Capabilities, seccomp-bpf, NoNewPrivileges

Timothée Ravier
siosm@floss.social - tim.siosm.fr/cours - github.com/travier

5e année cycle ingénieur, filière STI
Option Sécurité des Systèmes Embarqués et du Cloud (2SEC)

2024 - 2025

Sommaire

Capabilities

Capabilities (1)

- Utilisateur `root` tout puissant sur un système Linux
- Tous les autres utilisateurs non privilégiés par défaut
- *Capabilities* Linux : donner une partie des privilèges de `root` à des programmes non privilégiés
- Exemple :
  - `CAP_NET_BIND_SERVICE` : Bind un socket sur un port < 1024
  - `CAP_NET_ADMIN` : Configurer les interfaces réseaux
  - `CAP_SETUID` : Définir arbitrairement les UIDs d'un processus
  - `CAP_SYS_ADMIN` : Effectuer un grand nombre d'opération privilégiées
  - Voir capabilities(7) pour la liste complète

Capabilities (2)

- Peuvent être associées à des exécutables pour être ainsi transmises aux processus résultant
- Chaque *thread* possède son propre ensemble de *capabilities* (comme pour les UID/GID)
- Leur état est stocké dans plusieurs variables (nommés *sets*) :
  - *Effective* : le set utilisé pour les contrôles dans le noyau;
  - *Permitted* : limite actuelle des *capabilities* disponible
  - *Inheritable* : préservées lors d'un appel à `execve` et conservées si le fichier exécuté les possède
  - *Ambient* : préservées lors d'un appel à `execve`

Capabilities (3)

- Capabilities Bounding Set : limite définitive des *capabilities* disponibles pour un *thread*
- Ensemble de régles complexes pour déterminer quelles *capabilities* sont conservées, perdues ou transmises lors des appels à `execve`, `setuid`, etc.
- Voir capabilities(7)

Capabilities & `root`

- De nombreuses *capabilities* sont en pratique équivalente à `root` :
  - `CAP_SYS_ADMIN` : the new root
  - False Boundaries and Arbitrary Code Execution

Capabilities & *namespaces*

- `CLONE_NEWNET`, `CLONE_NEWNS`, `CLONE_NEWPID`, `CLONE_NEWUTS`, `CLONE_NEWIPC`, `CLONE_NEWTIME`, `CLONE_NEWCGROUP` :
  - Nécessite `CAP_SYS_ADMIN`
- `CLONE_NEWUSER` :
  - Mode non privilégié (*unprivileged*) disponible depuis le noyau 3.8
  - Parfois désactivé par défaut : Comportement variable suivant les distributions

seccomp & seccomp-bpf

seccomp

- Mode strict qui n'autorise un processus à ne faire usage qu'aux appels système `read`, `write` et `_exit`

seccomp-bpf

- Extension du modèle initial qui filtre les appels système disponibles pour un processus
- Filtre décrit sous forme de programme BPF (*Berkeley Packet Filter*)
- Si le processus utilise un appel système filtré, le noyau peut retourner un code d'erreur, envoyer un signal ou tuer le processus
- En pratique : utilisation de la libseccomp, des options de systemd ou du gestionnaire de conteneurs
- Possibilité de créer des listes de refus / authorisation

No New Privileges

Drapeau *No New Privileges*

- Mis en place pour un processus
- Définitif (ne peut pas être inversé)
- Ne peut plus obtenir plus de privilèges lors d'un appel à `execve`
- No New Privileges Flag (Documentation noyau Linux)

Suite

SELinux