
PKI et Certificats Numériques
Créé par : Rachid Bouselama
Superviseur : Mr Lahcen AIT IBOUREK

PKI et Certificats Numériques

Comprendre que la cryptographie ne sert à rien sans identité

Le Problème de l'Identité

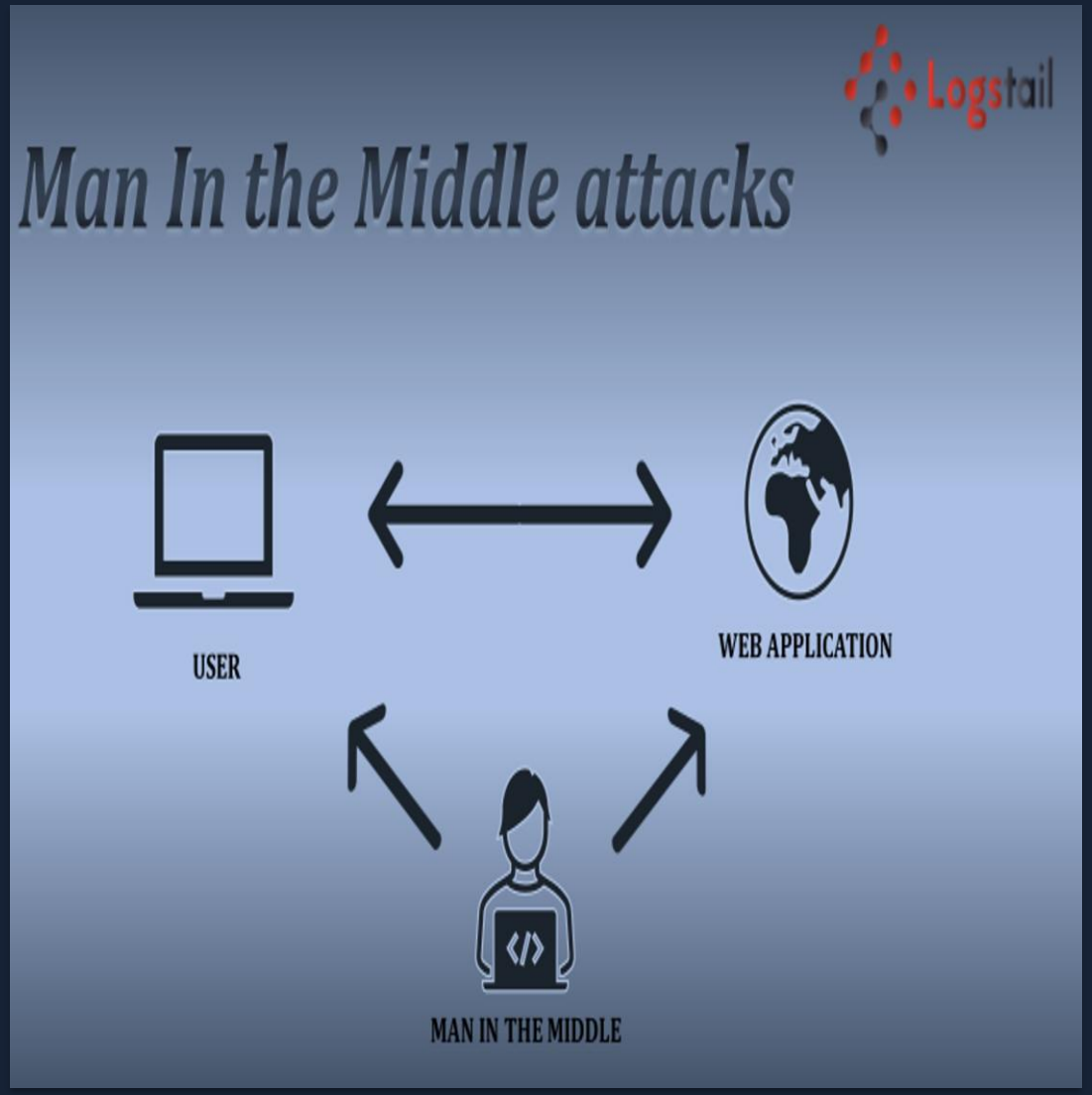
Attaque Man-in-the-Middle (MITM)

1. Alice demande la clé publique de Bob
2. Mallory intercepte et envoie sa propre clé
3. Mallory déchiffre, lit, rechiffre avec la vraie clé de Bob

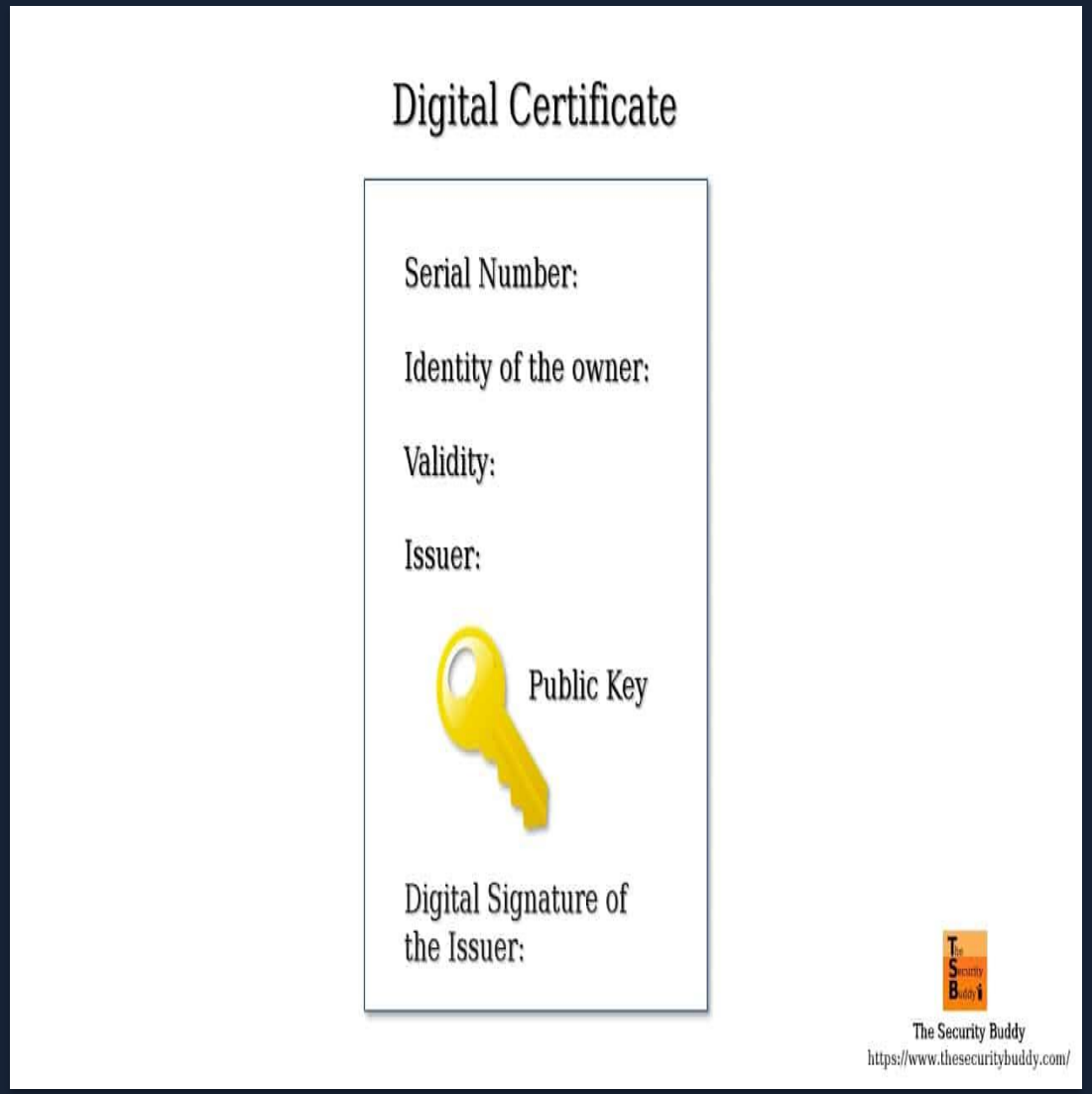
⚠

Une clé publique est une suite de bits anonyme.

Elle ne dit pas "J'appartiens à Bob".



La Solution: Le Certificat Numérique



Analogie: Le Passeport

Données (Nom, Photo) + Signature de l'État

Structure du Certificat X.509

Clé Publique

Comme la photo du passeport

Identité du Sujet

CN = Common Name (ex: www.google.com)

Signature de la CA

Authentification par l'Autorité

Date de Validité

Not Before / Not After

L'Infrastructure à Clés Publiques (PKI)

Ensemble de **Matériel** + **Logiciel** + **Procédures Humaines**

CA

Certificate Authority

- Entité de confiance suprême
- Clé privée très protégée
- Signe les certificats

RA

Registration Authority

- Guichetier de la PKI
- Vérifie les documents d'identité
- Kbis, Carte d'identité

VA

Validation Authority

- Service de validation
- Répond à la question
- Ce certificat est-il valide ?

La Chaîne de Confiance

Hiérarchie des Certificats

Root CA (Racine)

Auto-signée • Sommet de la pyramide

Intermediate CA

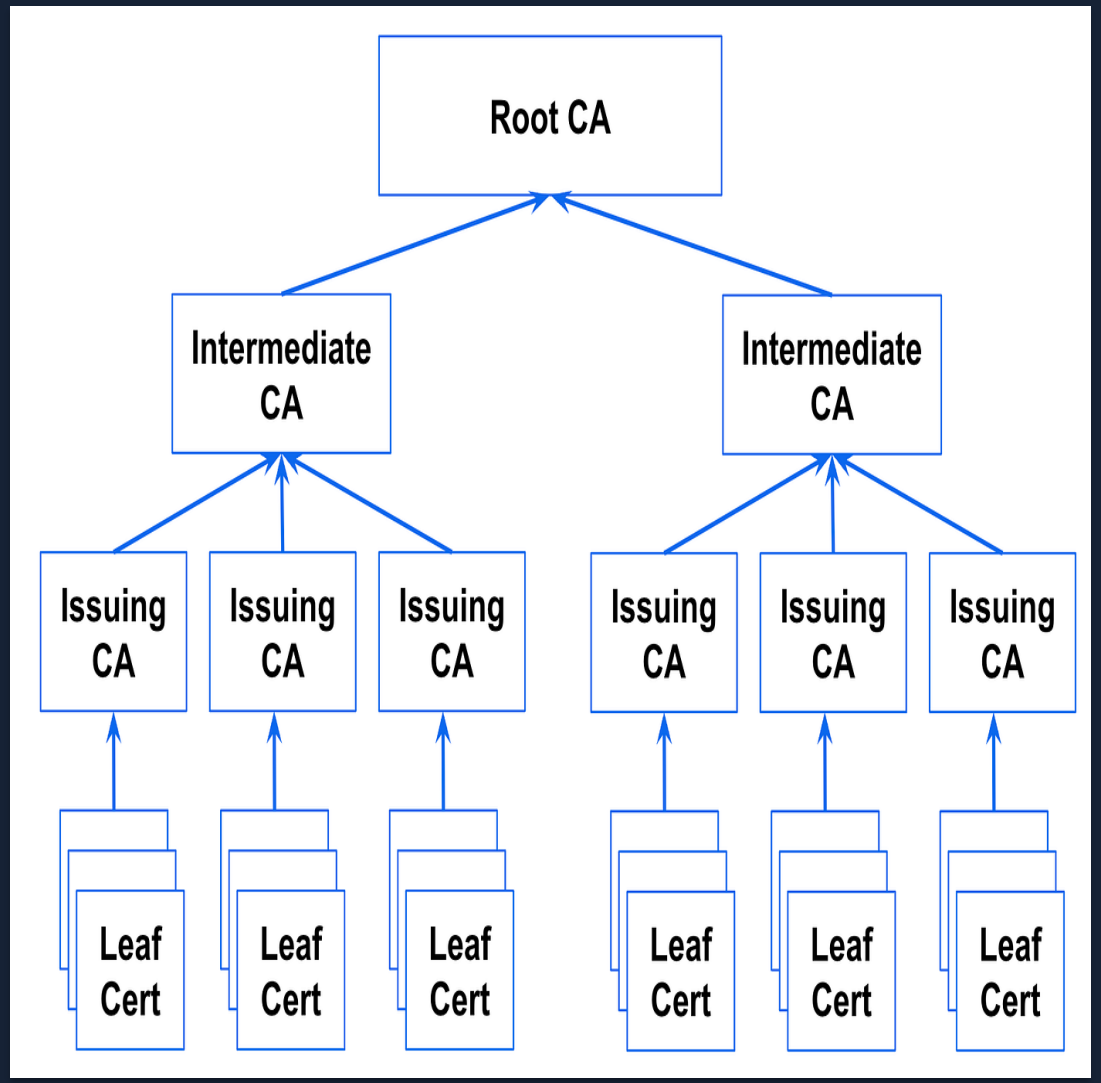
Signée par la Root • Protège la clé Root

End-Entity

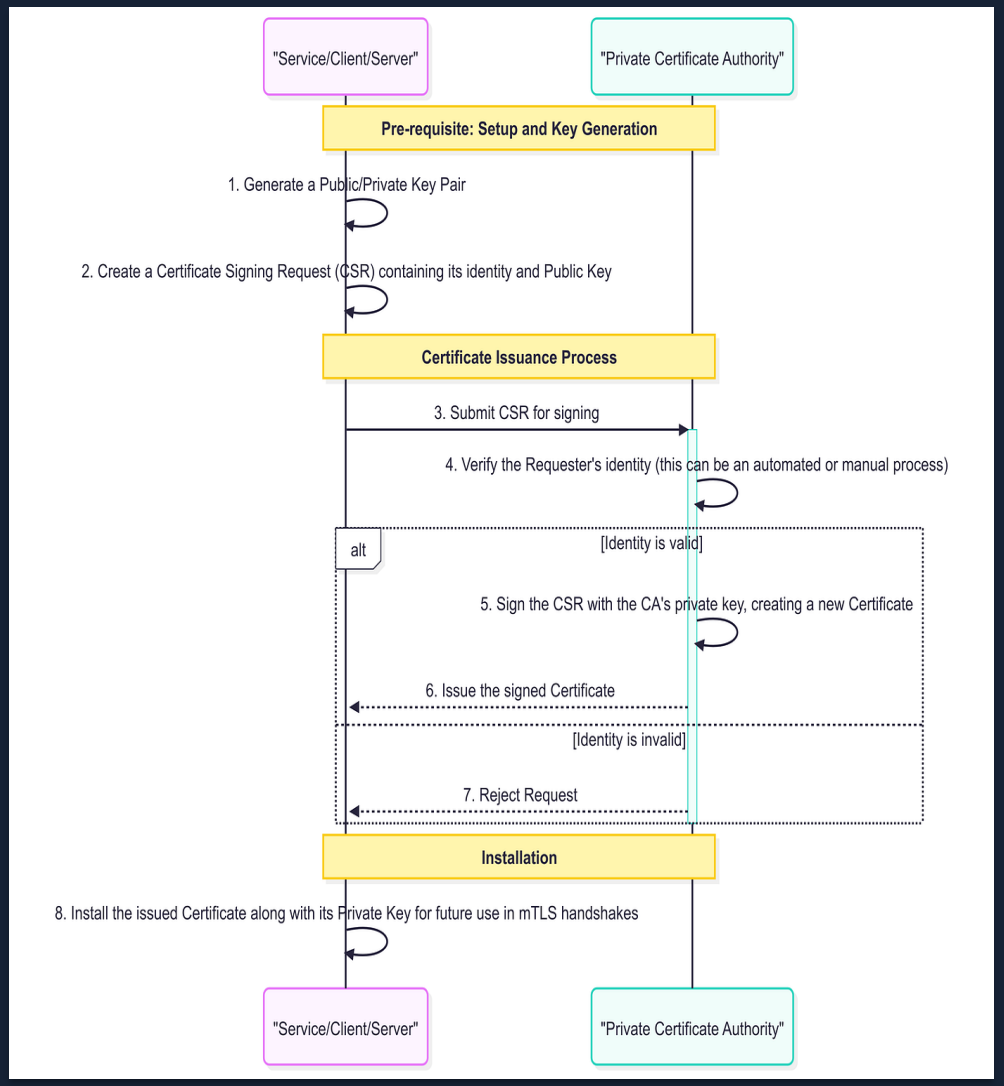
Certificat final • Serveur web, Utilisateur

Trust Store

~100 CAs de confiance intégrées par défaut dans OS et navigateurs



Le Cycle de Vie: Création et Émission



Le Processus CSR

1. Générer la paire de clés sur le serveur
2. Créer le fichier .csr avec clé publique + informations
3. Envoyer le CSR à la CA

🔒Garder la clé privée secrète!

4. CA vérifie, hache les infos et signe avec sa clé privée
5. CA renvoie le fichier .crt (Certificat)

Le Cycle de Vie: Révocation

Pourquoi Révoquer?

Clé privée volée • Certificat compromis

Méthodes de Révocation

CRL

Certificate Revocation List

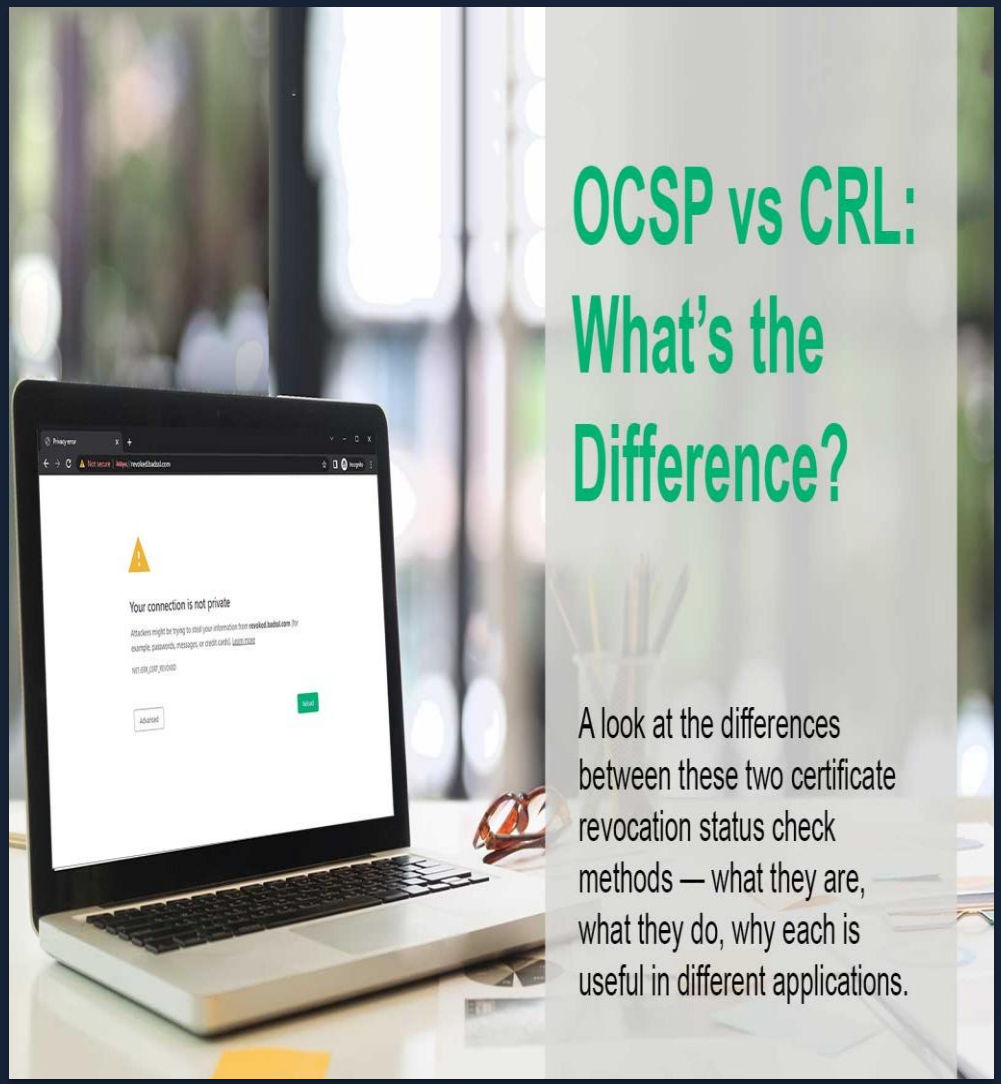
- ✕ Liste noire lourde
- ✕ Mise à jour lente

OCSP

Online Certificate Status Protocol

- ✓ Vérification temps réel
- ✓ Interroge la CA

Le navigateur demande: "Ce numéro de série est-il valide?"



Analyse de la Confiance

Concepts Fondamentaux

Trust Store

Liste des CAs de confiance intégrées dans OS et navigateurs

Vérification Navigateur

Chaîne de confiance visible dans les paramètres de sécurité

CA Personnalisée

Nécessite installation manuelle sur les postes clients

Démonstration visuelle: Hiérarchie des certificats dans un navigateur

Hiérarchie des Certificats

Travaux Pratiques: PKI et Certificats

Atelier Pratique - Construire une Mini-PKI

Durée: 2 heures

Environnement: Linux + OpenSSL

Sécurité et Cryptographie • Semaine 3

Atelier 1: Construire une Mini-PKI

Scénario

RSSI d'une entreprise • Sécuriser intranet.local

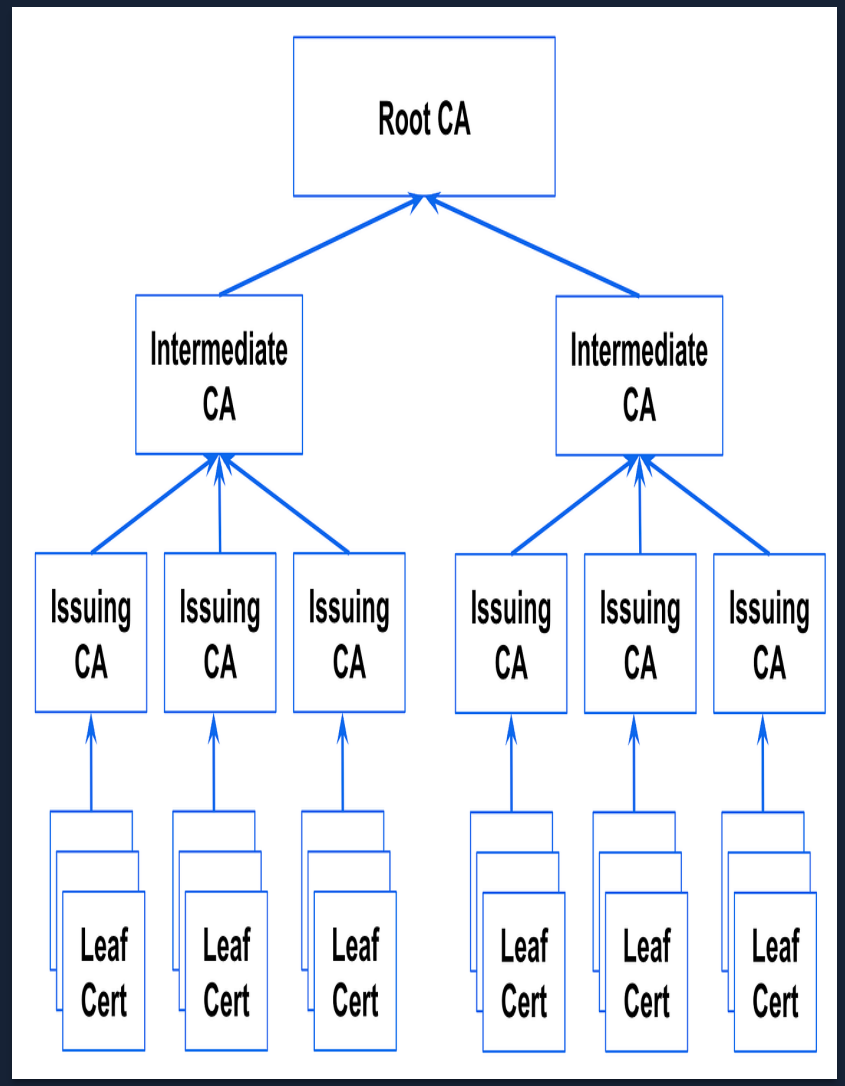
Durée

1h 15

Environnement

Linux + OpenSSL

1. Jouer le rôle de la CA (Autorité de Certification)
2. Jouer le rôle de l'Admin Serveur
3. Créer et signer des certificats



Étape 1: Devenir l'Autorité de Certification

1 **Générer la clé privée de la CA**

```
openssl genrsa -aes256 -out ca.key 4096
```

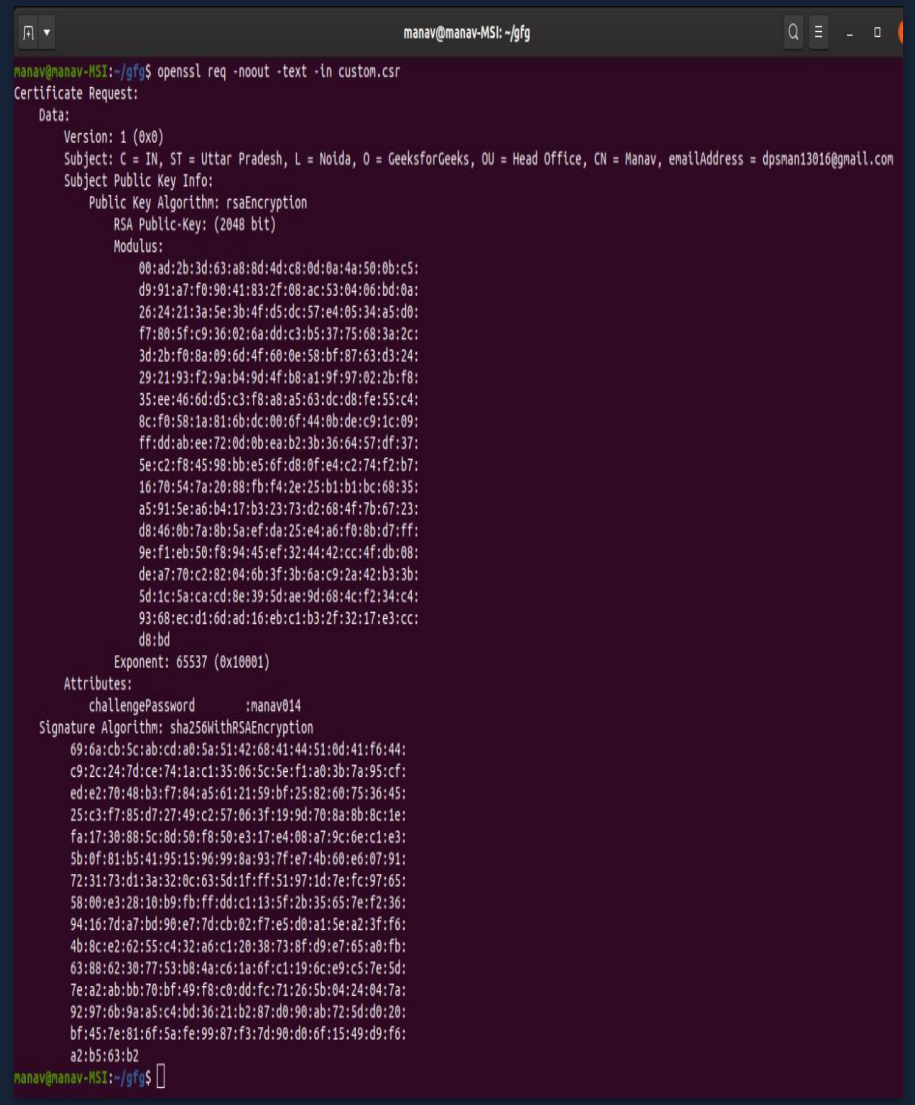
Mot de passe: **ca-password**

2 **Créer le certificat racine auto-signé**

```
openssl req -x509 -new -nodes -key ca.key
-sha256 -days 3650 -out ca.crt
```

Paramètres Importants

- CN = **"Ma Super CA Racine"**
- Validité: **10 ans**
- Algorithme: **SHA-256**



Windows PowerShell

```
root@Debian11-bouselama:~# openssl genrsa -aes256 -out ca.key 4096
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
4057C6F7627F0000:error:14000065:UI routines:UI_set_result_ex:result too small:../crypto/ui/ui_lib.c:899
:You must type in 4 to 1024 characters
4057C6F7627F0000:error:1400006B:UI routines:UI_process:processing error:../crypto/ui/ui_lib.c:552:while
 reading strings
4057C6F7627F0000:error:0480006D:PEM routines:PEM_def_callback:problems getting password:../crypto/pem/p
em_lib.c:62:
4057C6F7627F0000:error:07880109:common libcrypto routines:do_ui_passphrase:interrupted or cancelled:../
crypto/passphrase.c:178:
4057C6F7627F0000:error:1C80009F:Provider routines:p8info_to_encp8:unable to get passphrase:../providers
/implementations/encode_decode/encode_key2any.c:123:
root@Debian11-bouselama:~# openssl genrsa -aes256 -out ca.key 4096
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
root@Debian11-bouselama:~# openssl req -x509 -new -nodes -key ca.key -sha256 -days 3650 -out ca.crt
Enter pass phrase for ca.key:
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) [AU]:FR
State or Province Name (full name) [Some-State]:Ile-de-France
Locality Name (eg, city) []:Paris
Organization Name (eg, company) [Internet Widgits Pty Ltd]:My Company
Organizational Unit Name (eg, section) []:
```

Étape 2: Préparer le Serveur Web

1 **Générer la clé privée du serveur**

```
openssl genrsa -out serveur.key 2048
```

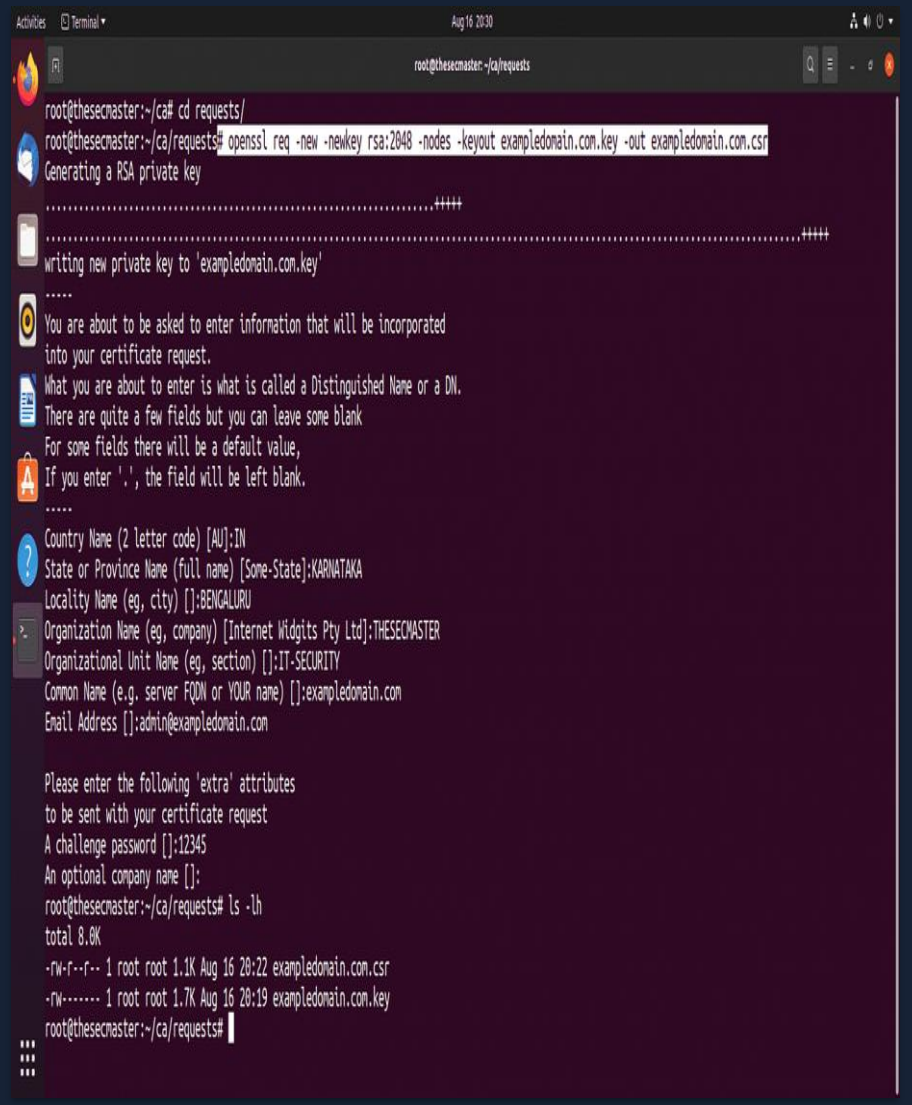
Sans mot de passe (Apache/Nginx démarre seul)

2 **Créer la demande de signature (CSR)**

```
openssl req -new -key serveur.key
-out serveur.csr
```

CRITIQUE

- CN = **"intranet.local"**
- NE PAS mettre de **"Challenge password"**



Windows PowerShell

```
        4b:24:96:b7:33:37:bd:78:58:3e:6c:09:c0:aa:1c:8d:2b:0d:
        4a:43:37:10:ae:dd:bc:8f:3d:83:5e:53:79:43:d4:ff:16:6f:
        df:f4:38:80:c4:84:a0:48
root@Debian11-bouselama:~# openssl genrsa -out serveur-votrenom.key 2048
root@Debian11-bouselama:~# openssl req -new -key serveur-rachid.key -out serveur-rachid.csr
Could not open file or uri for loading private key from serveur-rachid.key: No such file or directory
root@Debian11-bouselama:~# openssl genrsa -out serveur-rachid.key 2048
root@Debian11-bouselama:~# openssl req -new -key serveur-rachid.key -out serveur-rachid.csr
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) [AU]:FR
State or Province Name (full name) [Some-State]:France
Locality Name (eg, city) []:Paris
Organization Name (eg, company) [Internet Widgits Pty Ltd]:My Company
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:intranet.local
Email Address []:admin@mycompany.local

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
root@Debian11-bouselama:~# openssl req -in serveur-votrenom.csr -text -noout
Can't open "serveur-votrenom.csr" for reading, No such file or directory
40C7C5BF7F7F0000:error:80000002:system library:BIO_new_file:No such file or directory:../crypto/bio/bss
```

Étape 3: L'Acte de Certification

La CA signe le certificat

```
openssl x509 -req -in serveur.csr
-CA ca.crt -CAkey ca.key
-CAcreateserial -out serveur.crt
-days 365 -sha256
```

Mot de passe CA demandé ici

Vérification

1

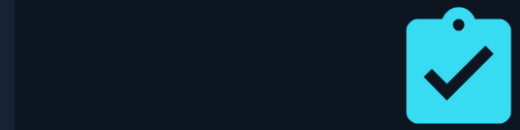
```
openssl x509 -in serveur.crt
-text -noout
```

2

```
openssl verify -CAfile ca.crt
serveur.crt
```

Résultat attendu:

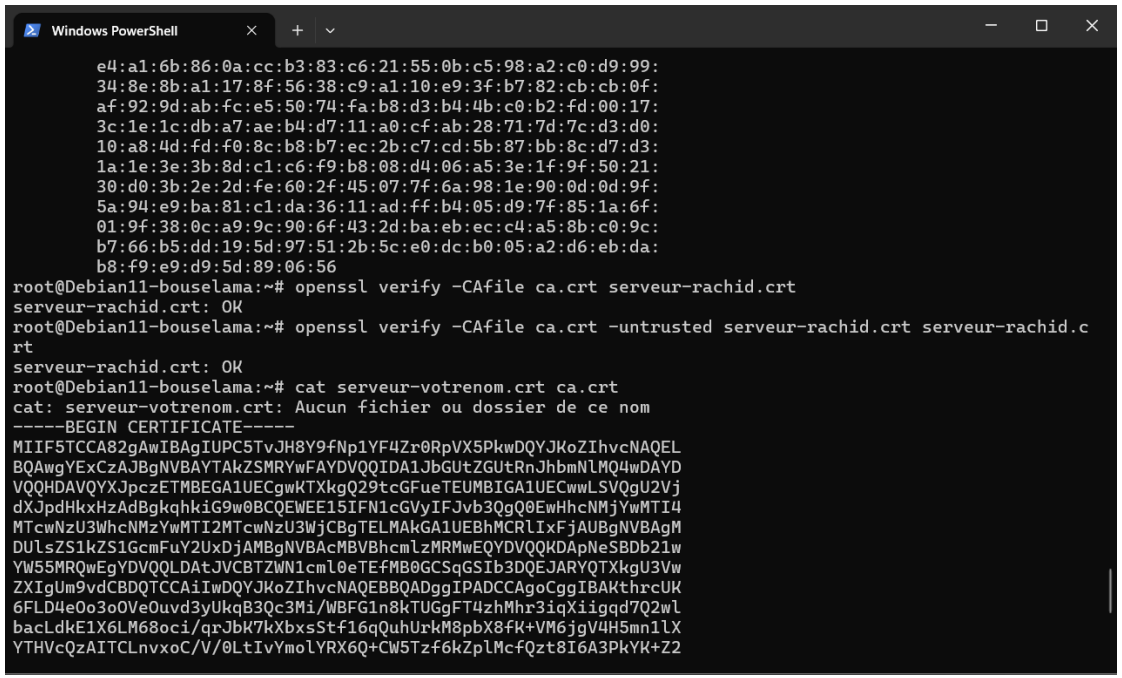
serveur.crt: OK

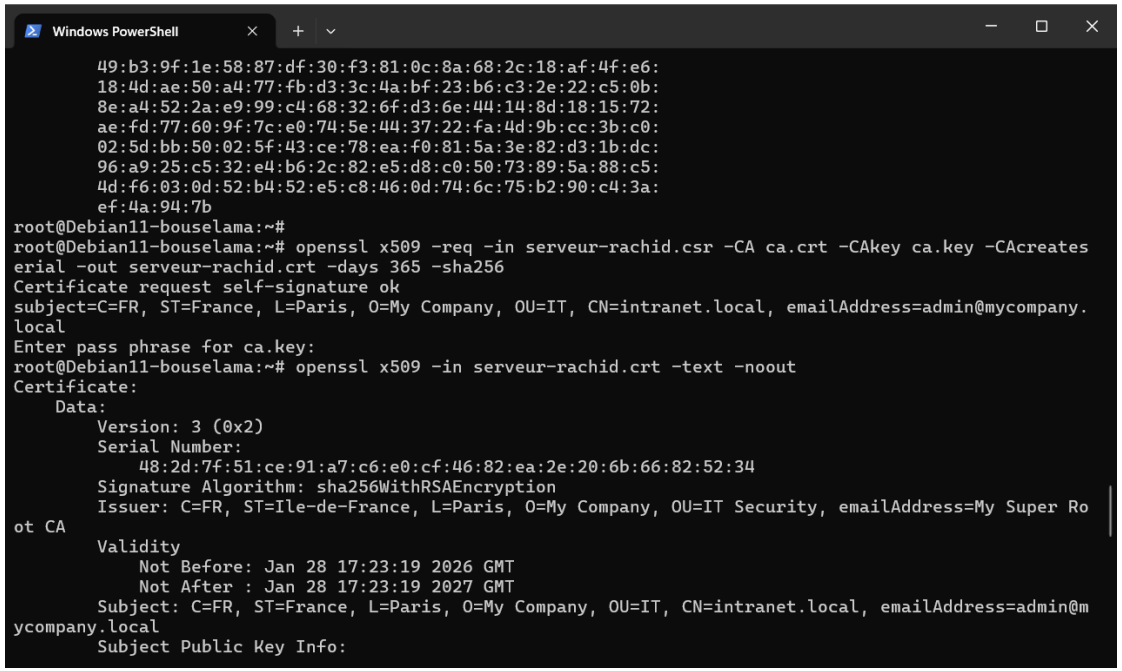


Certificat Signé!

Le serveur web est maintenant

authentifié par la CA


Windows PowerShell
e4:a1:6b:86:0a:cc:b3:83:c6:21:55:0b:c5:98:a2:c0:d9:99:
34:8e:8b:a1:17:8f:56:38:c9:a1:10:e9:3f:b7:82:cb:cb:0f:
af:92:9d:ab:fc:e5:50:74:fa:b8:d3:b4:4b:c0:b2:fd:00:17:
3c:1e:1c:db:a7:ae:b4:d7:11:a0:cf:ab:28:71:7d:7c:d3:d0:
10:a8:4d:fd:f0:8c:b8:b7:ec:2b:c7:cd:5b:87:bb:8c:d7:d3:
1a:1e:3e:3b:8d:c1:c6:f9:b8:08:d4:06:a5:3e:1f:9f:50:21:
30:d0:3b:2e:2d:fe:60:2f:45:07:7f:6a:98:1e:90:0d:0d:9f:
5a:94:e9:ba:81:c1:da:36:11:ad:ff:b4:05:d9:7f:85:1a:6f:
01:9f:38:0c:a9:9c:90:6f:43:2d:ba:eb:ec:c4:a5:8b:c0:9c:
b7:66:b5:dd:19:5d:97:51:2b:5c:e0:dc:b0:05:a2:d6:eb:da:
b8:f9:e9:d9:5d:89:06:56
root@Debian11-bouselama:~# openssl verify -CAfile ca.crt serveur-rachid.crt
serveur-rachid.crt: OK
root@Debian11-bouselama:~# openssl verify -CAfile ca.crt -untrusted serveur-rachid.crt serveur-rachid.crt
serveur-rachid.crt: OK
root@Debian11-bouselama:~# cat serveur-votrenom.crt ca.crt
cat: serveur-votrenom.crt: Aucun fichier ou dossier de ce nom
-----BEGIN CERTIFICATE-----
MIIF5TCCA82gAwIBAgIUPC5TvJH8Y9fNp1YF4Zr0RpVX5PkwDQYJKoZIhvcNAQEL
BQAwgYExCzAJBgNVBAYTAkZSMRYwFAYDVQQIDA1JbGUtZGUtRnJhbmNlMQ4wDAYD
VQQHDAVQYXJpczETMBEGA1UECgwKTXkgQ29tcGFueTEUMBIGA1UECwwLSVQgU2Vj
dXJpdHkxHzAdBgkqhkiG9w0BCQEWEE15IFN1cGVyIFJvb3QgQ0EwHhcNMjYwMTI4
MTcwNzU3WhcNMzYwMTI2MTcwNzU3WjCBgTELMAkGA1UEBhMCRlIxFjAUBgNVBAgM
DUlsZS1kZS1GcmFuY2UxDjAMBgNVBAcMBVBhcmlzMRMwEQYDVQQKDApNeSBDb21w
YW55MRQwEgYDVQQLDAtJVCBTZWN1cml0eTEfMB0GCSqGSIb3DQEJARYQTXkgU3Vw
ZXIgUm9vdCBDQTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBAKthrcUK
6FLD4eOo3oOVeOuvd3yUkqB3Qc3Mi/WBFG1n8kTUGgFT4zhMhr3iqXiigqd7Q2wl
bacLdkE1X6LM68oci/qrJbK7kXbxsStf16qQuhUrkM8pbX8fK+VM6jgV4H5mn1lX
YTHVcQzAITCLnvxoC/V/0LtIvYmolYRX6Q+CW5Tzf6kZplMcfQzt8I6A3PkYK+Z2


Windows PowerShell
49:b3:9f:1e:58:87:df:30:f3:81:0c:8a:68:2c:18:af:4f:e6:
18:4d:ae:50:a4:77:fb:d3:3c:4a:bf:23:b6:c3:2e:22:c5:0b:
8e:a4:52:2a:e9:99:c4:68:32:6f:d3:6e:44:14:8d:18:15:72:
ae:fd:77:60:9f:7c:e0:74:5e:44:37:22:fa:4d:9b:cc:3b:c0:
02:5d:bb:50:02:5f:43:ce:78:ea:f0:81:5a:3e:82:d3:1b:dc:
96:a9:25:c5:32:e4:b6:2c:82:e5:d8:c0:50:73:89:5a:88:c5:
4d:f6:03:0d:52:b4:52:e5:c8:46:0d:74:6c:75:b2:90:c4:3a:
ef:4a:94:7b
root@Debian11-bouselama:~#
root@Debian11-bouselama:~# openssl x509 -req -in serveur-rachid.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out serveur-rachid.crt -days 365 -sha256
Certificate request self-signature ok
subject=C=FR, ST=France, L=Paris, O=My Company, OU=IT, CN=intranet.local, emailAddress=admin@mycompany.local
Enter pass phrase for ca.key:
root@Debian11-bouselama:~# openssl x509 -in serveur-rachid.crt -text -noout
Certificate:
Data:
Version: 3 (0x2)
Serial Number:
48:2d:7f:51:ce:91:a7:c6:e0:cf:46:82:ea:2e:20:6b:66:82:52:34
Signature Algorithm: sha256WithRSAEncryption
Issuer: C=FR, ST=Ile-de-France, L=Paris, O=My Company, OU=IT Security, emailAddress=My Super Root CA
Validity
Not Before: Jan 28 17:23:19 2026 GMT
Not After : Jan 28 17:23:19 2027 GMT
Subject: C=FR, ST=France, L=Paris, O=My Company, OU=IT, CN=intranet.local, emailAddress=admin@mycompany.local
Subject Public Key Info:

Atelier 2: Analyse de Certificats Réels

45 min

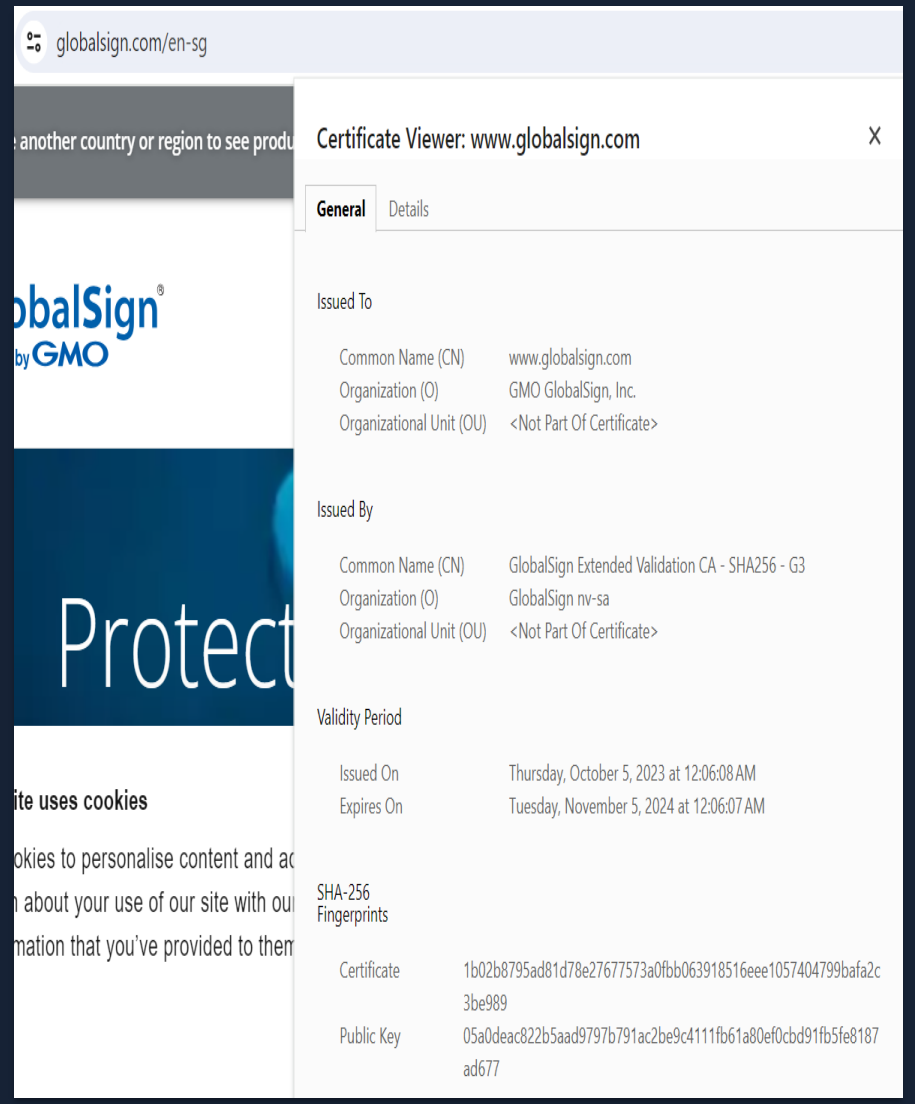
Étapes de l'analyse

1. Ouvrir un navigateur (Firefox ou Chrome)
2. Aller sur un site bancaire ou google.com
3. Clic cadenas > Connexion sécurisée > Afficher le certificat
4. Questions pour l'étudiant

Questions à explorer

- → Qui a émis le certificat (Issuer) ?
- → Quelle est la date d'expiration ?
- → Allez dans l'onglet Hiérarchie (ou Chemin d'accès)

Observer la chaîne: **Root CA → Intermediate CA → Leaf (google.com)**



Certificate Viewer: www.google.com

General | Details

Issued To

Common Name (CN)	www.google.com
Organization (O)	<Not Part Of Certificate>
Organizational Unit (OU)	<Not Part Of Certificate>

Issued By

Common Name (CN)	WR2
Organization (O)	Google Trust Services
Organizational Unit (OU)	<Not Part Of Certificate>

Validity Period

Issued On	Monday, December 29, 2025 at 8:53:09 PM
Expires On	Monday, March 23, 2026 at 8:53:08 PM

SHA-256 Fingerprints

Certificate	519352c62f1871ec3a8f29e17486ff6ca2ec1958652a9ca2dfe5866ae96a737e
Public Key	af9f02b0cbd9cef8b64e6c9d4910169647bcb0011fb4dcc71ed9782de945d2d3

Analyse Wireshark: Capture HTTPS

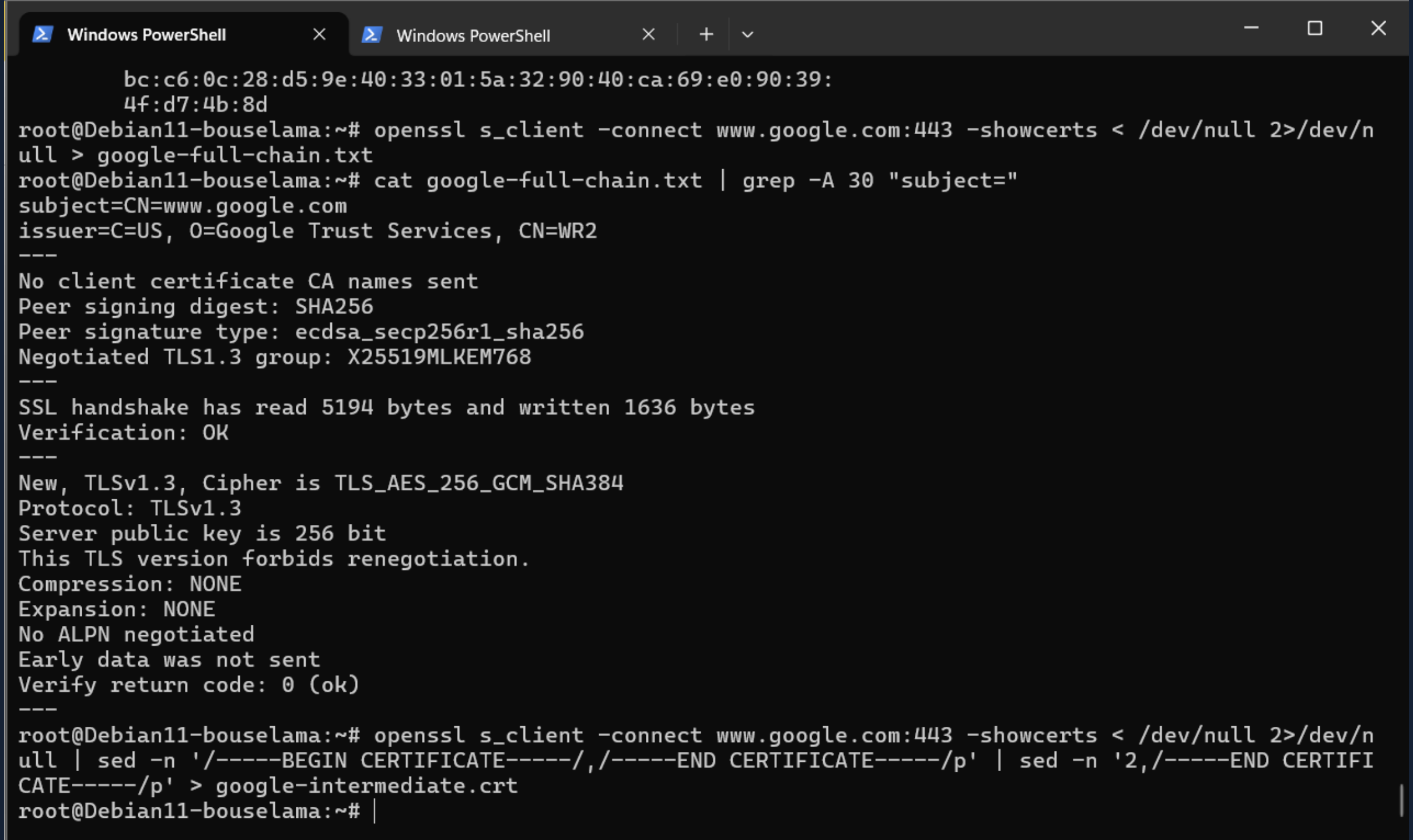
Optionnel

Étapes de l'analyse

1. Lancer Wireshark, démarrer la capture
2. Aller sur un site en HTTPS (ex: curl https://www.wikipedia.org)
3. Arrêter la capture, filtrer sur **"tls"**
4. Trouver le paquet **"Server Hello, Certificate"**
5. Déplier **TLS > Handshake Protocol > Certificates**

Point clé

Le serveur n'envoie pas un seul certificat, mais **toute la chaîne** (sauf le Root) pour aider le navigateur à faire le lien


Windows PowerShell
Windows PowerShell
bc:c6:0c:28:d5:9e:40:33:01:5a:32:90:40:ca:69:e0:90:39:
4f:d7:4b:8d
root@Debian11-bouselama:~# openssl s_client -connect www.google.com:443 -showcerts < /dev/null 2>/dev/null > google-full-chain.txt
root@Debian11-bouselama:~# cat google-full-chain.txt | grep -A 30 "subject="
subject=CN=www.google.com
issuer=C=US, O=Google Trust Services, CN=WR2

No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: ecdsa_secp256r1_sha256
Negotiated TLS1.3 group: X25519MLKEM768

SSL handshake has read 5194 bytes and written 1636 bytes
Verification: OK

New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Protocol: TLSv1.3
Server public key is 256 bit
This TLS version forbids renegotiation.
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 0 (ok)
---
root@Debian11-bouselama:~# openssl s_client -connect www.google.com:443 -showcerts < /dev/null 2>/dev/null | sed -n '/-----BEGIN CERTIFICATE-----/,/-----END CERTIFICATE-----/p' | sed -n '2,/-----END CERTIFICATE-----/p' > google-intermediate.crt
root@Debian11-bouselama:~#

Devoir et Points Clés

Devoir pour la semaine prochaine

1. Installer un serveur web simple (Apache ou Nginx) sur votre VM
2. Utiliser les certificats générés aujourd'hui pour activer le HTTPS
3. Chercher la différence entre TLS 1.2 et TLS 1.3

Astuce

Préparez-vous à démontrer le HTTPS en classe la semaine prochaine!

Points Clés à retenir

- **ca.key** = Secret absolu de la banque (NE JAMAIS PARTAGER)
- **serveur.key** = Secret du serveur web
- **serveur.csr** = Formulaire de demande
- **serveur.crt** = Le diplôme tamponné

Analogie

genrsa	= Naissance (ADN unique)
req (csr)	= Remplir formulaire à la mairie
x509 (sign)	= Tampon préfet sur passeport