Compte-rendu Expertise LSC - Mars 2019

Académie de Dijon

Version	1.0
Date	11/03/2019
Auteur	Soisik Froger
Référence	CR-201901210005-acdijon-lsc
Statut	Version finale

Table des matières

1 Contexte....3
1.1 Contexte....3
1.2 Besoins....3
2 Travaux réalisés....5
2.1 Ordonnancement des taches....5
2.2 Calcul du DN de destination....5
2.3 Déplacement des entrées dans la destination....5
2.4 Recommandations diverses....7
2.5 Authentification Kerberos....8
2.6 Délégation d'authentification OpenLDAP > AD via SASL....8

1 Contexte

1.1 Contexte

L'académie de Dijon utilise le logiciel LSC pour synchroniser des annuaires OpenLDAP et Samba4.

Deux synchronisations en mode asynchrone sont mises en place au sein de la solution « Eole » dont l'académie est éditrice nationale :

- Synchronisation mono-établissement ; les données sont importées à partir de fichier fournis par le ministère dans un Openldap local (paquet Ubuntu v2.4.42) et synchronisés vers un AD Samba4 local ;
- Synchronisation multi-établissement ; les données sont importées à partir de fichier fournis par le ministère dans un Openldap (paquet Ubuntu v2.4.42) et synchronisés vers un AD Samba4, tous deux mutualisés entre plusieurs établissements ;

L'académie de Dijon a mandaté la société Worteks pour une mission d'expertise de deux jours les 7 et 8 mars 2019 afin l'aider à résoudre des problématiques autour de ces synchronisations.

1.2 Besoins

Une évolution de la configuration de ces connecteurs est souhaitée pour :

- Gérer la problématique du multi-établissements en calculant la branche de destination de création d'un compte ou d'un groupe ;
- Rendre la synchronisation compatible avec le déplacement des entrées dans l'annuaire cible :
  - une tâche doit être dédiée à la création des entrées (avec calcul du DN) ;
  - une tâche dédiée aux modifications (qui ne modifiera pas le DN) ;
  - prendre en compte les potentiels changements de DN des utilisateurs dans la destination lors de la synchronisation des membres de groupes ;

Lors de la mission, d'autres besoins de conseil ont émergé pour les points suivants :

- Utiliser l'authentification Kerberos pour la connexion LDAP à l'annuaire Samba4 lors de la synchronisation LSC ;
- Explorer des solutions permettant aux utilisateurs de changer leur mot de passe directement dans Samba4 via leur poste de travail ; Les utilisateurs utilisent

actuellement une application qui met à jour le mot de passe dans les deux annuaires (Samba4 et OpenLDAP), la mise à jour des mots de passe sur le poste de travail est interdite.

- Question sur les durées de la synchronisation initiale en cas de fort volume de données : des temps de synchronisation très long ont été constatés avec sambatool (30 heures pour 100/200 000 comptes).

Par expérience, nous estimons que la synchronisation LSC devrait être plus rapide (une à deux heures en synchronisation initiale).

> Il faut veiller à positionner l'option « -i » de l'executable pour augmenter le timeout par défaut d'une synchronisation LSC (1 heure).

2 Travaux réalisés

2.1 Ordonnancement des taches

Quand une synchronisation LSC est lancée sans préciser une tache précise (mode *all*), les taches sont exécutées dans l'ordre alphabétique. La synchronisation des groupes nommée *Group* sera exécutée avant la tache de synchronisation des utilisateurs nommée *User*. Lors de la première synchronisation suivant l'ajout d'un nouvel utilisateur dans la source, les synchronisations des groupes référençant le nouvel utilisateur échouent, car l'utilisateur n'existe pas encore dans la destination. Un délai de 24 heures entre l'ajout de l'utilisateur et son obtention de droits d'accès est donc souvent observé.

Solution : Renommer la tache user en *t1_User* et la tache groupe en *t2_Group*, afin d'exécuter la tache *User* avant la tache *Group*, et prévenir ainsi l'absence d'un utilisateur lors de la synchronisation de l'attribut *member* des groupes.

2.2 Calcul du DN de destination

Le calcul dynamique du DN d'un utilisateur et d'un groupe est réalisé à partir du DN de la source via un script javascript dans la balise mainIdentifier. Exemple :

```
<propertiesBasedSyncOptions>
    <mainIdentifier>
    <![CDATA[
    var destOU = srcBean.getMainIdentifier().split("ou=")[4];
    var mainIdentifier = "cn=" +
srcBean.getDatasetFirstValueById("cn") + ",ou="+destOU+",dc=my-
domain,dc=com";
    mainIdentifier
    ]]>
    </mainIdentifier>
    […]
</propertiesBasedSyncOptions>
```

Pour voir les méthodes disponibles pour l'objet *srcBean* et *dstBean*, voir https://lsc-project.org/javadoc/2.1-SNAPSHOT/org/lsc/beans/LscBean.html.

La déduction dynamique de la branche de destination d'une entrée permet à l'AC Dijon de ne pas créer des taches de synchronisation spécifiques à chaque établissement lors de la génération du fichier via un template.

2.3 Déplacement des entrées dans la destination

L'objectif est de permettre le déplacement des utilisateurs et des groupes dans la destination après leur création.

Les ajustements suivants ont été appliqués aux scripts mono et multi établissements :

- Découplage de la tache de synchronisation des utilisateurs et des groupes en deux taches : création et modification;
- En création, calcul dynamique du DN de destination (voir plus haut). Seule la création est autorisée :

```
<propertiesBasedSyncOptions>
    <conditions>
         <create>true</create>
         <update>false</update>
         <delete>false</delete>
         <changeId>false</changeId>
    </conditions>
    […]
</propertiesBasedSyncOptions>
```

- En modification, le mainIdentifier est laissé vide et seule la modification (et la suppression pour le clean) sont autorisés :

```
<propertiesBasedSyncOptions>
    <mainIdentifier>""</mainIdentifier>
    […]
    <conditions>
         <create>false</create>
         <update>true</update>
         <delete>true</delete>
         <changeId>false</changeId>
    </conditions>
    […]
</propertiesBasedSyncOptions>
```

- La synchronisation de l'attribut "member" des groupes doit également prendre en compte la possibilité d'un déplacement d'un membre dans le DIT de destination. Pour cela, on effectue une recherche du membre dans la destination (*ldap*) et on utilise le DN renvoyé par la recherche pour reconstruire la liste des membres. Ex :

```
<propertiesBasedSyncOptions>
    […]
    <dataset>
         <name>member</name>
         <policy>FORCE</policy>
         <forceValues>
              <string>
              <![CDATA[
    var membersUid = srcBean.getDatasetValuesById("memberUid");
    var retMembersDn = [];
    for (var i=0; i<membersUid.size(); i++) {
```

```
          var uid = membersUid.get(i);
          var newDn = ldap.search("","(sAMAccountName="+uid+")");
          if (newDn.length > 0) {
               retMembersDn.push(newDn.get(0) + getContextDn());
          }
     }
     retMembersDn
               ]]>
               </string>
          </forceValues>
     </dataset>
     […]
</propertiesBasedSyncOptions>
```

La méthode search est documentée ici :
https://lsc-project.org/javadoc/2.1-SNAPSHOT/org/lsc/jndi/ScriptableJndiServices.html#search-java.lang.Object-java.lang.Object-

> Tip: Cette méthode ajoute le *ContextDn* de la destination à la base de la recherche. Elle retourne également des entrées amputées du *ContextDn*.

2.4 Recommandations diverses

Les recommandations suivantes sont proposées pour améliorer la synchronisation :

- Annoter (via un attribut particulier) les entrées créées par la synchronisation afin d'exclure tout risque de suppression d'objets systèmes dans la destination lors des phases de *clean,* en positionnant cette attribut dans le *getAllFilter* du *destinationService*. Ex :

```
<task>
     <ldapDestinationService>
          […]
          <getAllFilter>(&amp;(objectClass=User)
(extensionAttribute1=fromLSC))</getAllFilter>
          […]
     </ldapDestinationService>
     <propertiesBasedSyncOptions>
          […]
          <dataset>
               <name>extensionAttribute1</name>
               <policy>force</policy>
               <forceValues>
                    <string>"fromLSC"</string>
               </forceValues>
          </dataset>
          […]
     </propertiesBasedSyncOptions>
```

```
</task>
```

- Utiliser les constantes fournie par LSC pour le calcul de l'attribut *userAccountControl* des utilisateurs plûtot qu'une valeur en dure, et ne pas forcer la valeur en modification, pour ne pas écraser une action réalisée coté AD (ex: réactiver le compte après désactivation). Par exemple :

```
<dataset>
    <name>userAccountControl</name>
    <policy>KEEP</policy>
    <createValues>
        <string>AD.userAccountControlSet( "0",
[AD.UAC_SET_PASSWD_NOTREQD, AD.UAC_SET_NORMAL_ACCOUNT])</string>
    </createValues>
</dataset>
```

  - https://lsc-project.org/javadoc/2.1-SNAPSHOT/org/lsc/utils/directory/AD.html
  - https://lsc-project.org/documentation/2.1/configuration/syncoptions/activedirectory

- Connection à l'annuaire destination: Utiliser un compte de service dédié à LSC au lieu du compte Admin.

2.5 Authentification Kerberos

Objectif: authentifier le compte de service permettant de se connecter à l'annuaire de destination via Kerberos/GSSAPI afin de ne pas avoir le mot de passe en clair dans le fichier de configuration.

La procédure LSC est https://lsc-project.org/documentation/howto/kerberos.

Les tests réalisés n'ont pas permi de rendre cette authentification effective. Après investigation de l'AC Dijon, l'annuaire Samba n'est pas encore configuré pour autoriser une authentification SASL GSSAPI (option *SupportedSaslMechanisms*).

2.6 Délégation d'authentification OpenLDAP > AD via SASL

Cette possibilité est envisagée par AC Dijon pour permettre la mise à jour des mots de passe directement depuis le poste utilisateur et éviter de synchroniser les mots de passe et d'avoir à mettre en place des restrictions et une procédure spécifique pour leur mise à jour. Actuellement, une application spécifique permet aux utilisateurs de modifier leurs mots de passe à la fois dans OpenLDAP et dans Samba.

La mise en place d'une délégation SASL ayant des impacts importants en terme d'architecture, l'AC Dijon va étudier l'opportunité de la mettre en place.

Voir :

- http://www.openldap.org/doc/admin24/sasl.html
- https://ltb-project.org/documentation/general/sasl_delegation

Pour fonctionner, l'annuaire OpenLDAP doit avoir été compilé avec des options spécifiques autorisant le SASL *--enable-spasswd* et *--with-cyrus-sasl*. L'annuaire OpenLDAP utilisé par AC Dijon est la version distribuée par Ubuntu 16.04 (2.4.42). Après vérification des sources, l'annuaire n'est pas compilé avec ces options en 16.04, mais cela reste à vérifier pour la 18.04.

Deux solutions sont possibles :

- recompiler les sources OpenLDAP depuis le dépôt Ubuntu avec les bonnes options ;
- utiliser les paquets OpenLDAP LTB (ceux que nous utilisons dans nos installations) : https://ltb-project.org.

FIN DU DOCUMENT