

Projet 1 - Fail2ban

Installation de fail2ban

```
root@debian:~# apt install fail2ban
```

Mise en place du démarrage automatique après chaque redémarrage de la machine

```
root@debian:~# systemctl enable fail2ban
```

Adress ip

```
root@debian:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:56:ab:be:19 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 10.100.0.18/8 brd 10.255.255.255 scope global dynamic ens192
        valid_lft 634060sec preferred_lft 634060sec
    inet6 fe80::250:56ff:feab:be19/64 scope link
        valid_lft forever preferred_lft forever
root@debian:~#
```

Pour accéder au PowerShell, il faut autoriser l'accès (PermitRootLogin Yes)

```
GNU nano 7.2 /etc/ssh/sshd_config
PermitRootLogin yes
# This is the sshd server system-wide configuration file. See
# sshd_config(8) for more information.
```

Paramétrage général

Les paramètres de configuration doivent normalement être ajoutés dans le fichier principal de configuration : `/etc/fail2ban/jail.conf`. Cependant, ce fichier contient déjà de nombreux paramètres et peut être remplacé lors des mises à jour du logiciel. Il est donc recommandé de créer un fichier séparé dans le répertoire `/etc/fail2ban/jail.d/`. Tous les fichiers présents dans ce répertoire seront automatiquement inclus par le fichier principal de configuration.

De plus, si certains paramètres sont déjà définis dans `jail.conf`, ceux spécifiés dans le répertoire `jail.d/` auront toujours priorité et seront pris en compte.

Il suffit donc de créer un fichier comme `/etc/fail2ban/jail.d/monParametrage.conf` et d'y inscrire les paramètres souhaités

```

DEFAULT]

# Adresses IP non concernées par la surveillance. Ici par exemple on a :
# la boucle locale et l'adresse IP du DSI (10.100.0.99)
ignoreip = 127.0.0.1 10.100.0.99

# Période de temps de recherche dans les logs
findtime = 10m

# Durée de bannissement (en minute, heure, jour, semaine : m, h, d, w)
bantime = 1h
# Durée de bannissement grandissante
bantime.increment = true
# Le 2ème bannissement sera de 24h ; le 3ème de 48h ; le 4ème de 96h, etc.
bantime.factor = 24
# Durée maximale de bannissement
bantime.maxtime = 8w
]
# Nombre maximal de tentatives échouées avant bannissement (dans la période
# de temps fixée par findtime
maxretry = 3

```

Activation de la première prison :

```

[sshd]

enabled = true
port = 2222

logpath = /var/log/auth.log
maxretry = 2

```

Gestion des prisons

On peut vérifier les prisons mises en place :

```
fail2ban-client status
```

```

Status
|- Number of jail:      1
'-- Jail list:  sshd

```

Ou bien contrôler une prison spécifique (exemple SSH) :

```
fail2ban-client status sshd
```

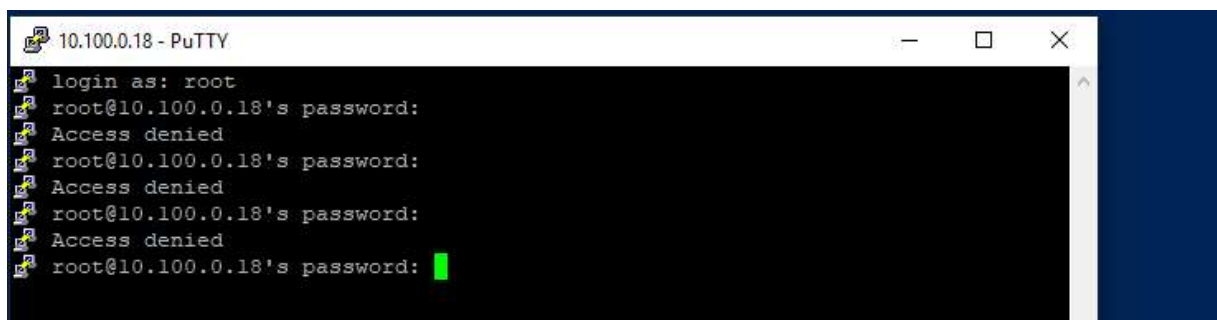
```

Status for the jail: sshd
|- Filter
| |- Currently failed: 0
| |- Total failed:     0
| '-- File list:       /var/log/auth.log
'-- Actions
   |- Currently banned: 0
   |- Total banned:     0
   '-- Banned IP list:

```

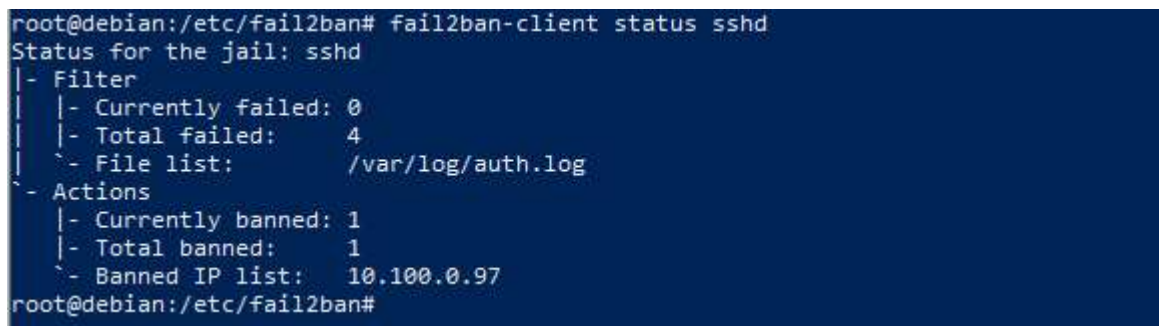
Test

1. Une fois la protection SSH en place pour bloquer les adresses IP après deux tentatives infructueuses de connexion (mots de passe incorrects), vous pouvez effectuer le test suivant

A screenshot of a PuTTY terminal window titled "10.100.0.18 - PuTTY". The terminal shows a series of failed SSH login attempts. The user attempts to log in as 'root' and provides a password. The system responds with "Access denied" for each attempt. The sequence of lines is: "login as: root", "root@10.100.0.18's password:", "Access denied", "root@10.100.0.18's password:", "Access denied", "root@10.100.0.18's password:", "Access denied", and finally "root@10.100.0.18's password:" followed by a green cursor. The terminal has a black background with white text.

```
10.100.0.18 - PuTTY
login as: root
root@10.100.0.18's password:
Access denied
root@10.100.0.18's password:
Access denied
root@10.100.0.18's password:
Access denied
root@10.100.0.18's password: 
```

2. Il est possible de constater qu'une adresse IP a été bannie, généralement pour une durée d'une heure, comme spécifié dans la configuration de notre protection SSH.

A screenshot of a terminal window showing the output of the command "fail2ban-client status sshd". The output displays the status for the 'sshd' jail, including the number of failed attempts, the file list, and the actions taken. The banned IP list shows "10.100.0.97". The terminal has a dark blue background with white text.

```
root@debian:/etc/fail2ban# fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|   |- Currently failed: 0
|   |- Total failed:     4
|   `-- File list:       /var/log/auth.log
`- Actions
    |- Currently banned: 1
    |- Total banned:     1
    `-- Banned IP list:  10.100.0.97
root@debian:/etc/fail2ban#
```

3. Les journaux (/var/log/fail2ban.log) montrent que Fail2Ban a détecté deux tentatives infructueuses correspondant à ses filtres et a bloqué l'IP après la troisième tentative.

```

ion Actions.ActionInfo.<lambda> at 0x7f7b32efd8a0}}': Error starting action Jail('sshd')/iptables-multiport: 'Script error'
2025-03-18 11:21:27,078 fail2ban.filter [5488]: INFO [sshd] Found 10.100.0.97 - 2025-03-18 11:21:26
2025-03-18 11:21:27,078 fail2ban.observer [5488]: INFO [sshd] Found 10.100.0.97, bad - 2025-03-18 11:21:26, 1 # -> 2.0, Ban
2025-03-18 11:21:27,229 fail2ban.actions [5488]: NOTICE [sshd] 10.100.0.97 already banned
root@debian:/etc/fail2ban#

```

2^{ème} prison de site internet

On install apache2

Après avoir installé Apache2 sur Debian, il est possible d'héberger un site web en utilisant l'adresse IP de la machine Debian 10.100.0.18

Lorsque qu'un utilisateur malveillant tente d'accéder à une page inexistante, comme par exemple <http://10.100.0.18/admin>, cela génère une erreur 404 qui est enregistrée dans les logs du serveur



Configuration d'une nouvelle règle, nommée « attaqueWeb », dans le fichier de configuration de Fail2Ban : `/etc/fail2ban/jail.d/monParametrage.conf`, en insérant les lignes suivantes :

```

Nano /etc/fail2ban/jail.d/monParametrage.conf

[attaqueWeb]
enabled = true

# On lui dit quel filtre utiliser pour détecter l'attaque (filtre qui va être créé dans l'étape suivante)

```

```
filter = attaqueWeb

# On utilise ce fichier de log pour rechercher les attaques
logpath = /var/log/apache2/access.log

# On bloque dès la 1ère tentative
maxretry = 0
```

Ensuite, nous définissons un filtre pour détecter la chaîne de caractères 'GET /admin HTTP/1.1' 404 dans les fichiers journaux. Pour cela, nous créons un fichier de filtre appelé `/etc/fail2ban/filter.d/attaqueWeb.conf` avec le contenu suivant

```
Nano /etc/fail2ban/filter.d/attaqueWeb.conf

[Definition]

failregex = ^<HOST>.*"GET /admin HTTP/1.1" 404.*

# Ou bien le filtre suivant si on veut bloquer /admin, ou /private ou /administration
#failregex = ^<HOST>.*"GET /(admin|private|administration) HTTP/1.1" 404.*

# Ou bien le filtre suivant si on veut bloquer quand la personne essaye /contact puis tout de suite après /about, puis /search
# Typiquement, ça serait l'uvre d'une énumération par force brute (comme avec Gobuster ) via le dictionnaire suivant :
# /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt ☺
#failregex = ^<HOST>.*"GET /contact HTTP/1.1" 404.*\n.*"GET /about HTTP/1.1" 404.*\n.*"GET /search HTTP/1.1" 404.*
```

On vérifie les prisons actives

```
lines 1-14/14 (END)
nano /etc/fail2ban/filter.d^C
root@debian:/etc/fail2ban# fail2ban-client status
Status
|- Number of jail:      2
`- Jail list:  attaqueWeb, sshd
root@debian:/etc/fail2ban#
```

On simule une tentative d'attaque en accédant à l'URL <http://10.100.0.18/admin>

(ce qui devrait entraîner un bannissement immédiat)

On vérifie la prison « attaqueWeb » : l'IP est bannie

```
root@debian:~# fail2ban-client status attaqueWeb
Status for the jail: attaqueWeb
|- Filter
| |- Currently failed: 0
| |- Total failed:    0
| `-- File list:      /var/log/apache2/access.log
`- Actions
  |- Currently banned: 1
  |- Total banned:    1
  `-- Banned IP list: 10.100.0.97
root@debian:~#
```

3^{ème} prison les attaques DDOS

Paramètre dans le fichier `/etc/fail2ban/jail.d/monParametrage.conf`

```
[ping]
enabled = true
port = any
protocol = icmp
logpath = /var/log/kern.log
maxretry = 20
bantime = 3600
findtime = 300
action = %(action_mwl)s
```

Parametre pour le filtre de

```
GNU nano 7.2 /etc/fail2ban/filter.d/ping.conf
[Definition]
failregex = .*ICMP.* SRC=<HOST>.*TYPE=8.*
ignoreregex =
```

Vérification des prisons

```
root@debian:/var/log# fail2ban-client status
Status
|- Number of jail:    3
`- Jail list:  attaqueWeb, ping, sshd
```

Cela ne suffit pas, il faut créer des règles de filtrage. Pour cela on utilise iptables, voici les règles à utiliser :

```
iptables -A INPUT -p icmp --icmp-type 8 -j LOG --log-prefix "ICMP: " --log-level 4
```

```
iptables -A INPUT -p icmp --icmp-type 8 -j LOG --log-prefix "PING-LOG: " --log-level 4
```

- `-A INPUT` → Ajoute la règle à la chaîne INPUT (trafic entrant).
- `-p icmp` → S'applique aux paquets ICMP.
- `--icmp-type 8` → Filtre uniquement les paquets ICMP de type 8, qui correspondent aux **requêtes ping (echo request)**.
- `-j LOG` → Journalise les paquets correspondants sans les bloquer.
- `--log-prefix "ICMP: "` et `--log-prefix "PING-LOG: "` → Ajoutent un préfixe pour distinguer les logs dans `/var/log/kern.log`.
- `--log-level 4` → Définit le niveau de journalisation (warning).

Test

```
root@debian:~# fail2ban-client status ping
Status for the jail: ping
|- Filter
| |- Currently failed: 1
| |- Total failed:    52
| `-- File list:      /var/log/kern.log
`- Actions
   |- Currently banned: 1
   |- Total banned:    1
   `-- Banned IP list:  10.100.0.97
root@debian:~#
```