# Sécuriser son serveur GLPI (HTTPS & Pare-feu)

**Introduction : Pourquoi cette étape est critique ?**

Dans le monde professionnel, ton serveur actuel est **dangereux**.

Pourquoi ? Parce que le protocole **HTTP** (HyperText Transfer Protocol) fait circuler les informations "en clair" sur le réseau.

Imagine la situation :

Tu es l'administrateur. Tu te connectes à GLPI avec ton mot de passe super complexe.

Si un pirate (ou un curieux) est connecté sur le même réseau que toi et lance un logiciel d'écoute comme Wireshark, il verra passer ton mot de passe tel quel, comme s'il lisait une carte postale sans enveloppe.

**Notre mission du jour :**

1. **Chiffrer les échanges** (HTTPS) pour que les données deviennent illisibles pour les pirates.
2. **Filtrer les accès** (Pare-feu) pour n'ouvrir que les portes strictement nécessaires.

## Partie 1 : Comprendre le HTTPS (La théorie simple)

Avant de taper des commandes, comprenons le concept.

- **HTTP (Port 80) :** C'est comme envoyer une **carte postale**. Le facteur (le réseau) et tous ceux qui manipulent le courrier peuvent lire le message écrit au dos.
- **HTTPS (Port 443) :** C'est comme envoyer une **lettre dans un coffre-fort blindé**. Seul le destinataire (le serveur) possède la clé pour ouvrir le coffre et lire le message.

Pour que cela fonctionne, nous avons besoin de deux choses :

1. Une **Clé privée** (conservée secrètement sur le serveur).
2. Un **Certificat public** (distribué aux visiteurs pour qu'ils puissent chiffrer les messages qu'ils t'envoient).

## Partie 2 : Mise en place du HTTPS (Certificat Auto-signé)

Dans une vraie entreprise, on achète un certificat validé par une autorité reconnue (comme *Let's Encrypt* ou *Digicert*). Pour ce TP, nous allons créer notre propre certificat : on dit qu'il est **auto-signé**.

C'est un peu comme fabriquer sa propre carte d'identité : elle est techniquement valide pour chiffrer, mais la police (le navigateur web) te dira qu'elle ne connait pas l'autorité qui l'a délivrée. C'est normal !

## Étape 2.1 : Activer le module SSL d'Apache

Apache possède un module pour gérer le chiffrement, mais il n'est pas activé par défaut.

1. Ouvre ton terminal (Ctrl+Alt+T).
2. Active le module SSL :

```
sudo a2enmod ssl
```

3. Pour que le changement soit pris en compte, redémarre Apache :

```
sudo systemctl restart apache2
```

## Étape 2.2 : Générer les clés et le certificat

Nous allons utiliser l'outil **OpenSSL** (« le couteau suisse de la cryptographie »).

Crée un dossier pour ranger proprement tes certificats :

```
sudo mkdir /etc/apache2/ssl
```

Génère le certificat et la clé en une seule commande (copie-colle tout ceci) :

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

**Déchiffrons cette commande :**

- req -x509 : On veut créer un certificat au standard X.509.
- -days 365 : Le certificat sera valide 1 an.
- -newkey rsa:2048 : On crée une clé de chiffrement RSA très robuste (2048 bits).
- -keyout et -out : Où ranger la clé (le secret) et le certificat (le public).

L'outil va te poser des questions (Pays, Ville, etc.). Comme c'est un TP local, tu peux appuyer sur **Entrée** pour tout passer, ou remplir "FR" pour le pays.

## Étape 2.3 : Configurer le VirtualHost Apache

Rappelle-toi, dans le TP précédent, nous avons configuré GLPI pour répondre sur le port 80[1]. Nous devons maintenant dire à Apache d'écouter aussi sur le port **443** (le port standard du HTTPS) et d'utiliser nos clés.

1. Crée un nouveau fichier de configuration pour la version sécurisée :

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

2. Colle le contenu suivant (c'est une adaptation de ta configuration précédente, avec la couche sécurité en plus) :

```
<VirtualHost *:443>
    ServerName localhost

    # On pointe vers le même dossier public que précédemment
    DocumentRoot /var/www/html/glpi/public

    # Activation du moteur de sécurité SSL
    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/glpi.crt
    SSLCertificateKeyFile /etc/apache2/ssl/glpi.key

    <Directory /var/www/html/glpi/public>
        Require all granted
        RewriteEngine On
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteCond %{REQUEST_FILENAME} !-d
        RewriteRule ^(.*)$ index.php [QSA,L]
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/glpi_ssl_error.log
    CustomLog ${APACHE_LOG_DIR}/glpi_ssl_access.log combined
</VirtualHost>
```

3. Enregistre (Ctrl+O, Entrée) et quitte (Ctrl+X).

## Étape 2.4 : Activer le site et tester

1. Active ce nouveau site sécurisé :

```
sudo a2ensite glpi-ssl.conf
```

2. Recharge Apache :

```
sudo systemctl reload apache2
```

Le moment de vérité :

Ouvre Firefox et tape : https://localhost (Note bien le s à la fin de http).

**L'alerte de sécurité !**

Tu vas voir un écran effrayant : "Attention : risque probable de sécurité".

C'est normal ! Firefox te prévient : "Je chiffre bien la connexion, MAIS je ne connais pas l'organisme qui a signé ce certificat" (puisque c'est toi !).

- Clique sur **Avancé**.
- Clique sur **Accepter le risque et poursuivre**.

Tu es maintenant connecté à GLPI avec un petit cadenas (parfois barré d'un triangle jaune selon les navigateurs car auto-signé), mais tes données sont chiffrées !

## Partie 3 (Bonus) : Le garde du corps (Pare-feu UFW)

Sécuriser le transport des données (HTTPS), c'est bien. Empêcher les intrus d'entrer, c'est mieux.

Nous allons utiliser **UFW** (Uncomplicated Firewall). Imagine-le comme un videur à l'entrée d'une boîte de nuit.

**La stratégie du "Tout refuser, sauf..."**

En sécurité, la meilleure politique est de tout fermer par défaut, et d'ouvrir seulement ce qui est nécessaire.

1. Vérifie l'état actuel du pare-feu :

```
sudo ufw status
```

*(Il devrait être "inactive").*

2. Ouvre les ports vitaux **AVANT** d'activer le pare-feu (sinon tu risques de te bloquer toi-même, surtout si tu travaillais en SSH à distance !) :
    - **Port 22 (SSH)** : Pour l'administration à distance (si tu l'utilises).

```
sudo ufw allow ssh
```

    - **Port 80 (HTTP)** : Pour rediriger les gens vers le site sécurisé.

```
sudo ufw allow http
```

    - **Port 443 (HTTPS)** : Pour l'accès sécurisé à GLPI.

```
sudo ufw allow https
```

3. Active le pare-feu :

```
sudo ufw enable
```

*(Réponds 'y' pour confirmer).*

4. Vérifie que le videur fait son travail :

```
sudo ufw status verbose
```

Tu devrais voir une liste où seuls les ports 22, 80 et 443 sont en "ALLOW". Tout le reste est bloqué.