Mission 15 : Fail2ban/Portsentry

Pré-requis

Afin de pouvoir réaliser cette mission il faut :

- Un serveur sous Debian 13
- Un routeur sous Debian 13
- La configuration IP des deux machines conforme au cahier des charges,
- La configuration DNS des deux machines conforme au cahier des charges,

Introduction

Nous allons installer Fail2ban qui est un mécanisme de protection automatique du serveur contre les attaques par brute force et Portsentry est une solution pour se protéger des scans de ports

> **Avantage de Fail2ban :**
> * Protège les serveurs web des attaques brutes forces.
> * Limite les tentatives de connexions.
> * Simple à configurer.

> **Avantage de Portsentry :**
> * Protège les serveur web des scans de port.
> * Renforce la sécurité.
> * Simple à configurer.

Documentation de l'installation et configuration de Fail2ban

Fail2ban

A.1 Wordpress.

Nous devons installer Fail2ban sur les serveurs webs, qui sont sensible d'être attaqué en brute force par des personnes malveillante.

Mettre à jour le conteneur :

```
apt update
apt upgrade
```

Installer Fail2ban

```
apt install fail2ban
```

Création d'un fichier jail.local

La création de ce fichier sert à garder la configuration des jails en cas de mise à jour des conteneur web.

```
cd /etc/fail2ban/
cp jail.conf jail.local
```

Configuration du fichier jail.local

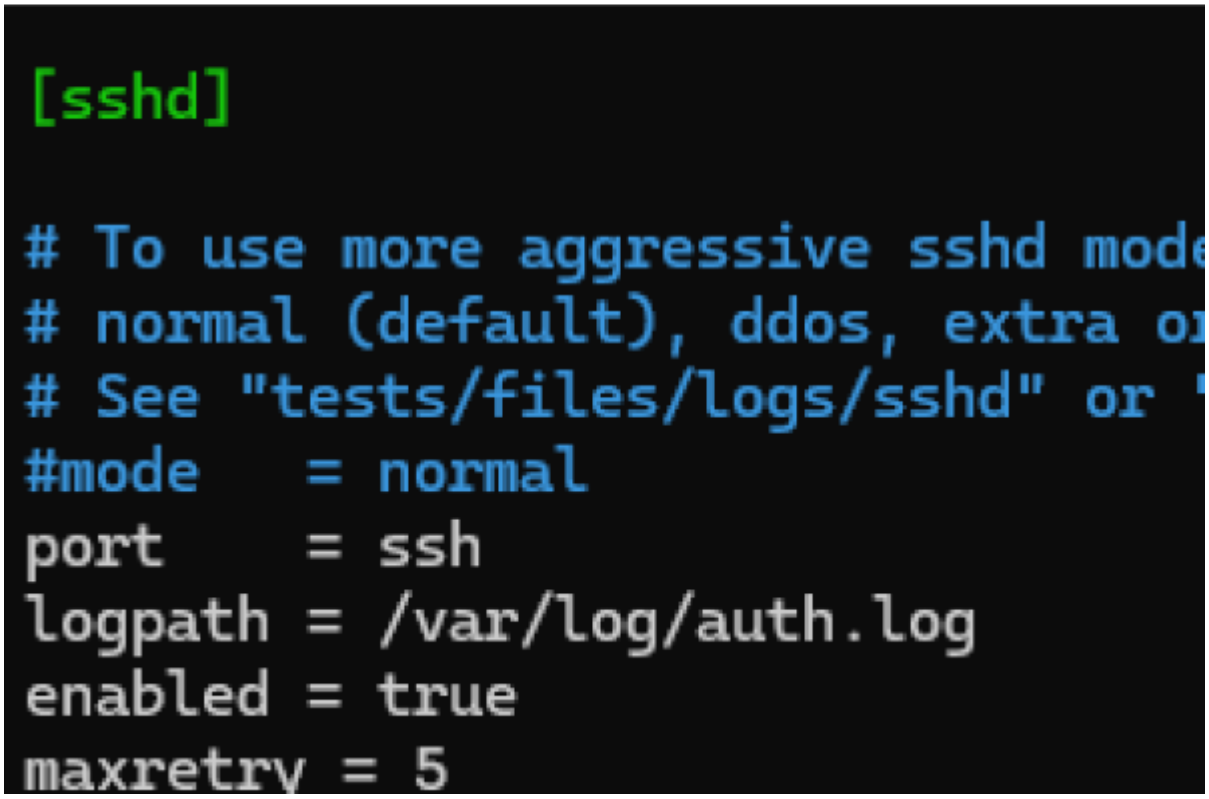
```
nano /etc/fail2ban/jail.local
```

Je vais ajouter une adresse IP dans la directive ignoreip afin que le service ne la bloque pas, ce qui me permettra d'effectuer des tests avec le nouveau conteneur que nous avons créé.

```
ignoreip = 127.0.0.1/8 ::1 10.31.250.2/22
```

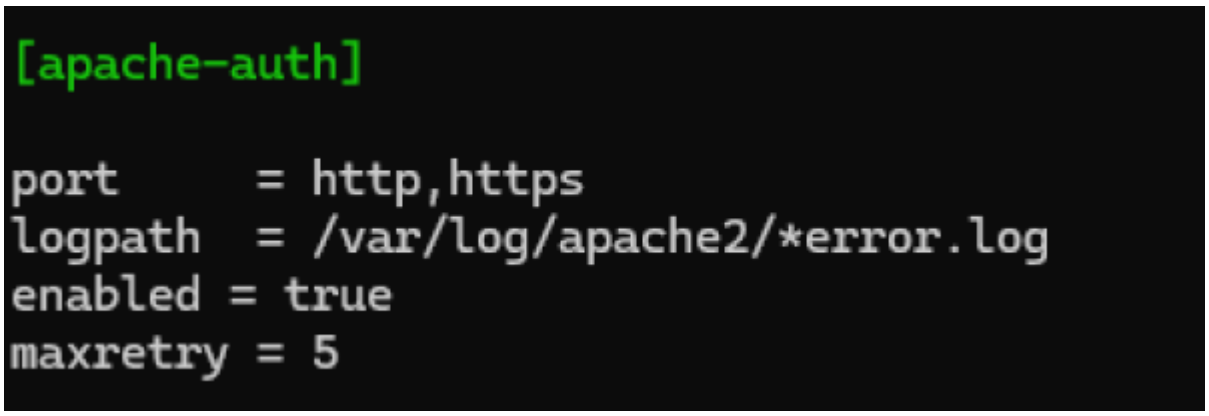
Dans la section de [sshd] ajouté les lignes suivantes :

```
port    = ssh
logpath = /var/log/auth.log
enabled = true
maxretry = 5
```



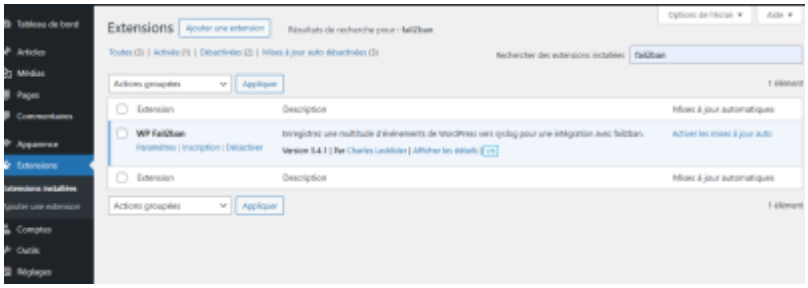
Ensuite dans la section suivante [apache auth] :

```
port     = http,https
logpath  = /var/log/apache2/*error.log
enabled = true
maxretry = 5
```



Configuration pour Wordpress

Pour la configuration de wordpress nous allons installer un plugin qui gere le fail2ban :
Sur le site de votre Wordpress : https://wordpresszone.usa.gsb.org installer le plugin suivant :

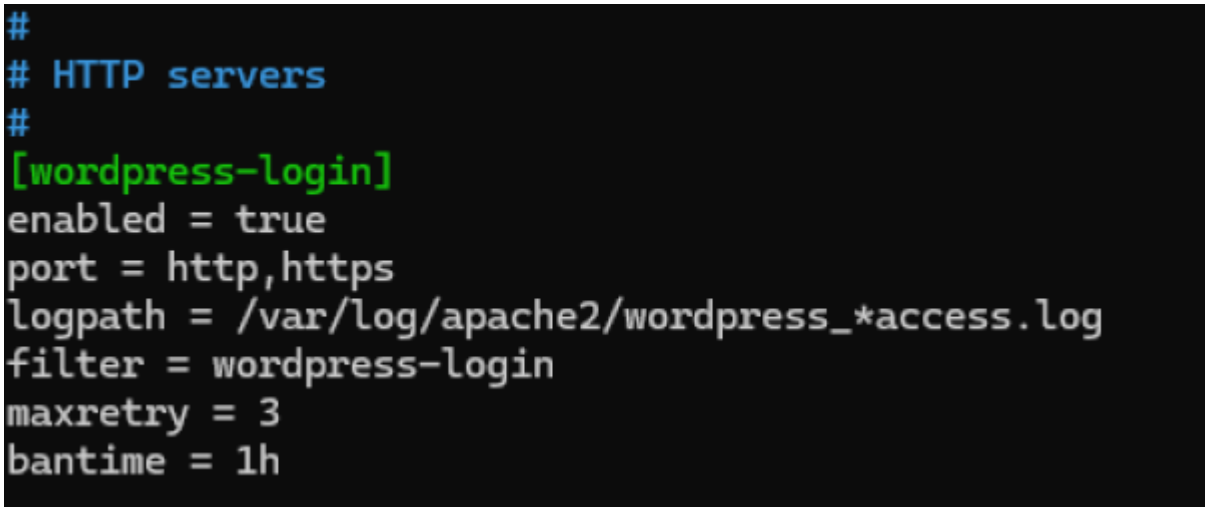


Ensuite dans le fichier suivant :

```
cd /home/htdocs/wordpress_zone/wordpress_zone/wp-content/plugins/wp-
fail2ban/filters.d#
```

cp * /etc/fail2ban/filter.d/ Nous allons ajouter une section pour les wordpress :

```
[wordpress-login]
enabled = true
port = http,https
logpath = /var/log/apache2/wordpress_*access.log
filter = wordpress-login
maxretry = 3
bantime = 1h
```



Ensuite nouss créer le filtre por les wordpress:

```
sudo nano /etc/fail2ban/filter.d/wordpress-login.conf
<code>
Ajouter les lignes suivante : \\
code>
[INCLUDES]
before = apache-common.conf
[Definition]
failregex = ^<HOST> -.*POST.*wp-login.php
ignoreregex =
```

Ainsi redémmaré Fail2ban :

```
sudo systemctl restart fail2ban
```

A.2 GLPI.

</callout>

A.3 Mise en place de la réplication.

Pour la mise en place d'une réplication entre les serveurs webs, j'ai décider d'utiliser rsync en créent des utilisateurs au nom de syncuser sur les deux webs qui peuvent se connecter entre eux avec une clé rsa. Ensuite je configure un crontab sur l'utilisateur du web 1 qui copie tout les fichier modifier au bout de 5 minute dans le fichier suivant : /var/www/
Tout d'abord créer les utilisateurs :

```
adduser syncuser
```

Ensuite créer une clé rsa sur web 1 dans le fichier suivant :

```
cd /home/syncuser/.ssh/
ssh-keygen -t rsa -b 4096
```

Puis copier la clé rsa dans l'utilisateur du web2

Configuration de la réplication

Nous allons utiliser rsync pour faire une réplication passive. Entre autre :
Web1 est la source
Web2 est la copie
Tout ce qui est fait sur web1 est répercuté sur web2
Tout ce qui est fait sur web2 n’affecte pas web1
Pour la réplication, nous allons écrire un script bash qui pernet de copier les fichiers que l'on souhaite dans les serveurs web choisi.

Script Bash

```
nano rsync_Almode.sh
```

```
#!/bin/bash


SERVERS=("10.31.252.84")  # ajoute ici les IP des autres serveurs si besoin


for TARGET in "${SERVERS[@]}"; do
    echo "Synchronisation vers $TARGET ..."


    rsync -az --delete -e "ssh -i /home/syncuser/.ssh/id_rsa" /var/www/html/
```

```
syncuser@$TARGET:/var/www/html/


    rsync -az --delete -e "ssh -i /home/syncuser/.ssh/id_rsa" --rsync-
path="sudo rsync" /etc/nginx/sites-available/ syn>


    rsync -az --delete -e "ssh -i /home/syncuser/.ssh/id_rsa" --rsync-
path="sudo rsync" /etc/nginx/sites-enabled/ syncu>
done
```

```
sudo crontab -e -u syncuser
```

Puis insérer la ligne suivante qui permet d'exécuter le script pour faire la réplication :

```
* * * * * /root/rsync_Almode.sh
```

Ensuite pour faire les test vous pouvez créer un fichier ou un dossier :

```
mkdir test
nano test suite
```

Ainsi le fichier et dossier crée seront ajouté sur le web2 .
Sur les serveurs webs; nous utilisons les comptes utilisateurs syncuser pour avoir plus de sécurité et ne pas tout passer sur le compte root.
Quelque fichier sont utilisé par le compte root; alors dans les fichiers suivant mettez la ligne suivantes :

```
visudo
```

```
syncuser ALL=(ALL) NOPASSWD: /usr/bin/rsync
```