Fail2Ban

Last updated by | Leslie Lemaire | 11 juin 2025 at 14:38 UTC+2

Mettre en place Fail2Ban sur Debian 12

Objectif

Renforcer la sécurité des serveurs en bannissant automatiquement les adresses IP effectuant des tentatives d'intrusion répétées, en complément de la supervision centralisée du projet CYNA SR.

Fail2Ban est déployé ici en complément des solutions suivantes :

- **Wazuh** : corrélation des alertes
- **Grafana / Loki** : visualisation des logs
- **Zabbix** : supervision de l'état du service Fail2Ban
- **pfSense** : filtrage réseau global

Installation

Mise à jour du système

```
sudo apt update && sudo apt upgrade -y
```

Installation de Fail2Ban

```
sudo apt install fail2ban -y
```

Vérification

```
fail2ban-client -V
```

Si pas encore installé, installation de Rsyslog

```
sudo apt install -y rsyslog
```

Configuration de base

Sauvegarde du fichier de configuration

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

> *Remarque : les modifications doivent se faire sur le fichier* `jail.local` .

Paramétrage global (`/etc/fail2ban/jail.local`)

Exemple de configuration de base :

```
[DEFAULT]

# Durée du bannissement
bantime = 1h

# Période pendant laquelle les tentatives sont comptabilisées
findtime = 10m

# Nombre de tentatives autorisées avant ban
maxretry = 5

# Backend utilisé
backend = systemd

# Envoi d'alertes par mail (à adapter si un système de mail est configuré)
destemail = root@localhost
sender = fail2ban@monserveur.local
mta = sendmail

# Action à réaliser (ban + mail avec logs)
action = %(action_mwl)s
```

Activation des protections

Protection SSH (`/etc/fail2ban/jail.local`)

```
[sshd]

enabled = true
port    = ssh
logpath = %(sshd_log)s
backend = systemd
```

Protection NGINX (utile dans le cadre de l'architecture CYNA SR)

```
[nginx-http-auth]

enabled  = true
port     = http,https
logpath  = /var/log/nginx/error.log
maxretry = 3
```

Autres jails possibles : `nginx-botsearch`, `nginx-noscript`, `apache-auth`, selon les besoins.

Démarrage et vérification

Redémarrer Fail2Ban

```
sudo systemctl restart fail2ban
```

Vérifier l'état du service

```
sudo systemctl status fail2ban
```

Vérifier les jails actives

```
sudo fail2ban-client status
```

Vérifier les IP bannies pour une jail spécifique

```
sudo fail2ban-client status sshd
```

Intégration avec l'infrastructure CYNA SR

Service	Complément Fail2Ban
pfSense	Filtrage réseau global (niveau pare-feu)
Wazuh	Remontée des logs Fail2Ban vers SIEM
Grafana / Loki	Visualisation en dashboard des bannissements
Zabbix	Supervision de l'état du service Fail2Ban (`systemctl status`)

Exemple de liaison Wazuh

- Activer la collecte des logs de `/var/log/fail2ban.log` sur l'agent Wazuh.
- Créer une règle personnalisée Wazuh pour alerter sur des IP bannies.

Surveillance continue

Superviser l'état du service dans Zabbix

- Créer un item Zabbix basé sur `systemctl is-active fail2ban` ou `systemctl status` .
- Créer un trigger pour alerter en cas de service inactif.

Sources

- https://fr-wiki.ikoula.com/fr/Mettre_en_place_fail2ban_sur_Debian
- https://slash-root.fr/fail2ban-installation-et-configuration/
- Pense-bête sécuriser Debian 12 avec Fail2Ban (PDF)