Mission 11 : Compréhension et utilisation des protocoles SSL/TLS

Introduction

Les protocoles SSL (Secure Sockets Layer) et TLS (Transport Layer Security) permettent de sécuriser les échanges entre un client et un serveur. Ils ajoutent une couche de protection qui garantit :

La confidentialité des données grâce au chiffrement

L'intégrité des informations transmises

L'authentification du serveur via l'utilisation de certificats

Qu'est-ce qu'un certificat SSL/TLS ?

Un certificat SSL/TLS est un fichier numérique qui atteste de l'identité d'un serveur ou d'un site Web. Il sert à établir une connexion sécurisée et chiffrée entre les deux parties.

Composition d'un certificat SSL/TLS

Un certificat contient plusieurs éléments importants, notamment : Un FQDN (nom de domaine du serveur)
La clé publique utilisée pour le chiffrement
Les dates de validité (émission et expiration)
Le nom de l'autorité de certification (AC) qui l'a délivré
La signature numérique de l'AC, qui garantit son authenticité

PARTIE A: HTTPS

I.Création et utilisation d'un certificat auto-signé

Dans cette partie , nous devons créer un certificat auto-signé en installant OpenSSL et créer un répertoire pour les certificats dans le serveur Web. Avec les commandes suivantes :

```
apt-get install openssl
mkdir /etc/ssl/localcerts # /etc et /ssl existent déjà
```

1. Génération du certificat

```
root@web: DIR=/etc/ssl/localcerts
root@web: openssl req -x509 -newkey rsa:4096 -nodes -keyout
$DIR/mydomainkey.key -out $DIR/mydomaincert.pem -days 365
```

Ce bloc de commande permet de produire à la fois une clé privée et un certificat auto-signé. Voici une présentation claire de chaque paramètre :

-newkey rsa:4096 : crée une nouvelle paire de clés en utilisant un algorithme RSA d'une taille de 4096 bits, offrant une sécurité renforcée.

-keyout : désigne l'emplacement où sera enregistrée la clé privée.

-out : indique le fichier qui accueillera le certificat généré.

-nodes : empêche l'ajout d'une phrase secrète, permettant ainsi au service de lire la clé sans interaction humaine.

-days 365 : fixe la durée de validité du certificat à un an.

⚠ ATTENTION la génération doit être effectuée directement dans le répertoire /etc/ssl/localcerts, afin d'assurer une organisation propre et conforme aux bonnes pratiques système. ⚠

Activation du Vhost SSL par défaut

Nous devons activer le Vhost pour apache2 et le configurer, avec la commande suivante :

```
root@web:a2ensite default-ssl
```

Activation du module ssl pour Apache

```
root@web:a2enmod ssl
```

II.Modifications à Effectuer sur le Vhost

1.Modification sur le Vhost par défaut SSL

Le chemin du fichier :**/etc/apache2/sites-available/default-ssl.conf** . Nous devons entrer dans le fichier et rajouter les deux directives :

```
SSLCertificateFile      /etc/ssl/localcerts/mydomaincert.pem
SSLCertificateKeyFile   /etc/ssl/localcerts/mydomainkey.key
```

.Test Avec Le Navigateur

Ici nous devons tester avec le **https** (wiki, intranet) de nos sites , et y accéder malgré l'interdiction
Cette erreur veut tout simplement dire que la page dans laquelle nous voulons nous rendre a un certificat.

⚠ VÉRIFICATION ESSENTIELLE DU PARE-FEU OPNsense

Avant toute mise en service du certificat, **il faut vérifier impérativement que le port 443 est ouvert et accessible** sur ke pare-feu **OPNsense**.

Ce port est indispensable pour :

- permettre aux navigateurs externes d'établir une connexion HTTPS,
- garantir l'accès sécurisé aux services web présents sur votre serveur,
- éviter tout blocage complet de la couche TLS malgré un certificat correctement généré,
- assurer la résolution correcte des redirections HTTP → HTTPS.

Si le port 443 n'est pas ouvert sur OPNsense, le certificat sera inutilisable, quel que soit sa qualité.