

Authentic 2 - Bug #1322

login_password_profile template assumes openid user if there's no password

28 mars 2012 10:48 - Frédéric Péters

Statut:	Rejeté	Début:	28 mars 2012
Priorité:	Bas	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:	future	Planning:	
Patch proposed:	Non		
Description			
login_password_profile.html has this snippet:			
<pre>{% if user.password == '' %} <p>{% trans "Set password" %}</p> {% else %}</pre>			
However it will also be true in case of LDAP users, and then:			
Caught NoReverseMatch while rendering: Reverse for 'authopenid_password_change' with arguments '()' and keyword arguments '{} ' not found.			

Historique

#1 - 19 avril 2012 12:20 - Jean Christophe André

I confirm I do hit this bug too, since I choose to base our authentication on our LDAP.

I would suggest to:

1. save the source of the User creation somewhere
2. create a class for each source type (LDAP, OpenID, registration, ...)
3. define in these classes a method for determining if the password is changeable or not (the LDAP one would test for a "Allow password change" checkbox from the LDAP source form)
4. use this method to test if the "Change/Set password" link should be displayed or not (would solve this bug)
5. define in these classes a method for changing the password (the LDAP one would ask the user for old and new passwords and then call ldap.bind_s and ldap.password_s)

Just my 2 cents...

#2 - 19 avril 2012 13:15 - Benjamin Dauvergne

redmine@entrouvert.com écrivait:

La demande [#1322](#) a été mise à jour par Jean Christophe André.

I confirm I do hit this bug too, since I choose to base our authentication on our LDAP.

I would suggest to:

1. save the source of the User creation somewhere
2. create a class for each source type (LDAP, OpenID, registration, ...)
3. define in these classes a method for determining if the password is changeable or not (the LDAP one would test for a "Allow password change" checkbox from the LDAP source form)
4. define in these classes a method for changing the password (the LDAP one would ask the user for old and new passwords and then call ldap.bind_s and ldap.password_s)

Just my 2 cents...

For LDAP there is a complication, some LDAP directorie (from Microsoft, you should know which one I'm talking about :)) have special way of changing the password and on some LDAP directories the userPasswd is not accessible (sometimes because it's not event in the LDAP tree, sometimes for security reasons), and in this case you must use the special command PasswordModify. On those two occurrences you must send to the LDAP server the old and the new password. The current password modification

form only works for setting the password without recalling the old password.

So my idea is:

- overload the LDAP backend to return User class object which a new synthesized set_password() method which use LDAP commands to update the passwd instead of modifying the local User object. Something like:

```
def get_user(self, user_id):
    user = super(NewLdapBackend, self).get_user(user_id)
    def set_password(self, new_password):
        ldap_user = self.ldap_user
        ldap = ldap_user.ldap
        try:
            self.ldap_user.connection.modify_s(ldap_user.dn.encode('utf-8'),
                [(ldap.MOD_REPLACE, 'userPassword', new_password.encode('utf-8'))])
        except ldap.LDAPError:
            pass
    user.set_password = set_password
    return user
```

This code could have more features:

- handle password hashing formats for LDAP
- call the old set_password() to also set the local passwd, allowing local connection to work if LDAP is down
- report LDAP errors to admin and maybe to user (a special logging domain with a handler able to post message to the user would be great for this)
- add an LDAP_PASSWORD_MODIFY_BEHAVIOUR setting to set, if we are on openldap with full access to userAttribute, on AD or if we use PasswordModify, in case of PasswordModify or AD the set_password() method synthesized we have an extra parameter current_password.
- modify the ChangePasswordForm to have an extra input field "Current password" if the set_password() method on the current user have the extra argument "current_password".

#3 - 19 avril 2012 13:24 - Benjamin Dauvergne

My answer is not really related to this ticket, I should open another one.

The correct answer for this ticket is to copy the view from django_authopenid.views as it is not really related to OpenID but to the case where an user does not have a password but is already logged. This new view should consider that in some cases the fact that the User does not have a usable password (i.e. stored password is '!') is not a sufficient reason to allow setting the password without giving the old (see the LDAP case). But this case maybe should be resolved by overloading has_usable_password() for LDAP users.

#4 - 27 avril 2012 14:49 - Mikaël Ates (de retour le 29 avril)

- Version cible mis à 2.0.2

#5 - 11 mai 2012 18:14 - Mikaël Ates (de retour le 29 avril)

- Version cible changé de 2.0.2 à 2.0.3

#6 - 06 mars 2015 15:33 - Benjamin Dauvergne

- Priorité changé de Normal à Bas

#7 - 06 mars 2015 16:34 - Benjamin Dauvergne

- Version cible 2.0.3 supprimé

- Patch proposed mis à Non

#8 - 06 mars 2015 16:37 - Benjamin Dauvergne

- Version cible mis à future

#9 - 10 mars 2016 22:35 - Benjamin Dauvergne

- Statut changé de Nouveau à Rejeté

authopenid is not part of authentic2 anymore.