COMPTE RENDU

Mise en place d'un hébergement mutualisé

Par Alexandre BEROT-ARMAND
BTS SIO 2
Lycée Louis PERGAUD

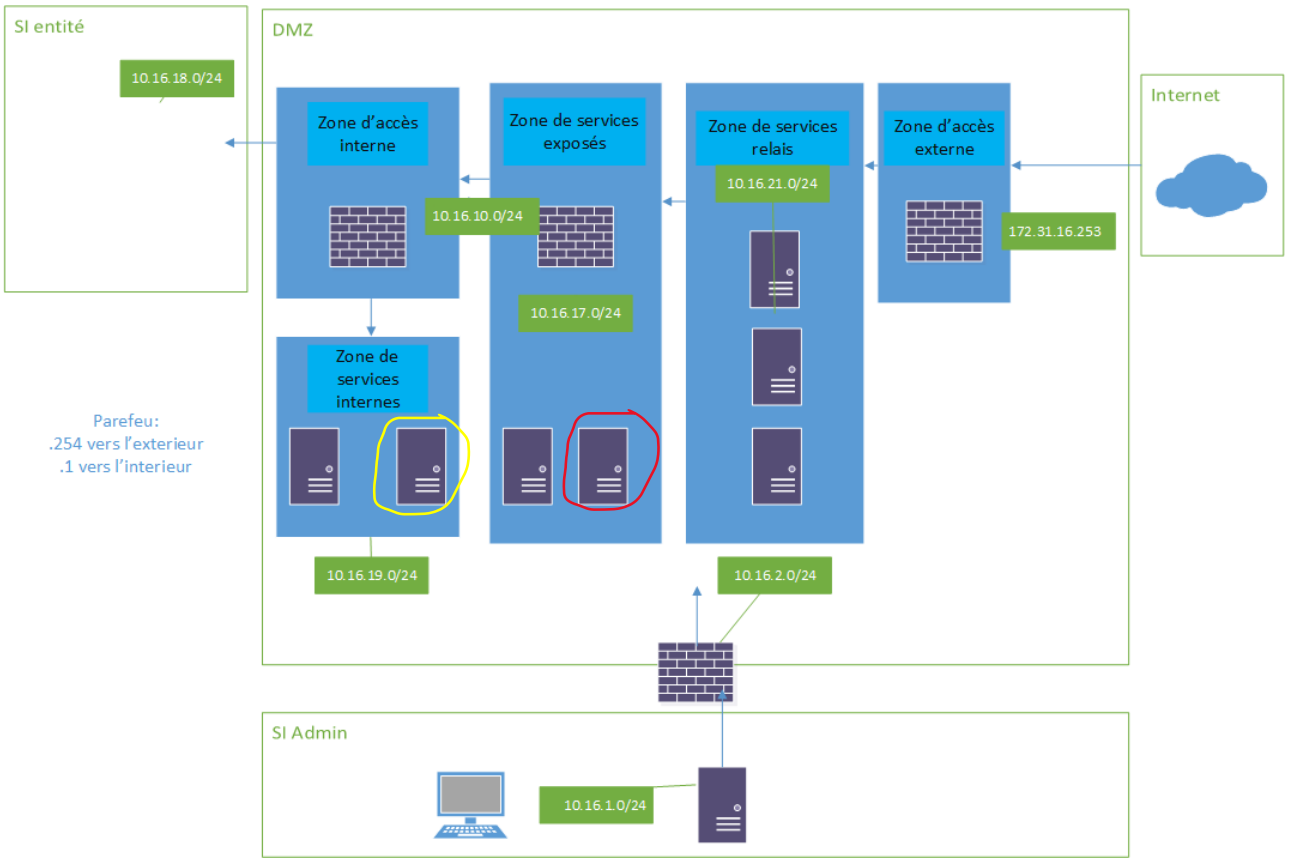
TABLE DES MATIERES

INTRODUCTION

Dans le TP précédent, nous avons vu comment rajouter des niveaux à notre DMZ pour en augmenter la sécurité.

Dans ce TP, nous allons dans un premier temps ajouter un serveur web dans la zone des services exposés puis, dans une seconde partie, un serveur de base de données pour le serveur web.

Voici l'emplacement des services concernés :



Le serveur en rouge est le serveur Web, celui en jaune la base de données. On note que les deux services sont dans deux partie différentes de la DMZ.
Sur le serveur WEB, il y aura 3 sites différents Pro, Extra et Rep.

N.B. : la configuration sera montrée sur le site Pro, mais le fonctionnement est le même pour les autres.

CREATION DE LA VM SERVEUR WEB ET PLACEMENT

Je parle de cette étape sans trop la détailler puisqu'à ce niveau, elle est devenue triviale. Une copie de Template d'une Debian sans interface graphique a été utilisée.

J'ai choisi de la faire à l'extérieur de la DMZ pour une question de rapidité mais il est possible de la positionner à son emplacement final puis de faire une ACL temporaire qui permet à la VM un accès à internet. Il faut aussi penser à installer un serveur Apache.

Ici, cette machine sera nommée Web-lab.

CREATION DES SITES DE BASE ET ACCES

Notre machine est installée, Apache est présent dessus et la machine est positionnée là où elle est censée être (Attention au VMBR).

La première chose à faire est de créer des dossiers pour les sites. Ainsi on va créer « /sites », puis dans le dossier « /sites » on va créer « /pro », « /rep » et « /extra ». Dans un souci de clarté, seule la configuration du site Pro sera donnée.

Dans chacun des dossiers, on crée un fichier HTML correspondant :

```
 GNU nano 5.4                                    /sites/pro/index.html
<html>
<body>
pro
</body>
</html>
```

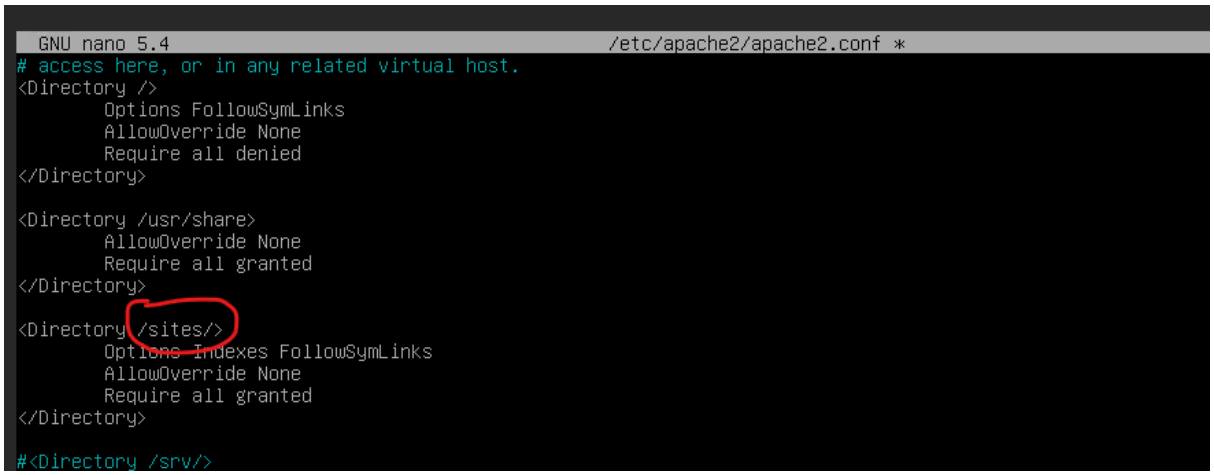
Puis on va aller dans le dossier « /etc/apache2/sites-enabled » pour trouver le fichier « 000-default.conf » :

```
root@debian:~# cd /etc/apache2/sites-enabled/
root@debian:/etc/apache2/sites-enabled# ls
000-default.conf
```

Ensuite on va copier ce fichier et changer les noms pour avoir « pro.conf », « extra.conf » et « rep.conf ». Ce fichier de configuration permet de donner les noms des sites ainsi que leurs emplacements (dans « /sites/ » actuellement) :

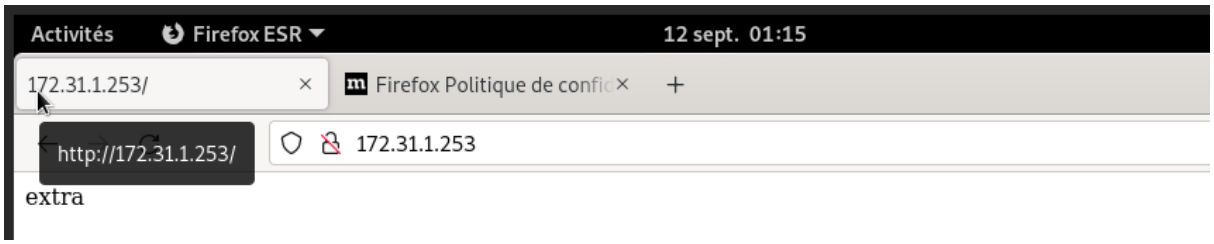
```
 GNU nano 5.4
<VirtualHost pro.gsb.fr:80>

        ServerAdmin webmaster@localhost
        DocumentRoot /sites/pro

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

Après cela, il faut autoriser le serveur apache à naviguer sur nos sites :



Puis on déclare les hôtes dans le DNS du serveur apache (/etc/hosts) :

On reboot le serveur et on vérifie que le serveur marche depuis la Debian externe :



Note : L'IP n'est pas bonne car j'ai oublié de faire le screen de la réussite de cette étape.

On note que le fichier host de la Debian externe n'a pas été modifié, donc il faut se connecter avec l'adresse IP du site. Comme les 3 sites ont la même adresse IP, il va choisir un site parmi les trois et n'ouvrir que celui-là.

Il suffit de modifier ce fichier (toujours /etc/hosts) en donnant l'adresse IP et le nom des sites pour régler ce problème :

```
172.31.16.253 pro.gsb.fr
172.31.16.253 extra.gsb.fr
172.31.16.253 rep.gsb.fr
```

Ainsi, lorsque l'on cherche http//:pro.gsb.fr sur Firefox :

Nous avons donc trois sites web fonctionnels.

CREATION DE LA VM BDD SERVEUR WEB ET CONFIGURATION DE MARIADB

Comme pour la machine juste avant, on utilisera un clone de Template Debian sans interface graphique. On place la machine puis on crée une ACL temporaire (ou un positionnement externe temporaire) pour les mises à jour. Elle sera nommée « bdd-lab » ici.

Il faut installer MariaDB sur cette machine et créer une base de données puis créer 3 tables (une par site) :

```
MariaDB [gsb]> create table `table_pro` ( id int(11) not null, value text not null);
Query OK, 0 rows affected (0.021 sec)

MariaDB [gsb]> alter table table_pro add primary key (id);
Query OK, 0 rows affected (0.036 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [gsb]> alter table table_pro modify id int(11) not null auto_increment;
Query OK, 0 rows affected (0.052 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [gsb]> commit;
Query OK, 0 rows affected (0.000 sec)
```

Puis changer l'adresse d'écoute TCP (dans /etc/mysql/mardiadb) :

```
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0
```

Puis on redémarre le service.

Suite à cela, il y a une période où il faut changer les pages web sur le serveur Web pour coller du PHP dedans pour que la bdd ait une utilité.

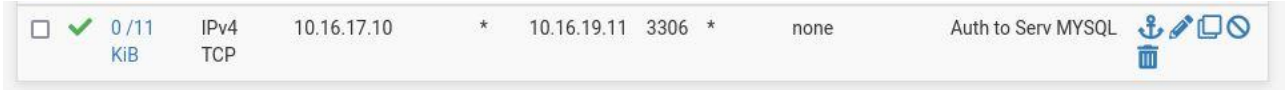
CONFIGURATION DE L'ACCES DE WEB-LAB A BDD-LAB

Web-lab et bdd-lab sont dans deux zones différentes de la DMZ séparées par deux pare-feux (vérifiable sur le diagramme du début).

Pour permettre l'accès du serveur à sa base de données, il faut donc autoriser les requêtes du serveur qui utilise le port de Mysql (le port 3306) ainsi que créer une route sur le PF intermédiaire pour lui donner la route vers la base de données qu'il ne connait pas. On commence par créer cette route :

Puis on ajoute l'autorisation de sortir les requêtes vers le port 3306 de la machine Web-lab vers la machine bdd-lab :

Enfin, nous ouvrons l'accès sur le PF interne du côté du WAN :



Puis on teste l'accès effectif de la BDD en testant les nouveaux sites comprenant du PHP :

Rappel : à ce stade, retirer les ACL temporaires pour les différentes configurations si on a oublié de le faire.

[FACULTATIF] CONFIGURATION DU RESEAU POUR UN ACCES SSH

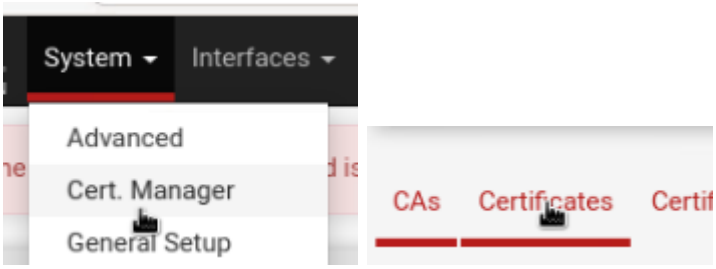
Pour la demande de signature de certificat entre le PF Admin et le Weblab, il est plus rapide d'utiliser une connexion SSH entre les deux pour copier les contenus des fichiers CSR.

Donc dans cette étape facultative, on configure une route entre la PF Admin et le réseau du Weblab (avec la PF intermédiaire comme passerelle) puis on crée une ACL qui autorise le passage SSH depuis le WAN du PF admin sur l'OTP2. Enfin on se connecte en SSH au serveur pour pouvoir copier les contenus des fichiers CSR

PASSAGE DES SITES EN HTTPS

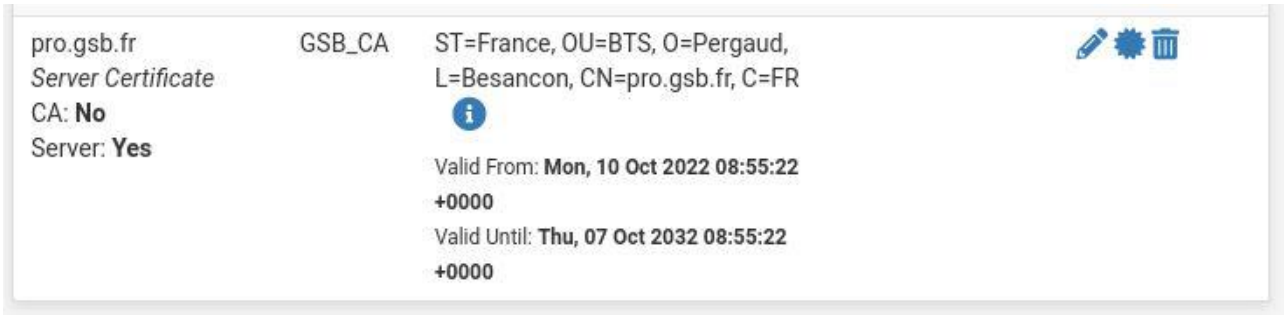
Après la création de trois sites en http qui accèdent à une base sur un serveur différent, il serait judicieux d'accroitre la sécurité en passant les sites en HTTPS. Pour cela, il faut faire plusieurs configurations différentes.

Commençons par créer une autorité d'administration, sachant qu'une PFsense peut faire le job et qu'il faut qu'elle soit relativement sécurisée, la PFsense admin fera l'affaire. On va donc sur la PFsense Admin :



Puis on crée une demande de signature.

Ensuite on crée un certificat par site via le CSR sur le Weblab que l'on copie sur la PF admin (d'où le SSH) :



On exporte le certificat sur le serveur :

```
root@debian-serveur:/home/user# nano /etc/apache2/tls/pro.gsb.fr.crt
root@debian-serveur:/home/user#
```

Puis on configure les virtual hosts pour indiquer les chemins des certificats et des clefs :

```
<VirtualHost pro.gsb.fr:443>

        ServerAdmin webmaster@localhost
        DocumentRoot /sites/pro
        ServerName pro.gsb.fr

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined


        SSLEngine on

        SSLCertificateFile /etc/apache2/tls/pro/pro.gsb.fr.crt

        SSLCertificateKeyFile /etc/apache2/tls/pro/pro.gsb.fr.key


</VirtualHost>
```

On vérifie que tous les fichiers sont présents :

```
./pro:
pro.gsb.fr.key  pro.gsb.fr.crt  pro.gsb.fr.csr
```

Et on fait le test final depuis la debian externe :

https://rep.gsb.fr

Logiciel GSB Rep

id	valeur
Pas encore d'enregistrement	

ajouter

https://extra.gsb.fr

Logiciel GSB Extra

id	valeur
Pas encore d'enregistrement	

ajouter

https://pro.gsb.fr

Logiciel GSB PRO

id	valeur
Pas encore d'enregistrement	

ajouter