Filter

Principe

Un filter est un fichier .conf dans /etc/fail2ban/filter.d/ qui contient une (ou plusieurs) expression(s) régulière(s) capable(s) de repérer, dans une ligne de log, une tentative échouée, et d'en extraire l'adresse IP via le groupe nommé <HOST>.

Structure minimale :

```
[Definition]
failregex = <ta regex avec <HOST>>
ignoreregex =
```

<HOST> est une macro fail2ban qui correspond à une regex IPv4/IPv6 déjà prête à l'emploi — pas besoin de l'écrire toi-même.

Méthode générale pour écrire un filter

1. **Identifier une ligne de log représentative** d'un échec d'authentification.
2. **Repérer ce qui varie** (timestamp, IP, user) vs ce qui est fixe (le message d'erreur).
3. **Écrire la regex**, en remplaçant l'IP par <HOST> et en échappant les caractères spéciaux (., [,], etc.).
4. **Tester** avec fail2ban-regex avant de créer la jail.

Exemple générique : filter SSH (fourni par défaut, pour référence)

/etc/fail2ban/filter.d/sshd.conf (déjà présent, à ne pas modifier) repère une ligne du type :

```
Jun 20 10:15:32 host sshd[1234]: Failed password for invalid user admin from 203.0.113.42 port 51234 ssh2
```

avec une regex de la forme :

```
failregex = ^%(__prefix_line)sFailed \S+ for .* from <HOST>(?: port \d+)?(?: ssh2)?\s*$
```

Fil rouge : filter Vaultwarden

Vaultwarden journalise les échecs de connexion via Docker (stdout → json-file). Une ligne typique ressemble à :

```
[2026-06-20 10:15:32.123][vaultwarden::api::identity][WARN] Username or password is incorrect. Try again. IP: 203.0.113.42. Username: user@example.com.
```

Créer /etc/fail2ban/filter.d/rakouns-vaultwarden.conf :

```
[Definition]
failregex = ^.*Username or password is incorrect\. Try again\. IP: <HOST>\..*$
            ^.*Username or password is incorrect\. IP: <HOST>\..*$
ignoreregex =
```

```
datepattern = ^\[%%Y-%%m-%%d %%H:%%M:%%S
```

> **Note —** Plusieurs lignes dans failregex (une par ligne) sont évaluées en OU logique — utile quand le message d'erreur varie légèrement selon la version du service.

Tester un filter avant de créer la jail

Toujours valider un filter sur un vrai log (ou un extrait) avant de l'attacher à une jail, pour éviter de générer un faux sentiment de sécurité avec un filter qui ne matche jamais — ou pire, qui matche trop large.

Test sur un fichier de log classique

```
sudo fail2ban-regex /var/log/auth.log /etc/fail2ban/filter.d/sshd.conf
```

Test sur les logs Docker (cas Vaultwarden)

Les logs Docker ne sont pas dans un fichier classique par défaut, il faut soit extraire un échantillon, soit pointer directement sur le fichier json-file du conteneur :

```
# Trouver le chemin du log JSON du conteneur
docker inspect --format='{{.LogPath}}' vaultwarden

# Tester le filter contre ce fichier
sudo fail2ban-regex /var/lib/docker/containers/<container_id>/<container_id>-json.log
/etc/fail2ban/filter.d/rakouns-vaultwarden.conf
```

Lecture du résultat

fail2ban-regex affiche un résumé du type :

```
Results
=======

Failregex: 3 total
|-  #) [# of hits] regular expression
|   1) [3] ^.*Username or password is incorrect\. Try again\. IP: <HOST>\..*$
`-

Ignoreregex: 0 total

Lines: 150 lines, 0 ignored, 3 matched, 147 missed
```

3 matched confirme que le filter fonctionne. Si tu obtiens 0 matched alors que tu sais qu'il y a des échecs dans le log, retravaille la regex (espace en trop, caractère spécial non échappé, format de date différent, etc.).

Revision #1
Created 2026-06-20 11:47:36 UTC by clement-derouet
Updated 2026-06-20 11:59:09 UTC by clement-derouet