Installer et configurer Fail2ban

- **Ressources** :
  - Debian 9 Stretch : sécuriser votre serveur avec Fail2ban
  - Wiki Debian - Fail2ban
- **Notes** : le support de Nftables par Fail2ban existe depuis la version 0.9.4

Installer Fail2ban

- Installer le paquet : `apt install fail2ban`
- Le fichier de config par défaut à **ne pas modifier** de *Fail2ban* est : `vi /etc/fail2ban/jail.conf`
- Habituellement, il est nécessaire de créer un fichier local `/etc/fail2ban/jail.local` pour éviter l'écrasement de la config lors des mises à jour. Mais sous Debian ce fichier existe déjà sous le nom : `vi /etc/fail2ban/jail.d/defaults-debian.conf` et contient déjà le service SSH activé.
  - Remplacer les 2 lignes existantes par le contenu suivant pour *db-srv*:

```
[DEFAULT]
# Emails
destemail = adminsys@<domaine-sinp>
sender = mailer@<domaine-sinp>

# Actions de banissement via Firewalld
banaction = firewallcmd-multiport
banaction_allports = firewallcmd-allports

# Actions à réaliser en cas de banissement : mwl (= ban & send an
e-mail with whois report and relevant log lines)
action = %(action_mwl)s

# Ajouter ses ip pour éviter de se faire bannir
# Ex.: ignoreip = 127.0.0.1/8 10.0.1.10 10.0.1.20 <ip-v4-db> <ip-
v4-web>
ignoreip = 127.0.0.1/8 <ip-v4-private-web> <ip-v4-private-db> <ip-
v4-db> <ip-v4-web> <ip-vpn-lpo>
# 1 jour de bannissement
#bantime = 86400
# 1 semaine de bannissement - Mise à jour 2020-12-04 [jpmilcent].
# bantime = 604800
# 1 an de banissement - Mise à jour 2021-01-06 [jpmilcent] => 31
536 000s
bantime = 31536000

[sshd]
enabled = true
port = <port-ssh>
```

```
[postfix]

enabled = true
port = smtp,submission
```

- Pour *web-srv* et *bkp-srv*, décommenter les sections concernant Nginx, une fois le serveur web installé et configuré :
  - Remplacer les 2 lignes existantes par le contenu suivant

```
[DEFAULT]
# Emails
destemail = adminsys@<domaine-sinp>
sender = mailer@<domaine-sinp>

# Actions de banissement via Firewalld
banaction = firewallcmd-multiport
banaction_allports = firewallcmd-allports

# Actions à réaliser en cas de banissement : mwl (= ban & send
an e-mail with whois report and relevant log lines)
action = %(action_mwl)s

# Ajouter ses ip pour éviter de se faire bannir
ignoreip = 127.0.0.1/8 <ip-v4-private-web> <ip-v4-private-db>
<ip-v4-private-bkp> <ip-v4-web> <ip-v4-db> <ip-v4-bkp>
# 1 jour de bannissement
#bantime = 86400
# 1 semaine de bannissement - Mise à jour 2020-12-04
[jpmilcent].
#bantime = 604800
# 1 an de bannissement - Mise à jour 2021-01-06 [jpmilcent] =>
31 536 000s
bantime = 31536000

[sshd]
backend=systemd
enabled = true
port = <port-ssh>

[postfix]
backend=systemd
enabled = true
port = smtp,submission

# Spécifique à l'instance : web-srv
# [nginx-http-auth]
# enabled = true
# port = http,https
# logpath = /var/log/nginx/error.log
```

```
# [nginx-limit-req]
# enabled = true
# port = http,https
# logpath = /var/log/nginx/error.log

# [nginx-botsearch]
# enabled = true
# port = http,https
# logpath = /var/log/nginx/error.log
# maxretry = 2
```

- Pour appliquer une nouvelle configuration, redémarrer le service *Fail2ban* : `systemctl restart fail2ban`
  - Vérifier le status du service : `systemctl status fail2ban`

Utilisations

- Document du client Fail2ban : https://manpages.debian.org/buster/fail2ban/fail2ban-client.1.en.html
- Vérifier son fonctionnement dans les logs : `vi /var/log/fail2ban.log`
- Consulter l'état d'une prison (ici avec comme nom de prison : *nginx-http-auth*) : `fail2ban-client status nginx-http-auth`
- Sortir de prison une IP : `fail2ban-client set <nom-de-prison> unbanip <IP-concernée>`
- Ajouter manuellement une IP à une prison : `fail2ban-client set <nom-de-la-prison> banip <IP-concernée>`
- Réinitialiser une prison : `fail2ban-client reload --unban <nom-de-prison>`
- Enregistrer dans un fichier txt les IPs bannies d'une prison : `fail2ban-client status <nom-de-prison> | grep "Banned IP list:" | awk -F: '{print $2}' | tr -s ' ' '\n' | grep -v '^$' > ips_bannies.txt`
- Tester un filtre sur un fichier de log : `fail2ban-regex <chemin-fichier-log> /etc/fail2ban/filter.d/<nom-filre>`
- Recharger un filtre après modification : `fail2ban-client reload <nom-de-prison>`
  - Vérifier que la regex du filtre est bien pris en compte en l'affichant : `fail2ban-client get <nom-de-prison> failregex`
  - Vérifier que l' *ignore regex* du filtre est bien pris en compte en l'affichant : `fail2ban-client get <nom-de-prison> ignoreregex`

Fai2ban et Wordpress

- Une fois Wordpress rendu indexable par les moteurs de recherche, nous risquons d'être attaqué sur la page de login de Wordpress. Si cette situation apparaissait, il serait nécessaire de mettre en place une règle Fail2ban.
- Pour vérifier si nous sommes attaqué, vérifier dans les logs du serveur web du jour à l'aide de la commande : `grep "POST /wp-login.php" /var/log/nginx/access.log`
- Dans cette situation, utiliser un des tutos suivant pour la mise en place :
  - Wordpress et les attaques brute force
  - WordPress & fail2ban : stopper la brute-force « POST /wp-login.php » : sans plugin WP.

    - Fail2ban pour wordpress pour se protéger des bruteforce : avec plugin WP-Fail2ban
    - Sécurisez votre WordPress avec Fail2ban : avec plugin WP-Fail2ban
- Ajouter un nouveau filtre : `vi /etc/fail2ban/filter.d/nginx-wordpress.conf`
    - Par défaut, Wordpress retourne une code 200 que le login réussisse ou échoue. Deux solutions :
        1. modifier Worpdress pour retourner un code 403 si le login échoue
        2. utiliser le code 200 dans la regexp de détection et ne pas mettre trop bas le nombre de tentative d'essai et la durée de rechercher (`findtime`)
    - Nous avons opté pour la solution 2.
    - Contenue du fichier :

```
[Definition]
# Fail2Ban configuration file
#
# Preserve brute force on Wordpress site
# Author: cam.lafit <cam.lafit@azerttyu.net>
# Source:
https://km.azerttyu.net/Wordpress-et-les-attaques-brute-force

file_preserved = wp-login\.php|xmlrpc\.php

# Option:  failregex
# Notes.:  Regexp to catch url to prevent bot connection
#          that it is your intent to block IPs which were driven
by
#          abovementioned bots.
# Values:  TEXT
#
failregex = ^<HOST> -.*"POST /(%(file_preserved)s)
HTTP/[12]\.[01]" 200 .*

# Option:  ignoreregex
# Notes.:  regex to ignore. If this regex matches, the line is
ignored.
# Values:  TEXT
#
ignoreregex =
```

- Tester le filtre : `fail2ban-regex --print-all-matched /var/log/nginx/access.log /etc/fail2ban/filter.d/nginx-wordpress.conf`
- Éditer le fichier de config : `vi /etc/fail2ban/jail.d/defaults-debian.conf`
    - Ajouter à la fin du fichier la nouvelle section suivante :

```
[nginx-wordpress]
# Banni toute IP ayant accédée à wp_login.php au moins 3 fois dans
un intervalle de 240 secondes (4mn)
enabled = true
port    = http,https
filter  = nginx-wordpress
logpath = /var/log/nginx/access.log
maxretry = 3
```

```
findtime = 240
```

- Recharger la config de *Fail2ban* : `systemctl reload fail2ban.service`
- Pour surveiller cette nouvelle prison : `watch fail2ban-client status nginx-wordpress`

Problème fail2ban-tmpfiles.conf points to /var/run/ instead of /run

Sous Debian 10, corriger le bug fail2ban-tmpfiles.conf points to /var/run/ instead of /run, en éditant les fichiers :10

- `vi /usr/lib/tmpfiles.d/fail2ban-tmpfiles.conf` et y remplacer le chemin comme suit :

```
D /run/fail2ban 0755 root root -
```

- `vi /lib/systemd/system/fail2ban.service` et y remplacer les occurrences de `/var/run/` par `/run/`.
- Prendre en compte les changements : `systemctl daemon-reload`
- Si le problème vient à se renouveller suite à une mise à jour de Fail2ban, utiliser la technique mise en place pour le service Docker sur l'instance *db-srv* pour surcharger le service *Fail2ban*.

Problème 'allowipv6' not defined in 'Definition'

- **Contexte** : le service SystemD fail2ban (`systemctl status fail2ban`) affiche le message

```
fail2ban.configreader   [874403]: WARNING 'allowipv6' not defined in
'Definition'. Using default one: 'auto'
```

- **Solution** : éditer le fichier `vi /etc/fail2ban/fail2ban.conf` et décommenter la ligne `allowipv6 = auto`.

Problème : failed during configuration: Have not found any log file for sshd jail

- **Contexte** : le service SystemD fail2ban (`systemctl status fail2ban`) affiche le message

```
failed during configuration: Have not found any log file for sshd jail
```

- **Solution** : Dans le fichier `/etc/fail2ban/jail.d/defaults-debian.conf` ajouter à la section `[sshd]` la propriété `backend=systemd`, idem pour la section `[postfix]`

Prison nginx-errors

- Mise en place d'une prison détectant un trop fort tôt de code d'erreurs HTTP : 302, 403, 404 et

500.
- Cette prison est SENSIBLE car elle peut rapidement prendre en compte des faux positifs. A surveiller avec : `watch fail2ban-client status nginx-errors` ou en directe avec `tail -f /var/log/fail2ban.log | grep "\[nginx-errors\] Ban"`
- Créer un fichier de filtre : `vi /etc/fail2ban/filter.d/nginx-errors.conf`
    - Contenu :

```
[Definition]
failregex = ^<HOST> \S+ - \[.*\] "(GET|POST|HEAD).*HTTP.*"
(302|403|404|500) .*$
ignoreregex =
```

    - Tester le bon fonctionnement de cette Regexp avec : `fail2ban-regex /var/log/nginx/access.log /etc/fail2ban/filter.d/nginx-errors.conf`
- Créer un fichier de prison : `vi /etc/fail2ban/jail.d/nginx-errors.conf`
    - Contenu :

```
[nginx-errors]
enabled = true
port = http,https
filter = nginx-errors
logpath = /var/log/nginx/access.log
maxretry = 6
findtime = 600
bantime = 1y
# Le paramètre ci-dessous peu être décommenté pour tester la prison sans bannir les ip
#banaction = dummy
```

- Recharger la config de *Fail2ban* : `systemctl restart fail2ban.service`

Vérification d'IP bannies

- Pour vérifier en lot des IPs abusives, vous pouvez utiliser le site : https://www.abuseipdb.com/bulk-check
- Le site IpInfo fournit également des informations : https://ipinfo.io/

From:
https://sinp-wiki.cbn-alpin.fr/ - **CBNA SINP**

Permanent link:
https://sinp-wiki.cbn-alpin.fr/serveurs/installation/fail2ban?rev=1790151486

Last update: **2026/09/23 08:18**