# AppArmor

Last updated by | Cerena Hostains | 23 juin 2025 at 11:16 UTC+2

## Installation et renforcement de service sur un serveur Debian 12 avec AppArmor

### Objectif

Configurer AppArmor pour renforcer la sécurité d'un service, ici nous prendrons en exemple le service : `zabbix_server` sur un serveur Debian 12.

### Introduction

AppArmor (Application Armor) est un système de contrôle d'accès obligatoire (MAC : Mandatory Access Control). Il permet de restreindre les capacités d'un processus à ce qui est strictement nécessaire. AppArmor est intégré nativement à Debian depuis la version 10 et repose sur LSM (Linux Security Module) dans le noyau Linux (≥ 2.6).

#### Intérêts d'AppArmor

- Protéger contre des vulnérabilités de type zero-day (Une vulnérabilité zero-day (ou 0-day) est une faille de sécurité inconnue du développeur du logiciel au moment où elle est découverte par un attaquant.)
- Suivre et cartographier les accès réels d'un service
- Détecter des comportements suspects ou inattendus
- Générer des alertes de sécurité

> Comparé à SELinux (présent sur les distributions Red Hat), AppArmor est souvent perçu comme bien plus simple à configurer.

### Prérequis

- Un serveur avec Zabbix Server installé
- Droits root
- Un snapshot système créé au préalable

### Installation des outils AppArmor

```
apt install apparmor-utils rsyslog
```

> Le paquet `apparmor-utils` apporte toutes les commandes nécessaires à la gestion des profils.

### Vérification de l'état d'AppArmor

Lister les profils actifs :

```
aa-status
```

Lister les processus non confinés (potentiellement exposés) :

```
aa-unconfined
```

> Variante avancée : `aa-unconfined --paranoid` affiche tous les processus avec port ouvert non renforcés.

## Création d'un profil AppArmor pour Zabbix

### Étape 1 : Générer une structure de profil

```
aa-autodep zabbix_server
```

> Cela crée un fichier `/etc/apparmor.d/usr.sbin.zabbix_server` avec des règles minimales.

Lire le fichier généré :

```
more /etc/apparmor.d/usr.sbin.zabbix_server
```

## Passage en mode complain

Ce mode permet d'enregistrer les violations sans les bloquer :

```
cd /etc/apparmor.d
aa-complain usr.sbin.zabbix_server
```

Vérifiez les changements avec :

```
aa-status
```

## Observation des logs

Lancez une lecture temps réel :

```
tail -f /var/log/syslog
```

Vous verrez des entrées similaires à :

```
apparmor="ALLOWED" operation="mknod" profile="/usr/sbin/zabbix_server" ...
```

### Utilisation du service pour enrichir les logs

Manipulez le service Zabbix :

```
service zabbix-server stop
service zabbix-server start
```

> Conseil : manipulez un maximum le service pour générer un maximum de logs.

### Génération interactive des règles avec `aa-logprof`

Lancez :

```
aa-logprof
```

Vous serez invité à autoriser ou refuser des accès. Exemple :

```
Profile: /usr/sbin/zabbix_server
Capability: setgid
Severity: 9
(A)llow / [(D)eny] / (I)gnore / Audi(t) / Abo(r)t / (F)inish
```

Tapez `A` pour autoriser, puis `S` pour sauvegarder les modifications.

Vérifiez les changements :

```
more /etc/apparmor.d/usr.sbin.zabbix_server
```

> Note : vous pouvez relancer `aa-logprof` autant de fois que nécessaire.

### Passage en mode enforce

Ce mode active la protection stricte (blocage + journalisation) :

```
aa-enforce usr.sbin.zabbix_server
```

Redémarrez le service :

```
service zabbix-server stop

service zabbix-server start
```

Vérifiez l'état :

```
aa-status
```

### Astuces supplémentaires

- Les chemins inscrits dans les profils doivent être génériques pour éviter des erreurs lors de la rotation de fichiers.

Exemple :

```
owner /var/log/zabbix/zabbix_server.log r,

owner /var/log/zabbix/zabbix_server.log w,
```

devient :

```
owner /var/log/zabbix/* rw,
```

- Le nom des fichiers de profils doit suivre ce format : chemin absolu avec `/` remplacés par `.`

Exemple : `/usr/sbin/zabbix_server → usr.sbin.zabbix_server`

### Dépannage

Si le service ne fonctionne plus :

- Consultez :

```
/var/log/syslog

/var/log/zabbix/zabbix_server.log
```

- Lancez à nouveau :

```
aa-logprof
```

> Certaines situations nécessitent un nouveau redémarrage du serveur pour observer tous les accès manquants.

### Visualisation JSON (bonus)

```
apparmor_status --json
```

### Table des Références

- Documentation AppArmor