n2m
tuto fail2ban

BELTRAMO Emeric

Juin 2021

Partie 1

commandes importantes

```
systemctl start fail2ban          #lancement de fail2ban
systemctl stop  fail2ban          #coupe fail2ban mais les ip restent ban
systemctl reload fail2ban         #relance fail2ban mais remet    0 les bans

fail2ban-client status            #voir les jail actifs
fail2ban-client status [Nom du jail]    #permet de savoir si une ip est ban sur un
                                        #jail

fail2ban-client set [nom du jail] banip [IP    bannir]   #bannir manuellement une
                                                         #ip sur un jail
fail2ban-client set [nom du jail] unbanip [IP concerné] #débannir une ip sur un
                                                         #jail
```

Partie 2

principe

fail2ban est une application qui analyse les logs de divers services (SSH, Apache, FTP...) en cherchant des correspondances entre des motifs définis dans ses filtres et les entrées des logs. Lorsqu'une correspondance est trouvée une ou plusieurs actions sont exécutées. Typiquement, fail2ban cherche des tentatives répétées de connexions infructueuses dans les fichiers journaux et procède à un bannissement en ajoutant une règle au pare-feu iptables ou nftables pour bannir l'adresse IP de la source. **Fail2ban ne doit pas être considéré comme un outil de sécurisation absolu d'un service**. Ses objectifs sont d'éviter de surcharger les logs du système avec des milliers de tentatives de connexion et de limiter la portée des attaques répétées provenant d'une même machine.Ceci va également rendre les attaques par force brute ou par dictionnaire beaucoup plus difficiles mais ce n'est pas une sécurité absolue contre ce type d'attaque.

source : https://doc.ubuntu-fr.org/fail2ban

Partie 3

Fonctionnement

3.1 jail

Fail2ban fonctionne sur le principe suivant:

- jail : un jail ou prison en français est un élément à configurer qui va se servir d'une règle pour analyser les logs et ainsi bannir ou non une ip sur les ports définit. On peut créer un ou plusieurs jail par service à protéger en fonction du type d'erreur que l'on repérer dans les logs.
- règle : une règle est un regex définit dans un fichier **.conf définit dans le dossier filter.d et portant le même nom que le jail qui l'exécute**. C'est à l'aide de cette règle que l'on va étudier les logs
-

3.2 installation

```
apt-get install fail2ban
systemctl start fail2ban
systemctl enable fail2ban
```

3.3 configuration

Une fois l'installation correctement effectuée il faut copier le fichier **jail.conf** en une version **jail.local** car se dernier peut être écrasé lors de mises à jours. C'est dans se fichier que l'on trouve les configurations par défauts ainsi que des jails déjà pré-configurés.

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

3.3.1 configuration par défaut

La configuration par défaut se trouve dans le fichier **jail.local**. Tous les éléments qui vont être énoncé dans cette partie disposent d'une valeur par défaut présenté dans ce fichier mais peuvent aussi être modifié pour chaque jail.

ingnoreip

Permet de spécifier des ip à whitelister. Ce qui signifie qu'elles ne seront jamais bannies. On y trouve biensur localhost mais il ne faut pas oublier d'ajouter celle du bureau ou d'autre qui pourraient être amener à travailler sur le serveur.

```
ignoreip = 127.0.0.1/8 86.194.13.127 93.27.238.202  82.64.14.25
```

bantime

Le ban time définie le temps de bannissement en cas de violation répété d'une règle

```
bantime  = 24h
```

findtime

Le findtime est le temps en secondes à partir duquel une anomalie est cherchée. Attention un temps trop élevé peut ralentir de façon significative le serveur car fail2ban doit analyser une plus grande partie des logs

```
findtime  = 30m
```

maxretry

La clause maxretry définit le nombre d'essais disponibles avant le bannissement d'une ip.

```
maxretry = 5
```

port

La clause port est définie pour chaque jail est permet d'indiquer sur quel port on souhaite bannir l'ip en cas de violation répété de la règle. On peut l'indiquer avec le nom du service par défaut qui est accessible sur ce port ou le numéro du port directement exemple:

```
[apache-badbots]
port     = http,https
logpath  = %(apache_access_log)s
bantime  = 48h
maxretry = 1
```

3.3.2 l'activation d'un jail

Comme vous pouvez le voir dans le dossier jail.local un certain nombre de jail sont déjà définit. De base pour éviter les bugs en cas de services non installé sur le serveur, chaque jail est désactivé. Pour les activer de façon durable il y a deux méthodes. La première est la moins lisible et consiste à **ajouter la clause enabled à chaque jail que l'on souhaite activer** exemple :

```

[sshd]
enabled = true
port    = ssh
logpath = %(sshd_log)s
backend = %(sshd_backend)s
bantime = 24h
maxretry = 5
```

La seconde méthode plus lisible consiste en la création d'un fichier en **.conf** dans le dossier **jail.d** et d'y ajouter la commande suivante ainsi toute les activations seront centralisée et on peut voir d'un coup d'oeil la configuration.

```
[sshd]
enabled = true

[apache-auth]
enabled = true

[apache-badbots]
enabled = true

[apache-noscript]
enabled = true

[apache-overflows]
enabled = true

[apache-nohome]
enabled = true

[apache-botsearch]
enabled = true

[apache-fakegooglebot]
enabled = true

[apache-modsecurity]
enabled = true

[apache-shellshock]
enabled = true
```

3.3.3 l'ajout d'un jail perso

Pour ajouter un jail personnalisé la première étape est d'ajouter sa configuration dans le fichier **defaults-debain.conf** ou **jail.local** suivant le choix que vous avez fait dans le point précédant. Exemple:

```
# Regle anti-Kevin...
[apache-admin]
enabled = true
port = http,https
filter = apache-admin
logpath = /var/log/apache*/error*.log
```

Ensuite il faut créer la règle correspondante dans le dossier **filter.d** en ajoutant un fichier du même nom que le jail avec l'extension **.conf**. Ce fichier doit être de la forme suivante. Le failregex définit le pattern qui sera recherché dans les logs. ignoreregex si un ligne correspond à ce regex elle sera ignorée.

```

# Fail2Ban configuration file
#
# Author: Cyril Jaquier
#
# $Revision: 471 $
#

[Definition]

# Option: failregex
# Notes.: regex to match the password failure messages in the logfile. The
# host must be matched by a group named "host". The tag "<HOST>" can
# be used for standard IP/hostname matching.
# Values: TEXT
# [client x.x.x.x] File does not exist: /home/www/admin/admin,
failregex = <HOST> -- \[.*?\] ".*?" 404
#
# Option: ignoreregex
# Notes.: regex to ignore. If this regex matches, the line is ignored.
# Values: TEXT
#
ignoreregex =
```