TP2 SELinux

Avant de commencer, vérifier qu'un patch `selinux-policy-devel` est bien installé sur votre machine – sinon l'installer. Il est aussi possible que sur certaines machines les `setools` ne soient pas installées – les installer aussi.

Exercice 1:

1. Écrire un petit programme C qui affiche un entier à la sortie std. Créer ensuite un module SELinux (fichiers `*.if, *.te, *.fc` pour votre programme, permettant qu'il puisse être lancé dans un shell par n'importe quel utilisateur :
   (a) Déclarer et créer le domaine pour le processus, ainsi que le type de l'exécutable.
   (b) Suivant l'exemple `/usr/share/selinux/devel/example.{if,te}` ajouter des appels aux macros m4 pour donner des droits à votre domaine.
   (c) Ajouter une "porte d'entrée" dans votre domaine, à partir du domaine `unconfined_t`.
   (d) Ne pas oublier de réétiqueter l'exécutable avant de le lancer en exécution, pour tester du bon comportement du domaine !
   (e) En modifiant le mode SELinux à permissif, lancez en exécution votre programme et, en notant les messages `avc : denied`, rajoutez les permissions nécessaires – utiliser plutôt les macros m4 définies chez `oss.tresys.com/docs/refpolicy/api`.
2. Modifier votre programme de sorte qu'il puisse demander l'entier à l'entrée std. Modifier en conséquence le module SELinux créé.
3. Votre programme devrait cette fois-ci se voir redirigée sa sortie std. dans un fichier du répertoire `/root/try/`. Créer un type fichier pour étiqueter ce fichier, et modifier en conséquence le module SELinux.
   Vous pouvez choisir plusieurs variantes de création du type fichier :
   – Fichier créé par votre programme (*attention* au type du répertoire dans lequel le fichier sera créé, et aux droits que votre domaine possède sur ce type !).
   – Fichier dans lequel votre programme pourrait rajouter des informations, mais sans avoir le droit de le créer – le seul `unconfined_t` pouvant le créer alors.
4. Modifier votre programme et le module SELinux associé de sorte qu'il lance un fils, et qu'il puisse récupérer son état lors de sa terminaison.
5. Modifier votre programme de sorte que le fils puisse faire `exec` un autre programme dans le domaine `unconfined_t`.

Vous povez essayer d'ajouter d'autres fonctionnalités à votre programme, et modifer le module SELinux en conséquence !