**SE Linux**

|  |  |  |  |
| --- | --- | --- | --- |
| **DESC** | | | |
| Security Enhanced Linux : Mécanisme de contrôle d’accès avancé  Concurrencé par GRSecurity | | | |
| **FILES** | | | |
| /etc/selinux/ | config | | |
|  | Targeted/ | modules/active/modules/ | Fichiers .pp binaire de module SE Linux |
|  |  | Policy/ | Politique de sécurité active |
|  |  | Contexts/files/file\_contexts.local | BDD standard correspondance  File -> type\_context  Les news fichiers et dir sont aussi  Inscrits ici |
| /var/log/ | messages | Logs d’accès refusés light | |
|  | Audit/audit.log | Logs d’accès refusés détaillés | |
| **QUICK START** | | | |
|  | | | |

**Généralités :**

* Il est aux processus et aux fichiers ce que le firewall est au réseau.
* On définit ce qu’un utilisateur ou un processus peut faire : Par exemple un processus ne sera autorisé à interagir qu’avec certains fichiers et processus : un utilisateur ne pourra exécuter n’importe quel processus ou scripts, uniquement ce dont il a besoin par exemple
* Par défaut, tout est non autorisé, c’est la politique de sécurité qui va autoriser.
* Sécurité dans Linux :
  + Sécurité DAC : première défense de Linux (Droits d’un fichier…)
  + SE Linux vient après, si la première ligne est franchie, et est implanté au niveau kernel.
* Chaque processus confiné à un/plusieurs domaines et chaque fichier est étiqueté (chaque fichier possède son etiquette dans la BDD SE Linux), et il y a correspondance entre les domaines (domain\_context) et les étiquettes (type\_context).
* Attention si SE Linux est disabled ou permissif, et si des fichiers sont créés pendant cette période, ils n’auront pas d’étiquette. Erreurs bloquantes si on réactive SE Linux : il faut tout ré étiqueter.

**Modes de SE Linux :**

* Enforcing : SE Linux assure la sécurité et logs les accès refusés
* Permissive : SE Linux n’assure pas la sécurité, aucun accès n’est restreint mais on log quand même les accès refusés (permet de tester SE Linux)
* Disabled

**Politique de sécurité de SE Linux :**

* Targeted : Permissions custom précises
* Minimum :
* Mls : multi level security protection. Mode de protection avancé, besoin de package additionnels.

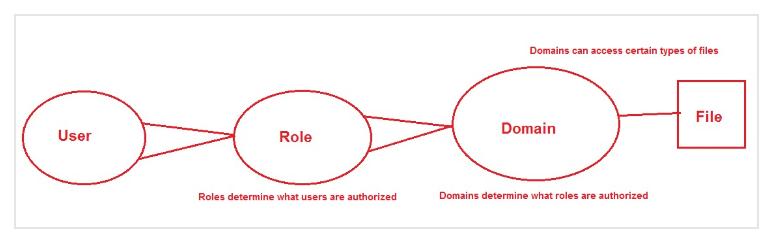
**Changement de mode : activé SE Linux à partir d’un état désactivé :**

* Mettre Permissive dans la conf
* Reboot
* Mettre enforcing dans la conf
* Reboot

Ou

* Set enforce 0|1|2|enforcing|permissive|disabled et pas besoin de reboot

**Fonctionnement général :**



* Un User Linux est mappé sur un SE Linux User.
* Rôle : passerelle entre un User et un domaine.
* Un domaine est le contexte de sécurité d’un processus. Un processus tourne dans un domaine, qui lui dit ce qu’il peut faire.
* Un type est le contexte de sécurité d’un fichier. Chaque fichier est étiqueté.
* Des règles d’accès font la correspondance entre les domaines et les types

Les processus et les fichiers héritent du contexte de sécurité de leur parent (un dossier est le parent d’un fichier)

Si un fichier est copié ailleurs, son contexte de sécurité n’est pas préservé et il hérite du contexte de sécurité du dossier de destination, sauf si flag –preserver=context est présent

**Contexte de sécurité :**

* D’un fichier : ls -Z

User :rôle :type\_context :MLS

MLS = criticité de la ressource

* D’un processus actif : ps -efZ

User :rôle :domain\_context :MLS

* D’un programme quelconque : Ls –Z /bin/…
* D’un utilisateur : Id -Z

User :rôle :domain :MLS

**Changement d’un contexte de sécurité de manière permanente d’un fichier :**

* Semanage pour MAJ la BDD SE Linux file -> type\_context
* Puis restorecon pour affecter le type\_context au fichier

**Règles d’accès :**

Allow <domain\_context> <type\_context> :<class> {<permissions>} ;

Class = la nature de la ressource (fichier, dossier, lien, périphérique, port…)

**Utilisateurs :**

Quand un utilisateur est créé, il est mappé par défaut au user local \_\_default\_\_, puis mappé au user SE Linux unconfined\_u.

**Booléens :**

Permettent de changer une politique SE Linux à haut niveau ( = quand t’y connais rien ^^)