Authentic 2 - Development #30125

backend SAML LDAP pour authentification auprès d'un IDP utilisant un annuaire accessible à Authentic

28 janvier 2019 10:30 - Serghei Mihai

Statut:	Rejeté	**Début:**	28 janvier 2019
Priorité:	Normal	**Echéance:**	
Assigné à:	Benjamin Dauvergne	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Un LemonLDAP utilisé par la collectivité, branché au même annuaire que Authentic, peut servir d'IDP SAML à Authentic.

Demandes liées:

Lié à django-mellon - Development #30541: django1.11: authenticate() got an u...	**Fermé**	**12 février 2019**

Historique

#1 - 28 janvier 2019 18:23 - Serghei Mihai

- Fichier 0001-backends-add-SAML-LDAP-authentication-backend-30125.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

L'idée est de spécifier dans la config LDAP l'entity_id de l'IDP qui repose sur le même annuaire.

L'IDP doit retourner les attributs sur lesquels se base authentic pour retrouver l'usager dans l'annuaire.
Il manque encore des tests: wip.

#2 - 29 janvier 2019 12:52 - Benjamin Dauvergne

- Sujet changé de backend SAML LDAP pour authentification auprès d'un IDP se reponsant sur le même annuaire que Authentic à backend SAML LDAP pour authentification auprès d'un IDP utilisant un annuaire accessible à Authentic

#3 - 30 janvier 2019 17:25 - Serghei Mihai

- Fichier 0001-backends-add-SAML-LDAP-authentication-backend-30125.patch ajouté

Patch à jour, avec tests.

#4 - 06 février 2019 10:50 - Benjamin Dauvergne

- Virer LDAP_ENABLE ne sert pas, si pas d'entity_id pas d'authent LDAP
- Renommer entity_id en saml_entity_id
- Le ENABLE_CONDITION ne sert à rien
- Virer tout src/authentic2_auth_saml/app_ldap_saml_settings.py
- Je ne comprends pas pourquoi dans SAMLLdapBackend n'utilise pas get_config()
- Renommer get_block_users() en get_users_with_config().
- Ne pas utiliser sync_ldap_users_filter, en fait il faut factoriser la recherche d'un utilisateur avec le code classique
- Ne pas rechercher le nom de l'attribut SAML via une regexp sur le filtrer, à la place avoir un attribut explicite saml_username_attribute et passer ce username au code normal de recherche d'un utilisateur (il faut minimiser le code qui diffère), en gros le authenticate, il devrait faire :

```
username = self.get_username_from_saml_attributes(saml_attributes)
user = super(SAMLLdapBackend, self).authenticate(username=username, without_password=True)
```

et ajouter la gestion d'un mode "sans mot de passe" ou bien éclater authenticate en deux morceaux réutilisables, la recherche des enregistrements LDAP et la validation des credentials.

#5 - 06 février 2019 10:55 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

- Virer LDAP_ENABLE ne sert pas, si pas d'entity_id pas d'authent LDAP
- Renommer entity_id en saml_entity_id
- Le ENABLE_CONDITION ne sert à rien
- Virer tout src/authentic2_auth_saml/app_ldap_saml_settings.py
- Je ne comprends pas pourquoi dans SAMLLdapBackend n'utilise pas get_config()
- Renommer get_block_users() en get_users_with_config().
- Ne pas utiliser sync_ldap_users_filter, en fait il faut factoriser la recherche d'un utilisateur avec le code classique
- Ne pas rechercher le nom de l'attribut SAML via une regexp sur le filtrer, à la place avoir un attribut explicite saml_username_attribute et passer ce username au code normal de recherche d'un utilisateur (il faut minimiser le code qui diffère), en gros le authenticate, il devrait faire :
  [...]

> et ajouter la gestion d'un mode "sans mot de passe" ou bien éclater authenticate en deux morceaux réutilisables, la recherche des enregistrements LDAP et la validation des credentials.

Et donc ce que je veux dire c'est dans le authenticate du backend de base faire un truc comme :

```
def authenticate(username=None, password=None):
    for dn in self.get_dn_from_username(username):
        if self.try_bind(dn, password):
           return self.get_user_from_dn(dn, password, conn, block)
```

#6 - 06 février 2019 15:55 - Serghei Mihai

Benjamin Dauvergne a écrit :

> et ajouter la gestion d'un mode "sans mot de passe" ou bien éclater authenticate en deux morceaux réutilisables, la recherche des enregistrements LDAP et la validation des credentials.

authenticate doit dans ce cas faire appel à un authenticate_with_block pour que la recherche/validation de l'usager se fasse uniquement dans l'annuaire qui correspond au fournisseur SAML, ok.

#7 - 06 février 2019 15:56 - Serghei Mihai

Au moment ou j'écris ça je me rends compte que authenticate_block existe déjà, donc certainement ça à refactoriser.

#8 - 06 février 2019 17:56 - Benjamin Dauvergne

Serghei Mihai a écrit :

> Benjamin Dauvergne a écrit :
>
> > et ajouter la gestion d'un mode "sans mot de passe" ou bien éclater authenticate en deux morceaux réutilisables, la recherche des enregistrements LDAP et la validation des credentials.
>
> authenticate doit dans ce cas faire appel à un authenticate_with_block pour que la recherche/validation de l'usager se fasse uniquement dans l'annuaire qui correspond au fournisseur SAML, ok.

Oui effectivement j'ai volontairement écris du code qui ne parle pas des blocks mais l'idée c'est effectivement de faire un get_dn_from_username(block, username) idéalement faudrait abstraire le concept d'un serveur LDAP et arrêter de passer block à tous les appels mais bon.

#9 - 12 février 2019 09:25 - Serghei Mihai

- Lié à Development #30541: django1.11: authenticate() got an unexpected keyword argument 'request' ajouté

#10 - 13 février 2019 11:31 - Serghei Mihai

- Fichier 0001-backends-add-SAML-LDAP-authentication-backend-30125.patch ajouté

Ok, un peu de séparation du code d'authentification. Tests sur jenkins passent.
Testé également en local avec un authentic qui tape dans un autre, it works.

#11 - 13 février 2019 13:57 - Benjamin Dauvergne

Serghei Mihai a écrit :

> Ok, un peu de séparation du code d'authentification. Tests sur jenkins passent.
> Testé également en local avec un authentic qui tape dans un autre, it works.

Le code dans try_bind a complètement changé, l'ordre n'est pas bon (pas de validation du mot de passe usager si if not block['connect_with_user_credentials']: c'est pas ça. T'as gagné le droit de faire un test de cet option.

#12 - 25 février 2019 10:34 - Benjamin Dauvergne

Je vais passer #30577 avant celui-ci par droit d'aînesse.

#13 - 04 mars 2019 00:05 - Serghei Mihai

- Fichier 0001-backends-add-SAML-LDAP-authentication-backend-30125.patch ajouté

Effectivement j'avais trop simplifié.

Il y a pourtant un test pour l'option connect_with_credentials (tests/test_ldap.py::test_no_connect_with_user_credentials) mais qui ne détectait pas d'erreur.
Je vais en rajouter, mais voici le patch à jour gardant au maximum la logique précedente.

#14 - 04 mars 2019 15:07 - Benjamin Dauvergne

Serghei Mihai a écrit :

> Effectivement j'avais trop simplifié.
>
> Il y a pourtant un test pour l'option connect_with_credentials (tests/test_ldap.py::test_no_connect_with_user_credentials) mais qui ne détectait pas d'erreur.
> Je vais en rajouter, mais voici le patch à jour gardant au maximum la logique précedente.

Oui le test existant n'est pas suffisant, il faudrait modifier les ACLs de la base LDAP pour voir le souci (qu'un bind user ne donne pas accès à suffisamment de choses genre ni nom, prénom ou email).

#15 - 04 mars 2019 18:21 - Serghei Mihai

En relisant et re-relisant le code je constate que dans le test il y a déjà l'ACL qui interdit la récupération des attributs lors d'un user bind:

```
@pytest.fixture
def slapd_strict_acl(slapd):
...
```

qui donc fait que get_ldap_attributes ne retourne rien et _return_user ne retourne pas d'objet User.

Comme la nouvelle méthode try_bind utilise la même logique:

```
        try:
            conn.simple_bind_s(dn, utf8_password)
            if not block['connect_with_user_credentials']:
                try:
                    self.bind(block, conn)
                except Exception as e:
                    log.exception(u'rebind failure after login bind')
                    return False
            return True
        except ldap.INVALID_CREDENTIALS:
```

le test ne detecte pas de regression.

#16 - 05 mars 2019 12:19 - Serghei Mihai

- Fichier 0001-backends-add-SAML-LDAP-authentication-backend-30125.patch ajouté

Ajout de tests sur la méthode authenticate_block pour tester l'authentification sans mot de passe et avec l'ACL interdisant l'accès aux attributs de l'usager.

#17 - 21 mars 2019 09:15 - Serghei Mihai

- Fichier 0001-backends-add-SAML-LDAP-authentication-backend-30125.patch ajouté

Patch rebasé sur master.

#18 - 21 mars 2019 12:02 - Benjamin Dauvergne

Serghei Mihai a écrit :

Patch rebasé sur master.

T'as ajouté un force_bytes et ça ne devrait pas marcher... Je regarde pourquoi ça marche quand même.

#19 - 21 mars 2019 12:04 - Serghei Mihai

Je revois aussi de mon côté.
Les builds précedents échouaient quand je faisais un force_bytes(password).
Mais rien sur le username, par ex.

#20 - 12 avril 2019 18:45 - Benjamin Dauvergne

- Assigné à changé de Serghei Mihai à Benjamin Dauvergne

#21 - 12 avril 2019 18:45 - Benjamin Dauvergne

- Fichier 0002-backends-add-SAML-LDAP-authentication-backend-30125.patch ajouté

- Fichier 0001-ldap-PEP8-code-style-30125.patch ajouté

Commencé par refaire le code style et virer le force_bytes qui me gênait.

#22 - 23 avril 2020 17:24 - Serghei Mihai

Benjamin, je pense qu'on peut rejeter car on a maintenant la possibilité de rechercher le compte via les attributs reçu de django-mellon.

#23 - 23 avril 2020 19:15 - Benjamin Dauvergne

- Statut changé de Solution proposée à Rejeté

Ok.

Fichiers

0001-backends-add-SAML-LDAP-authentication-backend-30125.patch	7,05 ko	28 janvier 2019	Serghei Mihai
0001-backends-add-SAML-LDAP-authentication-backend-30125.patch	10,8 ko	30 janvier 2019	Serghei Mihai
0001-backends-add-SAML-LDAP-authentication-backend-30125.patch	15 ko	13 février 2019	Serghei Mihai
0001-backends-add-SAML-LDAP-authentication-backend-30125.patch	16,4 ko	03 mars 2019	Serghei Mihai
0001-backends-add-SAML-LDAP-authentication-backend-30125.patch	16,8 ko	05 mars 2019	Serghei Mihai
0001-backends-add-SAML-LDAP-authentication-backend-30125.patch	16,8 ko	21 mars 2019	Serghei Mihai
0002-backends-add-SAML-LDAP-authentication-backend-30125.patch	17,3 ko	12 avril 2019	Benjamin Dauvergne
0001-ldap-PEP8-code-style-30125.patch	7,07 ko	12 avril 2019	Benjamin Dauvergne