Compte-rendu Mise en place d'une infrastructure Web et SFTP sécurisée en DMZ

Table des matières

1. Contexte et Objectifs

Le but de ce tp est de créer une infrastructure hébergeant un serveur web avec deux sites sécurisés. Voici les 3 axes de sécurisation :

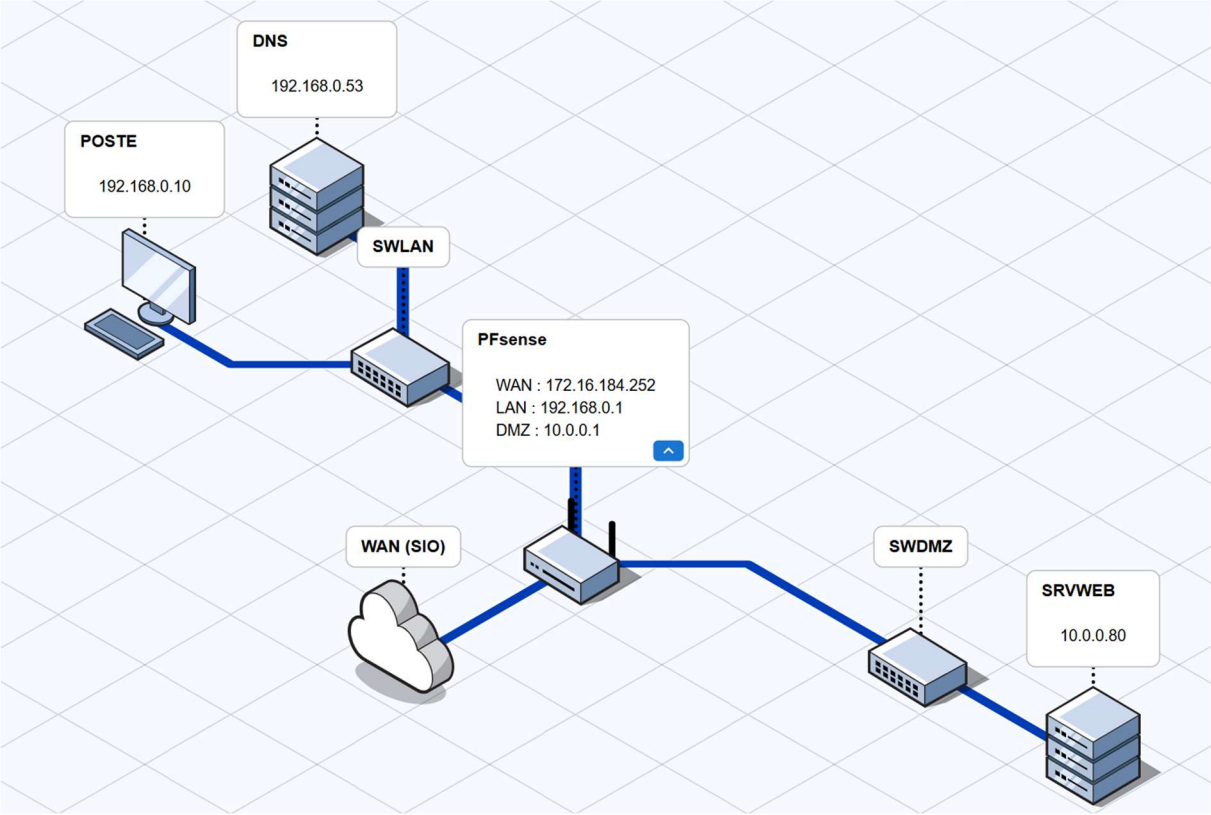
Routage : Isolation des zones via un pare-feu pfSense[3].

HTTPS : Publication des sites en HTTPS via le serveur web Apache[4].

SFTP : Mise en place d'un accès SFTP restreint pour les développeurs[5].

2. Architecture Réseau

L'infrastructure est répartie en deux zones dont voici le schéma.



Voici la configuration des interfaces du routeur pfSense et des équipements associés:

Zone	Interface pfSense	Réseau	Équipements / Description
WAN	172.16.184.252	172.16.0.0/16	Passerelle vers le réseau SIO / Internet
LAN	192.168.0.1	192.168.0.0/24	Réseau interne. PC Client et DNS
DMZ	10.0.0.1	10.0.0.0/24	Zone serveur. Serveur Debian

Note : Le serveur DNS utilisé sur le WAN est le 172.16.0.100

3. Configuration PFSense

Voici les règles de filtrage du routeur

Interface DMZ

Règle de passage de la DMZ vers Internet:

Action : Pass

Interface : DMZ

Source : DMZ net

Destination : WAN

Protocole : tcp

Interface LAN

LAN vers DMZ

Action : Pass

Source : LAN net

Destination : DMZ net

Protocole : any

Passons maintenant aux règles de NAT

HTTPS du wan au serveur WEB :

Interface : WAN

Protocole : TCP

Port destination : 443

IP de redirection : 10.0.0.80

Port de redirection : 443

4. Configuration du serveur Web

Voici le fichier /etc/network/interfaces (conf carte réseau)

address 10.0.0.80/24

gateway 10.0.0.1

dns 172.16.0.100

Mises a jour de la vm et installation des paquets

apt update

apt upgrade

apt install vim

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

systemctl enable apache2 mariadb

5. Déploiement des pages WEB

cd /var/www/html

unzip : ~/site1TPWEB.zip -d ./site1

unzip : ~/site2TPWEB.zip -d ./site2

chown -R www-data:www-data ./*

chmod -R 755 ./*

Les sites sont fournis directement dans le tp.

6. Bases de données GSB

Je vais utiliser les commandes fournis dans le tp:

mariaDB

Dans MariaDB :

CREATE DATABASE gsb1;

CREATE DATABASE gsb2;

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';

Je vais dorénavant créer les utilisateurs de la base de données

Dans MariaDB :

USE gsb1;

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

VALUES ('e21', 'LABARRE', 'Robin', 'labarrr', ' Azerty31',

'27 rue pierre bourthoumieux', '31300', 'Toulouse', '2025-06-18');

USE gsb2;

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

VALUES ('e21', 'LABARRE', 'Robin', 'labarrr', ' Azerty31',

'27 rue pierre bourthoumieux', '31300', 'Toulouse', '2025-06-18');

7. Connexion des bases de données

Dans les configurations php :

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";
```

8. DNS interne

J'ajoute un hote pour le site 1 et un alias pour le site 2

site1 ; 184.formsup ; 10.0.0.200

site2 ; 184.formsup hote site1.184.formsup

9. Configuration des Virtual hosts apache

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

```
<VirtualHost *:443>
```

```
    ServerName site1.184.formsup
    DocumentRoot /var/www/html/site1

    DirectoryIndex cAccueil.php
    <Directory /var/www/html/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 site2.184.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>
```

On active les sites

```
a2enmod ssl headers
a2ensite site1-ssl.conf site2-ssl.conf
```

Je configure aussi les hotes du port 80 pour rediriger vers https

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

```
<VirtualHost *:80>
    ServerName site1.184.formsup
    Redirect permanent / https://site1.184.formsup/
</VirtualHost>
```

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

```
<VirtualHost *:80>
    ServerName site2.184.formsup
    Redirect permanent / https://site2.184.formsup/
</VirtualHost>
```

On active les redirections

```
a2ensite site1-http.conf site2-http.conf
systemctl restart apache2
```

10. Certificats

mkdir -p /etc/ssl/localCA/{certs,crl,newcerts,private}

chmod 700 /etc/ssl/localCA/private

touch /etc/ssl/localCA/index.txt

echo 1000 | sudo tee /etc/ssl/localCA/serial >/dev/null

openssl genrsa -out /etc/ssl/localCA/private/ca.key.pem 4096

chmod 600 /etc/ssl/localCA/private/ca.key.pem

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=SIO184/OU=IT/CN=SIO184-Local-CA/emailAddress=admin@184.formsup"

Coté site 1

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=SIO184

OU=Web

CN=site1.184.formsup


[req_ext]

subjectAltName = @alt_names


[alt_names]

DNS.1 = site1.184.formsup

EOF


openssl genrsa -out /etc/ssl/private/site1.key.pem 2048

openssl req -new -key /etc/ssl/private/site1.key.pem \

 -out /etc/ssl/site1.csr.pem -config /etc/ssl/site1.cnf


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
```

On réitère l'opération pour le deuxième site

On met l'autorité de certification sur les ordinateurs car de base un certificat autosigné n'est pas valide on le rajoute donc manuellement

11. Accès SFTP sécurisé pour les développeurs

Un groupe pour chaque équipe de devs (comme si chaque site n'était pas a la même société) :

grp-dev-(1 / 2)

Utilisateurs :

dev-1/2-1, dev-1/2-2 // deux utilisateurs par groupe

On empeche l'acces au shell on ne laisse que le sftp de plus on fait sortir les utilisateurs dans leur répèrtoire réspectifs.

groupadd grp-dev-1/2

groupadd sftp-users

Pour le site 1

useradd -m -s /bin/false -G grp-dev-1,sftp-users dev-1-1

useradd -m -s /bin/false -G grp-dev-1,sftp-users dev-1-2

passwd dev-1-1

passwd dev-1-2

On recommence pour le site 2

chown -R root:grp-dev-1 /var/www/html/site(1/2)

chmod -R 770 /var/www/html/site(1/2)

Création des répertoires

mkdir /sftp-dev1 /sftp-dev2

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

chmod 755 /sftp-dev1 /sftp-dev2

sudo mkdir /sftp-dev(1/2)/site(1/2)

mount --bind /var/www/html/site(1/2) /sftp-dev(1/2)/site(1/2)

On va maintenant configurer le ssh via le fichier /etc/ssh/sshd_config, on rajoute :

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

Et on redemarre le service

systemctl restart sshd

Une fois ces manipulations effectuées on constate que tout est bien paramétré chaque utilisateur sort du sftp directement dans son répertoire (site 1 pour le grp dev1 et site2 pour le groupe dev2) donc, ils n'ont pas accès au shell et sont cloisonés dans leur répertoires

13. Conclusion

On a donc pu réaliser une architecture réseau avec une lan et une dmz séparés d'un routeur, de plus les deux sites sont mis en place sur des bases de données distinctes, configuré un DNS pour résoudre les noms de domaine, mis en place l'accès SFTP en bloquant le shell avec des répertoires bien restreints. Mais surtout sécurisé les sites avec des certificats autosignés ce qui fut la principale difficulté de ce tp pour moi, mettre en place le serveur de certificat m'as pris plusieurs heures car j'ai utilisé un serveur debian13 : une bonne partie des commandes fournies étaient déprécié et j'ai donc du épier des documentations et stackoverflow a la recherche d'utilisateurs avec les mêmes problèmes que moi