|  |  |
| --- | --- |
| AppArmor - Wikipedia | **AppArmor** |

# Présentation

Linux possède plusieurs outils permettant de sécuriser les applications.   
Toutefois, ces outils demandent parfois une configuration manuelle ou sont souvent méconnus.

Les modules de sécurité Linux (*Linux Security Modules* ou LSM) font partie de ces outils.  
Leur intégration dans le noyau de Linux leur donne ainsi une vue globale de tout le système.

Il existe plusieurs modules de sécurité Linux, SELinux et AppArmor étant les plus utilisés.

**Linux Security Modules**

SELinux est un système de sécurité pour Linux permettant de limiter les actions possibles en fonction d’une politique de sécurité. Il permet d’appliquer notamment différentes méthodes de contrôle d’accès obligatoire.   
SELinux possède un fonctionnement différent de celui d’Apparmor, mais il est également bien plus complet (et également plus complexe).   
Il se base sur des étiquettes attribuées à des ressources en plus de politiques.   
Celui-ci est activé par défaut sur Red Hat Enterprise Linux (et dérivés) et il est fortement recommandé de le laisser activé.

Apparmor est un autre module de sécurité et de contrôle d’accès obligatoire pour Linux.   
Celui-ci repose principalement sur des chemins de fichiers.   
Cependant, il peut limiter l’accès à de nombreuses autres ressources et fonctionalités.   
Sa configuration est uniquement textuelle et sous forme de profils par application.   
Combiné avec ses outils d’administration, il est simple et rapide à mettre en place.

Ces deux outils sont capables de fonctionner en mode alerte uniquement (*Complain/Permissive*), ce qui permet de ne pas perturber le bon fonctionnement des services.   
Les messages générés signalent toute infraction et fournissent une trace auditable.   
C’est pourquoi il est vivement conseillé d’activer ces modules même s’ils ne seront jamais en mode strict/défense. (**enforce**)

# AppArmor

AppArmor (*Application Armor*) est un **contrôle d'accès obligatoire** (*Mandatory Access Control* ou MAC) comme un système de sécurité avancé pour l’écosystème Linux.   
AppArmor permet à l'administrateur système d'associer à chaque programme un profil de sécurité qui restreint les capacités de ce dernier.  
Il s'agit plus précisément d'un outil qui permet de verrouiller les applications en limitant strictement leur accès aux seules ressources auxquelles elles ont droit sans perturber leur fonctionnement.

AppArmor est conçu pour fonctionner (et compléter) avec les autorisations de contrôle d'accès discrétionnaire (DAC) Unix standard tout en étant **facile à utiliser et à déployer**, en permettant à un administrateur de ne **confiner que des applications spécifiques**.   
Dans un système de contrôle d'accès obligatoire (MAC), le noyau impose des restrictions sur les chemins, les sockets, les ports et divers mécanismes d'entrée/sortie.   
AppArmor fait partie du noyau Linux depuis la version 2.6.36.

Alors que le noyau Linux offre une bonne isolation des utilisateurs et un contrôle renforcé des autorisations de fichiers, Le contrôle d’accès obligatoire comme AppArmor fournit des autorisations plus fines et une protection contre de nombreuses menaces inconnues.   
Si une vulnérabilité de sécurité est détectée dans le noyau Linux ou un autre démon système, un système AppArmor bien configuré peut empêcher l'accès aux chemins critiques qui pourraient être vulnérables au problème.

AppArmor peut fonctionner efficacement dans **deux modes** : **Strict** (enforce) et **Alerte** (complain).   
Le **mode strict est l'état de production par défaut** d'AppArmor, tandis que le mode **alerte** **est utile pour développer** **un ensemble** **de règles** basé sur des modèles de fonctionnement réels et pour **consigner les violations**.

AppArmor est configuré à l'aide de fichiers de texte brut dans un format relativement convivial et possède une courbe d'apprentissage plus courte que la plupart des autres systèmes de contrôle d'accès obligatoires (comme SELinux).

Essentiellement, AppArmor fournit des fonctionnalités de contrôle d’accès obligatoire à Linux et est utilisé pour **compléter la fonctionnalité** **traditionnelle DAC** (*Discretionary Access Control* ou MAC)) fournie par le système d'exploitation.   
À l'aide des outils AppArmor les plus élémentaires, un administrateur peut **créer et déployer des profils AppArmor** pour **restreindre l'accès à des processus spécifiques**.   
Par exemple, on pourrait **restreindre** **le fureteur** (*browser*) **Web** pour **permettre uniquement aux utilisateurs d'accéder** **aux fichiers de leurs répertoires personnels**.   
Cela empêcherait un scénario où Alice essaierait de télécharger ou de partager des fichiers appartenant à Bob à son insu.

# Mise en place

## Installation

Pour **installer AppArmor** sur Debian ...  
**>> sudo apt install apparmor apparmor-utils auditd**

**Remarque** ...  
On peut omettre auditd si on n'a pas besoin d'outils de génération de profil.

Si on souhaite **installer des profils de démarrage ou supplémentaires** ...  
**>> sudo apt install apparmor-profiles apparmor-profiles-extra**

Comme AppArmor est un **module du noyau Linux**, on **doit l'activer** ...  
**>> sudo mkdir --parents /etc/default/grub.d**

On crée par la suite le fichier /etc/default/grub.d/apparmor.cfg ...  
**>> sudo nano /etc/default/grub.d/apparmor.cfg  
GRUB\_CMDLINE\_LINUX\_DEFAULT="$GRUB\_CMDLINE\_LINUX\_DEFAULT apparmor=1 security=apparmor"**On enregistre et on quitte.

On **met à jour GRUB** ...  
**>> sudo update-grub**

On redémarre ensuite le système ...  
**>> sudo systemctl reboot**

Pour vérifier si **AppArmor est en fonction** …  
**>> sudo aa-enabled**

Une fois que l'on a redémarré le système, on peut vérifier si **AppArmor est activé** ...  
**>> sudo aa-status**

Cette commande **répertorie les profils** AppArmor chargés et **répertorie leur état de conformité actuel**   
(enforce, complain, ...)

On peut afficher la liste de programmes confinés par un profil AppArmor.   
Un programme confiné est un programme qui est affecté et limité (soit passivement, en mode alerte ou *complain*, soit activement en mode strict ou **enforce**) par AppArmor ...  
**>> sudo ps auxZ | grep -v '^unconfined'**

Une fois qu'AppArmor est en cours d'exécution, on peut s’assurer que le service s'exécute …  
**>> sudo systemctl status apparmor.service  
● apparmor.service - Load AppArmor profiles  
 Loaded: loaded (/lib/systemd/system/apparmor.service; enabled; vendor preset: enabled)  
 Active: active (exited) since Tue 2021-12-28 13:55:40 EST; 58s ago  
 Docs: man:apparmor(7)  
 https://gitlab.com/apparmor/apparmor/wikis/home/  
 Process: 240 ExecStart=/lib/apparmor/apparmor.systemd reload (code=exited, status=0/SUCC>  
 Main PID: 240 (code=exited, status=0/SUCCESS)  
 CPU: 142ms**

**Remarque** …  
Il est possible de **démarrer et d’activer AppArmor au démarrage** …  
**>> sudo systemctl enable --now apparmor.service  
Synchronizing state of apparmor.service with SysV service script with /lib/systemd/systemd-sysv-install.  
Executing: /lib/systemd/systemd-sysv-install enable apparmor**

Cette commande garantira que le système est toujours opérationnel avec AppArmor et prêt à appliquer les profils.

Maintenant que le service est en cours d'exécution, il est possible de commencer à utiliser les outils AppArmor   
(**aa-tools**) pour surveiller et gérer le système.

Pour commencer, on peut utiliser l'**outil** **aa-status** afin de **vérifier quels profils sont chargés** et quel **statut ils ont** **actuellement** …  
**>> sudo aa-status**ou **>> sudo apparmor\_status  
29 profiles are loaded.  
13 profiles are in enforce mode.  
 /usr/bin/man  
 /usr/bin/pidgin  
 /usr/bin/pidgin//sanitized\_helper  
 /usr/bin/totem  
…  
16 profiles are in complain mode.  
 /usr/bin/irssi  
 /usr/sbin/dnsmasq  
 /usr/sbin/dnsmasq//libvirt\_leaseshelper  
…  
0 processes have profiles defined.  
0 processes are in enforce mode.  
0 processes are in complain mode.  
0 processes are unconfined but have a profile defined.**

Il est possible de constater que **29 profils** ont été chargés par le système et sont dans un état d'exécution.   
Cela signifie qu'**AppArmor** **surveillera les processus qui correspondent à ces profils** et **décidera si une action spécifique** **est autorisée ou refusée** par la stratégie.

Une autre chose à souligner à propos de cette sortie est le statut alerte (complain) et non confiné (**unconfined**).

Lorsqu'un profil est en **mode alerte**, AppArmor lui permettra d'**effectuer** (presque) [[1]](#footnote-1) **toutes les tâches sans restriction**. Toutefois, il les **enregistrera dans le journal d'audit en tant qu'événements**.   
Ceci est utile lorsque l’on essaie de créer un profil pour une application mais que l’on ne sait pas à quoi elle doit accéder.

En revanche, le **statut non confiné** (unconfined) permettra au **programme d'effectuer n'importe quelle tâche et ne l'enregistrera pas**.   
Cela se produit généralement si un profil a été chargé après le démarrage d'une application, ce qui signifie qu'il s'exécute sans restriction depuis AppArmor.

Il est également important de noter que seulement les processus qui ont des profils sont répertoriés sous ce statut non confiné, tous les autres processus qui s'exécutent sur l’hôte mais n'ont pas de profil créé pour eux ne seront pas répertoriés sous aa-status.

# Utilisation

Le **répertoire /etc/apparmor.d** est l'endroit où se retrouvent les **profils d'AppArmor**.   
Ce répertoire peut être **utilisé pour manipuler le mode de l'ensemble des profils**.

On peut lancer tous les profils en mode strict (enforce) …  
**>> sudo aa-enforce /etc/apparmor.d/\*  
Profile for /etc/apparmor.d/abstractions not found, skipping  
Profile for /etc/apparmor.d/apache2.d not found, skipping  
Setting /etc/apparmor.d/lsb\_release to enforce mode.  
Setting /etc/apparmor.d/nvidia\_modprobe to enforce mode.  
Setting /etc/apparmor.d/sbin.klogd to enforce mode.  
Setting /etc/apparmor.d/sbin.syslogd to enforce mode.  
Setting /etc/apparmor.d/sbin.syslogd to enforce mode.   
…  
Setting /etc/apparmor.d/sbin.syslog-ng to enforce mode.**

On peut lancer tous les profils en mode alerte ou complain (ce qui est la recommandation) …  
**>> sudo aa-complain /etc/apparmor.d/\*  
Profile for /etc/apparmor.d/abstractions not found, skipping  
Profile for /etc/apparmor.d/apache2.d not found, skipping  
Setting /etc/apparmor.d/lsb\_release to complain mode.  
Setting /etc/apparmor.d/nvidia\_modprobe to complain mode.  
Setting /etc/apparmor.d/sbin.klogd to complain mode.  
Setting /etc/apparmor.d/sbin.syslogd to complain mode.   
…  
Setting /etc/apparmor.d/sbin.syslog-ng to complain mode.**

Pour mettre seulement le profil d'une application en marche …  
**>> sudo aa-enforce sbin.syslog-ng  
Setting /etc/apparmor.d/sbin.syslog-ng to enforce mode.**

La commande apparmor\_parser est utilisé pour charger un profil dans le noyau.   
Il peut aussi être utilisé pour recharger un profil déjà chargé en utilisant l'option --replace (-r).

Pour charger un profil …  
**>> sudo cat /etc/apparmor.d/<Nouveau profil> | sudo apparmor\_parser --add  
>> sudo cat /etc/apparmor.d/bin.ping | sudo apparmor\_parser --add**

**Remarque** …  
Le nouveau profile doit préalablement exister.

Pour recharger un profil existant …  
**>> sudo cat /etc/apparmor.d/<Profil> | sudo apparmor\_parser --replace  
>> sudo cat /etc/apparmor.d/sbin.syslog-ng | sudo apparmor\_parser --replace**

**Remarque** …  
Il est possible de remplacer les options --add par -a et --replace par -r.

Le service apparmor peut également être utilisé pour recharger tous les profils …  
**>> sudo systemctl reload-or-restart apparmor.service**

Il est aussi possible d’utiliser le répertoire /etc/apparmor.d/disable comma alternative à la commande apparmor\_parser --remove (-R) pour désactiver un profil …  
**>> sudo ln -s /etc/apparmor.d/<Profil>** **/etc/apparmor.d/disable/  
>> sudo ln -s /etc/apparmor.d/bin.ping /etc/apparmor.d/disable/**

**>> sudo apparmor\_parser --remove /etc/apparmor.d/<Profil>  
>> sudo apparmor\_parser --remove /etc/apparmor.d/bin.ping**

Pour activer à nouveau un profil désactivé, il faut supprimer le lien symbolique vers le profil dans le répertoire /etc/apparmor.d/disable/ et charger le profil en utilisant l'option --add …  
**>> sudo rm /etc/apparmor.d/disable/<Profil>  
>> sudo rm /etc/apparmor.d/disable/bin.ping**

**>> sudo cat /etc/apparmor.d/<Profil> | sudo apparmor\_parser –add  
>> sudo cat /etc/apparmor.d/bin.ping> | sudo apparmor\_parser –add**

# Création d'un profil

On utilise deux outils pour générer et ajouter des profils, aa-genprof et aa-logprof.

Le premier est utilisé pour surveiller et créer un profil pour une application la première fois qu'elle est exécutée (c'est-à-dire lors de la création du profil pour la première fois) afin qu'AppArmor puisse connaître les tendances des applications et vos demander quel comportement doit être adopté.

La second est utile une fois qu’il existe un profil existant et que l’on doit autoriser/refuser l'accès à certaines tâches qui ont déjà été enregistrées pendant les modes strict ou alerte.

Il existe deux manières de créer de nouveaux profils ...

* La première s'applique dans le cas d'un processus qui s'exécute simplement, pour une durée limitée ;
* La seconde s'applique aux processus de type démon (*daemon*), ou encore aux processus exécutés périodiquement ou sur une longue période.

**Processus à durée limitée**

Afin de créer un profil à durée limité, Il suffit de lancer l'application à l'aide de la commande aa-genprof.   
AppArmor détectera automatiquement les accès faits par l'application, générera le fichier profil approprié et le placera dans le répertoire /etc/apparmor.d.

Pour créer un profil à durée limitée …  
**>> sudo aa-genprof <Commande>  
>> sudo aa-genprof /usr/bin/uname  
Writing updated profile for /usr/bin/uname.  
Setting /usr/bin/uname to complain mode.  
Before you begin, you may wish to check if a profile already exists for the application you wish to confine. See the following wiki page for more information:** [**https://gitlab.com/apparmor/apparmor/wikis/Profiles**](https://gitlab.com/apparmor/apparmor/wikis/Profiles) **Création de profil: /usr/bin/uname  
…  
[(S)can system log for AppArmor events] / (F)inish**

On entre **f** afin de terminer l’opération.

**Processus de type démon**

Afin de créer un profil de type démon, on crée d'abord un profil temporaire à l'aide de la commande aa-autodep.   
On laisse tourner le processus à confiner durant le temps nécessaire, puis on corrige le fichier de profil à l'aide des messages laissés dans les fichiers log, grâce à la commande aa-logprof.

Au besoin, on continue à laisser tourner le processus et on relance la commande aa-logprof, jusqu'à ce que le fichier de profil soit adéquat. On peut alors placer ce fichier en mode strict (enforce).

Pour créer un profil de type démon …  
**>> sudo aa-autodep <Chemin du service>  
>> sudo aa-autodep /usr/sbin/nginx**

**>> sudo aa-logprof**

Pour la conception d'un plan de test, il est important de réfléchir à la manière dont l'application doit être examinée.   
Le plan de test devrait être divisé en tests unitaires.   
Chaque test unitaire devrait avoir une courte description et énumérer les étapes à suivre.

* Quelques tests unitaires standard sont …
* Démarrage du programme ;
* Arrêt du programme ;
* Rechargement du programme ;
* Test de toutes les commandes prises en charge.

# Mise à jour des profils

Lorsque le programme ne se comporte pas correctement, des messages d'audit sont journalisés.

L'application aa-logprof peut être utilisée pour analyser les messages d'audit AppArmor dans les fichiers journaux.

# Notifications de violations

Le mode strict (enforce) applique les règles et rapporte les tentatives de violation, alors que le mode alerte (complain) se contente d'enregistrer dans les journaux système les appels système qui auraient été bloqués, sans les bloquer réellement.

L’utilitaire aa-genprof n'est en fait qu'un petit script intelligent qui utilise aa-logprof.  
Il crée un profil vide, le charge en mode alerte, puis lance l’utilitaire aa-logprof.   
Ce dernier est un outil qui met à jour un profil en fonction des violations qui ont été enregistrées.   
On peut donc relancer cet outil plus tard, de manière à améliorer le profil nouvellement créé.

Une autre solution pourrait être l'installation du paquet logcheck, qui se chargera d'envoyer par courriel diverses informations contenues dans les logs, dont les violations signalées par AppArmor.

… <https://linux.die.net/man/8/logcheck> ...

Passer au niveau supérieur

Jusqu'à présent, vous en avez suffisamment fait avec AppArmor pour pouvoir l'utiliser au quotidien pour gérer les profils sur un seul système. Cependant, dans des environnements vastes ou en constante évolution, les outils utilisés jusqu'à présent ne suffisent pas, nous devons être en mesure d'adapter notre surveillance et notre gestion des profils ainsi que d'apprendre à différencier les bonnes politiques des mauvaises politiques. Pour l'instant, je veux vous laisser une réflexion intéressante sur AppArmor et nous essaierons de l'aborder dans la partie 2 de ce guide.

https://medium.com/information-and-technology/so-what-is-apparmor-64d7ae211ed

1. Le statut du mode alerte (*complain*) appliquera toujours les règles de refus explicites dans un profil. [↑](#footnote-ref-1)