# Metacat - Bug #138

## use LDAP as optional directory for authentication

09/22/2000 03:16 PM - Matt Jones

| | | | |
|---|---|---|---|
| **Status:** | Resolved | **Start date:** | 09/22/2000 |
| **Priority:** | Immediate | **Due date:** | |
| **Assignee:** | Jivka Bojilova | **% Done:** | 0% |
| **Category:** | metacat | **Estimated time:** | 0.00 hour |
| **Target version:** | Beta2 | **Spent time:** | 0.00 hour |
| **Bugzilla-Id:** | 138 | | |

**Description**

May be powerful to use an LDAP server or network to handle authentication rather than the current MCAT authentication mechanisms we're using. Probably need to create an abstract interface for authentication, so that we can implement different concrete classes to connect to each type of authentication system (e.g., one for MCAT authentication, one for LDAP authentication). The interface would have only a few methods, like "authenticateUser(username, password)", and "getGroups(username)", etc.

### History

#### #1 - 10/04/2000 02:50 PM - Matt Jones

Designed and implemented a new abstract authentication interface (AuthInterface.java) in metacat that allows the metadata server to utilize different backend authentication services, determined at runtime through properties in the metacat.properties file. Implemented an LDAP authentication service that works for authentication but doesn't yet implement all of the user and group listing capabilities of the srb system currently used in MetaCatSession.java. New classes for this service include AuthInterface.java, AuthLdap.java, AuthSession.java (will replace MetaCatSession.java). Also made modifications to MetaCatServlet.java to utilize this new authentication facility. Need to create an implementation of the interface for MCAT authentification that parallels the one done in AuthLdap.java (most of the code can be borrowed from MetaCatSession.java).

#### #2 - 10/04/2000 02:51 PM - Matt Jones

The changes described inthe previous comment are checked into CVS uder a separate branch with the tag "AUTH_LDAP".

#### #3 - 10/26/2000 02:39 PM - Matt Jones

Fixed the LDAP authentication adapter (AuthLdap.java) so that it now looks up the distinguished name for a user before attempting to do authentication. This is because the user's distinguished name can sometimes be based on their uid attribute, but sometimes be based on their cn (common name) attribute, or some other attribute. In order to authenticate, we must be able construct the distinguished name, so we have to look up the identifying attribute before trying to authenticate. Basically, this lets us authenticate agains both of the following records:

```
dn: uid=jones,o=NCEAS,c=US
```

and

```
dn: cn=Matt Jones,o=NCEAS,c=US
```

Effectively, this means that the user can type in their user id (uid), common name (cn), or surname (sn) and we'll still be able to authenticate them.

#### #4 - 02/14/2001 11:43 AM - Matt Jones

Basic functionality is now finished, but some of the interface functions (like getting group lists, etc) need to be completed still (They have null implementations now).

**#5 - 06/29/2001 06:20 PM - Jivka Bojilova**

Installed OpenLDAPv2.0.11 (support for LDAPv3) with TLS/SSL support on dev.

Here is the steps to do:
1. Install OpenSSL first for use from LDAP server for support of TLS/SSL.
- get openssl-0.9.5a-14-i386.rpm
and openssl-devel-0.9.5a-14-i386.rpm from RedHat Linux7 CD
- copy them to /tmp on dev
- install them as packages with
rpm ~~i openssl-0.9.5a-14-i386.rpm~~
~~rpm -i openssl-devel-0.9.5a-14-i386.rpm~~
to check if installed
rpm -q openssl
rpm -q openssl-devel

1A. Install OpenSSL from www.openssl.org (doesn't work for OpenLDAPv2.0.11)
- gunzip openssl-engine-0_9_6a_tar.gz in /usr/local/devtools on dev
- tar ~~xvf openssl-engine-0_9_6a_tar~~
cd openssl-engine-0.9.6a
- ./config
- make -> make test -> make install
This creates /usr/share/ssl, /usr/include/openssl dirs, and files
in /usr/bin, /usr/lib, ...

2. Generate self-signed x509 certificate for Metacat <-> LDAP communications
over SSL
- cd /usr/share/ssl
- generate new certificate request & key pair
openssl req ~~new -out REQ.pem -keyout KEY.pem~~
generate the self-signed x509 certificate from the cert request and the key
openssl req -x509 -in REQ.pem -key KEY.pem -out CERT.pem
These create 3 files in /usr/share/ssl
REQ.pem
KEY.pem
CERT.pem

3. Import the certificate on the client (Metacat) as trusted:
keytool ~~import -alias ldap -file /usr/share/ssl/CERT.pem -keystore~~
~~$JAVA_HOME/jre/lib/security/cacerts~~
~~[Note: The following should be preliminary done when configuring Metacat with~~
~~SSL support:~~
Install JSSE as extension on Metacat: copy jart.jar, jnet.jar, and jsse.jar
in $JAVA_HOME/jre/lib/ext and in $TOMCAT_HOME/lib (this won't be necessary with
JSDKv1.4)
- Register the SUN's JSSE provider by adding 1 line in
$JAVA_HOME/jre/lib/security/java.security as:
security.provider.2=com.sun.net.ssl.internal.ssl.Provider]

4. Install OpenLdap with TLS/SSL support (get the distribution from
www.openldap.org)
- download a copy to /user/local/devtools
- tar ~~xvf openldap-2_0_7.tgz~~
cd openldap-2.0.7
- CPPFLAGS=-I/usr/include; export CPPFLAGS
- LDFLAGS=-L/usr/lib; export LDFLAGS
- ./configure --prefix=/usr/local/devtools --with-tls --with-wrappers --with-
threads
- make depend -> make -> make test -> su root -> make install

5. Configure slapd (edit slapd.conf)
- cd /usr/local/devtools/etc/openldap
- vi ./slapd.conf
- insert 3 lines as:
include /usr/local/devtools/etc/openldap/schema/core.schema
include /usr/local/devtools/etc/openldap/schema/cosine.schema
include /usr/local/devtools/etc/openldap/schema/inetorgperson.schema
- edit suffix and rootdn as:
suffix "dc=ecoinformatics,dc=org"
rootdn "cn=Directory Manager,dc=ecoinformatics,dc=org"
- insert ldbm access control definitions as:
access to attr=userPassword
by anonymous auth
by dn="cn=Directory Manager,dc=ecoinformatics,dc=org" write
by * none

access to *
by dn="cn=Directory Manager,dc=ecoinformatics,dc=org" write
by * read
- insert 3 lines for TLS/SSL support as:
TLSCertificateFile /usr/share/ssl/CERT.pem
TLSCertificateKeyFile /usr/share/ssl/KEY.pem
TLSCACertificateFile /usr/share/ssl/CERT.pem

6. Run slapd (as root)
- cd /usr/local/devtools/libexec
- ./slapd [-h "ldap:/// ldaps:///"] [-d debug-level] [-u username/id]
where -h option is used to start two listeners: one for LDAP on default port 389 and the second for LDAP over SSL on default port 636

Note: LDAPv3 (as with this version of OpenLdapv2.0.11) supports Start TLS Extension.
The LDAP server must be set up with X.509 certificate (already did) and requires JSDKv1.4 on the client (Metacat) in order to use the extended interface in javax.naming.ldap for StartTLS support.
With TLS the same port 389 is used and does not require the second port as 636.

7. Restart Tomcat

#### #6 - 07/13/2001 03:35 PM - Jivka Bojilova

DONE

#### #7 - 03/27/2013 02:13 PM - Redmine Admin

Original Bugzilla ID was 138