Publik - Project management #20325

raccordement LDAP: clés reliant un compte LDAP à un compte créé dans authentic

28 November 2017 12:24 PM - Serghei Mihai

Status:	Nouveau	**Start date:**	28 November 2017
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0:00 hour
Target version:			
Patch proposed:	No	**Club:**	
Planning:			

Description

Ticket suite à une discussion privée avec Benjamin sur la situation suivante qui devrait donner suite à une page de doc mais je préfère expliquer et valider les choses ici d'abord.

Lors de la synchro avec l'annuaire le champ username du compte dans A2 est construit par la variable username_template dans la config LDAP.
Généralement dédans on mets un champ unique, genre uid issu de l'annuaire.

Dans certaines situations ce champ peut varier (malgré son nom) et donc il peut arriver qu'un doublon du compte LDAP soit créé dans authentic avec un username différent.

Le lien entre le compte authentic et le compte LDAP est fait par le modèle ExternalUserId qui utilise le paramètre external_id_tuples (avec la valeur par défaut 'uid').
Pour prendre en compte un nouveau attribut LDAP "unique" il est nécessaire de rajouter cet attribut dans external_id_tuples, par exemple:

```
{
...
"external_id_tuples": [["employeeNumber"], ["uid"]],
...
}
```

Cela evitera la création d'un compte doublon dans authentic.

History

#1 - 28 November 2017 12:40 PM - Benjamin Dauvergne

Alors les clés de configuration concernées sont :

```
266         # update username on all login, use with CAUTION !! only if you know that
267         # generated username are unique
268         'update_username': False,

269         # lookup existing user with an external id build with attributes
270         'lookups': ('external_id', 'username'),
271         'external_id_tuples': (('uid',), ('dn:noquote',),),
272         # keep password around so that Django authentification also work
273         'clean_external_id_on_update': True,
```

Il faut mettre update_username à True aussi, sinon le username restera constant après création de l'utilisateur, mais par contre par défaut on va nettoyer les anciens ExternalId à chaque nouvelle connexion et ne garder que le premier défini:

```
802         # if external_id lookup is used, update it
803         if 'external_id' in block['lookups'] \
804            and block.get('external_id_tuples') \
805            and block['external_id_tuples'][0]:
806             if not user.pk:
807                 user.save()
808                 user._changed = False
809             external_id = self.build_external_id(
810                 block['external_id_tuples'][0],
```

```
811                 attributes)
812             if external_id:
813                 new, created = UserExternalId.objects.get_or_create(
814                     user=user, external_id=external_id, source=block['realm'])
815                 if block['clean_external_id_on_update']:
816                     UserExternalId.objects \
817                         .exclude(id=new.id) \
818                         .filter(user=user, source=block['realm']) \
819                         .delete()
```

#2 - 28 November 2017 12:42 PM - Benjamin Dauvergne

Bon bien sûr tout cela se teste, on met pas en prod directement comme un bourrin, merci.