

Analyser les Logs Nginx (access.log, error.log)

1. Comprendre Fail2Ban et son fonctionnement

Fail2Ban surveille les logs du système et des services (comme Nginx) pour détecter des comportements malveillants. Lorsqu'une activité suspecte est identifiée, Fail2Ban peut : - **Bloquer l'adresse IP fautive** via iptables ou un pare-feu. - **Envoyer des alertes** ou exécuter des actions personnalisées.

Fail2Ban repose sur des **fichiers de configuration appelés jails**, qui contiennent des règles pour analyser les logs.

2. Configurer Nginx pour Fail2Ban

a) Assurez-vous que les logs Nginx sont correctement configurés

Vérifiez que les fichiers `access.log` et `error.log` sont activés et contiennent les informations nécessaires. Exemple de configuration dans `/etc/nginx/nginx.conf` :

```
http {  
    access_log /var/log/nginx/access.log;  
    error_log /var/log/nginx/error.log warn;  
}
```

Redémarrez Nginx pour appliquer les changements :

```
sudo systemctl restart nginx
```

3. Installer et configurer Fail2Ban

a) Installer Fail2Ban

Sur la plupart des distributions Linux, installez Fail2Ban avec le gestionnaire de paquets :

```
sudo apt update && sudo apt install fail2ban -y    # Sur Debian/Ubuntu  
sudo yum install fail2ban -y                       # Sur CentOS/RHEL
```

b) Configurer les jails de Fail2Ban

Les fichiers de configuration des jails se trouvent dans `/etc/fail2ban/jail.d/` ou `/etc/fail2ban/jail.local`. Créez ou modifiez une jail dédiée à Nginx :

```
sudo nano /etc/fail2ban/jail.d/nginx.local
```

Ajoutez les sections suivantes pour surveiller différents types d'activités :

i) Bloquer les erreurs de mot de passe ou tentatives répétées (exemple avec WordPress ou autres pages sensibles)

```
[nginx-auth]
enabled = true
port    = http,https
filter  = nginx-auth
logpath = /var/log/nginx/access.log
maxretry = 3
bantime = 600
```

ii) Bloquer les tentatives d'accès à des pages inexistantes (404)

```
[nginx-badbot]
enabled = true
port    = http,https
filter  = nginx-badbot
logpath = /var/log/nginx/access.log
findtime = 300
maxretry = 3
bantime = 20
```

iii) Bloquer les erreurs fréquentes dans error.log

```
[nginx-errors]
enabled = true
port    = http,https
filter  = nginx-errors
logpath = /var/log/nginx/error.log
maxretry = 5
bantime = 3600
```

4. Créer des filtres pour Nginx dans Fail2Ban

Fail2Ban utilise des filtres pour analyser les logs. Les filtres se trouvent dans /etc/fail2ban/filter.d/. Voici quelques exemples de filtres pour Nginx :

a) Filtre pour authentification échouée (nginx-auth.conf)

Créez un fichier /etc/fail2ban/filter.d/nginx-auth.conf :

[Definition]

```
# failregex = ^<HOST> - .* "GET /wp-login.php.*" 401
failregex = ^<HOST> - - \[.*\] "POST /login.*" 302 358
ignoreregex =
```

b) Filtre pour bots malveillants ou requêtes abusives (nginx-badbot.conf)

Créez un fichier /etc/fail2ban/filter.d/nginx-badbot.conf :

[Definition]

```
# failregex = ^<HOST> .* "(GET|POST) /.*" .* 403
failregex = ^<HOST> - - \[.*\] ".*" 404
ignoreregex =
```

c) Filtre pour erreurs multiples dans error.log (nginx-errors.conf)

Créez un fichier /etc/fail2ban/filter.d/nginx-errors.conf :

[Definition]

```
failregex = ^.* \[error\] .* client: <HOST>, server: .*, request: ".*", host: ".*"
ignoreregex =
```

Conclusion

Analyser les logs Nginx en parallèle avec **Fail2Ban** est une méthode puissante pour sécuriser votre serveur web contre les attaques comme les tentatives de brute-force, les explorations malveillantes ou les abus de requêtes.