

Certificats\_HTTPS\_PHPmyadmin

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Rédacteur** | **Changements par rapport à la version précédente** |
| 1.0 | 14/04/2025 | KETTERER  Anthony  BTS CIEL | Version initiale |
| 1.1 | 22/04/2025 | KETTERER  Anthony  BTS CIEL | Version corrigée |
| 1.2 | 25/04/2025 | MASSON SCIAUX Romaric  CMIL | Version corrigée |

Table des matières

1. Introduction

2. Prérequis

3. Explication du procédé

4. Conclusion

1. Introduction

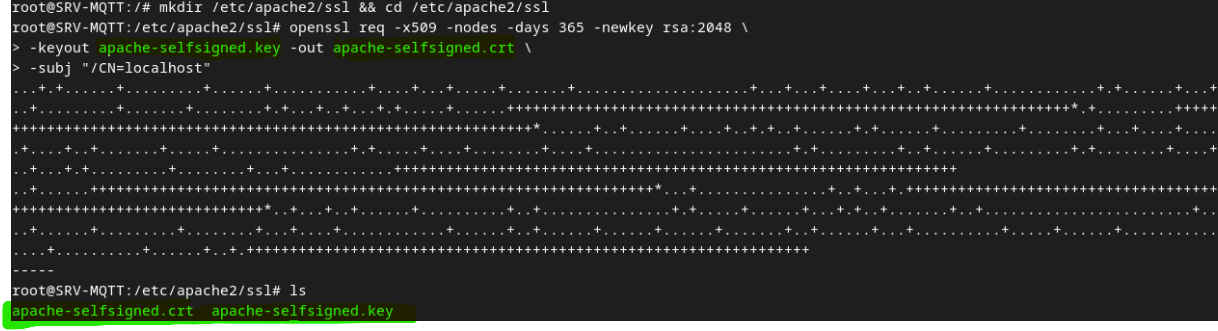
Ce document décrit la mise en place de certificats SSL/TLS afin de sécuriser la connexion web au site PHPmyadmin. Il affichera les valeurs récoltés par MariaDB sur son interface.

Cette documentation est destinée au professeurs/jurys notant l’ensemble de mon projet réalisé lors de ma deuxième année de BTS CIEL.

1. Prérequis

Ce document a pour but de sécuriser la connexion à PHPmyadmin hébergé sur un **OS Linux (Debian 12)**.

1. Explication du procédé

En premier lieu, nous créons un fichier où pouvoir stocker les clés futurement créée pour des raisons d’organisation. Une fois cela fait, la configuration peut démarrer.

**mkdir /etc/apache2/ssl && cd /etc/apache2/ssl**

* **mkdir /etc/apache2/ssl** : Crée un dossier nommé ssl dans /etc/apache2, qui contiendra les fichiers de certificat.
* **&& cd /etc/apache2/ssl** : Si la création du dossier réussit, on entre dans ce répertoire avec cd.

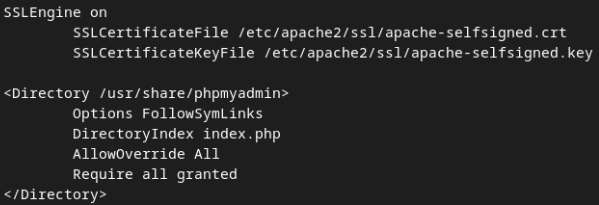
**openssl req -x509 -nodes -days 365 -newkey rsa:2048 \**

Commande openssl pour générer un **certificat auto-signé**.

* **req** : Lance une demande de certificat (Certificate Signing Request).
* **-x509** : Crée un certificat auto-signé directement au lieu d’une simple demande.
* **-nodes** : Ne chiffre pas la clé privée (No DES encryption), utile pour éviter d’entrer un mot de passe à chaque démarrage d’Apache.
* **-days 365** : Le certificat sera valide pendant 365 jours.
* **-newkey rsa:2048** : Génère une nouvelle paire de clés (privée et publique) RSA de 2048 bits.

**ls**

* Affiche les fichiers générés dans le dossier /etc/apache2/ssl :
  + **apache-selfsigned.crt** : Certificat auto-signé (utilisé par Apache pour la communication HTTPS).
  + **apache-selfsigned.key** : Clé privée correspondante.



**SSLEngine on** : Active le module SSL d’Apache pour ce VirtualHost, permettant d’utiliser HTTPS.

**SSLCertificateFile** : Chemin vers le fichier du **certificat SSL** (généré avec OpenSSL).  
Ici : /etc/apache2/ssl/apache-selfsigned.crt.

**SSLCertificateKeyFile** : Chemin vers la **clé privée** associée au certificat.  
Ici : /etc/apache2/ssl/apache-selfsigned.key.

**<Directory /usr/share/phpmyadmin>** : Débute la configuration pour le dossier contenant phpMyAdmin.

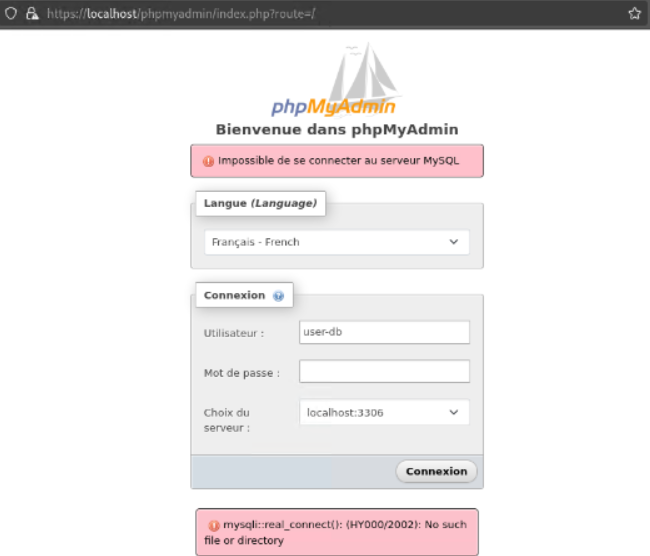
**Options FollowSymLinks** : Autorise le serveur à suivre les liens symboliques dans ce dossier.

**DirectoryIndex index.php** : Si aucun fichier n’est spécifié dans l’URL, Apache chargera index.php par défaut.

**AllowOverride All** : Autorise l’utilisation de fichiers .htaccess dans ce répertoire pour surcharger la configuration Apache localement.

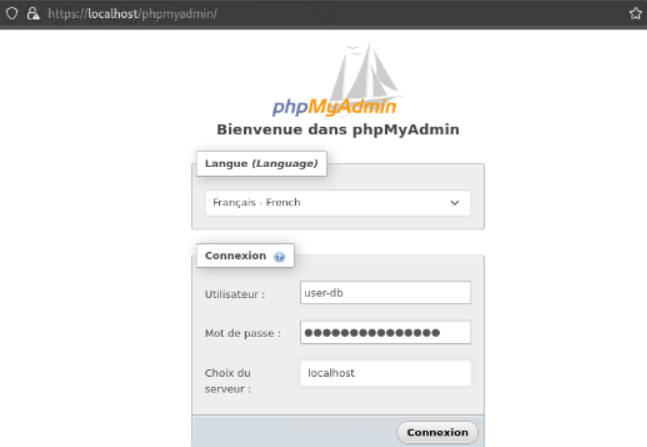
**Require all granted** : Permet l'accès à tous les utilisateurs (sinon, Apache peut bloquer l'accès par défaut).

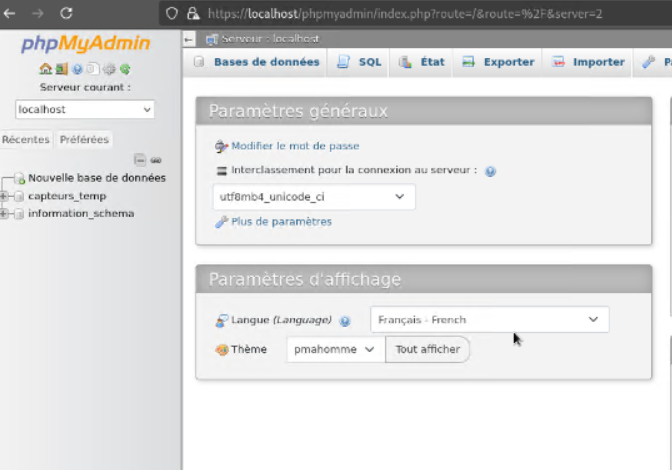
**</Directory> :** Termine la configuration.





Le fichier /etc/phpmyadmin/config.inc.php a été modifié afin de se connecter uniquement via un Socket Unix, non via TCP/IP. Cette connexion est plus sécurisée car il ne laisse aucune trace sur le réseau. Il nous faut donc nous connecter via le Socket Unix.







1. Conclusion

L’utilisation de certificats SSL/TLS générés avec OpenSSL est essentielle pour sécuriser la connexion à PHPMyAdmin via Apache2. En effet, PHPMyAdmin étant une interface d’administration de base de données accessible depuis un navigateur, elle est exposée à des risques importants si les échanges ne sont pas chiffrés. Les certificats permettent de chiffrer la communication entre le client et le serveur web, empêchant ainsi toute interception ou altération des données, notamment les identifiants de connexion à la base de données.

La mise en place d’un certificat avec OpenSSL, qu’il soit autosigné ou délivré par une autorité de certification, renforce donc la confidentialité et l’intégrité des informations échangées avec PHPMyAdmin. Cette étape s’inscrit dans une démarche globale de sécurisation des accès aux services critiques, et constitue un bon réflexe à adopter dans tout environnement de production ou d’administration distante.