ANNEXE - Apache2 // CA Local + TLS

Structure des dossiers d’Apache

Dossier (Chemin : /etc/apache2/)	Rôle
sites-available/	L'Entrepôt / Le Stock. C'est ici que l'on crée et stocke les fichiers de configuration des VirtualHosts (.conf). Ils sont présents, mais inactifs. Apache les ignore totalement.
sites-enabled/	La Production / Le Live. C'est ici que se trouvent les sites actifs. Apache ne lit QUE ce dossier au démarrage. Il contient des Liens Symboliques pointant vers sites-available.
mods-available/ & mods-enabled/	Même logique, mais pour les Modules (Fonctionnalités du serveur : SSL, PHP, Rewrite, etc.).
apache2.conf	Le fichier de configuration global. Il définit les règles universelles (Log, Timeout, Sécurité racine).

Le mécanisme d’activation Available / Enabled

C'est la beauté du système. On ne "copie" pas les fichiers. On crée un **Lien Symbolique** (un raccourci).

- **La Commande Magique :** `a2ensite` (Apache2 ENable SITE).
- **L'Inverse :** `a2dissite` (Apache2 DIsable SITE).

Le flux de travail (Workflow)

1. **Création :** On crée monsupersite.conf dans sites-available.
2. **Activation :** On tape a2ensite monsupersite.conf.
    - *Action système :* Linux crée un lien symbolique dans sites-enabled -> sites-available/monsite.conf.
3. **Application :** On recharge Apache (systemctl reload apache2).

Et voilà, votre site est accessible !

Quelques Best Practices !

Règle #1 : Ne jamais éditer dans sites-enabled

Si vous modifiez le fichier dans sites-enabled et que vous le supprimez par erreur, vous perdez votre configuration.

La bonne voie : On édite toujours dans sites-available. Le lien symbolique reflétera instantanément les changements.

Règle #2 : Nommage des fichiers

Un fichier de conf doit finir par .conf. Sinon, a2ensite risque de l'ignorer.

- ✕ mon-site-web
- ✓ mon-site-web.conf

Règle #3 : Le Test avant le Reload

Avant de redémarrer Apache (et potentiellement de planter tout le serveur de production parce que vous avez oublié un ;), on teste la syntaxe.

```
apache2ctl configtest
```

Si ça répond **"Syntax OK"**, vous pouvez recharger. Sinon, il vous dira exactement à quelle ligne vous avez échoué.

Règle #4 : Désactiver le défaut en Prod

Le fichier 000-default.conf est pratique pour tester "It Works!", mais en production, on le désactive pour ne pas exposer la structure du serveur par défaut.

```
a2dissite 000-default.conf
```

Installation d’Apache

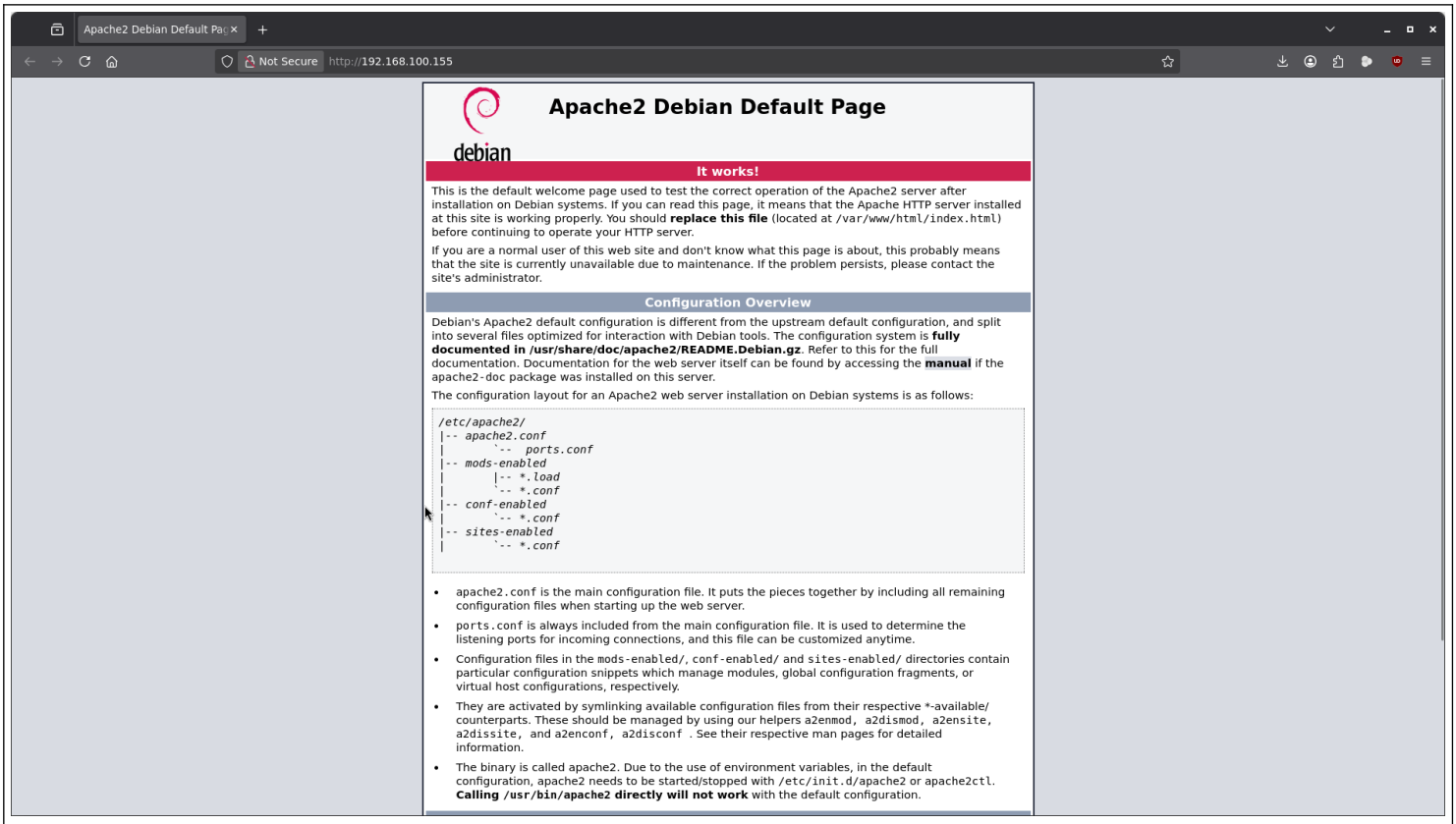
Mise à jour des dépôts && Installation du paquet Apache2

```
admin@lab-debian-23:~$ sudo apt update && apt install apache2
[sudo] password for admin:
Hit:1 http://deb.debian.org/debian trixie InRelease
Hit:2 http://security.debian.org/debian-security trixie-security InRelease
Hit:3 http://deb.debian.org/debian trixie-updates InRelease
All packages are up to date.
Installing:
  apache2

Installing dependencies:
  apache2-bin   apache2-utils  libaprutil1-dbd-sqlite3  libaprutil1t64  libldap-common  liblua5.4-0   librtmp1
libsasl2-modules     libssh2-1t64
  apache2-data  libapr1t64     libaprutil1-ldap         libcurl4t64     libldap2        libnghttp3-9  libsasl2-
2  libsasl2-modules-db  ssl-cert

Suggested packages:
  apache2-doc             | apache2-suexec-custom  www-browser                  | libsasl2-modules-gssapi-
heimdal  libsasl2-modules-otp
  apache2-suexec-pristine  ufw                     libsasl2-modules-gssapi-mit  libsasl2-modules-ldap
libsasl2-modules-sql

Summary:
  Upgrading: 0, Installing: 19, Removing: 0, Not Upgrading: 0
  Download size: 3,529 kB
  Space needed: 11.6 MB / 22.4 GB available

Continue? [Y/n]
```

Vérification du statut (Est-il vivant ?)

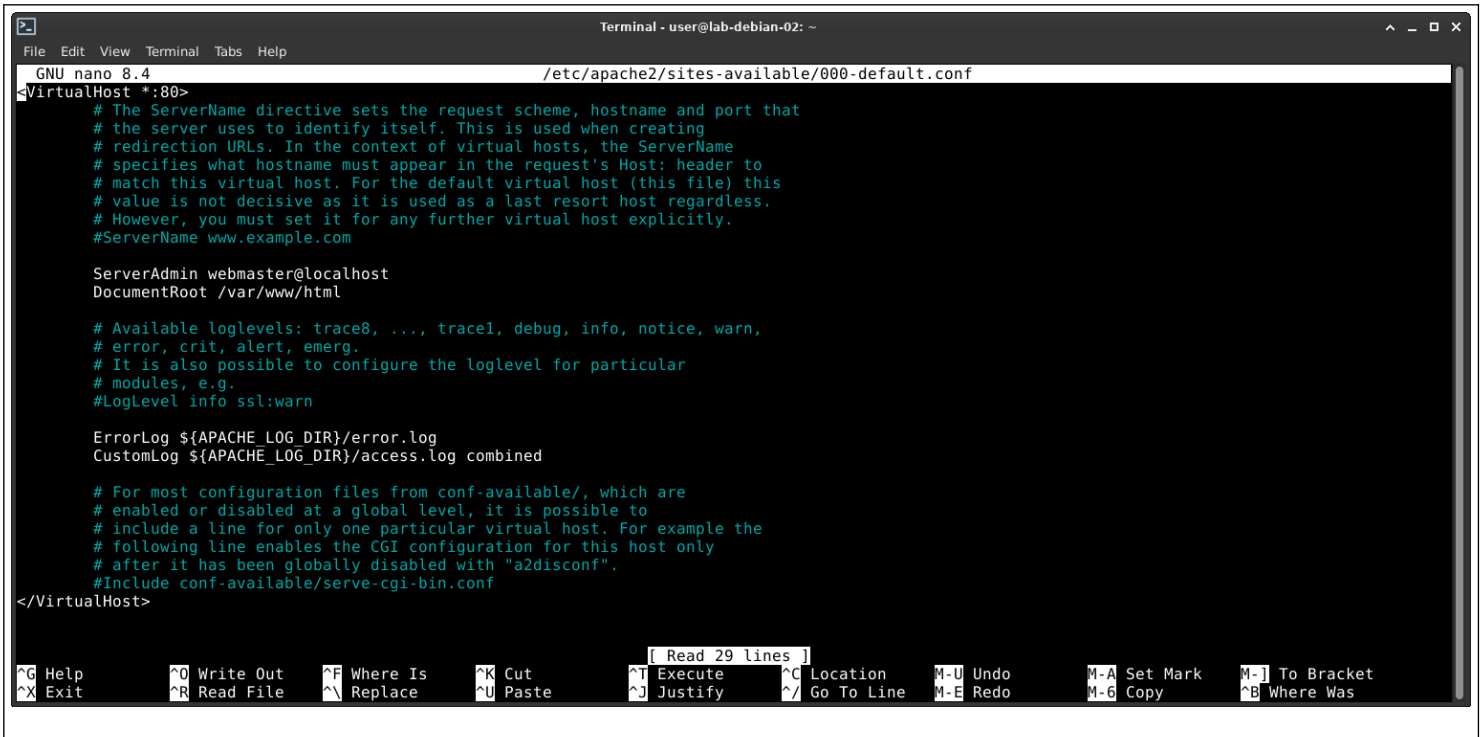
```
admin@lab-debian-23:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Fri 2026-01-16 14:10:04 CET; 1min 27s ago
 Invocation: 4fb9b7d464c54dcd895bbf3a6d2bd6ee
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 1501 (apache2)
      Tasks: 55 (limit: 4655)
     Memory: 5.7M (peak: 6.8M)
        CPU: 38ms
     CGroup: /system.slice/apache2.service
             ├─1501 /usr/sbin/apache2 -k start
             ├─1503 /usr/sbin/apache2 -k start
             └─1505 /usr/sbin/apache2 -k start

Jan 16 14:10:04 lab-debian-02 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jan 16 14:10:04 lab-debian-02 apachectl[1500]: AH00558: apache2: Could not reliably determine the server's
fully qualified domain name, using 127.0.1>
Jan 16 14:10:04 lab-debian-02 systemd[1]: Started apache2.service - The Apache HTTP Server.
```

Autre vérification directement avec le navigateur



Le responsable de l'affichage de cette pages par default d'Apache est le Virtual Host par default d'Apache !

Il se trouve ici : /etc/apache2/sites-available/000-default.conf



Le fichier 000-default.conf est un "Catch-All". Il attrape toutes les requêtes HTTP (Port 80) qui arrivent sur le serveur et qui ne correspondent à aucun autre `ServerName` spécifique. C'est le filet de sécurité par défaut.

Le fichier index.html quant à lui est placé dans le répertoire suivant :

```
admin@lab-debian-23:~$ ls -l /var/www/html/
total 12
-rw-r--r-- 1 root root 10703 Jan 16 14:10 index.html
```

Création de l’autorité de certifications (CA)

Structure que j’utilise

- my-pki/ : Racine du projet.
- my-pki/private/ : **TOP SECRET**. Contient les clés .key.
- my-pki/certs/ : **PUBLIC**. Contient les certificats .crt.
- my-pki/conf/ : Pour vos fichiers de configuration (.conf, .cnf).

Public Key Infrastructure

Création de la structure dans le dossier /etc

```
admin@lab-debian-23:~$ sudo mkdir -p /etc/my-pki/{private,certs,conf}
admin@lab-debian-23:~$ ls -l /etc/my-pki/
total 12
drwxrwxr-x 2 root root 4096 Jan 16 14:49 certs
drwxrwxr-x 2 root root 4096 Jan 16 14:49 conf
drwxrwxr-x 2 root root 4096 Jan 16 14:49 private
```

Sécurisation IMMEDIATE du dossier privé # Seul l'utilisateur propriétaire a le droit d'entrer. Les autres : "You shall not pass!"

```
admin@lab-debian-23:~$ sudo chmod 700 /etc//my-pki/private
admin@lab-debian-23:~$ ls -l /etc/my-pki/
total 12
drwxrwxr-x 2 root root 4096 Jan 16 14:49 certs
drwxrwxr-x 2 root root 4096 Jan 16 14:49 conf
drwx------ 2 root root 4096 Jan 16 14:49 private
```

Génération de l'Autorité (Root CA)

Nous allons maintenant créer l'entité qui aura droit de vie ou de mort (numérique) sur vos futurs certificats.

```
admin@lab-debian-23:~$ cd /etc/my-pki/
```

La Commande de Création

```
# Génération de la CA (Clé + Certificat)
admin@lab-debian-23:/etc/my-pki$ sudo openssl req -new -x509 -extensions v3_ca -nodes -keyout private/ca.key
-out certs/ca.crt

# req -new : Nouvelle demande
# -x509 : On veut un certificat final (pas juste une demande), car c'est la CA racine
# -nodes : Pas de mot de passe sur la clé (Pratique pour le lab, Risqué pour la Prod)
# -keyout : Où ranger la clé privée (DANS LE DOSSIER PRIVATE !)
# -out : Où ranger le certificat public (DANS LE DOSSIER CERTS !)
```

Le Formulaire d'Identité

OpenSSL va vous poser des questions. Comme c'est la CA, soyez solennel.

- **Country Name :** FR
- **Common Name (CN) :** C'est le plus important. Mettez quelque chose comme Ma Super CA Root ou Labo CA. **Ne mettez PAS** le nom de votre site web ici (sinon conflit garanti plus tard).

```
.........+++++++++++++++++++++++++++++++++++++++*...+...+..+...+++++++++++++++++++++++++++++++++++++++*....+...
+..........+.....+............+.+..+.......+........+...........+..+.......+....+...+...+......+......+...+.....
+....+..+......+..........+........+.........+............+......+....+......+............+..+..........
+.........+......+......+.....+....+.....+.+............+.....+.......+...........+...+....+...+...+..+....+...
+...........+.........+..........+...+.........+.....+.+...+...........+...+...+.........+...+.+......+........
+.......+...+..+.+.................+...+...+.......+.....+.......+..+.+.....+.........+.+...+..............+.
+......+...+...++++++
.......+......+.........+.........+.......++++++++++++++++++++++++++++++++++++++++*........+...+........+.+.....
+.+..+.+.....................+......++++++++++++++++++++++++++++++++++++++++*........+.............................
+...+.+..+.........+...+...+.......+......+.....+....+..+.+.........+........+......+.+...+.....+.
+..............+..............+.+..............+..........+...+.....+............+.+......+...+........
+.......+.....+.+.....+....+..+....+......+.....+...+..........+.....+...+.........+.........+.......+.....
+....+.....+...+......+....+...+.........+......+.++++++
-----
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                              <-- FR pour France
State or Province Name (full name) [Some-State]:Ariège            <-- Le département de votre CA
Locality Name (eg, city) []:Auzat                                 <-- La ville de votre CA
Organization Name (eg, company) [Internet Widgits Pty Ltd]:MyLocalPKI     <-- Le nom de votre structure globale
Organizational Unit Name (eg, section) []:IT Security             <-- Le service responsable
Common Name (e.g. server FQDN or YOUR name) []:MyLocalPKI Root CA       <--  Le nom de votre certificat
Email Address []:admin@mylocalpki.fr                              <-- L'adresse mail du responsable
```

Vérification

```
admin@lab-debian-23:/etc/my-pki$ ls -l *
certs:
total 4
-rw-rw-r-- 1 root root 1395 Jan 16 14:56 ca.crt

conf:
total 0

private:
total 4
-rw------- 1 root root 1704 Jan 16 14:53 ca.key
```

La Demande (CSR)

Maintenant, nous changeons de casquette. Nous ne sommes plus l'Autorité, nous sommes le **Webmaster** qui supplie pour avoir un certificat valide !

Le Fichier de Configuration (Le "Blueprint")

Pour éviter de retaper toutes les infos à chaque fois (et surtout pour ajouter les extensions modernes comme le SAN - Subject Alternative Name), nous allons créer un fichier de "recette".

```
admin@lab-debian-23:/etc/my-pki$ sudo nano conf/httpd-config-tls.conf
```

```
[req]
default_bits       = 2048      <-- LA FORCE BRUTE. 2048 bits est le standard actuel, + for Paranoïac Mode.
prompt             = no         <-- LE SILENCE. Empêche OpenSSL de poser des questions interactives.
default_md         = sha256    <-- L'EMPREINTE. L'algorithme de hachage.
distinguished_name = dn       <-- LE POINTEUR D'IDENTITÉ. OpenSSL va chercher les infos dans la section [dn].
req_extensions     = req_ext   <-- LE POINTEUR D'EXTENSIONS. OpenSSL va chercher les options techniques (SAN)
                                   dans [req_ext].
[dn]
C  = FR                <-- COUNTRY, Le pays (en 2 lettres).
ST = Ariège             <-- STATE, Le département ou la région.
L  = Auzat             <-- LOCALITY, La ville.
O  = Local Unsafe Lab    <-- ORGANIZATION, Le nom de votre structure.
CN = lab-unsafe.lan      <-- COMMON NAME, Historiquement, c'est le nom principal du serveur.
emailAddress = admin@lab-unsafe.fr    <-- EMAIL, Pour vous contacter quand le certificat expire ou que tout est
                                          planté.
[req_ext]
subjectAltName = @alt_names  <-- LA REDIRECTION, Dit simplement : "La liste des noms autorisés est dans la
                                 rubrique [alt_names]".
[alt_names]
DNS.1 = lab-unsafe.lan         <-- Le domaine principal que les gens taperont.
DNS.2 = www.lab-unsafe.lan    <-- Le sous-domaine www.
IP.1  = 192.168.100.155        <-- L'IP du server (Optionnel mais pratique).
```

Sauvegardez (Ctrl+O) et Quittez (Ctrl+X).

Vérification

```
admin@lab-debian-23:/etc/my-pki$ ls -l *
certs:
total 8
-rw-rw-r-- 1 root root 1489 Jan 16 15:09 ca.crt

conf:
total 4
-rw-rw-r-- 1 root root 378 Jan 16 15:40 httpd-config-tls.conf

private:
total 8
-rw------- 1 root root 1704 Jan 16 15:08 ca.key
```

Génération de la Clé et de la Demande

```
admin@lab-debian-23:/etc/my-pki$ sudo openssl req -new -nodes -out certs/httpd-tls.csr -keyout private/httpd-
tls.key -config conf/httpd-config-tls.conf
..+...+..+.......+.....+...+.+............+...+......+.............+......+...++++++++++++++++++++++++++++++++
+++++++*....+...+..+......+.......+...+........+....+..+...+.......+.....+...+...+.............+.+..++++++++++++
++++++++++++++++++++++++++*...........+...........+...+...................+.............+..............+..
+.........+.+..............+.................+.......+.....+...+......+.+.........+.....+..........
+...........+.+..............+......++++++
............+....++++++++++++++++++++++++++++++++++++*........++++++++++++++++++++++++++++++++++++*.+..+.
+..+.......+........+..........+...+...+...........+...+....................+.+...+..+.+........+......+....
+.....+....+..+.+.........................++++++
-----
```

On génère deux choses d'un coup :

1. **La Clé Privée (httpd-tls.key)** : La carte d'identité du serveur.
2. **La Demande de Signature (httpd-tls.csr)** : Le formulaire rempli à envoyer à la CA.

Vérification

```
admin@lab-debian-23:/etc/my-pki$ ls -l *
certs:
total 8
-rw-rw-r-- 1 root root 1489 Jan 16 15:09 ca.crt
-rw-rw-r-- 1 root root 1147 Jan 16 15:41 httpd-tls.csr

conf:
total 4
-rw-rw-r-- 1 root root 378 Jan 16 15:40 httpd-config-tls.conf

private:
total 8
-rw------- 1 root root 1704 Jan 16 15:08 ca.key
-rw------- 1 root root 1704 Jan 16 15:41 httpd-tls.key
```

CÔTÉ CA : La Signature (L'Adoubement)

Maintenant, retournement de situation. **Nous reprenons la casquette de l'Autorité (CA).**

La Commande de Signature

```
admin@lab-debian-23:/etc/my-pki$ sudo openssl x509 -req -days 90 -in certs/httpd-tls.csr -CA certs/ca.crt
-CAkey private/ca.key -CAcreateserial -out certs/httpd-tls.crt -extensions v3_req -copy_extensions copy
```

```
# openssl x509 : L'outil de gestion de certificats
# -req : On signe une demande (CSR)
# -days 90 : Validité de 3 mois
# -in : L'entrée (votre CSR)
# -CA : Le certificat du patron (CA)
# -CAkey : La clé du patron (CA)
# -CAcreateserial : Crée un petit fichier de suivi pour les numéros de série
# -out : La sortie (Le certificat final .crt)
# -extensions : La section spécifique à injecter
# -copy_extensions copy : L'instruction qui recopie les SAN
```

```
# Vérification
```

```
admin@lab-debian-23:/etc/my-pki$ ls -l *
certs:
total 16
-rw-rw-r-- 1 root root 1489 Jan 16 15:09 ca.crt
-rw-rw-r-- 1 root root   41 Jan 16 16:01 ca.srl
-rw-rw-r-- 1 root root 1509 Jan 16 16:01 httpd-tls.crt
-rw-rw-r-- 1 root root 1147 Jan 16 15:41 httpd-tls.csr

conf:
total 4
-rw-rw-r-- 1 root root 378 Jan 16 15:40 httpd-config-tls.conf

private:
total 8
-rw------- 1 root root 1704 Jan 16 15:08 ca.key
-rw------- 1 root root 1704 Jan 16 15:41 httpd-tls.key
```

La Vérification (Trust, but Verify)

Avant de crier victoire et de lancer Apache, on vérifie que le certificat contient bien tout ce qu'on a demandé.

Lire le contenu du certificat en texte clair

```
admin@lab-debian-23:/etc/my-pki$ sudo openssl x509 -in certs/httpd-tls.crt -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            58:25:bb:3e:9c:58:4e:ec:41:61:dc:1e:23:e1:2c:48:8f:02:68:34
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=FR, ST=AriÃ¨ge, L=Auzat, O=MyLocalPKI, OU=IT Security, CN=MyLocalPKI Root CA,
emailAddress=admin@mylocalpki.fr
        Validity
            Not Before: Jan 16 15:01:13 2026 GMT
            Not After : Apr 16 15:01:13 2026 GMT
        Subject: C=FR, ST=AriÃ¨ge, L=Auzat, O=Local Unsafe Lab, CN=lab-unsafe.lan, emailAddress=admin@lab-
unsafe.fr
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:c5:16:de:d5:55:83:45:ed:09:55:b7:68:4e:a3:
                    2f:47:fd:21:af:c5:ef:d9:18:01:22:a4:d4:38:fb:
                    58:9f:27:11:8b:a2:d9:40:43:5f:3c:6e:02:7a:57:
                    40:01:42:01:8e:e0:e4:2a:81:ff:1a:32:6b:fe:01:
                    2a:22:e7:66:69:3b:7a:7e:6a:15:cc:5b:52:a6:44:
                    43:a4:4e:83:b9:4d:13:45:fc:5c:97:88:3c:e7:9b:
                    20:82:43:14:27:c5:84:0b:0e:9e:ec:63:51:2c:9c:
                    4c:be:d1:5a:83:5d:aa:5f:1c:86:ba:e2:2d:e5:0c:
                    9a:89:8f:52:2d:25:9e:0e:1c:58:d1:4d:97:92:05:
                    ac:4b:f4:79:31:62:fe:0a:6a:99:9a:f8:4c:b1:0e:
                    ff:08:b8:f0:f9:27:c6:a0:f3:77:14:5a:c0:6d:44:
                    47:02:66:c4:54:e0:c0:c8:83:74:48:65:94:1b:86:
                    4d:06:2d:ad:d7:85:e7:1d:4e:e1:71:d5:c8:02:97:
                    c7:dd:fc:5e:e9:f8:b1:f0:53:34:b5:ca:f2:b5:f7:
                    13:54:f7:ea:25:8d:05:94:3e:96:f2:75:10:df:c6:
                    53:83:c5:8e:1a:b2:d0:53:43:7f:d0:97:e1:a2:0d:
                    96:96:4f:cc:c6:da:a5:13:b9:44:01:6c:d8:88:da:
                    8a:1d
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Alternative Name:
                DNS:lab-unsafe.lan, DNS:www.lab-unsafe.lan, IP Address:192.168.100.155
            X509v3 Subject Key Identifier:
                CA:1C:05:CA:FC:48:60:97:EC:A8:43:EF:F6:23:08:25:C5:85:C7:7F
            X509v3 Authority Key Identifier:
                48:E2:10:4A:C8:07:D8:88:9A:F8:F0:58:78:D3:5C:60:B0:F0:C2:C2
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        06:49:a2:8f:ae:60:b1:37:68:f8:8f:2e:2c:7c:08:74:9e:32:
        c8:3a:15:2d:55:b6:85:f2:83:16:e7:c5:96:1a:64:f8:48:a8:
        fd:0d:1c:3b:79:ee:45:5a:45:ea:4d:19:ba:c0:27:50:a6:d4:
        3d:66:f0:4d:d0:3a:87:e8:dc:58:b6:26:07:ac:92:df:c5:00:
        d1:46:d4:fa:0d:00:eb:54:14:68:32:8b:7c:47:39:78:6c:5e:
        36:4a:bd:5b:a9:03:c1:7c:49:af:74:43:8d:ed:3c:a6:b6:dc:
        87:35:94:ff:ec:55:21:0f:c4:4b:6a:da:6d:05:c2:84:30:55:
        43:dd:cd:62:b8:3e:9c:ba:81:73:8b:48:6c:f8:7e:2e:5d:5b:
        f2:b1:b3:ad:95:4b:1d:81:72:bd:7d:40:32:0a:56:ef:7f:ed:
        77:a5:a5:b5:42:48:e8:12:d9:a3:24:7e:79:a6:ad:ec:3b:3f:
        2c:a5:05:18:43:9a:42:65:e9:4c:41:f3:08:19:a9:d4:cb:44:
        2b:3e:0c:f8:f4:95:51:f4:90:ee:3a:41:0a:5d:85:d2:93:84:
        8a:05:da:ac:20:34:3b:71:a9:1f:82:1a:19:9e:f1:de:ef:73:
        1d:0f:76:6f:e5:98:25:54:4c:a6:bf:d2:85:9b:0e:2b:57:91:
        cf:53:36:d2
```

Maintenant nous avons notre certificat signé, il n’y a plus qu’à configurer Apache !

CÔTÉ Apache : L'Activation HTTPS

L'Échec Programmé (Le TP)

Le Piège : Activer le Site sans le Moteur

Pourquoi : si le fichier default-ssl.conf est entouré d'une balise <IfModule mod_ssl.c>.

- Apache lit le fichier, il se demande : "Est-ce que le module mod_ssl est chargé ?"
- Réponse : NON, il ignore tout le bloc. Le port 443 ne s'ouvre même pas. C'est un site fantôme.

Sinon, Apache refuse de redémarrer (Erreur de Syntaxe), car le module `mod_ssl` n'est pas chargé.

Préparation des Modules

1. Activation du module SSL (Indispensable)

```
admin@lab-debian-23:~$ sudo a2enmod ssl
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
```

2. Activation du modèle de site SSL par défaut

(Si ce n'est pas déjà fait)

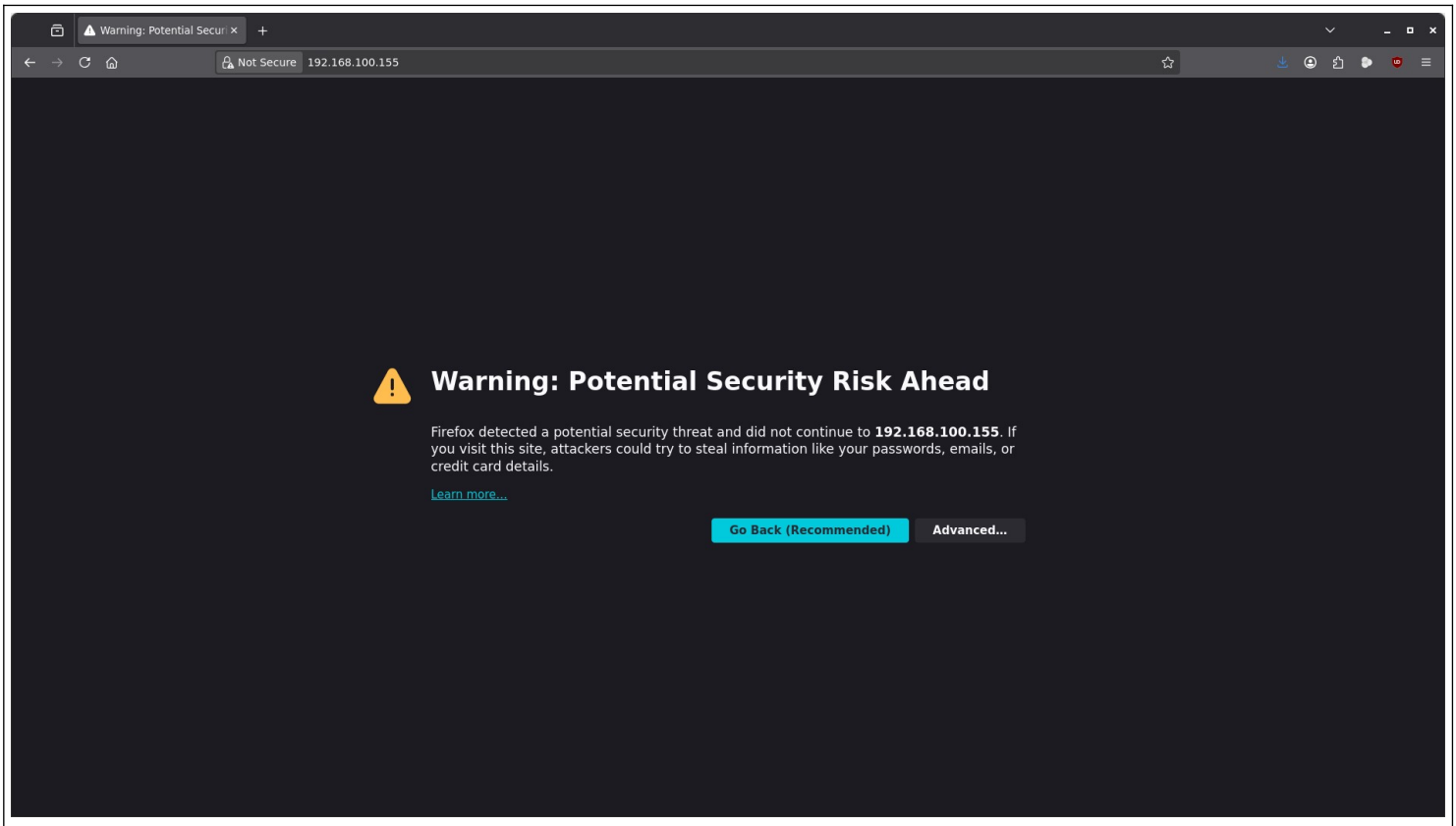
```
admin@lab-debian-23:~$ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

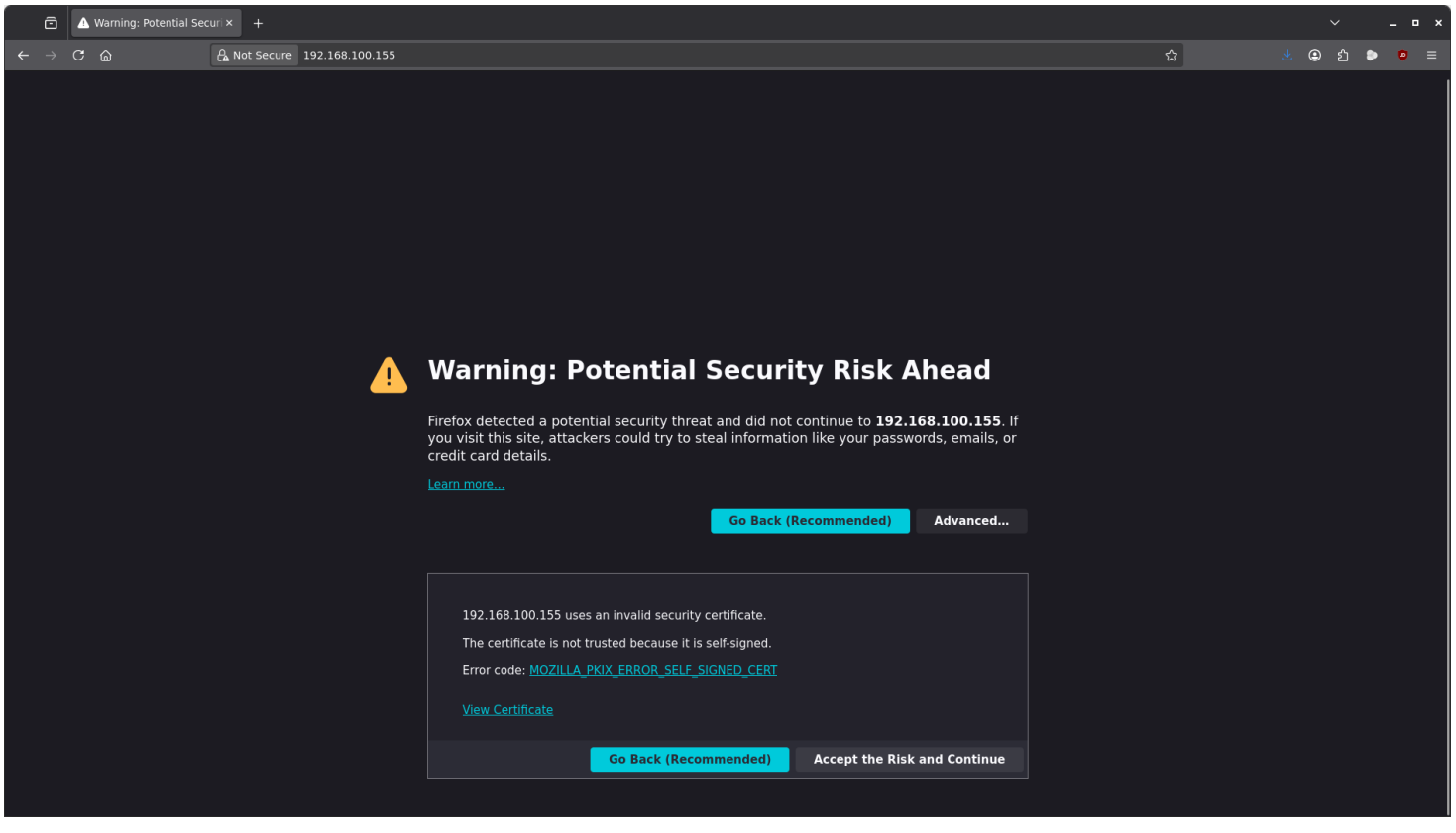
```
admin@lab-debian-23:~$ sudo systemctl restart apache2.service
```

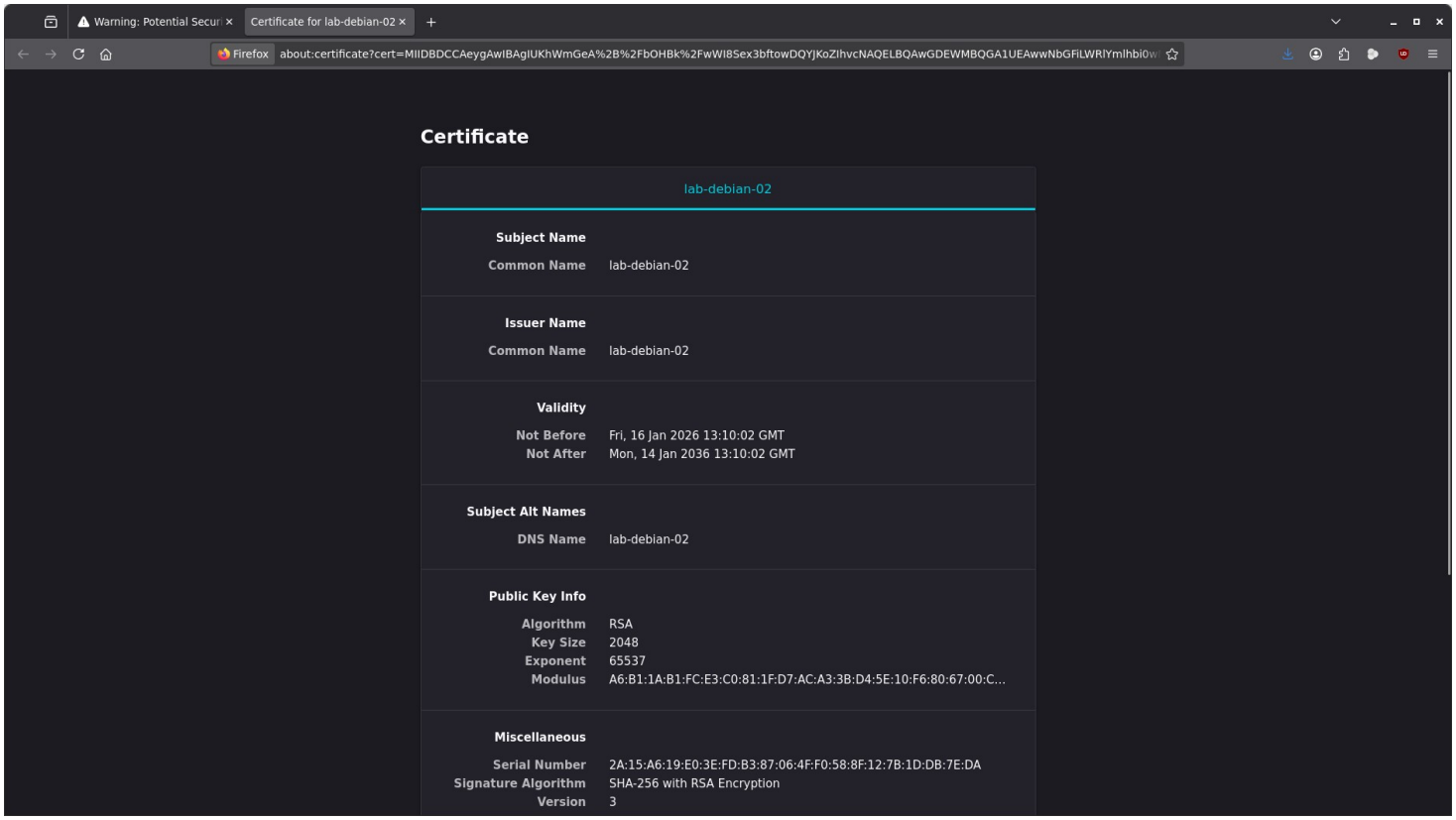
RELOAD (systemctl reload) : Le processus relie les fichiers de configuration (.conf) pour prendre en compte les changements (nouveau site, changement de dossier).

RESTART (systemctl restart) : On tue tous les processus (Kill). On libère la mémoire. On relance tout depuis zéro.

```
# On Reload pour de la config (texte). On Restart pour charger des modules (binaires).
```







Oops ! Ce n’est pas le certificats que nous avons générer …

Le fichier `/etc/apache2/sites-available/default-ssl.conf` pointe actuellement vers `/etc/ssl/certs/ssl-cert-snakeoil.pem`

> Apache génère automatiquement un certificat auto-signé "poubelle" (souvent appelé **Snake Oil** dans le jargon, en référence aux remèdes de charlatans) pour que le SSL fonctionne techniquement dès l'installation, même si l'identité est fausse.

L'INTÉGRATION FINALE (Le Virtual Host SSL)

Création du dossier du site (DocumentRoot)

```
admin@lab-debian-23:~$ sudo mkdir -p /var/www/lab-unsafe
```

Création d'une page d'accueil simple pour tester

```
admin@lab-debian-23:~$ sudo echo "<h1>Bienvenue dans le Labo Securise</h1><p>Certificat valide.</p>" >
/var/www/lab-unsafe/index.html
```

Attribution des droits (Apache doit pouvoir lire)

```
admin@lab-debian-23:~$ sudo chown -R www-data:www-data /var/www/lab-unsafe
```

Vérification

```
admin@lab-debian-23:~$ ls -l /var/www/lab-unsafe/
total 4
-rw-rw-r-- 1 www-data www-data 76 Jan 16 19:26 index.html
```

CRÉATION DU FICHIER DE CONFIGURATION

```
admin@lab-debian-23:~$ sudo nano /etc/apache2/sites-available/lab-unsafe.conf
```

```
<VirtualHost *:80>
    ServerName lab-unsafe.lan
    ServerAlias www.lab-unsafe.lan
    Redirect permanent / https://lab-unsafe.lan/

    ErrorLog ${APACHE_LOG_DIR}/lab-unsafe_error.log
    CustomLog ${APACHE_LOG_DIR}/lab-unsafe_access.log combined
</VirtualHost>

<IfModule mod_ssl.c>
    <VirtualHost *:443>
        ServerName lab-unsafe.lan
        ServerAlias www.lab-unsafe.lan

        ServerAdmin admin@lab-unsafe.fr
        DocumentRoot /var/www/lab-unsafe

        ErrorLog ${APACHE_LOG_DIR}/lab-unsafe_ssl_error.log
        CustomLog ${APACHE_LOG_DIR}/lab-unsafe_ssl_access.log combined

        SSLEngine on

        SSLCertificateFile      /etc/my-pki/certs/httpd-tls.crt
        SSLCertificateKeyFile   /etc/my-pki/private/httpd-tls.key

        SSLProtocol all -SSLv3 -TLSv1 -TLSv1.1     <-- J'accepte tout sauf ...

        SSLCipherSuite SSLCipherSuite HIGH:!aNULL:!MD5:!3DES  <-- On impose des algorithmes
        SSLHonorCipherOrder on     <-- Ordre : Le serveur décide, pas le client

    </VirtualHost>
</IfModule>
```

Sauvegardez (Ctrl+O) et Quittez (Ctrl+X).

ACTIVATION & NETTOYAGE

```
# On désactive les configurations par défaut (Nettoyage)
```

```
admin@lab-debian-23:~$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
admin@lab-debian-23:~$ sudo a2dissite default-ssl.conf
Site default-ssl disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

```
# On active NOTRE configuration propre
```

```
admin@lab-debian-23:~$ sudo a2ensite lab-unsafe.conf
Enabling site lab-unsafe.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

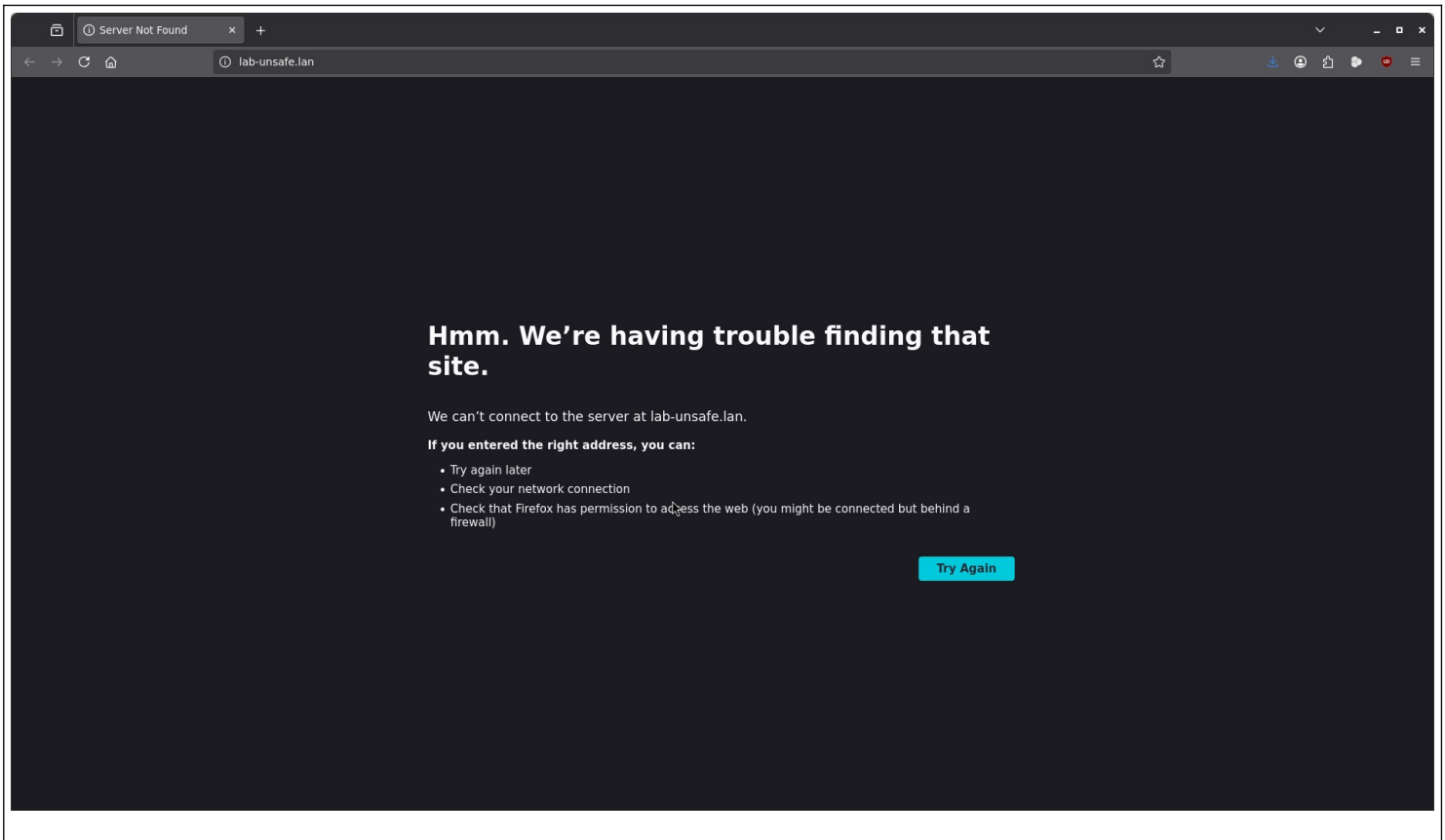
```
# Le Test de Syntaxe (Toujours !)
```

```
admin@lab-debian-23:~$ sudo apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set
the 'ServerName' directive globally to suppress this message
Syntax OK
```

```
# Redémarrage
```

```
admin@lab-debian-23:~$ sudo systemctl reload apache2.service
```

```
# Oops
```



LE DÉTAIL QUI TUE (DNS Menteur)

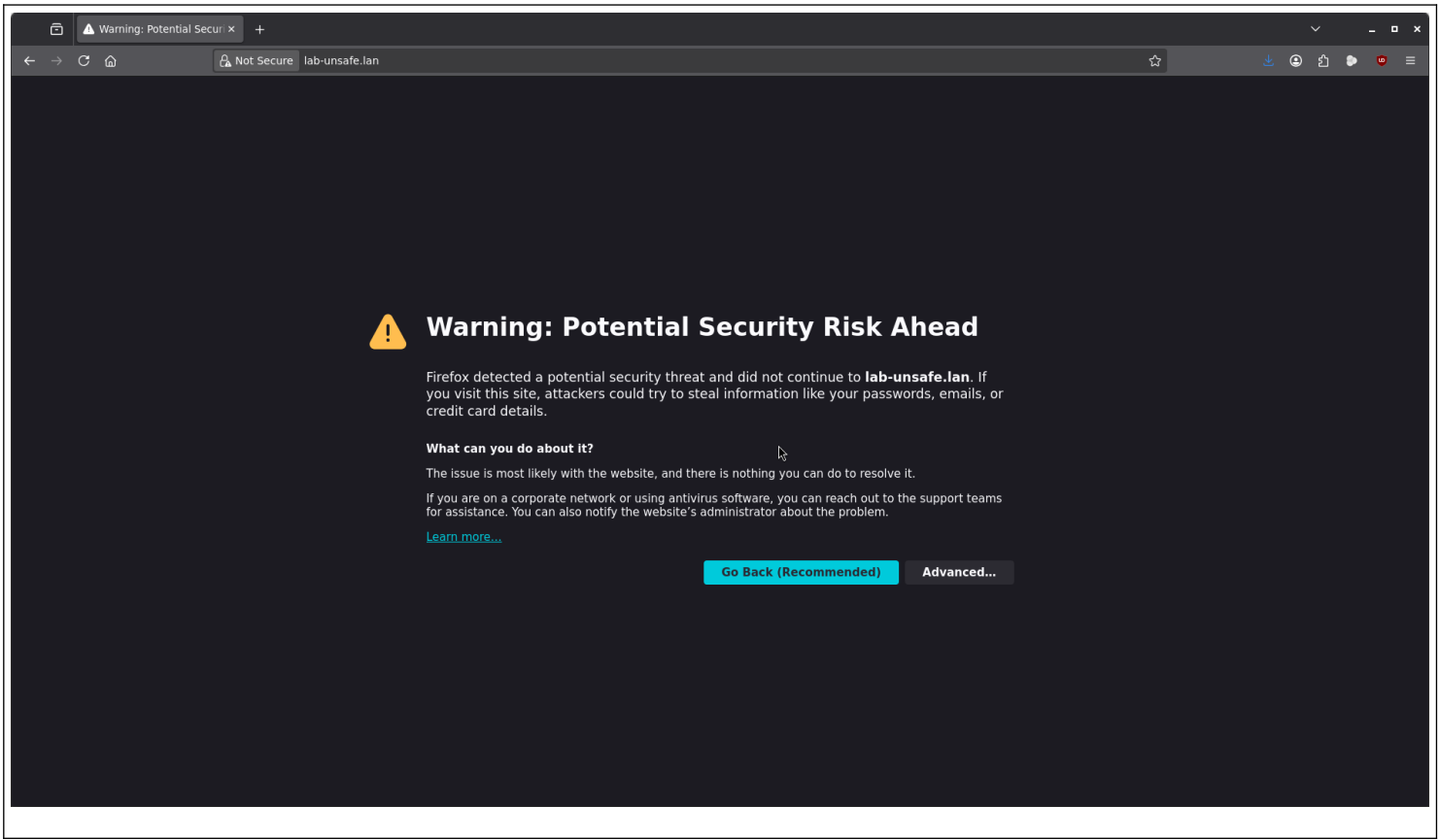
Votre certificat est fait pour lab-unsafe.lan. Mais ce domaine n'existe pas sur Internet. Si vous tapez ça dans votre navigateur, il ne trouvera rien.

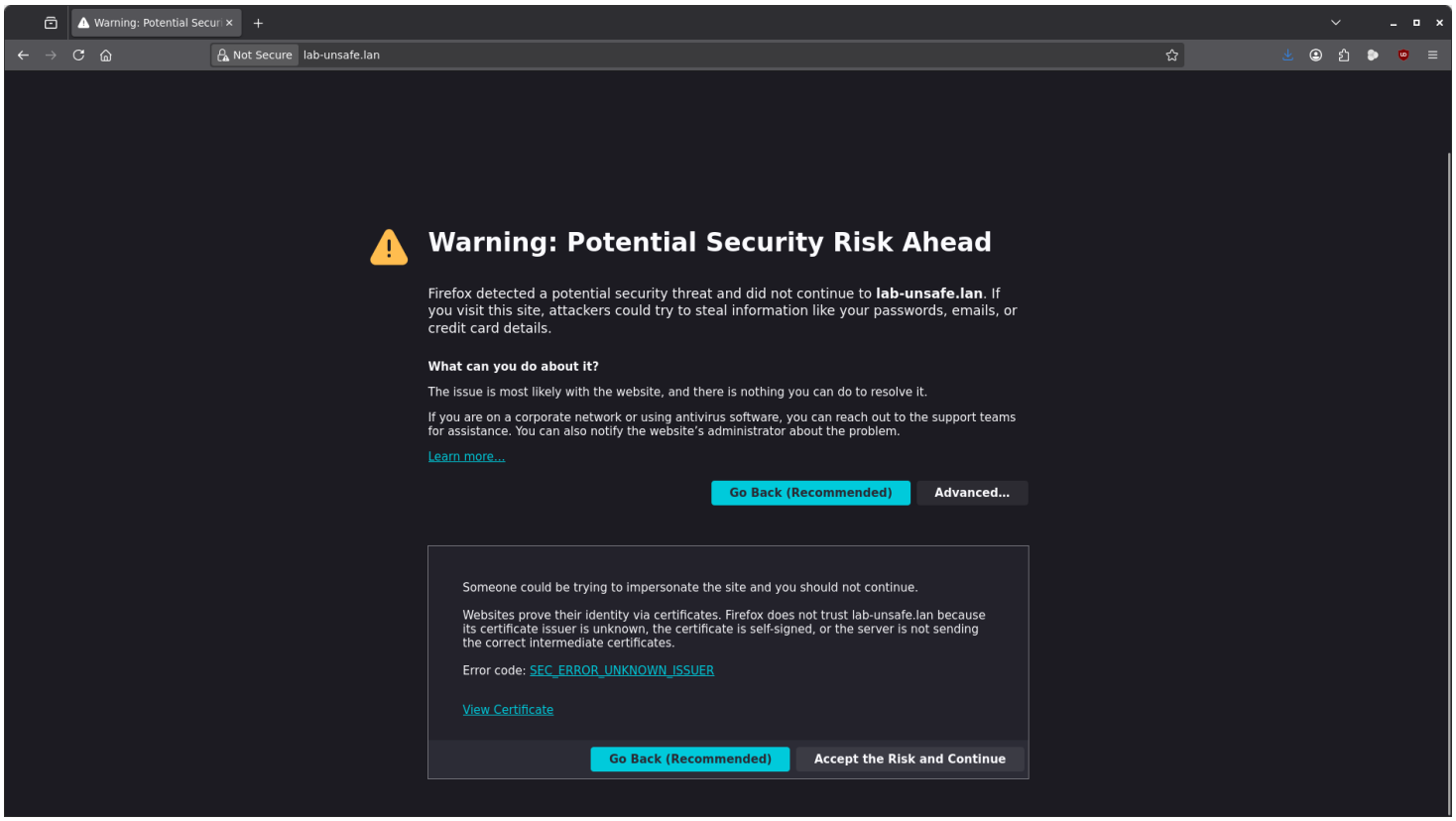
A faire sur la machine qui a le navigateur

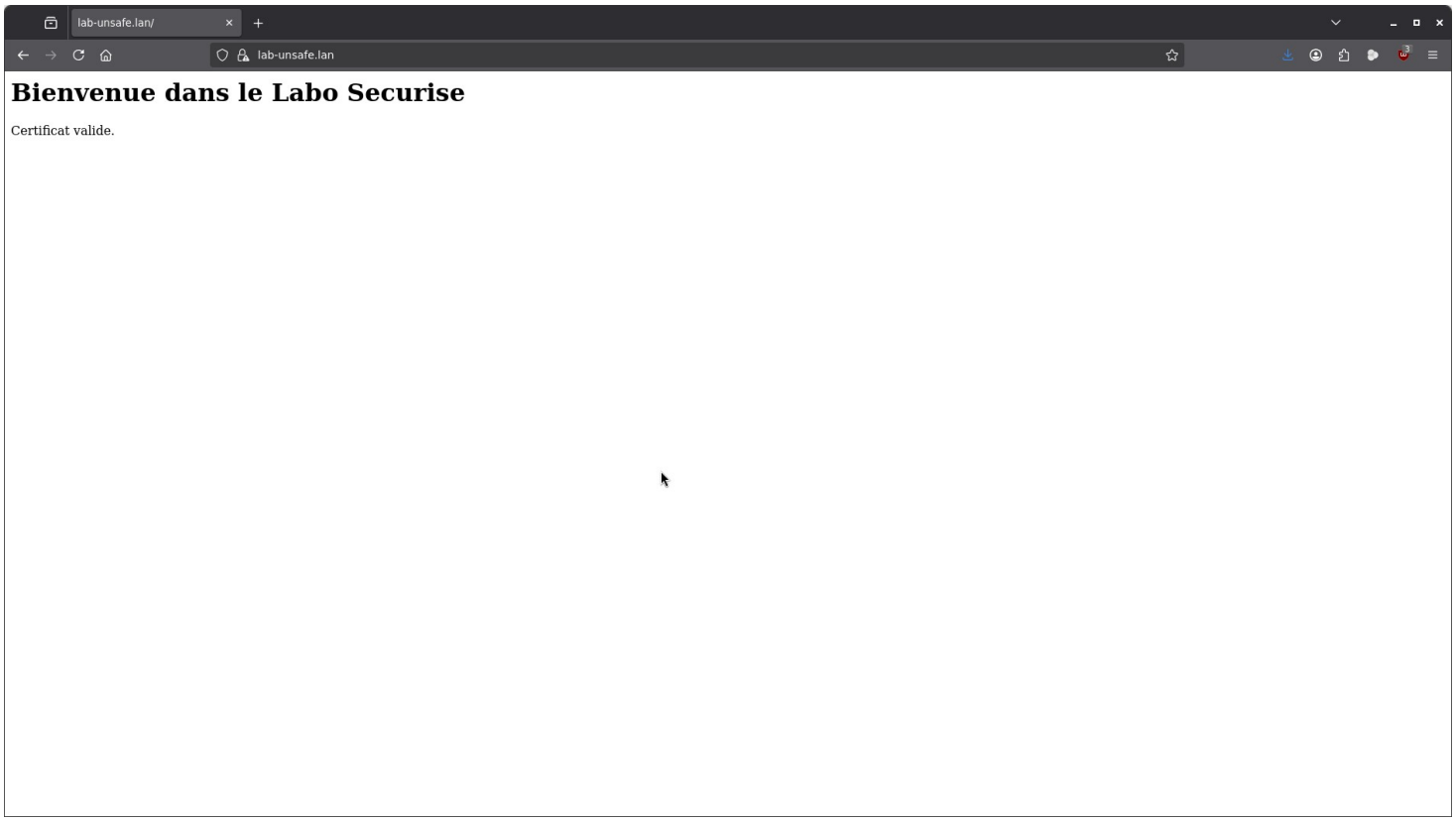
```
admin@lab-debian-23:~$ sudo nano /etc/hosts
```

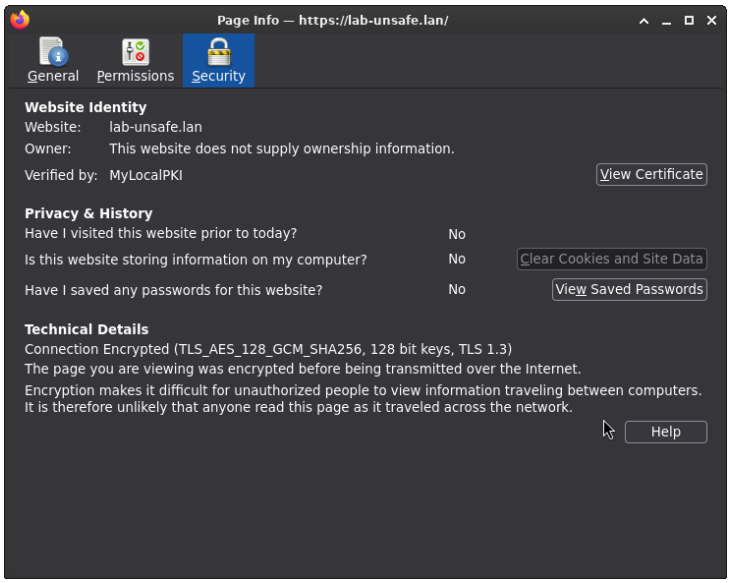
```
127.0.0.1       localhost
127.0.1.1       lab-debian-02

192.168.100.155 lab-unsafe.lan  www.lab-unsafe.lan
```

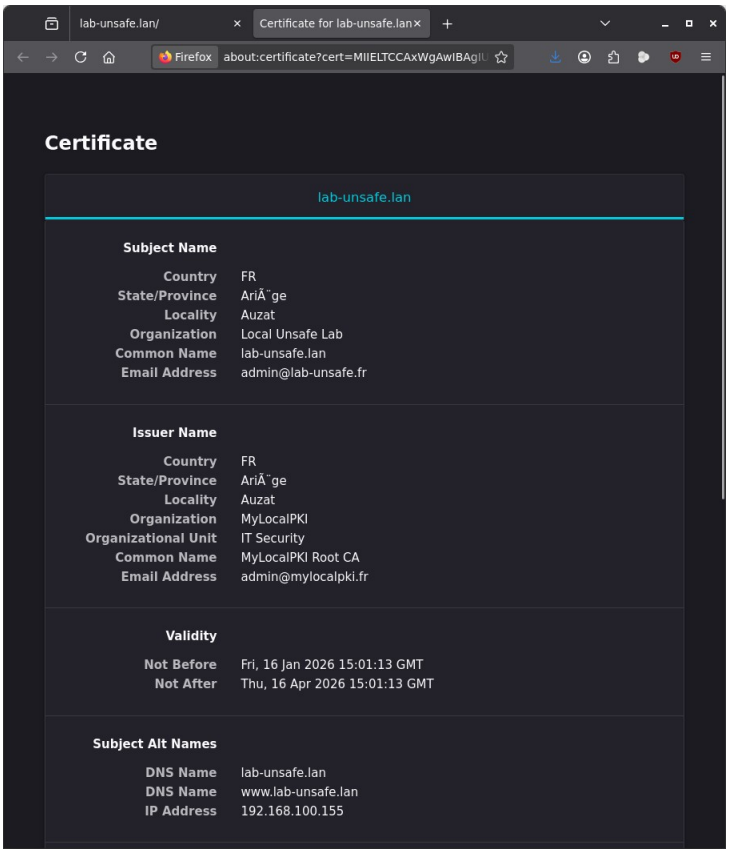
Sauvegardez (Ctrl+O) et Quittez (Ctrl+X).











"Nous avons sécurisé les données avec un chiffrement RSA 2048 bits de niveau militaire, capable de résister aux supercalculateurs de la NSA... mais visiblement, la technologie humaine n'est pas encore assez avancée pour gérer un accent grave."