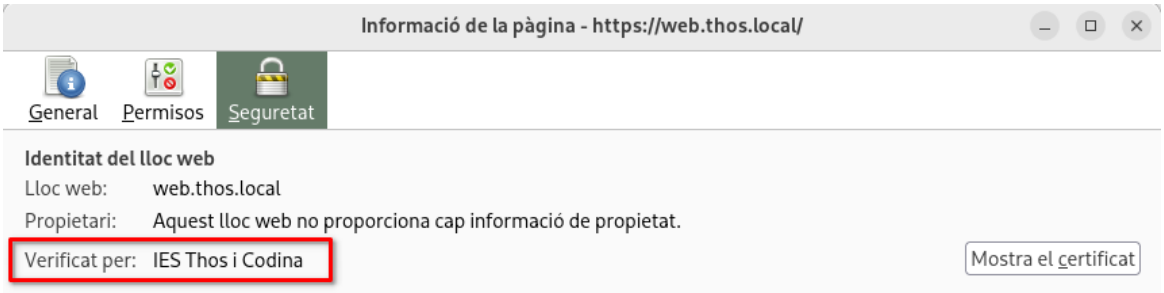

Figura 1: Verificat per: IES Thos i Codina

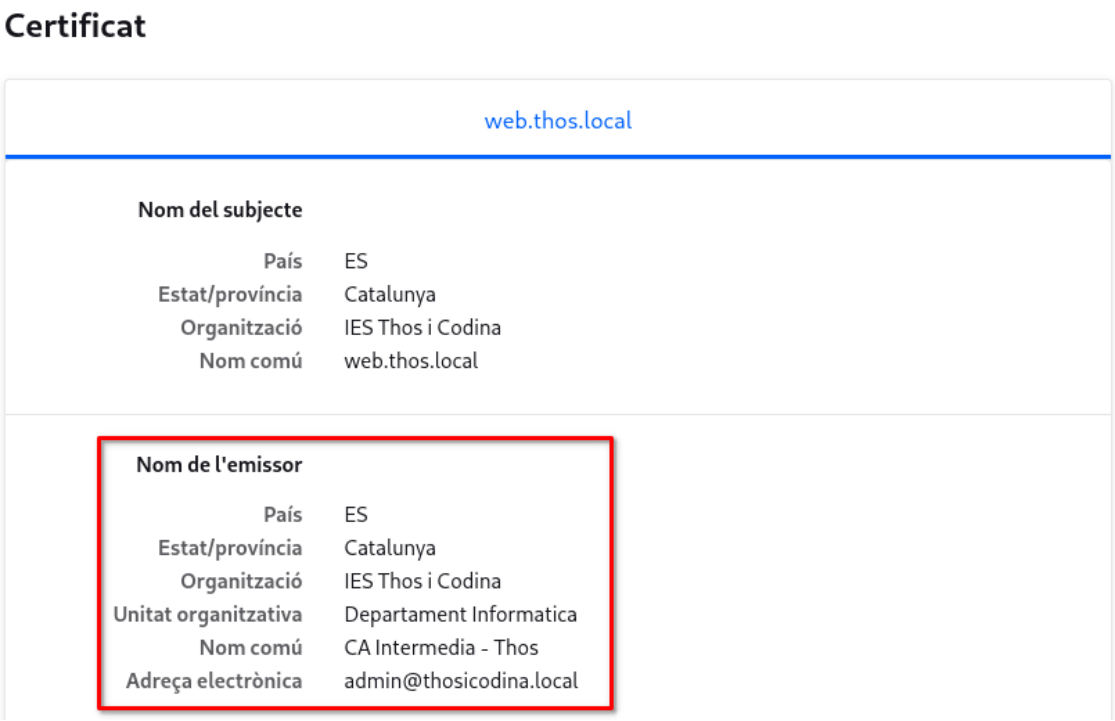

Figura 2: Emissor

11.5. Instal·lació de la CA al sistema Ubuntu

Per tal que el sistema i les aplicacions confiïn en la nostra CA privada:

```
# Copia el certificat de la CA al magatzem del sistema
sudo cp /opt/ca/intermediate-ca/certs/ca-chain.cert.pem \
    /usr/local/share/ca-certificates/ca-privada-thos.crt

# Actualitza el magatzem de CA
sudo update-ca-certificates

# Verifica que s'ha afegit
ls /etc/ssl/certs/ | grep thos

# Comprova amb curl
curl https://www.thos.local
```

11.6. Integració amb serveis del sistema

Apache2

```
SSLEngine on
SSLCertificateFile
 ↪  /opt/ca/intermediate-ca/certs/www.thos.local.cert.pem
SSLCertificateKeyFile
 ↪  /opt/ca/intermediate-ca/private/www.thos.local.key.pem
SSLCACertificateFile  /opt/ca/intermediate-ca/certs/ca-chain.cert.pem

# Verificació de client (opcional)
SSLVerifyClient require
SSLVerifyDepth 2
```

Nginx

```
ssl_certificate
 ↪  /opt/ca/intermediate-ca/certs/www.thos.local.cert.pem;
ssl_certificate_key
 ↪  /opt/ca/intermediate-ca/private/www.thos.local.key.pem;
ssl_trusted_certificate
 ↪  /opt/ca/intermediate-ca/certs/ca-chain.cert.pem;

# Verificació de client (opcional)
ssl_verify_client on;
ssl_client_certificate
 ↪  /opt/ca/intermediate-ca/certs/ca-chain.cert.pem;
```

12. Casos d'ús pràctics

12.1. CA per a un laboratori ASIX

Estructura recomanada per a un laboratori educatiu:

```
/opt/ca/
├── root-ca/                  ← CA arrel (guardada fora de línia)
│   ├── certs/ca.cert.pem
│   ├── private/ca.key.pem    (chmod 400)
│   └── openssl.cnf
└── intermediate-ca/          ← CA operativa del laboratori
    ├── certs/
    │   ├── ca-chain.cert.pem
    │   ├── www.thos.local.cert.pem
    │   ├── ldap.thos.local.cert.pem
    │   └── vpn.thos.local.cert.pem
    ├── crl/intermediate.crl.pem
    ├── private/              (chmod 700)
    └── openssl.cnf
```

12.2. Certificats per a Samba AD DC

```
# Certificat per al controlador de domini Samba
openssl req -new -sha256 \
    -key intermediate-ca/private/samba.key.pem \
    -subj "/C=ES/ST=Catalunya/O=THOS.LOCAL/CN=dc1.thos.local" \
    -out intermediate-ca/csr/samba.csr.pem

# Extensió específica per a Kerberos / LDAPS
cat > /tmp/samba_ext.cnf << 'EOF'
[ samba_cert ]
basicConstraints       = CA:FALSE
keyUsage               = critical, digitalSignature, keyEncipherment
extendedKeyUsage       = serverAuth, clientAuth
subjectAltName         = @alt_names

[ alt_names ]
DNS.1 = dc1.thos.local
DNS.2 = thos.local
IP.1  = 192.168.100.10
EOF
```

12.3. Certificats per a Kea DHCP + ISC Stork

```
# Certificat per a l'agent Stork
openssl req -new -sha256 \
    -key intermediate-ca/private/stork-agent.key.pem \
    -subj "/C=ES/O=Lab/CN=stork-agent.thos.local" \
    -out intermediate-ca/csr/stork-agent.csr.pem

openssl ca \
    -config intermediate-ca/openssl.cnf \
    -extensions usr_cert \
    -days 365 -notext -md sha256 \
    -in intermediate-ca/csr/stork-agent.csr.pem \
    -out intermediate-ca/certs/stork-agent.cert.pem
```

13. Bones pràctiques de seguretat

Protecció de claus privades

- Xifra sempre les claus privades de les CA amb AES-256 i una contrasenya forta
- La clau de la CA arrel ha d'estar fora de línia (idealment en un HSM o dispositiu desconnectat)
- Estableix sempre permisos `400` per a fitxers de claus privades
- Fes còpies de seguretat xifrades de les claus en ubicacions separades

Períodes de validesa

- No emetis mai certificats de servidor amb validesa superior a 398 dies
- Planifica la renovació dels certificats amb antelació (avisos a 30/15/7 dies)
- Estableix recordatoris en el calendari per renovar la CRL periòdicament

Algoritmes criptogràfics

- Usa **SHA-256** o superior per a totes les signatures (mai MD5 ni SHA-1)
- Per a noves instal·lacions, valora **ECDSA P-384** en comptes de RSA 4096
- A Ubuntu 26.04, OpenSSL 3.x incorpora suport per a **criptografia post-quàntica** (experimental)

Auditoria i registre

```
# Monitorar canvis a la base de dades de la CA
sudo auditctl -w /opt/ca/intermediate-ca/index.txt -p wa -k ca_audit
sudo auditctl -w /opt/ca/root-ca/index.txt -p wa -k ca_audit

# Veure els registres d'auditoria
sudo ausearch -k ca_audit
```

Còpies de seguretat

```
# Script de còpia de seguretat de la CA (sense claus en clar)
#!/bin/bash
BACKUP_DIR="/backup/ca-$(date +%Y%m%d)"
mkdir -p "$BACKUP_DIR"

# Copia tota l'estructura (les claus ja estan xifrades)
tar czf "${BACKUP_DIR}/ca-backup.tar.gz" /opt/ca/

# Verifica la còpia
tar tzf "${BACKUP_DIR}/ca-backup.tar.gz" | head -20
```

14. Resum d'ordres

Gestió de claus

Acció	Ordre
Genera clau RSA 4096 xifrada	`openssl genrsa -aes256 -out clau.key.pem 4096`
Genera clau ECDSA P-256	`openssl ecparam -name prime256v1 -genkey -noout -out clau.key.pem`
Verifica clau RSA	`openssl rsa -check -in clau.key.pem`
Treu xifratge de clau	`openssl rsa -in clau.enc.pem -out clau.pem`

Gestió de CSR

Acció	Ordre
Crea CSR interactiu	`openssl req -new -sha256 -key clau.key.pem -out sol.csr.pem`
Crea CSR no interactiu	`openssl req -new -sha256 -key clau.key.pem -subj "/C=ES/O=Org/CN=servidor" -out sol.csr.pem`
Verifica CSR	`openssl req -text -noout -verify -in sol.csr.pem`

Gestió de certificats

Acció	Ordre
Signa CSR amb CA	`openssl ca -config openssl.cnf -days 365 -notext -md sha256 -in sol.csr.pem -out cert.pem`
Crea certificat autosignat	`openssl req -x509 -days 365 -sha256 -key clau.key.pem -out cert.pem`
Verifica certificat	`openssl x509 -text -noout -in cert.pem`
Verifica cadena	`openssl verify -CAfile ca-chain.pem cert.pem`
Revoca certificat	`openssl ca -config openssl.cnf -revoke cert.pem -crl_reason keyCompromise`

Gestió de CRL i OCSP

Acció	Ordre
Generar CRL	`openssl ca -config openssl.cnf -gencrl -out crl.pem`
Verificar CRL	`openssl crl -text -noout -in crl.pem`
Iniciar respondedor OCSP	`openssl ocsp -port 2560 -index index.txt -CA ca-chain.pem -rkey ocsp.key.pem -rsigner ocsp.cert.pem`
Consultar OCSP	`openssl ocsp -url http://ocsp:2560 -CAfile ca-chain.pem -issuer inter.pem -cert cert.pem`

Conversió de formats

Acció	Ordre
PEM → DER	`openssl x509 -in cert.pem -outform DER -out cert.der`
DER → PEM	`openssl x509 -in cert.der -inform DER -out cert.pem`
PEM → PKCS#12	`openssl pkcs12 -export -inkey clau.pem -in cert.pem -certfile ca.pem -out bundle.p12`
PKCS#12 → PEM	`openssl pkcs12 -in bundle.p12 -nodes -out bundle.pem`

Versions d'aquest document

- HTML - ca-privada.html
- PDF - ca-privada.pdf
- ODT - ca-privada.odt
- MD - ca-privada.md