GHIRARDOTTI Emile

TP2 - Mise en place d'un serveur Web sécurisé en DMZ

1. Equipements

- Une DMZ dédiée à l'hébergement Web.
- Pare-feu PFSense jouant le rôle de routeur, pare-feu et DNS.
- Un serveur Debian en DMZ hébergeant deux applications PHP/MySQL fournis dans le TP :
    - site1.185.formsup
    - site2.185.formsup
- Une base de données dédiée par site (gsb1, gsb2) à partir des scripts GSB.
- Un accès aux sites en HTTPS à l'aide d'une autorité de certification locale.
- Un accès SFTP sécurisé pour des développeurs répartis en deux groupes :
    - grp-dev-1 → accès uniquement au site1
    - grp-dev-2 → accès uniquement au site2

2. Architecture réseau

2.1 Plan d'adressage IP

- LAN : 192.168.0.0/24
    - pfSense (LAN) : 192.168.0.254
    - Postes clients : 192.168.0.30
- DMZ : 10.0.0.0/24
    - PFSense (DMZ) : 10.0.0.254
    - Serveur Web Debian (servweb) : 10.0.0.200
- WAN : IP publique ou réseau pédagogique (172.16.185.254)

Passerelles :

- Clients LAN → gateway : 192.168.0.254
- Serveur DMZ → gateway : 10.0.0.254

3. Configuration PFSense

3.1 Interfaces

Dans Interfaces → Assignments :

- LAN : 192.168.0.254 /24
- DMZ : 10.0.0.254 /24
- WAN : 172.16.185.254 /16

3.2 NAT

Pare-feu → NAT → Sortant :

- Mode : Création automatique de règles NAT sortantes
- PFSense génère automatiquement le NAT pour LAN net et DMZ net vers le WAN.

3.3 Règles de filtrage

DMZ

Pare-feu → Règles → DMZ :

- Règle principale (DMZ → Internet) :
    - Action : Pass
    - Interface : DMZ
    - Source : DMZ net
    - Destination : any
    - Protocole : any
    - Description : DMZ vers Internet

Éventuellement, on peut ajouter :

- Blocage DMZ → LAN :
    - Action : Block
    - Source : DMZ net
    - Destination : LAN net

LAN

Pare-feu → Règles → LAN :

- Autoriser le LAN à accéder à la DMZ (administration, tests) :
    - Action : Pass
    - Source : LAN net
    - Destination : DMZ net
    - Protocole : any

3.4 NAT : accès WAN vers DMZ

Pare-feu > NAT > Transfert de port :

- HTTPS vers le serveur :
    - Interface : WAN
    - Protocole : TCP
    - Port destination : 443
    - IP de redirection : 10.0.0.200
    - Port de redirection : 443
    - Créer la règle de pare-feu associée

4. Configuration du serveur Debian en DMZ

4.1 Réseau

Configuration réseau : /etc/network/interfaces :

address 10.0.0.200/24

gateway 10.0.0.254

dns 172.16.0.100

4.2 Installation des services pour le serveur WEB

sudo apt update

sudo apt install -y apache2 mariadb-server php libapache2-mod-php php-mysql unzip

sudo systemctl enable apache2 mariadb --now

5. Déploiement des sites Web

5.1 Décompression

cd /var/www

sudo unzip : ~/site1.zip -d /var/www/site1

sudo unzip : ~/site2.zip -d /var/www/site2

sudo chown -R www-data:www-data /var/www/site1 /var/www/site2

sudo chmod -R 755 /var/www/site1 /var/www/site2

Les sites contiennent notamment des contrôleurs PHP (cAccueil.php, cSeConnecter.php, …) et les répertoires images/, include/, styles/.

6. Bases de données GSB

6.1 Création des bases et des comptes

Extrait des commandes du TP :

sudo mysql -u root -p

Dans MariaDB :

```
DROP DATABASE IF EXISTS gsb1;
DROP DATABASE IF EXISTS gsb2;

CREATE DATABASE gsb1 CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
CREATE DATABASE gsb2 CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;

CREATE USER 'gsb1_user'@'localhost' IDENTIFIED BY 'Azerty31';
CREATE USER 'gsb2_user'@'localhost' IDENTIFIED BY 'Azerty31';

GRANT ALL PRIVILEGES ON gsb1.* TO 'gsb1_user'@'localhost';
GRANT ALL PRIVILEGES ON gsb2.* TO 'gsb2_user'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

6.2 Import de la structure et des données

```
sudo mysql -u gsb1_user -p gsb1 < /var/www/gsb_structure.sql
sudo mysql -u gsb1_user -p gsb1 < /var/www/gsb_insert.sql

sudo mysql -u gsb2_user -p gsb2 < /var/www/gsb_structure.sql
sudo mysql -u gsb2_user -p gsb2 < /var/www/gsb_insert.sql
```

6.3 Création d’utilisateurs GSB

Dans MariaDB :

```
USE gsb1;

INSERT INTO Visiteur (id, nom, prenom, login, mdp, adresse, cp, ville,
dateEmbauche)

VALUES ('e99', 'OURLIAC', 'Julien', 'julien', ' Azerty31',

    '5 avenue dupont', '31000', 'Toulouse', '2025-01-01');


USE gsb2;

INSERT INTO Visiteur (id, nom, prenom, login, mdp, adresse, cp, ville,
dateEmbauche)

VALUES ('e99', 'OURLIAC', 'Julien', 'julien', ' Azerty31',

    '5 avenue dupont', '31000', 'Toulouse', '2025-01-01');
```

7. Connexion des sites aux bases

Dans les fichiers PHP de connexion des sites (ex. include/fct.inc.php), on configure :

site1 (gsb1)

```
$host = "localhost";

$user = "gsb1_user";

$pass = " Azerty31";

$db   = "gsb1";


$conn = mysqli_connect($host, $user, $pass, $db);

if (!$conn) {

   die("Échec de connexion : " . mysqli_connect_error());

}
```

site2 (gsb2)

```
$host = "localhost";

$user = "gsb2_user";

$pass = " Azerty31";
```

$db   = "gsb2";

Les pages de connexion (cSeConnecter.php) utilisent ensuite la table Visiteur.

8. DNS interne

8.1 DHCP LAN

Services → DHCP Server → LAN :

- DNS Servers :

On ajoute :

- Host : site1 – Domain : 185.formsup – IP : 10.0.0.200
- Host : site2 – Domain : 185.formsup – IP : 10.0.0.200
- (Optionnel) Host : servweb – Domain : 186.formsup – IP : 10.0.0.200

Tests depuis un poste LAN :

nslookup site1.185.formsup

nslookup site2.185.formsup

Les deux doivent pointer vers 10.0.0.200.

9. VirtualHosts Apache (HTTP/HTTPS)

9.1 VirtualHosts HTTPS

/etc/apache2/sites-available/site1-ssl.conf :

```
<VirtualHost *:443>
    ServerName site1.185.formsup
    DocumentRoot /var/www/site1

    DirectoryIndex cAccueil.php
    <Directory /var/www/site1>
        AllowOverride All
        Require all granted
    </Directory>

    SSLEngine on
    SSLCertificateFile      /etc/ssl/certs/site1.cert.pem
    SSLCertificateKeyFile   /etc/ssl/private/site1.key.pem
    SSLCACertificateFile    /etc/ssl/localCA/certs/ca.cert.pem

    ErrorLog ${APACHE_LOG_DIR}/site1_ssl_error.log
    CustomLog ${APACHE_LOG_DIR}/site1_ssl_access.log combined
</VirtualHost>
```

/etc/apache2/sites-available/site2-ssl.conf :

```
<VirtualHost *:443>
    ServerName site2.185.formsup
    DocumentRoot /var/www/site2

    DirectoryIndex cAccueil.php
    <Directory /var/www/site2>
        AllowOverride All
```

```
        Require all granted
    </Directory>

    SSLEngine on
    SSLCertificateFile      /etc/ssl/certs/site2.cert.pem
    SSLCertificateKeyFile   /etc/ssl/private/site2.key.pem
    SSLCACertificateFile    /etc/ssl/localCA/certs/ca.cert.pem

    ErrorLog ${APACHE_LOG_DIR}/site2_ssl_error.log
    CustomLog ${APACHE_LOG_DIR}/site2_ssl_access.log combined
</VirtualHost>
```

Activation :

```
sudo a2enmod ssl headers
sudo a2ensite site1-ssl.conf site2-ssl.conf
sudo systemctl reload apache2
```

9.2 Redirection HTTP vers HTTPS

/etc/apache2/sites-available/site1-http.conf :

```
<VirtualHost *:80>
    ServerName site1.185.formsup
    Redirect permanent / https://site1.185.formsup/
</VirtualHost>
```

/etc/apache2/sites-available/site2-http.conf :

```
<VirtualHost *:80>
    ServerName site2.185.formsup
    Redirect permanent / https://site2.185.formsup/
</VirtualHost>
```

Activation :

```
sudo a2ensite site1-http.conf site2-http.conf
```

sudo systemctl reload apache2

10. Autorité de certification locale (AC)

10.1 Structure

```
sudo mkdir -p /etc/ssl/localCA/{certs,crl,newcerts,private}
sudo chmod 700 /etc/ssl/localCA/private
sudo touch /etc/ssl/localCA/index.txt
echo 1000 | sudo tee /etc/ssl/localCA/serial >/dev/null
```

10.2 Clé et certificat de l'AC

```
sudo openssl genrsa -out /etc/ssl/localCA/private/ca.key.pem 4096
sudo chmod 600 /etc/ssl/localCA/private/ca.key.pem

sudo openssl req -x509 -new -sha256 -days 3650 \
  -key /etc/ssl/localCA/private/ca.key.pem \
  -out /etc/ssl/localCA/certs/ca.cert.pem \
  -subj "/C=FR/ST=Occitanie/L=Toulouse/O=SIO186/OU=IT/CN=SIO186-Local-CA/emailAddress=admin@186.formsup"
```

10.3 Certificats serveurs

Exemple pour site1.185.formsup :

```
sudo tee /etc/ssl/site1.cnf >/dev/null <<'EOF'
[req]
default_bits = 2048
prompt = no
default_md = sha256
distinguished_name = dn
req_extensions = req_ext

[dn]
C=FR
ST=Occitanie
```

```
L=Toulouse
O=SIO186
OU=Web
CN=site1.185.formsup

[req_ext]
subjectAltName = @alt_names

[alt_names]
DNS.1 = site1.185.formsup
EOF

sudo openssl genrsa -out /etc/ssl/private/site1.key.pem 2048
sudo openssl req -new -key /etc/ssl/private/site1.key.pem \
  -out /etc/ssl/site1.csr.pem -config /etc/ssl/site1.cnf

sudo openssl x509 -req -in /etc/ssl/site1.csr.pem \
  -CA /etc/ssl/localCA/certs/ca.cert.pem -CAkey /etc/ssl/localCA/private/ca.key.pem -
CAcreateserial \
  -out /etc/ssl/certs/site1.cert.pem -days 825 -sha256 \
  -extfile /etc/ssl/site1.cnf -extensions req_ext
```

Même principe pour site2.185.formsup.

10.4 Installation de l’AC sur les PC

Sous Windows :

1. Copier ca.cert.pem.
2. Double-clic → Installer le certificat.
3. Choisir Ordinateur local.
4. Stocker dans Autorités de certification racines de confiance.

Les navigateurs considèrent alors les certificats comme valides si le nom du site correspond au certificat.

11. Accès SFTP sécurisé pour les développeurs

11.1 Objectif

- Deux groupes de développeurs :
  - grp-dev-1 → accès uniquement au site1
  - grp-dev-2 → accès uniquement au site2
- Utilisateurs :
  - dev-1-1, dev-1-2 (groupe 1)
  - dev-2-1, dev-2-2 (groupe 2)
- Accès SFTP uniquement (pas de shell).
- Chaque groupe est chrooté dans son propre répertoire.

11.2 Groupes

```
sudo groupadd grp-dev-1
sudo groupadd grp-dev-2
sudo groupadd sftp-users
```

11.3 Utilisateurs développeurs

```
# Groupe 1 → site1
sudo useradd -m -s /bin/false -G grp-dev-1,sftp-users dev-1-1
sudo useradd -m -s /bin/false -G grp-dev-1,sftp-users dev-1-2
sudo passwd dev-1-1
sudo passwd dev-1-2

# Groupe 2 → site2
sudo useradd -m -s /bin/false -G grp-dev-2,sftp-users dev-2-1
sudo useradd -m -s /bin/false -G grp-dev-2,sftp-users dev-2-2
sudo passwd dev-2-1
sudo passwd dev-2-2
```

11.4 Droits sur les sites

sudo chown -R root:grp-dev-1 /var/www/site1

sudo chmod -R 770 /var/www/site1

sudo chown -R root:grp-dev-2 /var/www/site2

sudo chmod -R 770 /var/www/site2

11.5 Répertoires de chroot

Architecture :

/sftp-dev1/

site1/ (bind vers /var/www/site1)

/sftp-dev2/

site2/ (bind vers /var/www/site2)

Création :

sudo mkdir /sftp-dev1 /sftp-dev2

sudo chown root:root /sftp-dev1 /sftp-dev2

sudo chmod 755 /sftp-dev1 /sftp-dev2

sudo mkdir /sftp-dev1/site1

sudo mkdir /sftp-dev2/site2

sudo mount --bind /var/www/site1 /sftp-dev1/site1

sudo mount --bind /var/www/site2 /sftp-dev2/site2

Rendre les bind mounts permanents :

echo "/var/www/site1 /sftp-dev1/site1 none bind 0 0" | sudo tee -a /etc/fstab

echo "/var/www/site2 /sftp-dev2/site2 none bind 0 0" | sudo tee -a /etc/fstab

11.6 Configuration SSH

Éditer /etc/ssh/sshd_config et ajouter à la fin :

Match Group grp-dev-1

```
    ChrootDirectory /sftp-dev1
    ForceCommand internal-sftp
    X11Forwarding no
    AllowTcpForwarding no

Match Group grp-dev-2
    ChrootDirectory /sftp-dev2
    ForceCommand internal-sftp
    X11Forwarding no
    AllowTcpForwarding no
```

Redémarrer SSH :

```
sudo systemctl restart sshd
```

11.7 Résultat

- dev-1-1 / dev-1-2 :
    - chroot dans /sftp-dev1
    - ne voient que site1/
    - peuvent modifier les fichiers PHP/CSS/Images du site1.
- dev-2-1 / dev-2-2 :
    - chroot dans /sftp-dev2
    - ne voient que site2/
    - peuvent modifier le site2 uniquement.

Aucun utilisateur dev n’a accès à un shell (/bin/false) ni aux fichiers système.

12. Tests finaux

1. DNS :
2. nslookup site1.185.formsup
3. nslookup site2.185.formsup
4. HTTPS :
    - https://site1.185.formsup → cadenas valide, connexion possible avec un compte GSB.
    - https://site2.185.formsup idem.
5. SFTP :
    - Connexion avec dev-1-1 :
        - Hôte : sftp://servweb.185.formsup ou IP DMZ
        - Dossier racine : contient uniquement site1/.
    - Connexion avec dev-2-2 :
        - Ne voit que site2/.
6. Isolation DMZ :
    - Depuis la DMZ, le ping vers Internet fonctionne.
    - L’accès direct à LAN net est bloqué si la règle Block DMZ → LAN est activée.

13. Conclusion

Ce TP a permis de :

- Concevoir une architecture réseau segmentée (LAN / DMZ / WAN) avec pfSense.
- Configurer MariaDB avec deux bases séparées et des comptes dédiés.
- Mettre en place un DNS interne permettant l’accès par noms de domaine.
- Déployer un serveur Web Debian en DMZ, hébergeant deux applications PHP/MySQL.
- Sécuriser les sites en HTTPS via une autorité de certification locale.
- Implémenter un accès SFTP sécurisé, segmenté par groupes de développeurs, avec chroot et droits finement contrôlés.

L’ensemble reproduit une architecture professionnelle complète :
hébergement Web, réseau segmenté, chiffrement, contrôle d’accès et isolation des prestataires externes.