TD 1 - LIF12 système d'exploitation

Passage d'informations sur les pipes

25 avril 2017

I 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 de d'un membre du service informatique » ;
- <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

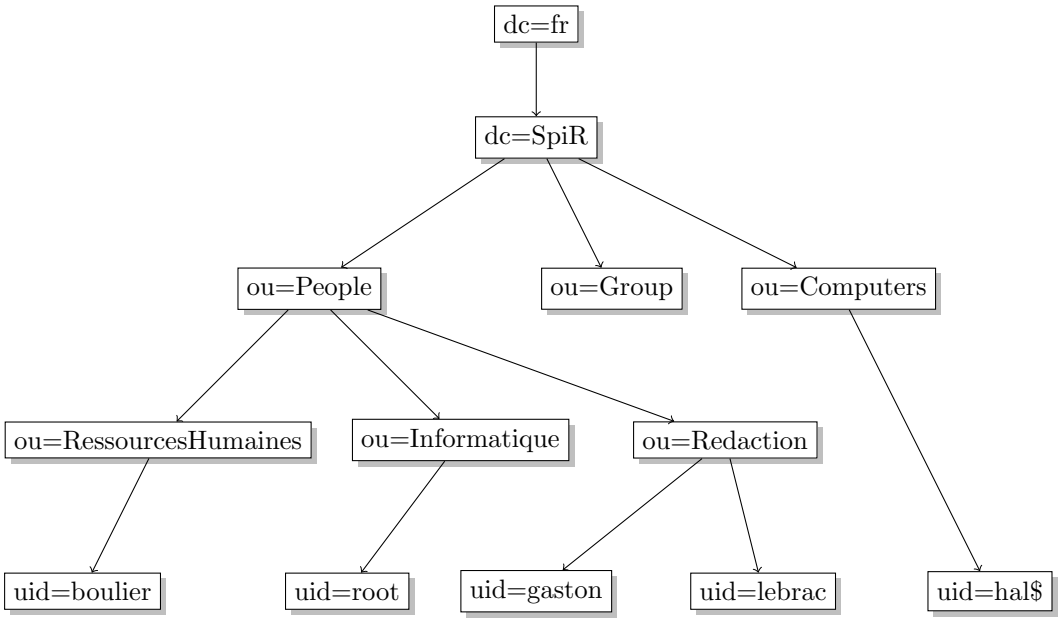

Figure 1 – Organisation de l'annuaire LDAP dans notre exemple.

— *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"	manage	*stop*
		"machines"	read	*stop*
		self	write	*stop*
		anonymous	auth	*stop*
		*	none	*stop*
2	*	"root"	manage	*stop*
		self	write	*stop*
		*	*none*	break
		*	*none*	*stop*
3	"infos employés"	"RH"	write	*stop*
		*	*none*	break
		*	*none*	*stop*
4	telephonenumber, displayName	*	read	*stop*
		*	*none*	*stop*
5	"infos perso"	*	none	*stop*
6	*	*	read	*stop*

Q.I.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 ?

Q.I.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.

Q.I.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 ?

II Lecture de données

De nombreux protocoles de communication ont besoin d'échanger des données de taille variables. Cela pose problème car lors de l'échange, celui qui lit les données doit avoir un moyen de reconnaître la fin du message.

L'une des méthodes pour résoudre ce problème est de choisir un fanion. C'est à dire un octet spécial (ou un groupe d'octets) qui marque la fin du message.

Q.II.1) - Donnez 2 autres méthodes permettant résoudre le même problème.

Pour pouvoir utiliser les fanions, il faut être capable de lire les données jusqu'à ce dernier, sans en oublier et sans lire de données en trop.

Q.II.2) - Donnez le code d'une fonction :

```
vector<char> read_until(int sock, char until);
```

Qui lit exactement de qui est envoyé sur la socket `sock` jusqu'au prochain caractère `until`

Q.II.3) - La solution proposée n'est pas très efficace. Comment peut-on faire mieu.