

Redmine - Defect #3253

LDAP Auth : Alias Dereference

2009-04-28 10:44 - Will aka S.Collado

<b>Status:</b>	New	<b>Start date:</b>	2009-04-28
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	100%
<b>Category:</b>	LDAP	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Affected version:</b>	0.8.3
<b>Resolution:</b>			
<b>Description</b>			
<p>Hello,</p> <p>I'm using last stable release of RedMine (0.8.3) with OpenLDAP 2.3.43 and I can't manage to get LDAP users authenticated.</p> <p>It seems that the ldap search is done with wrong parameter for alias dereference option.</p> <p>I made a ldapsearch from bash prompt with same filter as Redmine and all worked fine (my ldap.conf has Deref = always seen in ldapquery log as deref=3), I checked in OpenLDAP log and RedMine queries seems to be run with dereference set to "never" (seen in ldapquery log as deref=0).</p> <p>Do you know if there's a way to set this option in Redmine Source or settings files ?</p> <p>I'm not familiar with redmine source code and ruby in general, but i think it must be something related to Net::LDAP statements in app/models/auth_source_ldap.rb.</p> <p>Here is my server &amp; Redmine install info :</p> <div><div>Redmine version</div><div>Ruby version</div><div>RubyGems version</div><div>Rails version</div><div>Active Record version</div><div>Action Pack version</div><div>Active Resource version</div><div>Action Mailer version</div><div>Active Support version</div><div>0.8.3</div><div>1.8.7 (x86_64-linux)</div><div>1.3.1</div><div>2.1.2</div><div>2.1.2</div><div>2.1.2</div><div>2.1.2</div><div>2.1.2</div><div>2.1.2</div></div> <p>MySQL 5.0.58</p> <p>OpenLDAP 2.3.43</p> <p>On Linux CentOS 5.3</p> <p>Thanks in advance for your help.</p> <p>Regards.</p> <p>Will</p> <p>-----</p> <p>Sorry for my english, french user here ;)</p>			
<b>Related issues:</b>			
Related to Redmine - Patch #3358: Advanced LDAP authentication		New	2009-05-13
Related to Redmine - Feature #1913: LDAP - authenticate as user		Closed	2008-09-16
Related to Redmine - Defect #1420: LDAP authentication extremely flaky		Needs feedback	2008-06-10

History

## #1 - 2009-05-01 21:16 - Daniel Marczisovszky

- File `advanced_ldap_auth_0.8.3.diff` added

The attached patch replaces `auth_source_ldap.rb` and adds GUI options to allow alias dereferencing, custom search filter, `START_TLS`, server certificate validation level.

This patch includes Feature [#1913](#) by Adi Kriegisch.

The custom search filter works the same way as found in this patch. For example to search users whose `employeeType` contains `developer`, use this custom filter: `(&(uid=$login)(employeeType=developer))`

Searching is sub-tree, this could also solve Defect [#1954](#), but in the future it could be configurable from the GUI.

After applying this patch, run `rake db:migrate RAILS_ENV="production"`, as `auth_sources` table is modified in the database. (`starttls`, `filter`, `require_cert` and `dereference` columns are added)

The patch uses Ruby/LDAP, exactly the same way as Defect [#1420](#), so it should be installed first. On Debian systems, install the `libldap-ruby1.8` package.

## #2 - 2009-05-04 10:55 - Will aka S.Collado

Thanks for the patch, i'll test it this afternoon.

I suppose i must apply this patch to RedMine current trunk version (Stupid Question © inside ?)

## #3 - 2009-05-04 11:13 - Will aka S.Collado

Nevermind...

never post reply before finishing his mug of coffee.

Installing a fresh 0.8.3 redmine and reporting results here

## #4 - 2009-05-04 11:38 - Will aka S.Collado

- Installed Ruby/LDAP : OK
- Installed RedMine 0.8.3 : OK
- Patching RedMine 0.8.3 with diff file then `rake db:migrate` : OK

Testing :

I get a 500 error on [http://&lt;server&gt;:3000/auth\\_sources/new](http://&lt;server&gt;:3000/auth_sources/new)

extract from `production.log` :

```
Processing AuthSourcesController#new (for 81.252.183.66 at 2009-05-04 11:40:27) [GET]
  Session ID: 94b1ffce233cd05b54a5eb911d4b891b
  Parameters: {"action"=>"new", "controller"=>"auth_sources"}
Rendering template within layouts/base
Rendering auth_sources/new
```

```
ActionView::TemplateError (undefined method `starttls' for #<AuthSourceLdap:0x7fa45ccba4c8>) on line #14 of au
th_sources/_form.rhtml:
11: <p><label for="auth_source_port"><%=l(:field_port)%> <span class="required">*</span></label>
12: <%= text_field 'auth_source', 'port', :size => 6 %>
13: <%= check_box 'auth_source', 'tls' %> LDAPS
14: <%= check_box 'auth_source', 'starttls' %> START_TLS</p>
15:
16: <p><label for="auth_source_account"><%=l(:field_account)%></label>
17: <%= text_field 'auth_source', 'account' %></p>
```

```
/usr/local/lib/ruby/gems/1.8/gems/activerecord-2.1.2/lib/active_record/attribute_methods.rb:256:in `method_mis
sing'
```

Did I miss anything ?

## #5 - 2009-05-04 19:21 - Will aka S.Collado

So i took all from the beginning.

uninstalled rails, rake gems

re installing rails and rake with required versions as said on this page : <http://www.redmine.org/wiki/redmine/RedmineInstall>

This time, rake db:migrate worked as intended (i applied the patch before launching the first migration), now i can add a LDAP auth source but it's impossible to use it.

LDAP Error reported : Cannot contact the server

Settings :

server: localhost, port: 389

no TLS

filter: (objectClass=posixAccount)

Bind Account and password set up (my configuration does not allow anonymous searches).

I think i'll test LDAP connection with an home made ruby script (in fact, i should say "try to test" because i don't know ruby ... google is my friend).

Reporting result asap.

#### #6 - 2009-05-06 12:28 - Schanina Aether

I've the same issue please help us!

#### #7 - 2009-05-06 14:14 - Daniel Marczisovszky

This patch is my first met with Ruby as will, but I going to make it work on your Redmine ;)

Please create the ldap.rb on your server, run chmod 755 ldap.rb On my Debian box I had to install libldap-ruby1.8 beforehand (this is required both for this test script and the patch itself). To run it, simply start with ./ldap.rb Make sure that ruby is installed at /usr/bin/ruby or modify the script.

```
#!/usr/bin/ruby
require 'ldap'
conn = LDAP::Conn.new('ldap.integrity.hu', 389)
result = conn.bind("uid=marczi,ou=virtualUser,dc=integrity,dc=hu", "SECRET");
puts "Success!"
```

If it works, you should see the text "Success!", otherwise some Ruby error will appear.

#### #8 - 2009-05-06 15:19 - Will aka S.Collado

I tested your script modified like this :

```
#!/usr/local/bin/ruby
require 'ldap'
conn = LDAP::Conn.new('localhost', 389)
conn.set_option(LDAP::LDAP_OPT_PROTOCOL_VERSION, 3) # My OpenLDAP installation doesn't allow v2 protocol
result = conn.bind("cn=Will,ou=People,o=accounts,dc=local", "myPassword")
puts "Success!"
```

Result : "Success"

The DN that I used is not an alias. If I use an alias pointing to this DN I get "Invalid Credential" Error

#### #9 - 2009-05-06 16:14 - Will aka S.Collado

Will aka S.Collado wrote:

I tested your script modified like this :  
[...]

Result : "Success"

The DN that I used is not an alias. If I use an alias pointing to this DN I get "Invalid Credential" Error

I'll try to get Ruby/LDAP doc and to modify this script to do :

1. Connect to LDAP
2. Bind with "Bind/Root DN"
3. Set LDAP\_OPT\_DEREF to 3 (always)
4. define wanted filter, something like :

```
(&(objectClass=*)(uid=#username#))
```

5. launch an ldapSearch with defined filter

#### #10 - 2009-05-06 16:25 - Daniel Marczisovszky

Do you also get "Cannot contact the server" when you click the "Test" on the LDAP sources using DN cn=Will,ou=People,o=accounts,dc=local for binding? Dereferencing works only for searching, not for binding, I guess.

Could you please enable debug logging? To do this, open redmine/config/environments/production.rb and add this line: config.log\_level = :debug You may want to see app/models/auth\_source\_ldap.rb line 110: logger.debug "Bind as user #{ldap\_user}" if logger && logger.debug? You should restart the webserver and please check if "Bind as user" is logged or not.

Searching is not so complicated, you may use this: <http://ruby-ldap.sourceforge.net/rdoc/classes/LDAP/Conn.html#M000025>

#### #11 - 2009-05-06 16:55 - Will aka S.Collado

I still get "LdapError: Can't contact LDAP server" with "Test" link with my user DN

Dereferencing works only for searching, not for binding, I guess

Yeah I just remind that too

In LDAP, modes of alias dereferencing affect only the "search" operations

So impossible to bind from an Alias DN

Extract from the production logs :

```
Authenticating 'Will' against 'LDAP Podpilots'
LDAP-Auth with Admin User
Error during authentication: LdapError: Can't contact LDAP server
Rendering template within layouts/base
Rendering account/login
```

#### #12 - 2009-05-06 17:21 - Will aka S.Collado

I found some ldap search examples and changed the test script. This time i get what i want : attributes from aliased entry

here is the script :

```
#!/usr/local/bin/ruby
require 'ldap'
conn = LDAP::Conn.new('localhost', 389)
conn.set_option(LDAP::LDAP_OPT_PROTOCOL_VERSION, 3)
conn.set_option(LDAP::LDAP_OPT_DEREF, 3)
result = conn.bind("cn=bind,dc=local", "bindPass")

begin
  conn.search("ou=redmine,ou=Services,dc=local", LDAP::LDAP_SCOPE_SUBTREE, "(&(objectClass=*) (uid=Will))" ) { |entry|
    p entry.dn
    p entry.attrs
    p entry.vals('sn')
    p entry.vals('uid')
  }
end
puts "Success!"
```

It displays :

```
"cn=Will,ou=People,o=accounts,dc=local"
["loginShell", "sn", "objectClass", "gidNumber", "uid", "uidNumber", "cn", "homeDirectory", "userPassword", "mail"]
["S\303\251bastien Collado"]
["will"]
Success!
```

#### #13 - 2009-05-06 17:24 - Daniel Marczisovszky

Can you replace initialize\_ldap\_con(ldap\_user, ldap\_password) function in auth\_source\_ldap.rb with this:

```
def initialize_ldap_con(ldap_user, ldap_password)
  logger.debug "Connecting to #{self.host}:#{self.port}, tls=#{self.tls}" if logger && logger.debug?
  if self.tls
    conn = LDAP::SSLConn.new(self.host, self.port, self.starttls)
```

```

else
  conn = LDAP::Conn.new(self.host, self.port)
end
logger.debug "Dereference set option" if logger && logger.debug?
conn.set_option(LDAP::LDAP_OPT_DEREF, self.dereference)
logger.debug "Certificate set option" if logger && logger.debug?
conn.set_option(LDAP::LDAP_OPT_X_TLS_REQUIRE_CERT, self.require_cert)

logger.debug "Trying to bind" if logger && logger.debug?
if !ldap_user.blank? || !ldap_password.blank? then
  logger.debug "Bind as user #{ldap_user}" if logger && logger.debug?
  conn.bind(ldap_user, ldap_password)
else
  logger.debug "Anonymous bind" if logger && logger.debug?
  conn.bind
end
rescue LDAP::Error => text
  logger.debug "LDAP Connect Error: #{!}" if logger && logger.debug?
  raise
end

```

This adds more debug log, so I could see where does it fail.

#### #14 - 2009-05-06 17:26 - Daniel Marczisovszky

My guess is switching to LDAP v3 protocol is missing from my code, so you could also add that `set_option`

#### #15 - 2009-05-06 20:22 - Will aka S.Collado

I updated `auth_source_ldap/rb` with the new version of the function you gave. Still LDAP error,

Extract from production.log

```

Connecting to localhost:389, tls=false
Dereference set option
Certificate set option
LDAP Connect Error: Can't contact LDAP server

```

#### #16 - 2009-05-06 20:41 - Daniel Marczisovszky

Ok, so definitely the `conn.set_option(LDAP::LDAP_OPT_X_TLS_REQUIRE_CERT, self.require_cert)` line causes the trouble. Could you please comment it out? Moreover, can you add this line to your test script and run it from the command-line?

#### #17 - 2009-05-06 21:48 - Will aka S.Collado

I commented out the `require_cert` option line and now it connect to LDAP server (great ;) )

copy/pasting the line in my test script and bingo :

```
test.rb:6:in `set_option': Can't contact LDAP server (LDAP::ResultError)
```

Still unable to auth with an aliase DN but i think we're on the right way

#### #18 - 2009-05-06 22:03 - Daniel Marczisovszky

Are you still using `(objectClass=posixAccount)` as filter? If so, please change it to: `(&(objectClass=posixAccount)(uid=$login))` or whichever attribute you're using to authenticate.

#### #19 - 2009-05-06 22:17 - Will aka S.Collado

Filter changed to `(&(objectClass=posixAccount)(uid=$login))`, but redmine don't wants me to log in : incorrect login or password

On-the-fly user creation : on

Testing login/password directly against LDAP : OK

```

Authenticating 'Will' against 'LDAP Podpilots'
LDAP-Auth with Admin User
Connecting to localhost:389, tls=false
Dereference set option
Certificate set option
Trying to bind
Bind as user cn=bind,o=root
Search in DN: ou=Project,ou=Services,dc=local with filter: (&(objectClass=posixAccount)(uid=Will))

```

```
DN found for Will: cn=Will,ou=People,o=accounts,dc=local
Rendering template within layouts/base
Rendering account/login
```

#### #20 - 2009-05-06 22:30 - Daniel Marczisovszky

Can you replace this part in the authenticate function in auth\_source\_ldap.rb?

```
# authenticate user
ldap_con.unbind
begin
  result = ldap_con.bind(dn, password)
rescue LDAP::Error => bindError
  return nil
end
```

with this:

```
# authenticate user
ldap_con.unbind
begin
  logger.debug "Trying to login as #{dn}" if logger && logger.debug?
  result = ldap_con.bind(dn, password)
rescue LDAP::Error => bindError
  logger.debug "Login failed: #{bindError}" if logger && logger.debug?
  return nil
end
```

If you wish, you may log the password as well like this: "Trying to login as #{dn}, password: #{password}"

#### #21 - 2009-05-06 22:43 - Will aka S.Collado

Done :

```
Authenticating 'Will' against 'LDAP Podpilots'
LDAP-Auth with Admin User
Connecting to localhost:389, tls=false
Protocol version set option
Dereference set option
Trying to bind
Bind as user cn=bind,dc=local
Search in DN: ou=Project,ou=Services,dc=local with filter: (&(objectClass=posixAccount)(uid=Will))
DN found for Will: cn=Will,ou=People,o=accounts,dc=local
Trying to login as cn=Will,ou=People,o=accounts,dc=local, password: myPassword
Login failed: Protocol error
Rendering template within layouts/base
Rendering account/login
```

#### #22 - 2009-05-06 22:47 - Will aka S.Collado

That's pretty weird,  
OpenLDAP logs say :

```
conn=8066 op=0 RESULT tag=97 err=2 text=historical protocol version requested, use LDAPv3 instead
```

Seems that my  
conn.set\_option(LDAP::LDAP\_OPT\_PROTOCOL\_VERSION, 3) is ineffective

#### #23 - 2009-05-06 23:06 - Daniel Marczisovszky

Can you copy here your modification containing the conn.set\_option(LDAP::LDAP\_OPT\_PROTOCOL\_VERSION, 3)?

#### #24 - 2009-05-06 23:10 - Will aka S.Collado

```
if self.account.include? "$login" then
  logger.debug "LDAP-Auth with User login" if logger && logger.debug?
  ldap_con = initialize_ldap_con(self.account.sub("$login", encode(login)), password)
else
  logger.debug "LDAP-Auth with Admin User" if logger && logger.debug?
  ldap_con = initialize_ldap_con(self.account, self.account_password)
end
+ logger.debug "LDAP version : " + ldap_con.get_option(LDAP::LDAP_OPT_PROTOCOL_VERSION).to_s if logger && logger.debug?
```

```

    if self.filter.empty?
      filter = self.attr_login + "=" + encode(login)
    else
      filter = self.filter.gsub("$login", encode(login))
    end
  end
end

```

And

```

def initialize_ldap_con(ldap_user, ldap_password)
  logger.debug "Connecting to #{self.host}:#{self.port}, tls=#{self.tls}" if logger && logger.debug?
  if self.tls
    conn = LDAP::SSLConn.new(self.host, self.port, self.starttls)
  else
    conn = LDAP::Conn.new(self.host, self.port)
  end
+  logger.debug "Protocol version set option" if logger && logger.debug?
+  conn.set_option(LDAP::LDAP_OPT_PROTOCOL_VERSION, 3)
  logger.debug "Dereference set option" if logger && logger.debug?
  conn.set_option(LDAP::LDAP_OPT_DEREF, self.dereference)
  #logger.debug "Certificate set option" if logger && logger.debug?
  #conn.set_option(LDAP::LDAP_OPT_X_TLS_REQUIRE_CERT, self.require_cert)
end

```

#### #25 - 2009-05-06 23:12 - Will aka S.Collado

And here is the log generated by this code :

```

Authenticating 'Will' against 'LDAP Podpilots'
LDAP-Auth with Admin User
Connecting to localhost:389, tls=false
Protocol version set option
Dereference set option
Trying to bind
Bind as user cn=bind,dc=local
LDAP version : 3
Search in DN: ou=Project,ou=Services,dc=local with filter: (&(objectClass=posixAccount)(uid=Will))
DN found for Will: cn=Will,ou=People,o=accounts,dc=local
Trying to login as cn=Will,ou=People,o=accounts,dc=local, password: myPassword
Login failed: Protocol error
Rendering template within layouts/base
Rendering account/login

```

#### #26 - 2009-05-06 23:33 - Will aka S.Collado

I just added a logger line in rescue block for the Authenticating user section like this :

```

# authenticate user
  ldap_con.unbind
  begin
    logger.debug "Trying to login as #{dn}, password: #{password}" if logger && logger.debug?
    result = ldap_con.bind(dn, password)
  rescue LDAP::Error => bindError
    logger.debug "Login failed: #{bindError}" if logger && logger.debug?
    logger.debug "LDAP version : " + ldap_con.get_option(LDAP::LDAP_OPT_PROTOCOL_VERSION).to_s if logger &&
  logger.debug?
    return nil
  end
end

```

and see what it reports :

```

...
Trying to bind
Bind as user cn=bind,o=root
LDAP version : 3
Search in DN: ou=Project,ou=Services,o=root with filter: (&(objectClass=posixAccount)(uid=Will))
DN found for Will: cn=Will,ou=People,o=accounts,o=root
Trying to login as cn=Will,ou=People,o=accounts,o=root, password: 24275363*
Login failed: Protocol error
LDAP version : 2
...

```

LDAP version : 2 !?

Seems that ldap\_con.unbind reset this option, but I tried to add a line with ldap\_con.set\_option(LDAP::LDAP\_OPT\_PROTOCOL\_VERSION, 3) and it raise the error above :

```
Login failed: The LDAP handler has already unbound.
```

o\_O"

#### #27 - 2009-05-06 23:44 - Daniel Marczisovszky

Probably you're right and unbind should be avoided. Maybe this modification (this is uses an additional LDAP connection, the same way as the original auth\_ldap\_source.rb does in Redmine). The part after #authenticate\_user should be replaced in the authenticate function with this:

```
# authenticate user
# ldap_con.unbind
begin
  logger.debug "Trying to login as #{dn}" if logger && logger.debug?
  # result = ldap_con.bind(dn, password)
  initialize_ldap_con(dn, password)
rescue LDAP::Error => bindError
  logger.debug "Login failed: #{bindError}" if logger && logger.debug?
  return nil
end
```

Hope it works, I'll check it tomorrow. I-)

#### #28 - 2009-05-06 23:55 - Will aka S.Collado

- Status changed from New to Resolved

- % Done changed from 0 to 100

Victory !! \o/ It works like charm !

Many many thanks for your help !

Now RedMine can auth users from LDAP Directory, even for alias DN, hope this patch will be added to the trunk for future versions.

Thanks again for your help, tomorrow i will test that fix on another RedMine in production which needs this patch too.

#### #29 - 2009-05-07 14:51 - Daniel Marczisovszky

The lesson is that there should be a new field for selecting protocol V2 or V3. Moreover unbind should be avoided. In PHP and Java you can rebind by calling bind again (without unbind), so I'll create a new version and ask you to test it. If it does not work with Ruby/LDAP then the current solution will remain.

#### #30 - 2009-05-07 15:08 - Will aka S.Collado

No problem, I'll keep a test install of redmine dedicated to these tests.

Waiting for you patch ;)

#### #31 - 2009-05-07 15:42 - Daniel Marczisovszky

The patch with the protocol version will be ready only next week, but till then you may try this code:

```
# authenticate user
# ldap_con.unbind
begin
  logger.debug "Trying to login as #{dn}" if logger && logger.debug?
  result = ldap_con.bind(dn, password)
rescue LDAP::Error => bindError
  logger.debug "Login failed: #{bindError}" if logger && logger.debug?
  return nil
end
```

In this one unbind is removed, but instead of creating a new LDAP connection, it simply re-binds.

#### #32 - 2009-05-07 15:52 - Will aka S.Collado

Seems that it don't like re-bind :

```
Trying to login as cn=Will,ou=People,o=accounts,dc=local
Login failed: already bound.
```



#33 - 2009-05-12 15:25 - Will aka S.Collado

I think i'll roll back to the unbind / new connection solution.

#34 - 2009-05-12 15:52 - Daniel Marczisovszky

I think the unbind is not necessary, but the new connection method should be used. Re-binding on an existing connection definitely won't work. This evening I'll modify the patch to make it possible to select protocol version.

#35 - 2009-05-12 21:33 - Daniel Marczisovszky

- File *advanced\_ldap\_auth\_0.8.3.diff* added

This new version contains the LDAP protocol version dropdown box, but you have to recreate your database or add protocol\_version integer not null default '3' column to auth\_sources table.

#36 - 2009-05-12 22:40 - Will aka S.Collado

Daniel Marczisovszky wrote:

This new version contains the LDAP protocol version dropdown box, but you have to recreate your database or add protocol\_version integer not null default '3' column to auth\_sources table.

It works like charm :  
Diff merged to a fresh extracted RedMine 0.8.3 sources, installing database, adding ldap config, authenticating with LDAP aliased user : user created with success in database !

#37 - 2009-07-14 18:32 - Daniel Marczisovszky

Further versions of this patch will be uploaded to <http://www.redmine.org/issues/3358>

#38 - 2011-09-13 14:33 - Etienne Massip

- Category changed from *Accounts / authentication* to *LDAP*

#39 - 2013-01-15 22:30 - Jan Niggemann (redmine.org team member)

Closing this, status is resolved since 400 days and more (issue was last updated more than 400 days ago)...

#40 - 2013-01-15 22:37 - Jan Niggemann (redmine.org team member)

- Status changed from *Resolved* to *Closed*

#41 - 2013-01-19 20:13 - Etienne Massip

- Status changed from *Closed* to *New*

Files

advanced_ldap_auth_0.8.3.diff	8.34 KB	2009-05-01	Daniel Marczisovszky
advanced_ldap_auth_0.8.3.diff	9.93 KB	2009-05-12	Daniel Marczisovszky