Contrôle d'accès e le POSIX Access Control Lists(ACL)

CS435 - Administration de Système

Dan Pham et Fabrício Nascimento

Octobre 2009

Introduction

Quand l'objective c'est contrôle l'accès sur les données dans une système de fichiers, il y a plusieurs formes de règlement. Par défaut, les systèmes POSIX (Portable Operatin System Interface)[2, 3] ont une mécanisme que permettre de associer chaque entité avec un ensemble de règle, lequel est composé par une séquence d'octet que exprime le droit du propriétaire, de son groupe et des autres utilisateurs.

Ce mode traditionnel est assez simple et capable de adresser les problème plus fréquents. Par contre, il pose des limitation aux administrateur de système, lesquels fréquemment doivent employer quelques configuration non évidentes afin d'être capable de exprimer ces besoins. Par exemple les application comme le serveur FTP Proftp[4] ont ces exclusive façon de résoudre ces problèmes de droits pour accéder les objets du système de fichier.

À couse de remédier ces limitation présente les UNIX permettent d'employer les ACL.

Cette article présente une exposition sur les ACL POSIX, ces mode de fonctionnement, ces clefs de succès et désavantages. Le texte est fortement basé en l'article de Andreas Gruembacher[1] dont a été dans l'équipe que as ajouté le support aux ACL dans le noyaux Linux pour les système de fichier ext2 et ext3, lequel est le système de fichier plus utilisé dans les monde UNIX.

Le POSIX 1003.1

Traditionnellement les système qui implémentions les patron POSIX avaient une système simple et puissante de permission, quand même, certain problèmes ont arrive, et éventuellement les problèmes ont apparaître.

Après savoir la nécessite de régler sur le domaine de sécurise et non seulement les ACL, une groupe a était forme pendant la définition de la famille de patron POSIX 1003.1. Les premières documents POSIX qui ont été considère ces question étaient les document 1003.1e (*System Application Programming Interface*) et 1003.2c (*Shell and Utilities*), cependant, la première approximation de ce sujet était trop ambitieuse. Les groupe responsable pour le patronisation avait centre ces effort dans une tas assez grande de choses, lesquelles comprenant *Access Control Lists* (ACL), *Audit*, *Capability*, *Mandatory Access Control* (MAC), et *Information Labeling*[1].

En Janvier de 1998[1] le financement était fini, par contre, le travaille n'était pas prés. De toute façon le dixsèptieme brouillon a été publique quand même[5].

Donc après cette an, les système UNIX appelé "*trusted*" (Trusted Solaris, Trusted Irix, Trusted AIX ont été développer avec quelque parts de la documentation 17. Ces systèmes ne sont pas complètement compatible entre eux.

Aujourd'hui la plupart des système UNIX et UNIX-like support ACL. Ces implémentassions sont usuellement compatible avec le document 17. Le projet TrustedBSD aussi avait ajouter les ACL sur les système BSD. Les ACL e les MAC FreeBSD-RELEASE avaient apparu en 2003.

Les base des ACL sont lancé sur le système traditionnel présent usuellement dans presque tous les système UNIX, alors, avant de préciser sur les ACL on parlerais du modelé traditionnel.

Système de permission Traditionnel

Le modelé traditionnel POSIX offre trois group de utilisateur qui sont le propriétaire, le group e les autres. Chaque group a une octet que indique les permission de lecture (**r**ead), écrire (**w**rite) et exécution (e**x**ecute). La première classe fournit les permission pour le utilisateur que rempli le rôle de propriétaire, ensuite, vient les droits pour le groupe principal du propriétaire enfin les droites pour touts les autres utilisateurs.

Après les trois octets peut venir le *Set User Id*, *Set Group Id* et *Sticky bit*, lesquelles sont utilisé dans certain cases. Il faut faire attention avec le *Sticky Bit*, il permit les utilisateur normale d'exécuter les utilitaire comment le administrateur(*root*), par contre, quelque manque de sécurité peut compromettre le système entière.

Seulement le *root* peut créer les groups e changer les association de groupes. Celui-là que aussi peut changer les propriétaire.

Les ACL

Les types de ACL	
Type d'entrée	**format**
Propriétaire	user : :rwx
Utilisateur nommée	user :name :rwx
Groupe propriétaire	group : :rwx
Groupe nommée	group :name :rwx
Masque	mask : :rwx
Autres	other : :rwx

Dans une modèle de de sécurise ACL, si quelque agent faire une requête pour acceder aux donnés, il faut consulte les ACL pour une entrée que permetre l'operation demandé.

Linux

ACL Implementation in Linux

Conclusion

Références

[1] Andreas Gruenbacher, *POSIX Acess Control Lists on Linux.* http ://www.suse.de/ agruen/acl/linux-acls/online/, 2003.

Algorithm 1 Vérifie se une utilisateur peut ou ne peut pas accéder une objet du système de fichier

if the user ID of the process is the owner **then**
  the owner entry determines access
else if the user ID of the process matches the qualifier in one of the named user entries **then**
  this entry determines access
else if one of the group IDs of the process matches the owning group and the owning group entry contains the requested permissions **then**
  this entry determines access
else if one of the group IDs of the process matches the qualifier of one of the named group entries and this entry contains the requested permissions **then**
  this entry determines access
else if one of the group IDs of the process matches the owning group or any of the named group entries, but neither the owning group entry nor any of the matching named group entries contains the requested permissions **then**
  this determines that access is denied
else
  the other entry determines access.
end if
if the matching entry resulting from this selection is the owner or other entry and it contains the requested permissions **then**
  access is granted
else if the matching entry is a named user, owning group, or named group entry and this entry contains the requested permissions and the mask entry also contains the requested permissions (or there is no mask entry) **then**
  access is granted
else
  access is denied.
end if

[2] IEEE Std 1003.1-2001 (Open Group Technical Standard, Issue 6), Standard for Information Technology–Portable Operating System Interface (POSIX) 2001. ISBN 0-7381-3010-9. http ://www.ieee.org/

[3] IEEE 1003.1e and 1003.2c : Draft Standard for Information Technology–Portable Operating System Interface (POSIX)–Part 1 : System Application Program Interface (API) and Part 2 : Shell and Utilities, draft 17 (withdrawn). October 1997. http ://wt.xpilot.org/publications/posix.1e/

[4] Mark Lowes : Proftpd : A User's Guide March 31, 2003. http ://proftpd.linux.co.uk/

[5] Winfried Trümper : Summary about Posix.1e. Publicly available copies of POSIX 1003.1e/1003.2c. February 28, 1999. http ://wt.xpilot.org/publications/posix.1e/