jail.conf, fail2ban.conf

Configuration pour le serveur fail2ban

fail2ban a 4 types de fichier de configuration :

fail2ban.conf Configuration globale de fail2ban

filter.d/*.conf Filtres spécifiant comment détecter les erreurs d'authentification

action.d/*.conf Actions définissant les commandes pour bannir et débannir les adresses IP

jail.conf Les jails définissent les combinaisons de filtre avec les actions

Format des fichiers de configuration

Les fichiers .conf sont distribués par Fail2Ban. Il est recommandé que ces fichiers restent inchangés pour simplifier les mises à jours. Si nécessaire, les personnalisations devraient être fournies dans des fichiers .local. Par exemple, pour activer le jail [ssh-iptables-ipset] spécifié dans jail.conf, créer un fichier jail.local contenant :

```
jail.local
 [ssh-iptables-ipset]

 enabled = true
```

tail.d et fail2ban.d En plus des fichier .local, pour jail.conf et fail2ban.conf, il y a un répertoire correspondant contenant des fichier .conf additionnels. L'ordre de configuration des jail est :

[ENDSECTION]

[SECTION] name="-" table="listes" imbrication="1"

jail.conf jaid.d/*.conf

jail.local jail.d/*.local

Les fichiers de configuration ont des sections et des paire nom = valeur. Les fichiers de configuration peuvent inclure d'autres fichiers de configuration, qui sont souvent utilisé dans les filtres et actions, en utilisant les directives before et after. En utilisant les mécanismes d'interpolation de chaîne Python, d'autres définitions sont permise et peuvent ensuite être utilisées dans d'autres définitions sous la forme %(name)s :

```
baduseragents = IE|wget
failregex = %(known/failregex)s
```

Additionnellement à l'interpolation $(known/parameter)s, qui ne fonctionne pas pour les paramètres init de filtre et action, un tag d'interpolation <known/parameter> peut être utilisé. Cela permet d'étendre un paramètre un filtre ou action directement dans le jail sans créer de filtre séparément.

```
# filter.d/test.conf:
[Init]
test.method = GET
baduseragents = IE|wget
[Definition]
failregex = ^%(__prefix_line)\s+"<test.method>"\s+test\s+regexp\s+-\s+useragent=(?:<baduseragents>)
# jail.local:
[test]
# use filter "test", overwrite method to "POST" and extend known bad agents with "badagent":
```

```
filter = test[test.method=POST, baduseragents="badagent|<known/baduseragents>"]
```

fail2ban.conf

Ces fichier ont une section, [Definition]. Les éléments sont :

loglevel Niveau de verbosité des logs de sortie : CRITICAL, ERROR, WARNING, NOTICE, INFO, DEBUG. Défaut : ERROR

logtarget Cible des logs : filename, SYSLOG, STDERR ou STDOUT. Défaut : STDERR

socket Fichier socket. Défaut : /var/run/fail2ban/fail2ban.sock. utilisé pour la communication avec les serveur fail2ban

pidfile Fichier pid. Défaut : /var/run/fail2ban/fail2ban.pid

dbfile nom de la base de données. Défaut : /var/lib/fail2ban/fail2bn.sqlite3. Contient les données persistante.

dbpurgeage age de purge de la base, en secondes. Défaut : 86400

jail.conf

Les options suivantes sont applicables dans les jail. Elles apparaissent dans une section spécifiant le nom du jail ou dans la section [DEFAULT]

filter Nom du filtre - nom d'un fichier dans /etc/fail2ban/filter.d/, sans l'extension

logpath Nom des fichiers de log à surveiller, séparés par des newline.

logencoding Encodage des fichiers de logs. Défaut : auto (utilise la locale système)

banaction Action pour le bannissement (défaut : iptables-multiport).

banaction_allports Idem, mais pour certains jails "allports" signifie "pam-generic" ou "recidive". Défaut : iptables-allports

action Actions dans /etc/fail2ban/action.d, sans l'extension

ignoreip Liste des IP à ne pas bannir. Peut inclure un masque cidr

ignorecommand La commande qui est exécutée pour déterminer si l'IP candidate pour le bannissement ne devrait pas être bannie

bantime Durée du ban

findtime Interval de temps, en secondes, avant l'heure courante où les erreurs comptent comme un ban

maxretry Nombre d'erreur qui se produisent dans findtime pour bannir une IP

backend Backend à utiliser pour détecter les changements dans le logpath. Défaut : auto, qui tente, dans l'ordre, pyinotify, gamin, systemd, polling.

usedns Utilise DNS pour résoudre les noms d'hôte qui apparaîssent dans les logs.

failregex Expression régulière Python à ajouter aux failregex du filtre.

ignoreregex expression régulière, si la ligne de log match, à ne pas considérer.

action.d/*.conf

Les fichiers action spécifie quelles commande sont exécutées pour bannir et débannir une adresse IP. Les fichiers action ont 2 section, Definition et Init. La section Init définis des paramètres qui peuvent être écrasés pour un jail particulier. Les commandes suivantes peuvent être présents dans la section Definition

actionstart Commande à exécuter quand le jail démarre

actionstop Commande à exécuter quand le jail s'arrête

actioncheck Commande à exécuter avant tout autre action

actionban Commande à exécuter pour bannir l'adresse IP

actionunban Commande à exécuter pour débannir l'adresse IP

La section Init permet de définir des actions spécifiques à l'action. Les éléments spéciaux suivants peuvent être définis dans la section Init :

timeout Délai max en seconde qu'une commande peut mettre à s'exécuter, avant d'être terminée

Les commandes spécifiées dans la section Definition sont exécutées via un shell système donc les redirections shell et contrôle de process sont autorisés. Les commandes doivent retourner 0, sinon une erreur est loggée. De plus, si actioncheck qui avec un status non-0, il est considéré que le status du firewall a changé et fail2ban doit se réinitialiser. Les tags sont entre <>. Tous les éléments de la section Init sont des tags qui sont remplacés dans les commandes action. Plus d'une commande est autorisée. Chaque commande doit être sur une ligne séparée et indenté avec des espaces blanc.

Tags d'action

Les tags suivants sont substitués dans l'actionban, actionunban, et actioncheck.

ip IPv6 à bannir

failures Nombre de fois que l'erreur se produit dans le fichier de log

ipfailures idem, mais le total de toutes les erreurs pour cette IP dans tous les jails, depuis la base persistante

ipjailfailures idem, mais le total basé sur les erreus de lIP pour le jail courant

time temp (epoch) du ban

matches Chaîne concaténées des lignes du fichier de log des matchs qui génèrent le ban. De nombreux caractères interprétés par le shell sont échappés pour éviter les injection

ipmatches idem, mais inclus toutes les lignes pour l'IP qui sont contenus avec la base persistante

ipjailmatches idem, mais les matches sont limités pour l'ip et pour le jail courant

Fichiers d'action Python

Les actions basées sur python peuvent également être utilisés, où le nom de fichier doit être [actionname].pv. Le fichier python doit contenir une variable Action, qui pointe vers une classe python. Cette classe doit implémenter une interface minimum tel que décris dans fail2ban.server.action.ActionBase.

filter.d/*.conf

Ces fichiers sont utilisés pour identifier les tentatives d'authentification échouées dans les fichiers de log et pour extraire l'adresse IP de l'hôte. La principale section est la section Definition. Il y a 2 définitions de filtre définie dans cette section :

failregex Expression régulière pour matcher les tentatives échouées.

ignoreregex Expression régulière pour identifier les entrées de log qui doivent être ignorées.

Similairement aux actions, les filtres on une section Init qui peut écraser jail.conf/jail.local. Les éléments qui peuvent s'y trouver sont :

maxlines Nombre maximum de lignes à mettre en tampon pour matcher les expressions régulières multilignes

datepattern Spécifie une motif/regex de date personnalisé pour le détecteur de date

journalmatch Spécifie le journal systemd utilisé pour filtrer les entrées du journal.

Les filtres peuvent également avoire une section INCLUDES, utilisée pour lire d'autres configurations :

before Indique que ce fichier est lu avant la section Definition

after Indisque que ce fichier est lu après la section Definition