3 - Mise en place de Fail2Ban

Fail2Ban est le must have et permet de bannir les machines qui tentent de pirater votre serveur. Il se base sur les logs du serveur et en fonctions de règles établies afin de bannir les tentavies d'accès non autorisées.

Setup de Fail2Ban

Commençons par installer le paquet de fail2ban:

```
apt-get install -f fail2ban
```

On crée un fichier `jail.local` qui nous servira a paramétrer les différentes "prisons" en cas d'attaques.

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

Puis on édite le fichier pour mettre nos paramètres de sécurités.

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

On modifiera les valeurs suivantes:

```
# L'ip avec laquelle vous accédez au serveur habituellement si fixe
ignoreip = 127.0.0.1/8 XXX.XXX.XXX.XXX  # <= REMPLACER ICI !

# Le nombre de temps qu'un utilisateur sera banni par défaut si il enfreint une règle
bantime = 7d # -1 est un ban permanent 7d équivaut à 7 jours

# La période de temps ou les "maxretry" seront comptabilisés
findtime = 1d

# maxretry le nombre d'essais ratés avant qu'un utilisateur soit banni
maxretry = 5

# Les emails de fail2ban sont envoyés sur l'email défini pour l'utilisateur root
destemail = root

# On change la ligne: action = %(action_)s pour recevoir un mail lors d'un ban
action = %(action_mwl)s

# Activation de fail2ban sur le ssh
[sshd]
```

```
enabled = true
port = 14009 # Votre port SSH
logpath = %(sshd_log)s
backend = %(sshd_backend)s
maxretry = 3 # Si on veut override le maxretry par défaut

# port = les ports à bloquer au moyen des règles iptables
# logpath = l'emplacement des fichiers de log à surveiller
# backend = le moteur de surveillance des logs.

################################################
# SI VOUS AVEZ INSTALLE ET ALLEZ UTILISER NGINX AVEC CLOUDFLARE
# utilisez real_ip_header CF-Connecting-IP; dans vos host !!
[nginx-auth]
# Désactiver en mettant false si Nginx n'est pas installé
enabled  = true
port  = http,https
filter   = nginx-auth
# Chemin vers les error log peut varier si vous utilisez npm
logpath  = /var/log/nginx/*error.log
banaction = iptables-multiport

[nginx-badbots]
# Désactiver en mettant false si Nginx n'est pas installé
enabled  = true
port  = http,https
filter = nginx-badbots
# Chemin vers les access log peut varier si vous utilisez npm
logpath = /var/log/nginx/*access.log
banaction = iptables-multiport
################################################

# Si le ban par défaut n'est pas permanent vous pouvez ajouter la jail recidive
# Si le pirate récidive le temps de ban sera augmenté
[recidive]

enabled = true
logpath  = /var/log/fail2ban.log
banaction = %(banaction_allports)s
bantime  = 3w
```

```
findtime = 7d

# Il vous suffit ensuite de rajouter "true" sur les services que vous voulez surveillez par la suite,
# et rajoutez au besoin les services absents
```

On va ensuite modifier la valeur de `dbpurge` pour l'ajuster, pour cela nous allons éditer le fichier de configuration fail2ban.

```
nano /etc/fail2ban/fail2ban.conf
```

Puis nous allons ajuster les différentes lignes si besoin:

```
# Pour éviter une boucle infini si le loglevel est en DEBUG lorsque la jail recidive est activée
loglevel = INFO

# On ajuste la valeur de dbpurge
dbpurgeage = 3w
```

Une fois cela fait et sauvegardé, nous allons rajouter nos filtres pour Nginx en accord avec les jails créés précédemments.

!! Utilisateurs Cloudflare !!

On notera que si vous utilisez Cloudflare en proxy vous ne verrez pas les vraies IPs des visiteurs. De ce fait vous devez utiliser `real_ip_header CF-Connecting-IP;` dans vos proxy host afin de récupérer la vraie IP !

Pour le premier filtre de la jail `nginx-auth` :

```
nano /etc/fail2ban/filter.d/nginx-auth.conf
```

Puis on copie colle dans le fichier sans modifier:

```
## FICHIER /etc/fail2ban/filter.d/nginx-auth.conf ##
[Definition]

failregex = no user/password was provided for basic authentication.*client: <HOST>
            user .* was not found in.*client: <HOST>
            user .* password mismatch.*client: <HOST>
```

```
ignoreregex =
```

Pour le second filtre de la jail `nginx-badbots`. Il servira a détecter les bots qui cherchent des fichiers de configuration vulnérables et failles sur votre serveur nginx:

```
nano /etc/fail2ban/filter.d/nginx-badbots.conf
```

Puis on copie colle dans le fichier sans rien modifier:

```
# Fail2Ban configuration file
# Author: Patrik 'Sikevux' Greco <sikevux@sikevux.se>

[Definition]

# Option: failregex
# Notes.: regex to match access attempts to setup.php
# Values: TEXT

failregex = ^<HOST> .*?"GET.*?\/setup\.php.*?" .*?

# Anti w00tw00t
            ^<HOST> .*?"GET .*w00tw00t.* 400

# try to access to directory
            ^<HOST> .*?"GET .*admin.* 403
            ^<HOST> .*?"GET .*admin.* 404
            ^<HOST> .*?"GET .*install.* 404
            ^<HOST> .*?"GET .*dbadmin.* 404
            ^<HOST> .*?"GET .*myadmin.* 404
            ^<HOST> .*?"GET .*MyAdmin.* 404
            ^<HOST> .*?"GET .*mysql.* 404
            ^<HOST> .*?"GET .*websql.* 404
            ^<HOST> .*?"GET .*webdb.* 404
            ^<HOST> .*?"GET .*webadmin.* 404
            ^<HOST> .*?"GET \/pma\/.* 404
            ^<HOST> .*?"GET .*phppath.* 404
            ^<HOST> .*?"GET .*admm.* 404
            ^<HOST> .*?"GET .*databaseadmin.* 404
            ^<HOST> .*?"GET .*mysqlmanager.* 404
            ^<HOST> .*?"GET .*phpMyAdmin.* 404
```

```
            ^<HOST> .*?"GET .*xampp.* 404
            ^<HOST> .*?"GET .*sqlmanager.* 404
            ^<HOST> .*?"GET .*wp-content.* 404
            ^<HOST> .*?"GET .*wp-login.* 404
            ^<HOST> .*?"GET .*typo3.* 404
            ^<HOST> .*?"HEAD .*manager.* 404
            ^<HOST> .*?"GET .*manager.* 404
            ^<HOST> .*?"HEAD .*blackcat.* 404
            ^<HOST> .*?"HEAD .*sprawdza.php.* 404
            ^<HOST> .*?"GET .*HNAP1.* 404
            ^<HOST> .*?"GET .*vtigercrm.* 404
            ^<HOST> .*?"GET .*cgi-bin.* 404
            ^<HOST> .*?"GET .*webdav.* 404
            ^<HOST> .*?"GET .*web-console.* 404
            ^<HOST> .*?"GET .*manager.* 404
# Option: ignoreregex
# Notes.: regex to ignore. If this regex matches, the line is ignored.
# Values: TEXT
#
ignoreregex =
```

Il ne nous reste plus qu'a redémarrer fail2ban :) ! Vous recevrez plusieurs mails par la même occasion mais pas de panique rien de très important cela sera plus calme par la suite.

Une fois tout cela fait on redémarre fail2ban

```
systemctl restart fail2ban
```

On vérifie que le service est bien lancé:

```
systemctl status fail2ban
```

Quelques Utilitaires pour fail2ban:
Pour voir les jails actives on peut utiliser:

```
fail2ban-client status
```

Pour voir le status d'une jail en particulier il suffit d'utiliser son nom (exemple avec `sshd`):

```
fail2ban-client status sshd
```

Pour débannir un client remplacer les fields `NOM_DE_LA_JAIL` et `IP_A_DEBANNIR`, ne pas oublier de débannir de la jail `recidive` si elle est active:

```
fail2ban-client set NOM_DE_LA_JAIL unbanip IP_A_DEBANNIR
```

(Optionnel)
Vous pouvez personnaliser vos messages d'alertes dans `/etc/fail2ban/action.d/` en modifiants ces différents fichiers:
Prenez le temps de lire les fichiers ils sont en anglais mais relativement simple à comprendre et personnaliser

```
nano /etc/fail2ban/action.d/sendmail.conf
nano /etc/fail2ban/action.d/sendmail-whois.conf
nano /etc/fail2ban/action.d/sendmail-whois-lines.conf
```

Une fois tout cela fait on redémarre fail2ban

```
systemctl restart fail2ban
```

Voilà votre magnifique douanier Fail2Ban est actif !