Installation d'un serveur Web

Cette mission a été réalisé lors de mon alternance dans l'entreprise Wendling Adhésif.

I- Contexte

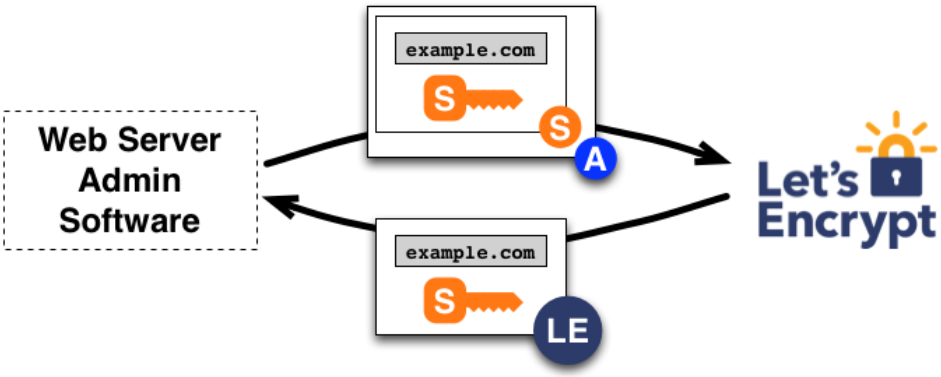
L'entreprise Wendling souhaitant faire appel à des API externes, se voit dans l'obligation d'installer un certificat SSL pour son serveur. Ce changement devient réellement important dans l'optique de pouvoir faire appel à des API de transporteurs et de vente (Ebay). Il n'y a pas de budget alloué à ce projet mis à part les coûts en ressources humaines. Les outils disponibles pour cette tâche sont :

- Un serveur avec tous les composants web installés (Apache/PHP/MySQL)
- Un terminal connecté en SSH pour naviguer sur le serveur.

Les contraintes ici sont de ne pas altérer les autres services durant l'installation du certificat.

II- La réponse au besoin

Afin de répondre à ce besoin, il a fallu choisir une solution rapide et adaptée à ce besoin. L'idée retenu a été de faire appel à Let's Encrypt avec certbot étant donné que c'est une solution rapide et facile à mettre en place. Le principe est de faire appel à un script qui vérifie que l'on est bien propriétaire du site web, et, si c'est le cas, instal le certificat.



III- Les étapes du projet

a. *L'installation de certbot*

Tout d'abord il nous faut installer snap :

```
sudo apt update
```

```
sudo apt install snapd
```

A partir d'ici, il nous manque certaines dépendances de snap, c'est pourquoi on installe le "core"

```
sudo snap install core
```

On s'assures que l'on possède la dernière version de snap et de ces dépendances :

```
sudo snap refresh core
```

A partir de cette étape, nous pouvons installer certbot avec un niveau d'isolation "--classic" étant donné que certbot doit être autorisé à modifier des fichiers de configuration pour installer correctement son certificat. Ilm faudra aussi créer un lien symbolique dans le /usr/bin pour permettre d'utiliser la commande certbot depuis n'importe où.

```
sudo snap install --classic certbot
```

```
sudo ln -s /snap/bin/certbot /usr/bin/certbot
```

Certbot est prêt à l'emploi.

b. Vérification du Virtual Host

Pour vérifier la configuration du Virtual Host, il faut se rendre dans /etc/apache2/sites-available/nomdedomaine.conf.

D'ici nous pouvons avoir une architecture de Virtual Host qui ressemble à ceci :

```
<VirtualHost *:80>
    ServerName www.it-connect.local
    ServerAlias it-connect.local
    DocumentRoot /var/www/it-connect/html
    ErrorLog /var/www/it-connect/logs/error.log
    CustomLog /var/www/it-connect/logs/access.log combined
</VirtualHost>
```

Après avoir rempli le Virtual Host, nous pouvons faire un test afin de vérifier que la syntax est correct grâce à cette commande :

```
sudo apache2ctl configtest
```

S'il est écrit que la Syntax est OK, nous pouvons recharger apache.

c. *Autoriser l'HTTPS dans le firewall ufw*

Dans le cadre de ce projet, nous utilisons un parefeu avec la commande "ufw", pour vérifier ce qui est autorisé, nous pouvons taper la commande suivante :

```
sudo ufw status
```

```
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
WWW                        ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
WWW (v6)                   ALLOW       Anywhere (v6)
```

Dans cette capture d'écran, nous pouvons remarquer que seule la connexion en SSH et la connexion en HTTP sont autorisés. Il nous faut donc autoriser le HTTPS avec les commandes :

```
sudo ufw allow 'WWW Full'
sudo ufw delete allow 'WWW'
```

WWW Full autorise le HTTP et le HTTPS, nous pouvons donc supprimer les droits WWW qui eux n'autorisent que le HTTP. Pour vérifier que ça a bien marché, on réaffiche le statut et si WWW Full apparaît à la place de WWW, alors ça a fonctionné.

d. *Obtention du certificat*

Maintenant que toutes les configurations ont été vérifiés ou modifiés, il est temps de faire une demande de certificat SSL pour obtenir le HTTPS. Pour ce fait, il suffit simplement de taper la commande :

```
sudo certbot --apache -d your_domain -d www.your_domain
```

Après avoir lancé cette commande, il devrait être demander une adresse e-mail et d'accepter les conditions d'utilisations de Let's Encrypt.

Si tout à bien fonctionné, le terminal devrait nous afficher un message de félicitation comme ceci :

```
Congratulations! Your certificate and chain have been saved at:
```

e. <u>Renouvellement automatique du certificat</u>

Maintenant que le certificat à bel et bien été installé, il nous faut pouvoir le renouveler automatiquement. Pour ça, il nous suffit simplement de taper la commande :

```
sudo certbot renew --dry-run
```