

Scribe - Tâche #34603

Scénario # 34409 (Partiellement Réalisé): EoleAD : création de comptes préfixés (POC)

POC vérification de la disponibilité du login

15/09/2022 17:10 - Joël Cuissinat

Statut:	Fermé	Début:	15/09/2022
Priorité:	Normal	Echéance:	
Assigné à:	Joël Cuissinat	% réalisé:	0%
Version cible:	Livraison MENSUR 31/03/2023	Temps estimé:	0.00 heure
		Temps passé:	0.00 heure
Description			

Historique

#1 - 15/09/2022 17:19 - Joël Cuissinat

En conservant le code actuel :

- si le compte est retrouvé dans l'OpenLDAP local, il s'agit d'une mise à jour d'un compte existant
- si le compte n'est pas retrouvé, on doit vérifier la disponibilité du login proposé "en central" et "boucler" jusqu'à en trouver un de disponible
- contrairement à ce qui est fait actuellement (synchronisation LSC délayée), il faudrait réserver le login en question au niveau central immédiatement, la "fiche utilisateur" pouvant être mise à jour ultérieurement

#2 - 19/09/2022 09:47 - Joël Cuissinat

- Statut changé de Nouveau à En cours

#3 - 19/09/2022 09:47 - Joël Cuissinat

- Assigné à mis à Joël Cuissinat

#4 - 19/09/2022 09:59 - Joël Cuissinat

Toutes les demandes sont centralisées dans la fonction `_gen_new_login` qui :

- génère le login souhaité (base)
- vérifie sa disponibilité dans l'OpenLDAP local et les comptes système (fonction `_is_available_name` de `scribe/eoleldap.py`)
- en cas d'indisponibilité, on ajoute un chiffre et on l'incrmente jusqu'à trouver une disponibilité

C'est dans cette fonction qu'on pourrait ajouter une vérification sur l'AD "centralisé", une requête sur l'existence ou non du **CN** suffirait.

FIXME : Actuellement les responsables ne sont pas synchronisés vers AD, peut-on accepter des doublons "locaux" pour ces comptes ? Sachant que c'est déjà le cas actuellement avec l'annuaire consolidé Seshat.

#5 - 19/09/2022 10:22 - Joël Cuissinat

Exemple de code python3 vérifiant la disponibilité d'un CN dans un AD :

```
import sys
import ldap
login = sys.argv[1]
AD_ADDRESS = '192.0.2.2'
```

```

AD_BASE = 'DC=domscribe,DC=ac-test,DC=fr'
AD_USERDN = 'CN=eole-synchro,CN=Users,DC=domscribe,DC=ac-test,DC=fr'
AD_PWD = 'RilAjpg3ZhIH9bueyCu701J13w5ZpMINTSuFgYfPYxK'
ldap.set_option(ldap.OPT_REFERRALS, 0)
ldap.set_option(ldap.OPT_X_TLS_REQUIRE_CERT, ldap.OPT_X_TLS_NEVER)
connexion = ldap.initialize('ldaps://{}'.format(AD_ADDRESS))
connexion.simple_bind_s(AD_USERDN, AD_PWD)
ldb_filter = 'cn={}'
res = connexion.search(AD_BASE, ldap.SCOPE_SUBTREE, ldb_filter.format(login), ['cn'])
if connexion.result(res)[1][0][0] is None:
    print('login {} disponible'.format(login))
else:
    print('login {} INdisponible'.format(login))
connexion.unbind_s()

```

#6 - 19/09/2022 10:25 - Joël Cuissinat

Mais puisqu'on veut une réservation, on peut directement essayer de créer un compte utilisateur minimaliste (dans la bonne OU). Je viens de tester le code suivant, LSC arrive bien à reprendre la main derrière :

```

import sys
import ldap
login = sys.argv[1]
AD_ADDRESS = '192.0.2.2'
AD_BASE = 'DC=domscribe,DC=ac-test,DC=fr'
AD_USERDN = 'CN=eole-synchro,CN=Users,DC=domscribe,DC=ac-test,DC=fr'
AD_PWD = 'RilAjpg3ZhIH9bueyCu701J13w5ZpMINTSuFgYfPYxK'
user_dn = 'CN={},CN=Users,{}'.format(login, AD_BASE)
user_name = login.encode()
datas = [(('objectClass', b'user')),
         (('cn', user_name)),
         (('info', b'EOLE')),
         (('sAMAccountName', user_name)),
        ]
ldap.set_option(ldap.OPT_REFERRALS, 0)
ldap.set_option(ldap.OPT_X_TLS_REQUIRE_CERT, ldap.OPT_X_TLS_NEVER)
connexion = ldap.initialize('ldaps://{}'.format(AD_ADDRESS))
connexion.simple_bind_s(AD_USERDN, AD_PWD)
try:
    connexion.add_s(user_dn, datas)
    print('login {} réservé'.format(login))
except ldap.ALREADY_EXISTS:
    print('login {} INdisponible'.format(login))
connexion.unbind_s()

```

On pourrait presque envisager de généraliser ce fonctionnement car cela accélère le temps de création d'un utilisateur unique (pas à attendre que LSC se décide à créer l'utilisateur dans AD)

#7 - 28/10/2022 10:31 - Joël Cuissinat

- Statut changé de *En cours* à *Résolu*

#8 - 29/03/2023 10:41 - Joël Cuissinat

- Statut changé de *Résolu* à *Fermé*

- Restant à faire (heures) mis à 0.0