Authentification LDAP

Le but du TP est de mettre en place le mécanisme d'authentification en utilisant un annuaire global.

Deux étapes, mettre en place l'annuaire puis configurer l'authentification.

Installation du serveur LDAP

Installez le serveur openldap.

Configuration d'icelui

Configurez icelui en modifiant le fichier /etc/openldap/slapd.conf :

Incluez les schémas qui vont bien (ceux que vous avez trouvé en TD),

include /etc/openldap/schema/core.schema

include /etc/openldap/schema/cosine.schema

include /etc/openldap/schema/inetorgperson.schema

include /etc/openldap/schema/nis.schema

Créez les objets qui vont bien

ldapadd -x -v -D cn=Manager,dc=my-domain,dc=com -W < init.ldif

avec init.ldif :

dn: dc=my-domain,dc=com

dc: my-domain

o: my-domain.com

objectClass: top

objectClass: dcObject

objectClass: organization

dn: ou=people,dc=my-domain,dc=com

dc: my-domain

ou: people

objectClass: top

objectClass: dcObject

objectClass: organizationalUnit

dn: ou=group,dc=my-domain,dc=com

dc: my-domain

ou: group

objectClass: top

objectClass: dcObject

objectClass: organizationalUnit

Population de l'annuaire

Ajoutez un groupe d'utilisateurs avec ldapadd

ldapadd -x -v -D cn=Manager,dc=my- domain,dc=com -W < group.ldif

group.ldif :

dn: cn=totogroup,ou=group,dc=my- domain,dc=com

objectClass: top

objectClass: dcObject

objectClass: posixGroup

dc: my- domain

cn: totogroup

gidNumber:1001

Ajoutez dans votre annuaire un utilisateur non présents sur la machine cliente en utilisant ldapadd et un autre en utilisant gq.

ldapadd -x -v -D cn=Manager,dc=my- domain,dc=com -W < user.ldif

user.ldif

dn: uid=toto,ou=people,dc=my- domain,dc=com

loginShell: /bin/bash

objectClass: top

objectClass: dcObject

objectClass: person

objectClass: posixAccount

objectClass: shadowAccount

dc: my- domain

cn: toto

uid: toto

uidNumber: 1001

homeDirectory: /home/toto

sn: Gilbert Toto

gidNumber: 1001

loginShell: /bin/bash

userPassword: {MD5}9x2+UmKKP4OnerSUgXUlxg==

le mot de passe est fabriqué avec slappasswd

Vous devez instancier 3 objets pour refléter les 3 fichiers qui servent à l'authentification :

/etc/passwd => posixAccount

/etc/shadow => shadowAccount

/etc/group => posixGroup

Configuration nnsswitch

!!! ATTENTION !!!

Pendant vos tests, gardez toujours une console ouverte, sous root. Si vous faîtes une erreur (ça ne vous arrivera pas à vous, mais votre binôme peut la faire) vous ne pourrez peut- être plus vous connecter.

/etc/nsswitch

passwd: files ldap

shadow: files ldap

group: files ldap

Et testez un login

<CTRL><ALT><F2> pour ouvrir une console texte et se logger.

Si vous faîtes un su en étant root, le système ne vous demande pas le password et le test n'est pas complet.

Configuration pam

Sauvegardez le fichier /etc/pam.d/system- auth

Authconfig

Comparez l'ancien system- auth avec le nouveau, regardez /etc/ldap.conf et essayez d'expliquer comment ça marche. Utilisez la doc

http://www.kernel.org/pub/linux/libs/pam/Linux- PAM-html/pam.html

Ajout d'une ligne « sufficient /lib/security/$ISA/pam_ldap.so »

pour auth, password et session après « sufficient /lib/security/$ISA/pam_unix.so »

Testez (en capturant les trames bien sûr) et expliquez ce qui se passe.

Rien ne vous choque ?

Le bind est anonyme et on peut lire le mot de passe crypté. Ca veut dire que n'importe qui peut obtenir la liste des mots de passe cryptés et prendre son temps pour les craquer !

Il faut donc configurer les access lists, comme expliqué dans le cours :

access to attribute=userPassword # accès à l'attribut « userPassord », quel que soit l'objet

by dn="cn=Manager,dc=example,dc=com" write # l'administrateur peut écrire

```
        by self write #l'utilisateur concerné par l'objet peur écrire
        by anonymous auth #un utilisateur anonyme peut essayer s'authentifier
        by * none #et le reste du monde va se faire voir
# everything else is read- only
access to *
        by dn="cn=Manager,dc=example,dc=com" write
        by * read
```

A partir de ce moment, le serveur ne renvoie plus de mot de passe. Le client fini par se lasser et se « bind » avec son dn et son mot de passe, si le serveur accepte le bind, le client considère que le mot de passe est bon.

C'est déjà mieux, on ne peut plus lister les mots de passe. Il ne reste plus qu'à chiffrer la liaison. Ca marche comme https à la différence que comme on ne peut pas demander à l'utilisateur si il accepte le certificat, il faut absolument que celui-ci soit valide et corresponde bien au « basedn ».

Il faut donc le créer avec rigueur ou bien indiquer à pam dans /etc/ldap.conf de ne pas vérifier le certificat, ce qui est un peu léger. Ca chiffre la liaison mais ne garanti pas que c'est le bon serveur qui réponds.