Configuration HTTPS des Sites Web

Partie A

Introduction

Ce rapport présente le processus de configuration du protocole HTTPS pour les sites web d'une organisation.
Le HTTPS (Hypertext Transfer Protocol Secure) est une extension sécurisée du protocole HTTP utilisé pour
la communication sécurisée sur Internet. Nous avons suivi une méthodologie étape par étape pour mettre
en place HTTPS sur les serveurs Apache hébergeant les sites web de l'organisation.

Installation et Configuration de OpenSSL

Pour commencer, nous avons installé OpenSSL sur le conteneur web en utilisant la commande

```
apt install openssl
```

Ensuite, un répertoire localcerts a été créé pour stocker les certificats générés.

Génération du Certificat SSL

Nous avons généré un certificat SSL auto-signé en utilisant la commande suivante,
en intégrant le chemin dans une variable pour une utilisation simplifiée :

```
openssl req -x509 -newkey rsa:4096 -nodes -keyout $DIR/mydomainkey.key -out $DIR/mydomaincert.pem -days 365
```

Explication des paramètres :

newkey rsa:4096 : Crée une clé RSA de 4096 bits.

keyout : Spécifie le fichier de sortie pour la clé privée.

out : Spécifie le fichier de sortie pour le certificat.

nodes : Indique qu'aucune phrase de passe n'est utilisée pour le déverrouillage.

days 365 : Définit la durée de validité du certificat à 365 jours.

Configuration d'Apache pour HTTPS

Nous avons modifié le fichier default-ssl.conf pour activer le Virtual Host SSL et le module SSL pour

Apache. Les étapes comprenaient :

- Décommenter la ligne SSLEngine on.
- Modifier les chemins des certificats SSL.

Activation du Virtual Host SSL et du Module SSL

Après avoir effectué les modifications, les commandes suivantes ont été utilisées pour activer le Virtual Host SSL et le module SSL dans Apache :

1. a2ensite default-ssl : Active le Virtual Host SSL.
2. a2enmod ssl : Active le module SSL.

Redémarrage d'Apache et Vérification des Ports

Nous avons redémarré Apache à l'aide de la commande :

```
systemctl restart apache2
```

et vérifié que le port 443 était ouvert en utilisant la commande :

```
netstat -nat
```

Configuration des Autres Sites

Nous avons répliqué le processus de configuration HTTPS pour les autres sites de l'organisation, notamment Intranet, Extranet, WWW et Wiki. Pour chaque site, nous avons :

- Copié et modifié le Virtual Host pour le port 443.
- Ajouté les directives SSL pour spécifier les chemins des certificats SSL.

Test de Fonctionnalité

Une fois toutes les configurations effectuées, nous avons testé la fonctionnalité en accédant aux sites web via HTTPS.
Le succès du test garantit que les sites web sont désormais accessibles de manière sécurisée.

Conclusion

Ce rapport détaille le processus complet de configuration du protocole HTTPS pour les sites web de l'organisation.
La mise en œuvre de HTTPS renforce la sécurité des communications et protège la confidentialité des utilisateurs lors de l'accès aux sites web.

Virtual hosts :

WWW

```
<VirtualHost *:80>
    ServerName m2l.org
    ServerAlias www.m2l.org
```

```
    DocumentRoot /var/www/html/www.m2l.org

    ErrorLog /var/log/apache2/www.m2l.org-error.log
    CustomLog /var/log/apache2/www.m2l.org-access.log combined

    <Directory /var/www/html/www.m2l.org/>
        Require all granted
    </Directory>
</VirtualHost>

<VirtualHost *:443>
    ServerName m2l.org
    ServerAlias www.m2l.org
    DocumentRoot /var/www/html/www.m2l.org

    ErrorLog /var/log/apache2/www.m2l.org-error.log
    CustomLog /var/log/apache2/www.m2l.org-access.log combined

    <Directory /var/www/html/www.m2l.org/>
        Require all granted
    </Directory>

SSLEngine on
SSLCertificateFile      /etc/ssl/localcerts/mydomaincert.pem
SSLCertificateKeyFile   /etc/ssl/localcerts/mydomainkey.key

</VirtualHost>
```

Intra

```
<VirtualHost *:80>
    ServerName intranet.m2l.org
    DocumentRoot /var/www/html/intranet.m2l.org

    ErrorLog /var/log/apache2/intranet.m2l.org-error.log
    CustomLog /var/log/apache2/intranet.m2l.org-access.log combined

    <Directory /var/www/html/intranet.m2l.org/>
        Require all granted
        AllowOverride All
    </Directory>
</VirtualHost>

<VirtualHost *:443>
    ServerName intranet.m2l.org
    DocumentRoot /var/www/html/intranet.m2l.org

    ErrorLog /var/log/apache2/intranet.m2l.org-error.log
    CustomLog /var/log/apache2/intranet.m2l.org-access.log combined

    <Directory /var/www/html/intranet.m2l.org/>
```

```
        Require all granted
        AllowOverride All
    </Directory>

SSLEngine on
SSLCertificateFile      /etc/ssl/localcerts/mydomaincert.pem
SSLCertificateKeyFile   /etc/ssl/localcerts/mydomainkey.key

</VirtualHost>
```

extra

```
<VirtualHost *:80>
    ServerName extranet.m2l.org
    DocumentRoot /var/www/html/extranet.m2l.org

    ErrorLog /var/log/apache2/extranet.m2l.org-error.log
    CustomLog /var/log/apache2/extranet.m2l.org-access.log combined

    <Directory /var/www/html/extranet.m2l.org/>
        Require all granted
    </Directory>
</VirtualHost>

<VirtualHost *:443>
    ServerName extranet.m2l.org
    DocumentRoot /var/www/html/extranet.m2l.org

    ErrorLog /var/log/apache2/extranet.m2l.org-error.log
    CustomLog /var/log/apache2/extranet.m2l.org-access.log combined

    <Directory /var/www/html/extranet.m2l.org/>
        Require all granted
    </Directory>

SSLEngine on
SSLCertificateFile      /etc/ssl/localcerts/mydomaincert.pem
SSLCertificateKeyFile   /etc/ssl/localcerts/mydomainkey.key

</VirtualHost>
```

wiki

```
<VirtualHost *:80>
    ServerName wiki.m2l.org
    DocumentRoot /var/www/html/wiki.m2l.org

    ErrorLog /var/log/apache2/wiki.m2l.org-error.log
    CustomLog /var/log/apache2/wiki.m2l.org-access.log combined

    <Directory /var/www/html/wiki.m2l.org/>
```

```
        Require all granted
    </Directory>
</VirtualHost>

<VirtualHost *:443>
    ServerName wiki.m2l.org
    DocumentRoot /var/www/html/wiki.m2l.org

    ErrorLog /var/log/apache2/wiki.m2l.org-error.log
    CustomLog /var/log/apache2/wiki.m2l.org-access.log combined

    <Directory /var/www/html/wiki.m2l.org/>
        Require all granted
    </Directory>
SSLEngine on
SSLCertificateFile      /etc/ssl/localcerts/mydomaincert.pem
SSLCertificateKeyFile   /etc/ssl/localcerts/mydomainkey.key

</VirtualHost>
```

Image de test :

Configuration FTPS

Partie B

Introduction

Ce rapport détaille la configuration du protocole FTPS sur le conteneur FTP d'une infrastructure. FTPS est une extension sécurisée du protocole FTP, utilisant TLS (Transport Layer Security) ou SSL (Secure Sockets Layer) pour sécuriser les transferts de fichiers.

Configuration de FTPS

La configuration de FTPS a suivi un processus méthodique, comprenant les étapes suivantes :

- 1 Création du Répertoire SSL : Un répertoire nommé ssl a été créé dans le répertoire /etc/proftpd/ afin de stocker les certificats nécessaires.

```
mkdir /etc/proftpd/ssl/
```

- 2 Génération du Certificat SSL : Un certificat SSL auto-signé a été généré à l'aide d'OpenSSL, garantissant l'authenticité et la sécurité des communications.

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

- 3 Configuration de ProFTPD : Les fichiers de configuration de ProFTPD ont été ajustés pour

activer et paramétrer le support TLS/SSL. Cela inclut l'activation du moteur TLS la spécification des chemins vers les certificats SSL, et la définition des options TLS.

1. Dans /etc/proftpd/proftpd.conf, le TLS a été activé.
2. Dans /etc/proftpd/tls.conf, les paramètres suivants ont été configurés :
3. TLSEngine a été activé.
4. Le chemin pour TLSLog a été modifié pour enregistrer les connexions chiffrées.
5. Les chemins pour TLSRSACertificateFile et TLSRSACertificateKeyFile ont été configurés pour spécifier les certificats SSL.
6. TLSOptions a été décommenté pour définir les options TLS.

- 4 Redémarrage du Service : Une fois les configurations effectuées, le service ProFTPD a été redémarré pour appliquer les changements et intégrer le support FTPS.

```
systemctl restart proftpd
```

- 5 Test de Fonctionnalité avec FileZilla : Pour vérifier le bon fonctionnement du système, FileZilla a été utilisé pour établir une connexion FTPS. L'identification du certificat SSL lors de la connexion confirme le succès de la configuration.
- Accéder à “Fichier” > “Gestionnaire de Sites” dans FileZilla.
- Entrer l'hôte, l'utilisateur et le mot de passe.
- Lors de la connexion, le certificat SSL doit être affiché pour validation.

Conclusion

La configuration réussie de FTPS renforce la sécurité des transferts de fichiers en assurant le cryptage des données échangées entre le client et le serveur.
Cette implémentation sécurisée garantit la confidentialité des informations lors des échanges de fichiers, renforçant ainsi la sécurité globale du système.

Image de test :