TD 3 - ASR5 système d'exploitation

Protocols et droits

24 mars 2018

I Placement de blocs de fichiers

Nous utilisons un système de fichiers avec une table de blocs libres. Après un formatage, la table des blocs d'une partition est : 100000000000000. Le premier bloc est utilisé par le répertoire racine de la partition.

Pour placer les fichiers, le système essaye en priorité de remplir les zones libres les plus petites. Donnez l'occupation des blocs pour le scénario suivant :

Q.I.1) - Écriture du fichier A qui réclame 6 blocs.

table	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
contenu	R	A1	A2	A3	A4	A5	A6								

Q.I.2) - Écriture du fichier B qui réclame 5 blocs.

table	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
contenu	R	A1	A2	A3	A4	A5	A6	B1	B2	B3	B4	B5			

Q.I.3) - Suppression du fichier A.

table	1	0	0	0	0	0	0	1	1	1	1	1	0	0	0
contenu	R							B1	B2	B3	B4	B5			

Q.I.4) - Écriture du fichier C qui réclame 8 blocs.

table	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
contenu	R	C4	C5	C6	C7	C8		B1	B2	B3	B4	B5	C1	C2	C3

Q.I.5) - Suppression du fichier B.

table	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
contenu	R	C4	C5	C6	C7	C8							C1	C2	C3

II Droit unix

Sur le système considéré, il y a trois utilisateurs :

— gmw qui fait partie du groupe utilisateurs ;

— asw qui fait partie des groupes utilisateurs et developpeurs ;

— scw qui ne fait partie d'aucun de ces deux groupes.

Q.II.1) - Représentez dans une matrice les possibilités d'accès des fichiers et répertoires suivants pour chaque utilisateur.

```
-rw-r-----. 1 gmw developpeurs     27 janv.  2 11:06 donnees.txt
-rw-r--r--. 1 gmw utilisateurs     24 janv.  2 10:46 PPP-Notes
-rwxr-sr-x. 1 asw developpeurs 152392 janv.  2 10:47 prog1
-rw-rw----. 1 asw utilisateurs 164488 janv.  2 10:55 project.t
-rw-r-----. 1 asw developpeurs 118581 janv.  2 10:49 splash.png
```

	donnees.txt	PPP-Notes	prog1	project.t	splash.png
gsw	rw	rw	rx	rw	-
asw	r	r	rwx	rw	rw
scw	-	r	rx	-	-

Q.II.2) - Que signifie le bit s du fichier prog1, à quoi cela peut-il servir ? donnez un exemple d'utilisation.

C'est le bit setgid : la commande est exécutée avec le gid du fichier et pas celui du lanceur. Cela permet d'acquérir temporairement les droits correspondants. Ici par exemple, quel que soit l'utilisateur lançant le programme prog1, prog1 pourra lire le fichier splash.png (pour afficher l'icone alors que ce fichier est protégé par exemple). La commande sudo, qui appartient à root, fonctionne selon un principe similaire, mais avec un bit setsuid.

Exemple :
$ whereis passwd
passwd: /usr/bin/passwd /bin/passwd /etc/passwd /usr/share/man/man5/passwd.5.bz2 /usr/share/man/man1/pas
$ l /usr/bin/passwd
lrwxrwxrwx 1 root root 11 21 oct. 2015 /usr/bin/passwd −> /bin/passwd
$ l /bin/passwd
−rws−−x−−x 1 root root 46984 21 oct. 2015 /bin/passwd
Qu'en est-il de sudo ?

Plus d'info :
`http://docs.oracle.com/cd/E19683-01/816-4883/secfile-69/index.html`

Q.II.3) - Expliquer comment partager localement des fichiers avec au moins une personne, sans qu'elle(s) ai(en)t accès à l'ensemble de vos fichiers stockés localement.

Question à laisser en suspens, pour pratique des étudiants.
Le répertoire doit être exécutable, pas readable, et les personnes doivent connaître les noms des fichiers. Peut-on mieux faire, avec les réponses ci-dessus ?

III ACL ldap

Un annuaire électronique est une base de données spécialisée, dont la fonction première est de retourner un ou plusieurs attributs d'un objet grâce à des fonctions de recherche multi-critères. Aujourd'hui, on utilise souvent les annuaires basés sur LDAP (*Lightweight Directory Access Protocol*). Un annuaire LDAP contient des objets, organisés de façon hiérarchique (arborescente), comme cela est illustré par la figure 1. Chaque objet peut posséder un certain nombre d'attributs, dont certains sont normalisés, comme par exemple :

- dc pour *domain component* : identifie une composante de l'annuaire ;
- dn pour *distinguished name* : nom distinct, identifiant unique de l'objet, il contient les dn des ancêtres ;
- cn pour *common name* : nom commun de l'objet ;
- o pour *organisation* : nom de l'organisation ;
- ou pour *organisation unit* : branche de l'organisation ;
- ...

Un annuaire LDAP permet à certains de s'authentifier et de manipuler les informations. Ces manipulations sont limitées par des directives d'accès sous la forme d'ACLs. Pour définir une directive d'accès vous devez utiliser :

access to <what> [by <who> [<access>] [<control>]]+

Les 4 parties importantes de la directive sont :

- <what> spécifie l'objet cible par exemple « le mot de passe de tout le monde » ou « la photo d'un membre du service informatique » ;

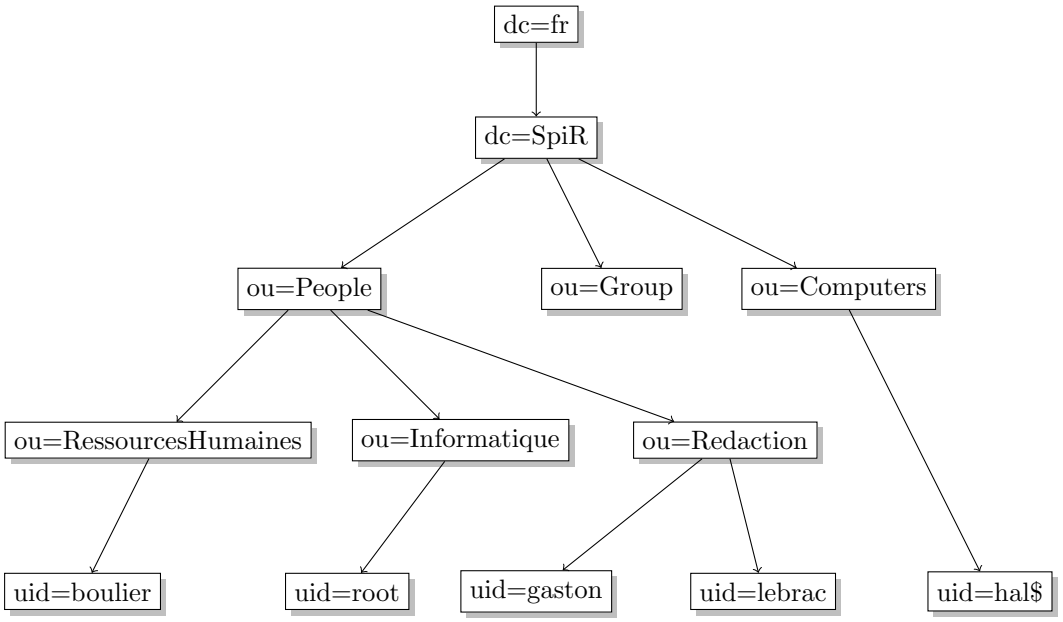

Figure 1 – Organisation de l'annuaire LDAP dans notre exemple.

— <who> l'objet à qui le droit est attribué, par exemple, l'administrateur ou n'importe qu'elle ordinateur ;
— <access> le droit attribué (none|auth|compare|search|**read**|**write**|manage) ces droits sont croissant, celui qui a le droit de lecture (**read**) peut forcement comparer (compare).
— <control> est permet de choisir l'action à faire ensuite. Si rien n'est précisé, c'est stop (arreter d'évaluer les ACLs), mais il est possible de choisir continue (évalue la suite de la même directive) ou break (passe à la directive suivante).

Pour tester si un objet a accès à une certaine cible, la liste est évaluée par ordre d'écriture. Si une clause <what> correspond à la cible demandée, les champs <who> sont évalués. Lorsqu'un des champs correspond à l'objet authentifié, le droit obtenu est celui qui est spécifié par <access>. Le test s'arrête alors, sauf si cela est spécifié par <control>.

On considère que chaque directive est terminée implicitement par un by * none stop refusant l'accès aux objets qui ne sont pas nommés dans la directive. De même, la liste des ACL est supposée se terminer par

```
access to *
  by * none
```

Ce qui signifie que lorsque rien n'est précisé, personne n'a d'accès.

L'idée de l'exercice n'est pas de connaitre la syntaxe des ACL propre à openldap mais uniquement leur fonctionnement. C'est pourquoi nous allons utiliser les abréviations suivantes

— *root* l'utilisateur administrateur : la définition serait exactement dn.exact="uid=root,ou=Informatique,ou=People,dc=SpiR,dc=fr" ;
— *machines* descendants de l'objet `Computer` : dn.children="ou=Computer, dc=SpiR, dc=fr" ;
— *employés* tous les déscendants de `People` : dn.children="ou=People, dc=SpiR, dc=fr" ;
— *infos employés* **tous** les attributs des employés ;
— *infos perso* les attributs représentant les informations personnelles : homePostalAddress,homePhone,street ;
— *RH* tous les employés déscendant de `RessourcesHumaines`.

nous proposons le tableau de directives suivant (les directives en *italique* sont implicites) :

directive	**what**	**who**	**access**	control
1	userPassword	"root" "machines" self anonymous *	manage read write auth none	*stop* *stop* *stop* *stop* *stop*
2	*	"root" self * *	manage write *none* *none*	*stop* *stop* break *stop*
3	"infos employés"	"RH" * *	write *none* *none*	*stop* break *stop*
4	telephonenumber, displayName	* *	read *none*	*stop* *stop*
5	"infos perso"	*	none	*stop*
6	*	*	read	*stop*

Q.III.1) - Pour les communications internes, les telephones intérogent l'annuaire grâce à une connection annonyme. Peuvent-ils connaitre l'identité de celui qui appel ? Quelles directives sont concernées ?

Les téléphones doivent rechercher le nom (displayName) d'un utilisateur à partir de son numéro de téléphone (telephoneNumber).

- la directive 1 ne correspond pas car elle concerne le mot de passe ;
- la directive 2 correspond car elle concerne tout
  - l'auteur de la demande est anonyme et donc la première ACE qui s'applique est la 3e : ∗ none break
  - elle n'accord aucun droit (none)
  - le control associé est break ce qui signifie qu'on continue à lire les règles suivantes.
- la directive 3 s'applique car elle concerne tous les attributs des employés. Comme la précédente elle se termine par ∗ none break
- la directive 4 s'applique et donne ∗ **read** stop ce qui accord le droit de lecture sur les 2 attributs et stop la lecture des ACLs.

Au final, un utilisateur anonyme comme un téléphone à le droit de lire le numéro de telephone et le nom des employés. Il peut donc aussi faire une recherche en fonction du numero afin d'obtenir ce nom (le droit de recherche est inférieur à celui de lecture). Donc l'affichage du nom est possible.

Q.III.2) - Que se passe-t'il si on retire la ligne « self, write » de la seconde directive (celle concernant *) ? Que peuvent faire sur leurs propres données :

- l'utilisateur uid=gaston,ou=Redaction,ou=People,dc=SpiR,dc=fr
- l'utilisateur uid=boulier,ou=RessourcesHumaines,ou=People,dc=SpiR,dc=fr

Donnez le droit maximum obtenu pour chacun des attributs telephonenumber (le téléhone de travail), homePostalAddress (l'adresse personnelle), userPassword et jpegPhoto.

Si on retire la directive, un utilisateur n'a plus le droit de modifier ses propres données par défaut. Il peut toujours modifier sont propre mot de passe (directive 1) mais pour les autres champs, il n'y a pas de droit spécial accordé sur ses données. Donc :

- Gaston, peut :
  - modifier sont mot de passe userPassword (directive 1)
  - n'obtient rien par la directive 2 (il n'est pas root)
  - n'obtient rien par la directive 3 il n'est pas dans Ressource Humaine)
  - peut lire son nom et son numero de teléphode telephonenumber (directive 4)
  - n'a aucun droit sur ses informations personnelles : donc ne peux ni lire ni même comparer la valeur du champs homePostalAddress (directive 5)
  - peut lire le reste (directive 6) donc peut lire jpegPhoto.
- M. Boulier peut :
  - modifier sont mot de passe userPassword (directive 1)
  - n'obtient rien par la directive 2 (il n'est pas *root*)
  - obtient le droit d'écrire ses informations personnelles car il est dans *RH* (directive 3) il peut donc écrire telephonenumber, homePostalAddress, et jpegPhoto.

Q.III.3) - Sous Active Directory, un utilisateur non authentifié (annonymous) ne peut pas consulter le contenu de l'arbre. La seule possiblilité offerte est de s'authentifier ce qui impose de fournir son login et son mot de passe et de demander au serveur ldap de les vérifier (droit auth sur le mot de passe).
Que faut-il changer aux ACL proposées pour obtenir le même comportement ?

Entre les directives 1 et 2 il faut ajouter une directive :

directive	**what**	**who**	**access**	control
1.5	*	anonymous	none	stop
		*	none	break

Ce qui ajoute une règle concernant tous les objets. Cette dernière n'accorde aucun droit à *anonymous* et arrête la lecture des ACL. Pour tous les autres utilisateurs, on n'accorde rien non plus, mais on continue la lecture des ACL. Les droit ce ces dernier ne sont donc pas modifié par cette règle. Le droit d'*anonymous* est bien réduit à la seul authentification (acquis par la directive 1) sans aucun autre.