Exemple concret d'utilisation et d'activation d'un module Apache : `mod_ssl`

Le module `mod_ssl` permet à Apache de prendre en charge HTTPS en utilisant des certificats SSL/TLS. Cela garantit que les communications entre le client (navigateur) et le serveur sont chiffrées.

Contexte :

Vous avez un site web accessible en HTTP et vous voulez le sécuriser avec HTTPS en utilisant un certificat SSL auto-signé.

Étapes d'utilisation et d'activation de `mod_ssl`

1. Activer le module `mod_ssl`

Sur Ubuntu/Debian

- Activez le module avec la commande suivante :

```
sudo a2enmod ssl
```

- Rechargez Apache pour activer le module :

```
sudo systemctl reload apache2
```

Sur CentOS/RHEL

- Assurez-vous que le module est chargé dans le fichier `/etc/httpd/conf/httpd.conf` ou `/etc/httpd/conf.modules.d/00-ssl.conf` :

```
LoadModule ssl_module modules/mod_ssl.so
```

- Rechargez Apache pour appliquer les changements :

```
sudo systemctl reload httpd
```

---

2. Créer un certificat SSL auto-signé

Générez un certificat SSL auto-signé avec **OpenSSL**.

1. **Créer le répertoire pour les certificats** :

```
sudo mkdir -p /etc/apache2/ssl
```

2. **Générer une clé privée et un certificat** :

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
-keyout /etc/apache2/ssl/apache-selfsigned.key \
-out /etc/apache2/ssl/apache-selfsigned.crt
```

3. **Répondre aux questions** :
   - Country Name : `FR`
   - State or Province Name : `Ile-de-France`
   - Locality Name : `Paris`
   - Organization Name : `MyCompany`
   - Organizational Unit Name : `IT`
   - Common Name : `example.com`
   - Email Address : `admin@example.com`

---

3. Configurer le Virtual Host pour HTTPS

Ajoutez un Virtual Host spécifique à HTTPS dans le fichier de configuration approprié. Sur Ubuntu/Debian, cela pourrait être `/etc/apache2/sites-available/default-ssl.conf`. Si ce fichier n'existe pas, créez-en un.

```
sudo nano /etc/apache2/sites-available/ssl-example.conf
```

Exemple de configuration :

```
<VirtualHost *:443>
    ServerName example.com
    DocumentRoot /var/www/html

    SSLEngine On
    SSLCertificateFile /etc/apache2/ssl/apache-selfsigned.crt
    SSLCertificateKeyFile /etc/apache2/ssl/apache-selfsigned.key

    <Directory /var/www/html>
        AllowOverride All
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
```

```
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

---

4. Activer le Virtual Host pour HTTPS

Sur Ubuntu/Debian

- Activez le fichier de configuration :

```
sudo a2ensite ssl-example.conf
```

- Rechargez Apache :

```
sudo systemctl reload apache2
```

Sur CentOS/RHEL

- Ajoutez le Virtual Host dans `/etc/httpd/conf.d/ssl-example.conf`.
- Rechargez Apache :

```
sudo systemctl reload httpd
```

---

5. Tester le serveur HTTPS

- Accédez à votre site via HTTPS :

```
https://example.com
```

- Si vous utilisez un certificat auto-signé, le navigateur affichera un avertissement. Acceptez l'exception pour accéder au site.

---

6. (Optionnel) Rediriger le trafic HTTP vers HTTPS

Ajoutez une redirection dans votre Virtual Host HTTP (fichier `/etc/apache2/sites-available/000-default.conf`) :

```
<VirtualHost *:80>
    ServerName example.com
```

```
    Redirect permanent / https://example.com/
</VirtualHost>
```

Rechargez Apache :

```
sudo systemctl reload apache2
```

Vérifications

1. **Tester la configuration Apache** :

   ```
   apache2ctl configtest
   ```

   Sortie attendue :

   ```
   Syntax OK
   ```

2. **Vérifier les modules activés** :

   ```
   apache2ctl -M | grep ssl
   ```

   Sortie attendue :

   ```
   ssl_module (shared)
   ```

3. **Tester le certificat** : Avec OpenSSL, testez la connexion SSL :

   ```
   openssl s_client -connect example.com:443
   ```

Résumé des étapes

1. Activez le module `mod_ssl`.
2. Créez un certificat SSL auto-signé avec OpenSSL.
3. Configurez un Virtual Host pour HTTPS.
4. Testez l'accès HTTPS.
5. (Optionnel) Configurez une redirection HTTP vers HTTPS.

Avantages concrets de `mod_ssl`

- **Sécurité renforcée** : Le chiffrement SSL/TLS protège les données sensibles échangées.
- **Conformité** : Obligatoire pour les sites qui traitent des données personnelles (RGPD, PCI DSS).
- **SEO** : Les moteurs de recherche comme Google privilégient les sites en HTTPS.