TP2: Sécurité OS avec SELinux sous Red Hat 7

D.E. MENACER

Le contrôle d'accès sous Linux

- Le système doit fournir :
  - La **disponibilité** : continuité de services, fournir l'accès aux services
  - L'**intégrité** : Les données ne doivent pas être altérées
  - La **confidentialité** : utilisable que par des personnes identifiées

Les modèles de sécurité sous Linux

- DAC
- MAC
- RBAC
- DTE
- LSM

DAC

- Le modèle **DAC** - Discretional Access Control est le mode de sécurité classique sous Unix/Linux
- Les utilisateurs et les groupes d'utilisateur sont propriétaires des fichiers et répertoires sur le système
- Les propriétaires et root peuvent modifier la politique d'accès aux données.
- Confiance en l'utilisateur, pas de type de données (secret d'états, secret, confidentiel, public, ..), pas de rôle utilisateur (admin, DSI, chef, …)
- Les processus ou programmes sont exécutés par un propriétaire ou un groupe propriétaire : accès à toutes les données du propriétaire
- Les listes de contrôle d'accès, ACL permettent une gestion plus fine des autorisations sur les fichiers, pas les processus

MAC

- Le modèle **MAC** -Mandatory Access Control - Contrôle d'accès obligatoire
- Dans ce modèle, on **labélise** (on pose une **étiquète** sur) les données (fichiers, socket, ..) et les processus :
  - **Top secret**: Un utilisateur accrédité public ne pourra donc pas avoir accès aux données top secret
  - **Confidentiel**
  - **Départements** (marketing, direction, …)
  - **Public**
- La politique interdit tous sauf si on autorise

DTE

- Le **DTE** – Domain and Type Enforcement utilisé par Selinux sur le même principe que le MAC.
- Le DTE labélise des objets
- Contrôles d'accès entre les sujets (processus, domaines) et les objets (fichiers, répertoires, sockets, ..)
- Le DTE les confine dans un **domaine** qui en limite les actions
- **Exemple**: Un serveur Web qui s'exécute en root n'aura accès qu'aux domaines HTTPD (libraires, pages web, fichier de configuration du serveur)

RBAC

- Le modèle **RBAC** - Role-Based Access Control - Contrôle d'accès par **rôle**
- Définit des rôles pour les utilisateurs, processus :
  - Administrateur internet
  - Administrateur de base de données
- Définit des contrôles d'accès à ces rôles
- **Exemples**: Commande **sudo**, SGBD

LSM

- **LSM** - Linux Security modules
- Intégré au noyau depuis la version 2.6, LSM est une API qui vérifie la conformité des règles de sécurité
- Introduit initialement pour SELinux, il est utilisé également par **AppArmor**
- **Security-Enhanced Linux – SELinux: permet de définir des politiques de sécurité de type DTE**
- **Apparmor**: Permet de définir des politiques de sécurité de types **MAC** (concurrent de SELinux)

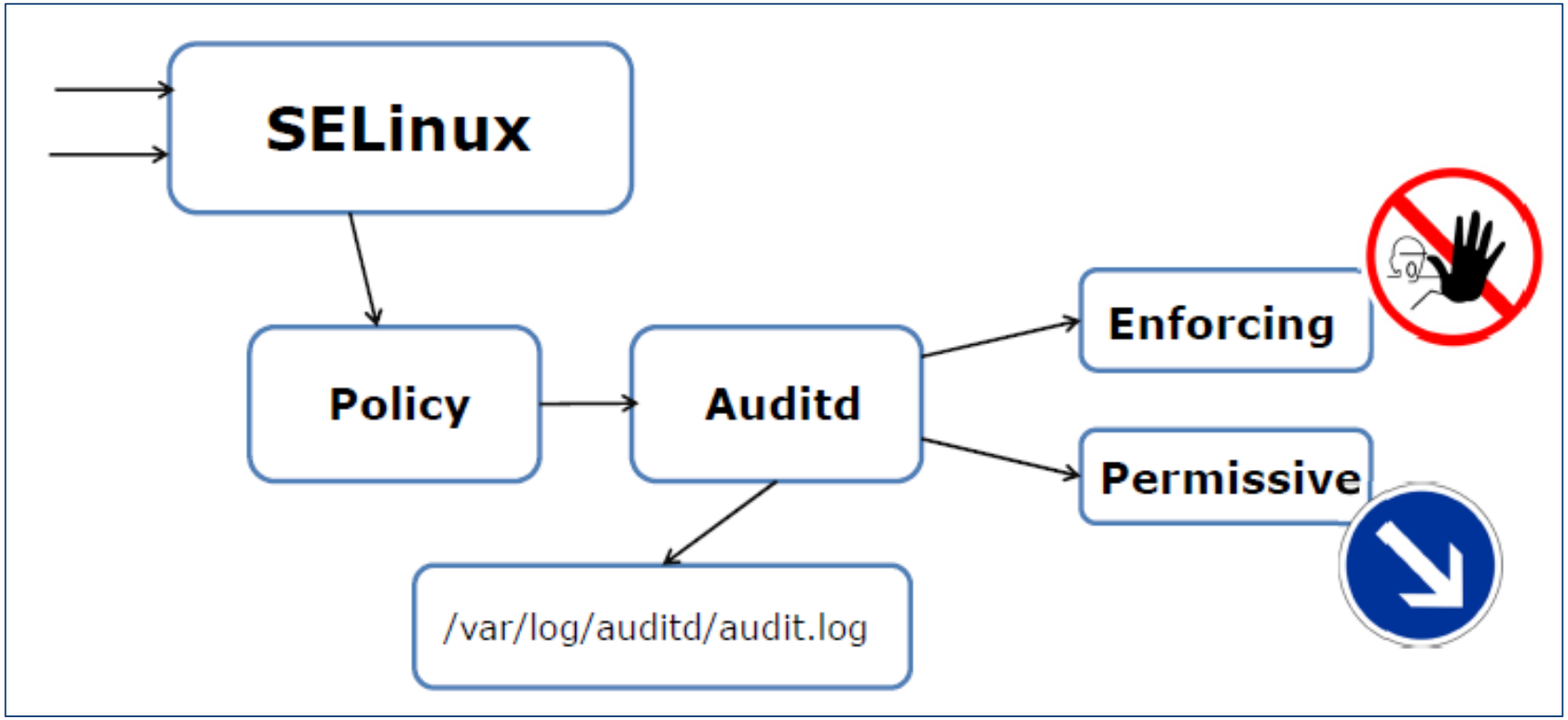
Architecture

Modèle	Niveau implémentation	Exemples
RBAC	Application	sudo
LSM	Module Noyau	Apparmor SELinux
MAC DTE	Extensions Système de fichiers Noyau: Extension module Processus Noyau: Extensions module sockets	Apparmor SELinux
DAC	Système de fichiers: fichiers et répertoires	Modes Unix ACL

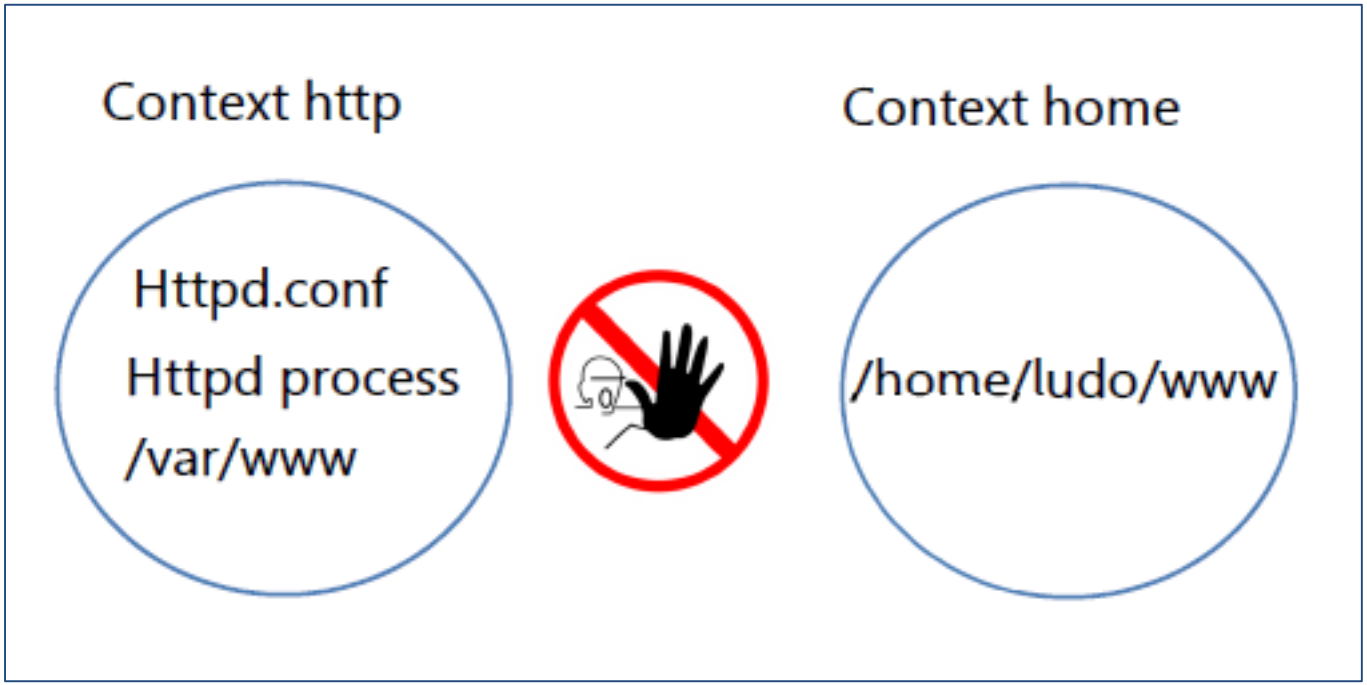
SELinux

- Security-Enhanced Linux (ou SELinux) est une architecture de sécurité de type **MAC**
- Intégrée dans le noyau 2.6.x à l'aide des modules **LSM**
- Projet de la **NSA** (National Security Agency)
- Activement développée par Red Hat
- Disponible sur Red hat, Debian, Gentoo, Ubuntu …

Le modèle Linux



Le modèle Linux



Les modes SELinux

- **Enforcing** : Application des règles SeLinux
- **Permissive** : mode de déboguage. Les règles sont logguées mais ne bloquent pas les accès
- **Disabled** : SELinux Désactivé
- Modification des modes

```
# getenforce
# setenforce 1
# setenforce 0
# vi /etc/selinux/config
```

Les modes SELinux

- **Les contextes SELinux avec l'option –Z concernent 3 entités:**
  - **Fichiers**
  - **Processus**
  - **Sockets**
- **L'affichage produit les informations sur les:**
  - **Utilisateurs**
  - **Rôles**
  - **Labels**

 # ls -Z

-rw-------. 1 system_u:object_r:admin_home_t:s0 root root 1558 23 nov. 09:56 anaconda-ks.cfg

 # ps -auxZ

system_u:system_r:sshd_t:s0-s0:c0.c1023 root Ss 11:19 0:00 /usr/sbin/sshd -D

 # netstat –Zatunp

tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 1356/sshd system_u:system_r:sshd_t

TP SELinux

- **Objectif**: comprendre le fonctionnement de SELinux à travers un **use case**: publication de pages Web.
- **Mise en place**: Une machine RHEL7 avec serveur Web Apache installé
  - Yum install httpd
- **Durée**: 2H

Préparation

- Configuration d’une page Web HTML
- Création de la page HTML

Configuration d’une page html

- Créer, dans /var/www/html, un fichier index.html:
- $ su –lc ‘mkdir /var/www/html/selinux && chown -R user:user /var/www/html/selinux’
- $ su –lc ‘touch /var/www/html/selinux/index.html’

Page html

- **Insérer le code html suivant dans index.html:**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xml:lang="fr">
<head>
    <title>Test de SELinux</title>
    <meta http-equiv="Content-Type" content="text/html;
    charset=UTF-8" />
</head>
<body>
    <p>SELinux me permet d'afficher cette page </p>
</body>
</html>
```

Afficher les contextes

- $ ls -alZ /var/www/html/selinux/

drwxr-xr-x user user
unconfined_u:object_r:httpd_sys_content_t:s0
.

drwxr-xr-x root root
system_u:object_r:httpd_sys_content_t:s0 ..

-rw-rw-r-- user user
unconfined_u:object_r:httpd_sys_content_t:s0
index.html

Test URL

- Dans le navigateur (Firefox), tapez: http://localhost/selinux/index.html
- L'URL fonctionne normalement.

Changement de contexte

- $ cp /var/www/html/selinux/index.html /tmp/selinux.html && mv /tmp/selinux.html /var/www/html/selinux/

Afficher les contextes

- $ ls -alZ /var/www/html/selinux/

drwxr-xr-x user user
unconfined_u:object_r:httpd_sys_content_t:s0 .

drwxr-xr-x root root
system_u:object_r:httpd_sys_content_t:s0 ..

-rw-rw-r-- user user
unconfined_u:object_r:httpd_sys_content_t:s0
index.html

-rw-rw-r-- user user
unconfined_u:object_r:user_tmp_t:s0
selinux.html

Nouveau test

- Dans le navigateur, tapez: http://localhost/selinux/selinux.html.
- Le navigateur affiche:

Forbidden - You don't have permission to access /selinux/selinux.html on this server

- *Pourquoi ce message?*

Corriger le contexte

- Deux possibilités généralement pour corriger un contexte de fichier erroné :
  - restaurer le contexte par défaut du chemin
  - spécifier un contexte « manuellement »

Trouver le contexte adéquat

- $ matchpathcon /var/www/html/selinux/selinux.html
- /var/www/html/selinux/selinux.html system_u:object_r:httpd_sys_content_t:s0

Méthode 1: Restaurer le contexte

- Commande **restorecon**: restaurer le contexte d'origine
- $ su -lc 'restorecon -v /var/www/html/selinux/selinux.html'
  restorecon reset /var/www/html/selinux/selinux.html context unconfined_u:object_r:user_tmp_t:s0->system_u:object_r:httpd_sys_content_t:s0

Nouveau test

- l'URL http://localhost/selinux/selinux.html

fonctionne normalement.

Méthode 2: Modification manuelle du contexte

- Pour Apache, si vous souhaitez placer des fichiers ou dossiers qui doivent être accessibles en dehors de /var/www, vous aurez à en modifier le contexte SELinux, sinon Apache ne pourra pas y accéder.
- Deux solutions sont possibles :
  - **chcon**
  - **semanage**

CHCON

- **Commande chcon:** chcon permet de changer le contexte SELinux d'un fichier ou dossier donné, mais pas de façon permanente

$ su -lc 'chcon -t httpd_sys_content_t /var/www/html/selinux/selinux.html‘

- **Inconvénient**: **chcon** permet le changement temporaire; il sera perdu lors du prochain étiquetage (redémarrage de la machine, restauration de contextes).

SEMANAGE

- **semanage**, en revanche, ne permet pas de modifier directement le contexte d'un fichier ou d'un dossier, mais de définir un contexte par défaut qui sera appliqué par **restorecon** ou lors d'un ré-étiquetage du système de fichiers.
- Cette solution est donc à privilégier lorsque l'on souhaite placer un contexte qui restera permanent.

Chemins semanage

- Les chemins de semanage sont stockés dans */etc/selinux/targeted/contexts/files/file_contexts.local*

Tests semanage

- **Déplacez le dossier** */var/www/html/selinux* vers */srv/web/selinux/*

mkdir /srv/web

mv /var/www/html/selinux /srv/web/

Tests semanage

- **Création des liens:**

```
# cd /var/www/html
# ln -s /srv/web/selinux .
# ls -alZ /srv/web/
drwxr-xr-x  root    root    unconfined_u:object_r:var_t:s0
    .
drwxr-xr-x  root    root    system_u:object_r:var_t:s0      ..
drwxr-xr-x  trasher trasher
    unconfined_u:object_r:httpd_sys_content_t:s0 selinux
```

Tests semanage

- Les URL http://localhost/selinux/selinux.html

et http://localhost/selinux/index.html
fonctionnent normalement car la commande mv conserve le contexte et les droits des fichiers et dossiers

Tests semanage

- Si nous lançons restorecon:

restorecon -R -v /srv/web

restorecon reset /srv/web/selinux context unconfined_u:object_r:httpd_sys_content_t:s0->system_u:object_r:var_t:s0

restorecon reset /srv/web/selinux/selinux.html context system_u:object_r:httpd_sys_content_t:s0->system_u:object_r:var_t:s0

restorecon reset /srv/web/selinux/index.html context unconfined_u:object_r:httpd_sys_content_t:s0->system_u:object_r:var_t:s0

- **Les pages ne sont plus accessibles**

Tests semanage

- Définissons donc à l'aide de ***semanage*** un contexte par défaut pour le dossier /srv/web, ainsi que ses descendants :

```
# semanage fcontext -a -t httpd_sys_content_t
'/srv/web(/.*)?'
```

Tests semanage

- **Appliquons le contexte par défaut:**

 # restorecon -R -v /srv/web

restorecon reset /srv/web context
unconfined_u:object_r:var_t:s0->system_u:object_r:httpd_sys_content_t:s0

restorecon reset /srv/web/selinux context
system_u:object_r:var_t:s0->system_u:object_r:httpd_sys_content_t:s0

restorecon reset /srv/web/selinux/selinux.html context
system_u:object_r:var_t:s0->system_u:object_r:httpd_sys_content_t:s0

restorecon reset /srv/web/selinux/index.html context
system_u:object_r:var_t:s0->system_u:object_r:httpd_sys_content_t:s0

Fin du test semanage

- Les pages http://localhost/selinux/selinux.html

et http://localhost/selinux/index.html sont de nouveau accessibles.