```
Documentation valable pour AbulÉdu Pro 1.6
```

Les ACLs et HackD

Présentation

Les permissions et les partages de fichiers sur AbulÉdu 1.6 sont implémentés grâce une fonctionnalité puissante des derniers noyaux et systèmes de fichiers Linux appelée « les **ACL**s » (*Access Control List*, ou Listes de Contrôles d'Accès). Ces ACLs permettent d'étendre le système de permissions sur les fichiers (auparavant limité à “utilisateur propriétaire”, “groupe” et “tous les autres”) : grâce à eux il est possible de fixer des permissions spécifiques pour autant de groupes ou de comptes utilisateurs dont on a besoin.

Le système des ACLs permet aussi l'héritage de permissions : on supersède la notion de *umask* (définie pour chaque utilisateur, mais indépendante du chemin où il travaille), et l'on peut définir des permissions différentes qui se propagent (ou pas) dans des sous-répertoires indépendamment de l'utilisateur qui manipule les fichiers.

Problèmes sous-jacents

- L'utilisation d'ACLs étend les possibilités d'affinage du système de permissions mais entraine pas mal de complications si on ne le gère pas avec rigueur. Il est donc indispensable d'avoir un système de vérification ou de “retour à la normale”. De même, les permissions “classiques” ugo affichées sous UNIX n'ont plus la même signification, et un `chmod 777` n'a plus du tout les mêmes conséquences que dans l'ancien modèle. Il est donc nécessaire d'avoir des outils de haut niveau pour corriger rapidement les problèmes sans handicaper les administrateurs avec l'apprentissage de notions compliquées.
- Nous mettons immédiatement de côté les ACLs pour des comptes utilisateurs, et n'utiliserons que les ACLs pour les groupes. Ceci permettra d'éviter les manipulations répétées des méta-données du système de fichiers. Pour autoriser un certain individu à accéder d'une certaine manière à certaines données, nous n'aurons ainsi qu'à le rendre membre du groupe de ceux qui ont ces permissions-là, et non à appliquer récursivement la permission individuelle sur toutes les données.
- Certaines assomptions et décisions du système (ainsi que de nombreux bugs d'applications) vont dans le sens contraire des fonctionnalités que nous voulons. Par exemple, la décision de conserver autant que possible les attributs d'un fichier lors de son déplacement ou de sa copie pose un problème clair lorsque l'utilisateur veut partager un de ses documents : chez lui le document a des droits restreints, et lorsqu'il le copie dans un répertoire de partage, il s'attend à ce que ça suffise pour que le fichier soit partagé, mais ce n'est pas le cas car les permissions d'origine (restrictives) ont été conservées. Une manipulation supplémentaire est nécessaire pour réellement donner les droits d'accès aux personnes concernées. Mais la manipulation supplémentaire est freinée par un manque de connaissances et de maîtrise certain du système d'ACLs par un utilisateur lambda. D'où la nécessité d'un système d'accompagnement des utilisateurs, afin de faciliter leur vie dans ce cas-là.

HackD : Horizon Auto Check Daemon

C'est pour répondre à tous ces problèmes que HackD a été créé. Ce démon surveille tous les répertoires de groupes à l'affut du moindre changement. Lorsqu'un évènement qui l'intéresse survient (dans un contexte précis), par exemple l'apparition d'un fichier ou d'un répertoire, le démon applique aussitôt les permissions (sous forme d'ACLs) nécessaires au partage effectif du fichier ou répertoire (en respect de la notion de permissivité).

HackD est très léger en ressources système : pendant un `rsync` sur plusieurs centaines de méga de données, il reste en dessous de 0.1% d'utilisation de l'UC. Ceci est possible grâce à l'utilisation d'une autre technologie récente : `inotify`. Pour surveiller les répertoires partagés, HackD demande simplement au noyau de l'informer de tout changement qui survient à l'intérieur de ces répertoires. Comme le noyau est un passage obligé dans ces cas-là, HackD est prévenu instantanément et agit tellement vite que l'utilisateur ne le remarque même pas. Le service est ainsi rendu : les bonnes permissions sont appliquées tellement vite que pour un humain c'est instantané.