Metacat - Bug #5262

EML document owner can't read the document

12/21/2010 06:11 PM - Jing Tao

Status:	New	**Start date:**	12/21/2010
Priority:	Normal	**Due date:**	
Assignee:	ben leinfelder	**% Done:**	0%
Category:	metacat	**Estimated time:**	0.00 hour
Target version:	2.x.y	**Spent time:**	0.00 hour
Bugzilla-Id:	5262		

Description

An eml document was inserted into metacat.
There is no access rule in the document, so only the owner can read the document.

However, it seems the document can't read the document.

Here is the record in xml_documents. it shows kepler user is the owner:
test=# select * from xml_documents where docid ='doc.1292983505983';
docid | rootnodeid | docname | doctype | user_owner | user_updated | server_location | rev | date_created | date_updated | public_access | catalog_id
---
doc.1292983505983 | 2498 | eml | eml://ecoinformatics.org/eml-2.1.0 | uid=kepler,o=unaffiliated,dc=ecoinformatics,dc=org | uid=kepler,o=unaffiliated,dc=ecoinformatics,dc=org | 1 | 1 | 2010-12-21 | 2010-12-21 | 0 | 16

However, when I tried to read the document as kepler and metacat give me the error:
User uid=kepler,o=unaffiliated,dc=ecoinformatics,dc=org does not have permission to read the document with the docid doc.1292983505983.1

My metacat version is 1.9.3

I tried this in dev.nceas.ucsb metacat and got the same result.

History

#1 - 12/21/2010 10:11 PM - ben leinfelder

What was the EML document?
Are there any records in the xml_access table for this document?

#2 - 12/22/2010 09:31 AM - Jing Tao

In dev metacat, the eml document is is doc.1292979786139.1. And owner is uid=maanand,o=unaffiliated,dc=ecoinformatics,dc=org.

This document was generated by a workflow which was created by manish from SDSC. The worklfow will upload metadata (eml format) and data from dataturbine to a metacat.

There is no any record in xml_access table for the eml.

#3 - 12/22/2010 10:09 AM - ben leinfelder

I tried this on two local machines (running from svn trunk) and then on dev.nceas.ucsb.edu. In al cases, the owner was able to view the document.
For dev I had to edit the online distribution URL so that I didn't appear to be messing with an existing data table.
Here's the example on dev (login as kepler user first):
http://dev.nceas.ucsb.edu/knb/metacat?action=read&docid=brl.1.1&qformat=default

#4 - 02/14/2011 11:42 AM - Jing Tao

Somehow, in EcogridWriter actor, author put an extra space after uid=kepler,o=unaffiliated,dc=ecoinformatics,dc=org.

So the "uid=kepler,o=unaffiliated,dc=ecoinformatics,dc=org " was recorded as the owner. "uid=kepler,o=unaffiliated,dc=ecoinformatics,dc=org" is not recognized as the owner.

We need to trim the string before we record it to the xml_documents table

#5 - 02/14/2011 11:54 AM - Matt Jones

Trimming strings on input like this is always dangerous. My guess is that the bug arises because someone through a 'trim' in on the original login field, which allowed someone to authenticate incorrectly in the first place. a user DN should be an exact match to what is in LDAP (without any trimming), and if it is not it should be rejected both for login and for any access control checks.

Jing -- can you check and see if someone can log in successfully if they have a space following their DN, and if so stop that from happening?

Throwing in a trim() call here or there just eliminates any precision we had and creates a bunch of corner cases that we have to deal with in the code wherever that field is referenced.

#6 - 02/14/2011 01:47 PM - Jing Tao

Hi, Matt:

Yes, the user name with extra space can login. The username field in EcogridWriter is used for login. The workflow working means the the username (with extra space) and password can login successfully.

I just browsed the code and didn't find any trimming in username. But I will take a close look.

#7 - 02/14/2011 02:48 PM - Jing Tao

Hrrm, it seems our code doesn't trim the username.

In AuthLdap.authenticateTLS:

I added a information line to display the userName.

```
ctx.addToEnvironment(Context.SECURITY_AUTHENTICATION, "simple");
System.out.println("===============The userDN is "+userDN+".")
ctx.addToEnvironment(Context.SECURITY_PRINCIPAL, userDN);
ctx.addToEnvironment(Context.SECURITY_CREDENTIALS, password);
ctx.reconnect(null);
```

The output is:
===============The userDN is uid=tao,o=NCEAS,dc=ecoinformatics,dc=org .

So our metacat doesn't trim the username variable, the correct username was passed to the javax.naming.ldap.LdapContext object. But the login still succeed with the username having extra space.

This means we can't modify it since trimming doesn't happen in our source code space.

Did I miss anything?

#8 - 10/26/2011 04:11 PM - Matt Jones

Probably solution is to use Java's canonicalization routines to normalize the DN, as Ben has done on the DataONE DNs. For now it will rarely be encountered so delaying work on it.

#9 - 03/27/2013 02:29 PM - Redmine Admin

Original Bugzilla ID was 5262