Controle de Accés e le POSIX Access Control Lists(ACL)

NE410 - Administration de Systéme

Dan Pham et Fabrício Nascimento

October 2009

Introduction

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

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

À couse de remédier ces limitation presenté les unix permettent l'employ de les ACL.

Cette article presenté une exposition sur les ACL POSIX, ces mode de fonctionnement, ces clés de succes et désavantages. Le texte est fortement basé en l'article de Andreas Gruembacher[1] dont a été dans l'equipe 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.

Les systèmes POSIX

Système Tradicional

Le modele traditional 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 execution (e**x**ecute). La premiere classe fournit les permission pour le utilisateur que rempli le role de propriétaire, ensuite, vien les droits pour le groupe principal du propriétaire enfim les droites pour touts les autres utilisateurs.

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

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

Les ACL

En Janvier de 1998[1]

En Janvier de 1998[1]

Pour remédier à ce limitation "trusted" UNIX systems comme Trusted Solaris, Trusted Irix, Trusted AIX ont été developer avec

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.

Algorithm 1 Verifie se une utilisateur peut ou ne peut pas acceder 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

Conclusion

Références

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

[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/