Gestion d'identités PSL – Exploitation LDAP

Entr'ouvert SCOP – http ://www.entrouvert.com

Table des matières

1 Service `slapd`

Le serveur LDAP est `slapd` du projet OpenLDAP.

1.1 Arrêt et démarrage du service

`slapd` est démarré lors du boot de la machine et arrêté lors d'un *shutdown*. En dehors de ces moments, les commandes suivantes sont disponibles :

- `service slapd status` : état du service
- `service slapd stop` : arrêt du service

- `service slapd start` : démarrage du service
- `service slapd restart` : arrêt puis redémarrage du service

1.2 Logs

`slapd` envoie ses logs systèmes dans syslog, on y voit principalement le démarrage ou l'arrêt du service. Par défaut c'est enregistré dans `/var/log/syslog`.

Les logs des requêtes sont enregistrés dans la base LDAP, sous différents suffixes :

- `cn=config-accesslog` : pour les accès à la configuration
- `cn=accesslog,<suffixe>` : pour les bases créées ensuite, par exemple `cn=accesslog,dc=univ-fooba`

L'accès à ces logs se fait via un client LDAP, par exemple avec `ldapsearch` :

```
# ldapsearch -Y EXTERNAL -H ldapi :// -b "cn=config-accesslog" \
  "(&(reqStart>=20150212091000.000000Z)(reqEnd<=20150212091500.000000Z))"
```

renvoie la liste des requêtes sur la configuration entre 9h10 et 9h15 le 12 février 2015.

2 Commande slapd-supann

Le pilotage bas niveau des données et configurations de l'annuaire LDAP s'effectue via une commande spécifique développée dans le cadre de ce projet : `slapd-supann`

```
# slapd-supann help
syntaxe : slapd-supann commande ...

commandes disponibles :
help        cette aide
import      import d'un ou plusieurs fichiers LDIF
metasync    synchronise un annuaire distant dans le méta-annuaire local
newdb       création d'une nouvelle base, avec un nouveau suffixe
reset       mise à zéro complète
restore     restauration des données depuis un répertoire
save        sauvegarde de la configuration et des données
```

2.1 Mise à zéro (*reset*)

La commande `reset` met à zéro la configuration du LDAP (conformance norme SUPANN 2009) ainsi que toute les données :

```
# slapd-supann reset
```

Rappel : cette commande **efface toutes les données LDAP**, y compris les configurations.

2.2 Ajout d'une base (*newdb*)

La commande `newdb` permet d'ajouter une base dans le LDAP, typiquement avec le suffixe de l'établissement `dc=quelquechose,dc=fr`.

La commande est interactive, elle pose quelques questions puis créé une nouvelle base dans l'annuaire, avec un administrateur dédié dont il faudra saisir le mot de passe. Exemple de réponses aux questions :

```
# slapd-supann newdb
Suffixe de la base à créer (exemple : dc=dauphine,dc=fr) :
-> dc=quelquechose,dc=fr

Choisir un mot de passe administrateur (uid=admin,ou=people,dc=quelquechose,dc=fr) :
->
Une nouvelle fois :
->

Nom de l'organisation (o=...) :
uniquement des majuscules, sans accent
Exemple : ENS
-> QUELQUECHOSE

Code de l'établissement, préfixé par son origine (supannEtablissement={ORIG}CODE)
Exemples :
 {UAI}0350936C          Université de Rennes 1
 {SIRET}18004312700067  AMUE
 {CNRS}MOY1400          Délégation régionale de Toulouse du CNRS
-> {UAI}0610000X

Récapitulatif :
 Suffixe : dc=quelquechose,dc=fr
     Nom : QUELQUECHOSE
Code UAI : {UAI}0610000X
DN entité établissement : supannCodeEntite=QUELQUECHOSE,ou=structures,dc=quelquechose,

Créer cette base ? (taper oui)
-> oui

Chargement de la définition de la nouvelle base annuaire (/tmp/newdbsUdiiW.ldif) :
(add) olcDatabase={1}mdb,cn=config
```

```
(add) olcDatabase={1}mdb,cn=config
(add) olcOverlay={0}syncprov,olcDatabase={1}mdb,cn=config
(add) olcOverlay={1}accesslog,olcDatabase={2}mdb,cn=config
(add) olcOverlay={2}refint,olcDatabase={2}mdb,cn=config
(add) olcOverlay={3}constraint,olcDatabase={2}mdb,cn=config
(add) olcOverlay={4}unique,olcDatabase={2}mdb,cn=config
(add) dc=quelquechose,dc=fr
(add) ou=people,dc=quelquechose,dc=fr
(add) uid=admin,ou=people,dc=quelquechose,dc=fr
(add) ou=structures,dc=quelquechose,dc=fr
(add) supannCodeEntite=QUELQUECHOSE,ou=structures,dc=quelquechose,dc=fr
(add) ou=groups,dc=quelquechose,dc=fr
(add) cn=admin,ou=groups,dc=quelquechose,dc=fr
OK
#
```

Résultat :

- une base `dc=quelquechose,dc=fr` est ajoutée dans l'annuaire
- l'administrateur attribué à cette base est `uid=admin,ou=people,dc=quelquechose,dc=fr` avec le mot de passe choisi lors de la commande `newdb`.
- les logs des requêtes sur cette base sont dans `cn=accesslog,dc=quelquechose,dc=fr`

2.3 Import de données (*import*)

La commande `import` permet d'importer des fichiers LDIF dans l'annuaire. Syntaxe :

```
# slapd-supann import fichier1.ldif fichier2.ldif ...
```

- les fichiers LDIF doivent être correctement formatés et contenir des données à la norme SUPANN 2009
- les mots de passes en clair dans les fichier LDIF sont automatiquement chiffrés en utilisant l'algorithme SHA1 salé.
- `import` effectue une mise à jour des enregistrement déjà chargés

Exemple d'un fichier à importer :

```
# cat import.ldif             # exemple de fichier à importer
dn : uid=bdauvergne,ou=people,dc=quelquechose,dc=fr
objectClass : inetOrgPerson
objectClass : eduPerson
objectClass : supannPerson
givenName : Benjamin
sn : Dauvergne
cn : Dauvergne Benjamin
```

```
displayName : Benjamin Dauvergne
supannCivilite : M.
supannEtablissement : {UAI}0610000X
supannListeRouge : FALSE
preferredLanguage : fr
supannMailPerso : bdauvergne@entrouvert.com
eduPersonNickname : bdauvergne
uid : bdauvergne
supannAliasLogin : bdauvergne
eduPersonPrincipalName : bdauvergne@quelquechose.fr
mail : bdauvergne@quelquechose.fr
```

Lancement de l'import :

```
# slapd-supann import import.ldif
- added entry uid=bdauvergne,ou=people,dc=quelquechose,dc=fr
  - supanncivilite : M.
  - displayname : Benjamin Dauvergne
  - cn : Dauvergne Benjamin
  - objectclass : inetOrgPerson, eduPerson, supannPerson
  - edupersonnickname : bdauvergne
  - supannmailperso : bdauvergne@entrouvert.com
  - preferredlanguage : fr
  - edupersonprincipalname : bdauvergne@quelquechose.fr
  - sn : Dauvergne
  - supannetablissement : {UAI}0610000X
  - mail : bdauvergne@quelquechose.fr
  - givenname : Benjamin
  - supannlisterouge : FALSE
  - supannaliaslogin : bdauvergne
  - uid : bdauvergne
```

Résultat visible dans la base, par exemple en utilisant `ldapsearch` :

```
# ldapsearch -Y EXTERNAL -H ldapi :// -b "dc=quelquechose,dc=fr" "uid=bdauvergne"
```

Note : pour voir tous les logs concernant l'entrée `uid=bdauvergne,ou=people,dc=quelquechose,dc=fr` on regarderait dans `cn=accesslog,dc=quelquechose,dc=fr` :

```
# ldapsearch -Y EXTERNAL -H ldapi :// -b "cn=accesslog,dc=quelquechose,dc=fr" \
  "reqDN=uid=bdauvergne,ou=people,dc=quelquechose,dc=fr"
```

2.4 Sauvegarde (*save*)

La commande *save* créée un répertoire de sauvegarde dans lequel toute la **configuration** et toutes les **données** de toutes les bases de l'annuaire sont enregistrées.

Ce répertoire est destiné à la commande `restore`, pour restaurer l'annuaire dans l'état exact du `save`.

Par défaut, le répertoire est créé dans `/var/backups/`.

Exemple d'exécution :

```
# slapd-supann save
Sauvegarde de la configuration et des données slapd
dans le répertoire /var/backups/slapd-save-20150219T122440
  Export de la configuration dans /var/backups/slapd-save-20150219T122440/config.ldif
ok
  Export de le base 1 ..
ok
  Export de le base 2 ..
ok
(...)
Sauvegarde des certificats SSL ..ok
Efface les fichiers vides ..
ok

/var/backups/slapd-save-20150219T122440 contient :
total 132
-rw-r--r-- 1 root root 57122 févr. 19 12 :24 config.ldif
-rw-r--r-- 1 root root 40349 févr. 19 12 :24 db-1.ldif
-rw-r--r-- 1 root root  4480 févr. 19 12 :24 db-2.ldif
-rw-r--r-- 1 root root   431 févr. 19 12 :24 db-4.ldif
-rw-r--r-- 1 root root 14937 févr. 19 12 :24 db-5.ldif
-rw-r----- 1 root root  1704 févr. 19 12 :24 slapd.key
-rw-r--r-- 1 root root  1038 févr. 19 12 :24 slapd.pem
```

Note : la commande `save` essaye de sauvegarder toutes les bases présentes, mais certaines peuvent être vides voire inexistantes. Il peut donc y avoir d'**éventuels affichages de messages d'erreurs** lors de l'exécution de la commande, sans autre conséquence.

Attention : le répertoire de sauvegarde contient la clé privée `slapd.key`, stockée en clair. À ne pas mettre entre toutes les mains...

2.5 Restauration (*restore*)

La commande `restore` permet de remettre le serveur LDAP dans l'état exact de la sauvegarde. Il faut fournir en argument de la commande un répertoire crée par la commande `save`.

Attention : la commande `restore` **efface complètement** et définitivement la configuration et toutes les données actuelle du serveur.

Exemple d'exécution :

```
# slapd-supann restore /var/backups/slapd-save-20150219T122440/

  *************
  *           *   La configuration et toutes les données
  * ATTENTION *   de l'annuaire LDAP vont être définitivement
  *           *   effacées. Avez-vous fait un backup ?
  *************

Confirmez la MISE A ZÉRO COMPLÈTE avant restauration.

Tapez oui en toutes lettres : oui
[ ok ] Stopping OpenLDAP : slapd.
Effacement des données actuelles ..ok

Restauration du config.ldif ..
_#################### 100.00% eta   none elapsed            none fast !
Closing DB...

Restauration des certificats SSL ..
ok

Restauration de la base 1 ..
-#################### 100.00% eta   none elapsed                 spd  39.4 k/s
Closing DB...
Restauration de la base 2 ..
_#################### 100.00% eta   none elapsed            none fast !
Closing DB...
Restauration de la base 4 ..
_#################### 100.00% eta   none elapsed            none fast !
Closing DB...
Restauration de la base 5 ..
_#################### 100.00% eta   none elapsed            none fast !
Closing DB...
```

```
[ ok ] Starting OpenLDAP : slapd.
#
```

2.6 Mise à jour du paramétrage d'une base (*resetdb*)

La commande `resetdb` permet de remettre tout ou parti de la configuration d'une base à son paramétrage initiale en matière d'ACLs, de configuration des indexs ou de contrainte sur les formats des attributs.

Attention : la commande `restore` **efface complètement** et définitivement la certaines configurations comme les ACLs, le consentement pour certaines de ces mises à jour étant demandé dans le doute il vaut mieux refuser, notamment si vous avec développé de nouvelles règles d'accès locales.

Exemple d'exécution :

```
# slapd-supann resetdb dc=college-de-france,dc=fr

La réinitialisation des ACLs supprimera vos ACLs locales les remplaçant par le
standard PSL, à ne faire qu'en connaissance de cause.
Voulez-vous remettre à zéro les ACLs ? (y/n) n

La réinitialisation des directives olcDBIndex supprimera vos règles
d'indexation locales, si vous utilisez des attributs locaux qui nécessitent une
indexation cela pourrait produire des ralentissements.
Voulez-vous réinitialiser les directives olcDbIndex ? (y/n) n

- contrainte supannAliasLogin retiré
- contrainte pslBadgeCSN ajouté

Réinitialisation de la base olcDatabase={2}mdb,cn=config pour le suffixe dc=college-de
```

2.7 Méta-annuaire `o=meta` (*metasync*)

L'alimentation du méta-annuaire *o=meta* s'effecture au travers de la commande `metasync` qui permet de synchroniser un annuaire distant dans une branche `dc=<distant>,o=meta`.

L'annuaire distant doit être **strictement SUPANN 2009**, c'est-à-dire :

- la racine utilise les classes *organization*, *dcObject*, *eduOrg* et *supannOrg*
- les sous-branches organisationelles `ou=people`, `ou=structures` et `ou=groups` utilisent la classe *organizationalUnit*
- les groupes utilisent les classes *groupOfNames* et *supannGroupe*

- les utilisateurs utilisent les classes *inetOrgPerson*, *eduPerson* et *supannPerson*
- les entités utilisent les classes *supannEntite* et *organizationUnit* pour les sous-entités ou *supannEntite*, *organization*, *eduOrg*, *supannOrg* pour les entités racines
- **aucune autre classe n'est utilisée**

Cette dernière contrainte est importante, par exemple `metasync` ne pourra pas opérer sur un Active Direcory "supannisé" de façon partielle laissant apparaitre une classe *user*.

Syntaxe de la commande :

```
slapd-supann metasync [--quiet] [--fake] ldap_uri ldap_newbasedn
                      ldap_basedn [ldap_binddn] [ldap_bindpwd]
```

- `ldap_uri` est l'URI du LDAP distant, par exemple `ldap ://ldap.univ-test.fr/`
- `ldap_newbasedn` est l'emplacement local où l'annuaire sera synchronisé. Le suffixe doit absolument être **`o=meta`**, par exemple `dc=univ-test,o=meta`. Attention, le suffixe ne doit contenir **qu'un seul niveau par rapport au dn de base du méta-annaire** `o=meta` (`dc=univ-test,dc=fr,o=meta` ne marchera pas)
- `ldap_basedn` le base DN de l'annuaire distant, ex. : `dc=univ-test,dc=fr`
- `ldap_binddn` et `ldap_bindpwd` : identifiants et mots de passe pour accéder à l'annuaire distant (optionnels)
- option `--quiet` : limite l'affichage aux seules erreurs
- option `--fake` : calcule et affiche les actions nécessaires à la synchronisation mais ne les effectue pas

Exemple d'exécution :

```
# slapd-supann metasync ldap ://ldap.univ-test.fr dc=univ-test,o=meta \
      dc=univ-test,dc=fr uid=admin,ou=people,dc=univ-test,dc=fr motdepasse
Synchronizing LDAP directory at uid=admin,ou=people,dc=univ-test,dc=fr locally.
BaseDN : dc=univ-test,dc=fr
BindDN : uid=admin,ou=people,dc=univ-test,dc=fr
BindPWD : admin
Actions :
- Create dc=univ-test,o=meta
- Create ou=groups,dc=univ-test,o=meta
- Create ou=people,dc=univ-test,o=meta
- Create ou=structures,dc=univ-test,o=meta
- Create cn=admin,ou=groups,dc=univ-test,o=meta
- Create uid=admin,ou=people,dc=univ-test,o=meta
- Create uid=bdauvergne,ou=people,dc=univ-test,o=meta
- Create supannCodeEntite=test1,ou=structures,dc=univ-test,o=meta
- Create supannCodeEntite=test2,ou=structures,dc=univ-test,o=meta
- Create supannCodeEntite=racine,ou=structures,dc=univ-test,o=meta
- Create supannCodeEntite=test3,ou=structures,dc=univ-test,o=meta
- Create supannCodeEntite=test4,ou=structures,dc=univ-test,o=meta
```

```
Waiting for completion..  done
```

Note technique : l'outil de synchronisation utilise l'extension LDAP `PagedResult` permettant de parcourir sans limitation la plupart des serveurs LDAP. Pour OpenLDAP la limitation par défaut s'applique même avec cette extension. Pour permettra la synchronisaton on ajoutera la ligne de configuration suivante dans la section de la base concernée sur le serveur distant :

```
limits * size.prtotal=unlimited
```

Rappel : attention à l'horloge des machines, à la fois celle de la présente solution mais aussi celle de l'annuaire distant. Toutes les machines en jeu doivent être en permanence **parfaitement à l'heure** sous peine de grave dysfonctionnement.

La lecture du méta-annuaire se fait via des comptes de lecture dans la branche ou=readers,o=meta. Un premier compte est initialisé nommé uid=reader,ou=readers,o=meta ayant le mot de passe « reader ». Pour changer ce mot de passe on pourra faire (en tant que root) :

```
ldappasswd -Y EXTERNAL -H ldapi :// -s "new-password" uid=reader,ou=readers,o=meta
```

Pour créer d'autres utiliateurs on utilisera ldapvi, en s'inspirant des utilisateurs existant :

```
ldapvi -b ou=readers,o=meta
```

3 Mise à jour

La mise à jour du système doit être effectuée aussi fréquement que possible, typiquement une fois par jour (mises à jour de sécurité Debian). Entr'ouvert informera aussi le projet en cas de mise à jour urgente de sécurité à effectuer sur les composants mis en jeu par la solution.

La procédure de mise à jour est la suivante, en **deux étapes**.

Mise à jour de la liste des logiciels disponibles sur les dépôts de la solution (Debian et Entr'ouvert) :

```
# apt-get update
```

Mise à jour des paquets qui ont une version plus récente que celle installée :

```
# apt-get upgrade
```

Il est possible que des versions futures de la solution nécessitent l'installation de nouveaux paquets, dans ce cas Entr'ouvert mettra à jour les dépendances de ses paquets et il faudra utiliser la commande `apt-get dist-upgrade`.

4 Ajout du schéma pslPerson

Le schéma pslPerson a été développé pour l'établissement PSL en vue de stocker certains attributs spécifiques à leur population, dans un premier temps l'attribut pslBadgeCSN. Pour ajouter ce schéma à une base existante il faut suivre le déroulé suivant (en supposant le préfix de la base comme étant dc=etablissement,dc=fr). Cette procédure n'est appliquable que sur un annuaire OpenLDAP mis en place via les outils du projet LDAP SUPANN de PSL.

Tout d'abord charger le nouveau schéma dans la configuration d'OpenLDAP :

```
# slapd-supann load-psl-schema
```

Ajouter à la base concernée la contrainte sur le format du nouvel attribut pslBadgeCSN :

```
# slapd-supann resetdb dc=etablissement,dc=fr
```

La commande resetacl a été renommée en resetdb, elle n'écrase plus systématiquement les règles d'ACL mais requière une confirmation désormais, ici on s'en servira simplement pour l'ajout de la contrainte.

Ajouter la classe pslPerson à tous les objets supannPerson de l'annuaire (si un traitement plus compliqué devait être entrepris il faudrait le faire autrement) :

```
# slapd-supann add-psl-person ou=people,dc=etablissement,dc=fr
```

---

5 Historique du document

20150217 tnoel – première version

20150624 bdauvergne – ajout documentation accès en lecture au méta-annuaire

20180503 bdauvergne – ajout documentation sur resetdb (anciennement resetacl)

20180503 bdauvergne – ajout documentation sur ajout du schéma pslPerson