firejail : sécuriser le lancement d'applications via "un bac à sable" ou une "prison"

- Objet : du tuto Fix Me!
- Niveau requis : Normal
  débutant, avisé
- Commentaires : Lancer des applications non sécurisée dans un environnement sécurisé (sandbox (Bac à sable) ou Jail (prison))
- Débutant, à savoir : Utiliser GNU/Linux en ligne de commande, tout commence là !.
- Suivi :
  à-placer
  - Création par cyrille 26/02/2021
  - Testé par cyrille le 10/02/2021,fiche
- Commentaires sur le forum : Lien vers le forum concernant ce tuto [1] Fix Me!

Nota :

Contributeurs, les Fix Me! sont là pour vous aider, supprimez-les une fois le problème corrigé ou le champ rempli !

Késako

Parfois, il est nécessaire utiliser des applications qui n'ont pas été bien testées dans différents environnements, mais vous devez les utiliser. Dans de tels cas, il est normal de se préoccuper de la sécurité de votre système. Une chose qui peut être faite sous Linux est d'utiliser des applications dans un **bac à sable** (en anglais "**Sandboxing**). Personnellement je préfère le terme de **prison** (***jail***), mais ça n'engage que moi ;)

Exécuter une application dans un bac à sable offre la possibilité de lancer cette application dans un environnement limité. De cette façon, l'application dispose d'une quantité accrue de ressources, nécessaires à son exécution. Grâce à **Firejail**, vous pouvez exécuter en toute sécurité des applications non fiables sous Linux.

Firejail est une application **SUID** (Set Owner User ID) qui réduit l'exposition aux failles de sécurité.

Il permet à un processus et à tous ses descendants d'être exécuté dans une prison (un jail) afin de garantir la sécurité du système.

Installation

Depuis les dépôts

```
apt install firejail
```

```
sudo apt install firejail
```

Depuis le git

```
git clone https://github.com/netblue30/firejail.git
cd firejail
```

```
./configure && make && sudo make install-strip
```

Syntaxe de base

```
firejail <OPTIONS> <APPLICATION>
```

Par exemple

```
firejail mousepad
```

Attention : Sans aucune option, le bac à sable se compose d'un système de fichiers construit dans un nouvel espace de noms de montage et de nouveaux espaces de noms PID et UTS. Les espaces de noms IPC, réseau et utilisateur peuvent être ajoutés à l'aide de la ligne de commande.

Le système de fichiers Firejail par défaut est basé sur le système de fichiers hôte avec les principaux répertoires système montés en lecture seule. Ces répertoires sont **/etc, /var, /usr, /bin, /sbin, /lib, /lib32, /libx32** et **/lib64**. Seuls **/home** et **/tmp** sont accessibles en écriture

Usage avancé

Les fichiers de configuration

Afin de sécuriser le système, il y a toute une flopée d'options (voir le **man de firejail** pour en avoir un aperçu) qui peuvent s'ajouter aux profils de base.

Lors de l'exécution, **Firejail** cherche d'abord dans **~/.config/firejail/** un profil éponyme au nom de l'application et s'il n'en trouve pas, il cherche dans **/etc/firejail/**.

Si aucun profil approprié n'est trouvé, Firejail utilisera un profil par défaut. Ce profil par défaut est assez restrictif. Au cas où l'application ne fonctionne pas, utilisez l'option **–noprofile** pour le désactiver.

Les profils par défaut

Ce sont ces profils qui donneront les privilèges au lancement d'une application via firejail.

Ces profils sont localisés dans

```
ls /etc/firejail/ | wc
   1101    1101   20335
```

Plus de mille profils sont disponibles... Un profil éponyme à l'application lancé pour les plus communes. Et aussi des héritages de profil que nous laisserons ici de côté.

Ce sont ces profils qui définiront les règles de sécurité d'ouverture / fermeture des portes de la prison.

Attention, il ne faut pas modifier ces profils sous **/etc/firejail** car en cas de mise à jour de firejail ils seront écrasés par les nouveaux. Dans ce cas, il suffit de copier le profil à modifier sous **~/.config/firejail/**

Lancement sans nom d'application

SI vous omettez de préciser le nom de l'application, Firejail démarre le shell préféré de l'utilisateur.

```
firejail
Reading profile /etc/firejail/default.profile
Reading profile /etc/firejail/disable-common.inc
Reading profile /etc/firejail/disable-passwdmgr.inc
Reading profile /etc/firejail/disable-programs.inc
Warning: networking feature is disabled in Firejail configuration file

** Note: you can use --noprofile to disable default.profile **

Parent pid 211751, child pid 211752
Child process initialized in 78.39 ms
```

Ainsi lancer, le **/home** de l'$USER sera accessible comme d'habitude , un ls vous le confirmera.

Pour sortir de ce bac à sable, saisissez simplement l'option **exit**

Lister les "bacs à sable"

Pour lister les bacs à sable en cours d'exécution :

```
firejail --list
211751:ragnarok::firejail
```

Les fichiers inclus dans les profils

Les lignes commençant par un hachage (#) sont des commentaires. Au début, les profils comprennent des inclusions defichiers, et à la fin on peut en trouver d'autres. Vous pouvez les examiner vous-même, mais pour les essentielles :

fichiers inclus	Rôle
disable-mgmt.inc	rend les commandes de gestion du système inaccessibles (répertoires / sbin et / usr / sbin, et quelques commandes)
disable-secret.inc	rend les fichiers secrets inaccessibles dans votre répertoire personnel (clés SSH, trousseaux de clés Gnome et KDE, clés GPG, etc.)

disable-common.inc	rend les fichiers inaccessibles à partir d'autres navigateurs, avec la ligne “noblacklist $ {HOME} /. mozilla” ci-dessus garantissant que les fichiers pour Firefox ne sont pas rendus inaccessibles (= liste noire).
disable-devel.inc	rend les commandes de développement inaccessibles (comme les compilateurs, les outils de débogage, les outils de script, etc.)
whitelist-common.inc	rend accessibles les fichiers et répertoires communs dont la plupart des programmes graphiques auront besoin

La ligne “seccomp” active un filtre pour les appels système que le programme peut effectuer. La ligne «protocole» adapte en outre le filtre d'appel système pour la mise en réseau. Voici les options des différentes lignes :

Lignes	Rôle
“netfilter”	Un filtre réseau par défaut est activé si vous configurez un nouvel espace de noms réseau.
“tracelog”	Toute violation où le programme essaie d'accéder à des fichiers ou répertoires sur liste noire sera enregistrée dans /var/log/syslog .
“noroot”	Désactive l'utilisateur root dans le bac à sable.
“whitelist”	Rend accessible les fichiers et répertoires qui seraient utilisés le programme. Les modifications apportées aux fichiers et répertoires de la liste blanche sont persistantes, tout le reste écrit dans votre répertoire personnel est supprimé lorsque le bac à sable est fermé.

L'option --private

Sans doute la plus intéressante. C'est cette option qui va enfermer votre application à l'intérieure d'une prison.

En effet, cette option permet de monter **/root** et **/home/$USER** dans les systèmes de fichiers temporaires. A la fermeture du bac à sable toutes les modifications seront ignorées. Par conséquent, l'application se lancera comme si c'était la première fois qu'on la lançait sur un système vierge, les fichiers de configuration ne seront pas conséquent pas lus.

Exemple:

```
firejail --private claws-mail
```

Afin de monter un répertoire dans le bac à sable, il suffit de sélectionner son emplacement dans private

```
firejail --private --whitelist=/home/$USER/Documents  gimp
```

Ainsi le répertoire Documents de l'$USER sera monté en lecture / écriture et les modifications seront gardées après fermeture de la prison

Les principales options

Option	Action
–blacklist=dirname	Rend le répertoire ou le fichier inaccessible
–cpu=cpu-number	Définit les cœurs de CPU que le programme pourra utiliser
–net=none	Refuse l'accès au réseau du programme

–private	Donne au programme une copie privée de votre répertoire personnel qui est jeté après la fermeture du programme
–private=directory	Utilise le répertoire donné comme répertoire personnel du programme, il n'est pas supprimé après la fermeture du programme
–tmpfs = dirname -	onne au programme un répertoire vide pour le répertoire donné qui est rejeté après la fermeture du programme

Le cas de Firefox

Par exemple, voici les règles de sécurité mises en place par le lancement de firefox dans un bac à sable. Firejail désactivera plusieurs moyens par lesquels Firefox peut obtenir les privilèges root;

- Cela fait même en sorte que le compte root n'existe même pas dans le bac à sable;
- Cela rend les profils firejail inaccessibles;
- Cela rend inaccessibles les répertoires binaires système et plusieurs binaires système sur n'importe quel répertoire de votre chemin;
- Il rend les clés secrètes (trousseaux de clés de session, clés GnuPG, clés SSH, etc.) inaccessibles;
- Cela rend vos fichiers utilisateur d'autres navigateurs Web, clients FTP et programmes de discussion inaccessibles;
- Il rend les fichiers d'historique de votre console inaccessibles (historique de bash, less, etc.);
- Firefox ne peut pas voir les autres processus sur votre système;
- Firefox ne peut écrire des fichiers que dans les répertoires / home, / tmp et / var (s'il y est autorisé) et tous les autres répertoires sont immuables.

Client GUI

Pour ceux que la ligne de commande rebute, il y a un outil graphique (merci de l'avoir précisé @fiche)

```
apt-get install firetools
```

Et les clients sont :

```
firejail-ui
```

```
firetools
```

|| ||

|| ||

|| ||

Un peu de lecture

Beaucoup de documentations mais essentiellement en anglais

- https://firejail.wordpress.com/
- https://linuxhint.com/install_firejail_ubuntu/
- https://wiki.archlinux.org/index.php/firejail

```
man firefail
```

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !