Redmine - Defect #6925

ldap authentication can fail when multiple entries are returned

2010-11-17 15:10 - Bart Vanbrabant

Status:	New	**Start date:**	2010-11-17
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	LDAP	**Estimated time:**	0.00 hour
Target version:			
Resolution:		**Affected version:**	1.0.2

Description

I've setup redmine against the ldap directory that returns multiple entries for the "uid=<login>" query. The authentication code uses the dn of the latest entry to perform the ldap bind.

For some users the entry with the shadowAccount class is returned last, for other first and for them the authentication fails.

I have worked around this by changing the filter on line 106 of file app/models/auth_source_ldap.rb to

```
object_filter = Net::LDAP::Filter.eq( "objectClass", "shadowAccount" )
```

For openldap this is valid because you can not bind against a directory that does not have this entry. Normally we would not hit this because they are specified under a two different ou's under root. In this case we need to specify the rootdn as basedn because we need entries from a third ou.

The cleanest solution would be to add an extra filter that can be used in the ldap source configuration.

History

#1 - 2011-08-06 03:49 - Ilya I

Same problem, but I use 0.9.3 from Ubuntu repos. I solved it differently - rearranging the code to try each returned entry until successful.
The new code in app/models/auth_source_ldap.rb looks like this:

```
  def authenticate(login, password)
    return nil if login.blank? || password.blank?
    attrs = []
    # get user's DN
    ldap_con = initialize_ldap_con(self.account, self.account_password)
    login_filter = Net::LDAP::Filter.eq( self.attr_login, login )
    object_filter = Net::LDAP::Filter.eq( "objectClass", "*" )
    dn = String.new
    ldap_con.search( :base => self.base_dn,
                     :filter => object_filter & login_filter,
                     # only ask for the DN if on-the-fly registration is disabled
                     :attributes=> (onthefly_register? ? ['dn', self.attr_firstname, self.attr_lastname, self.
attr_mail] : ['dn'])) do |entry|
      dn = entry.dn
# IVA2K fix for OpenLDAP:
      unless dn.empty?
        attrs = [:firstname => AuthSourceLdap.get_attr(entry, self.attr_firstname),
                 :lastname => AuthSourceLdap.get_attr(entry, self.attr_lastname),
                 :mail => AuthSourceLdap.get_attr(entry, self.attr_mail),
                 :auth_source_id => self.id ] if onthefly_register?
        logger.debug "DN found for #{login}: #{dn}" if logger && logger.debug?
        # authenticate user
        ldap_con = initialize_ldap_con(dn, password)
        if ldap_con.bind
          # return user's attributes
          logger.debug "Authentication successful for '#{login}'" if logger && logger.debug?
          return attrs
        end
      end
    end
    return nil
```

```
rescue  Net::LDAP::LdapError => text
  raise "LdapError: " + text
end
```