Metacat - Bug #968

Access control for eml2 documents

01/23/2003 03:19 PM - Jing Tao

Status:	Resolved	**Start date:**	01/23/2003
Priority:	Normal	**Due date:**	
Assignee:	Saurabh Garg	**% Done:**	0%
Category:	metacat	**Estimated time:**	0.00 hour
Target version:	1.5	**Spent time:**	0.00 hour
Bugzilla-Id:	968		

Description

Access control for eml2 documents is pretty complicated.

In access control submodule, the access rules control the documents access, this can be handled as same as eml beta6. But in additional meta data part, user can specify another access control for a subtree of the document. This control is not base on docid and is base on node id. So we maybe need to change the table structure of xml_access table: adding a new field named nodeid which is foreign key of xml_nodes table.

When we access control, it should be base on sql rather than java code for performance issue.

Related issues:

Is duplicate of Metacat - Bug #967: Access control for eml2 documents	**Resolved**	**01/23/2003**
Blocks EML - Bug #1132: fix access control rule ambiguities	**Resolved**	**08/15/2003**

History

#1 - 01/23/2003 03:30 PM - Jing Tao

- Bug 967 has been marked as a duplicate of this bug. ***

#2 - 03/05/2003 11:13 AM - Jing Tao

In xml_access table, 3 new fields - subtreeid, startnodeid and endnodeid were added. They are used as subtree access control.

Currently, metacat can pull out access control info for both top document level and subtree level(in additionalmetadata part) and write the info to xml_access table.

If user try to read a document, metacat will check both document level and subtree level access contol. If a subtree is not allow to a user, he will only part of document.

In search, the return field access control was applied too. If the return field is in a subtree that the user could not access, the search result would NOT show it's value.

So now, read and search access control is almost done.

But, we still have a big issue in writting access control for subtree. If a subtree is set to a user only can read, but top doucment the user can write and read. It is difficult to know if the user update the subtree which he doesn't have a permission to write when he submit a newer version documents.

Another issue is: citation, software and protocol moudle can be include in dataset module, but these modules can have access elemment in itself too. So in a dataset module, it is possible to have lots of access elements(they come from citation, software or protocol) except datset's access element and access element in additionalmetadata. How do we handle these extral access?

#3 - 04/30/2003 06:16 PM - Jing Tao

Currently, metacat only support access control for top level. The others rule will be ignored.

#4 - 04/30/2004 03:09 PM - Saurabh Garg

Main points based on IRC chat on April 30th, 2004. The chat text is also copied below.

1. if /eml/access (in 2.0.1), or /eml/dataset/access (in 2.0.0) is present, we follow the rules laid out there for the whole package
2. /eml/citation/access, /eml/software/access and /eml/protocol/access (in 2.0.0) are always ignored. These along with /eml/dataset/access will be not allowed from 2.0.1
3. if there is an additional rule set for the data only (location to be determined), then that can override the spec from the top level
4. new documents being inserted can be eml 2.0.0

Regarding point 3, I agree to the location talked about in bug 1132. i.e. the diagram below
<dataTable>
<physical>
<distribution>
<access>...</access> <-- defines access to the data object
in inline, online, or offline
elements (ie, not the metadata
itself, just the data)
<inline>...</inline>
</distribution>
</physical>
</dataTable>

I think this would be a better place as compared to <additionalMetadata>

I am not sure yet what would be the best way of proceeding to implement this solution. I will have to see metacat code and how xml_access table is used. Watch out for more in this space..

##################
[11:17] <sid> is it possible to set read access to metadata and not allow read access to the data itself
[11:17] <jing> good question.
[11:17] <jing> you mean in eml2?
[11:18] <sid> i guess it is dependent on eml2
[11:18] <jing> currently we don't support that.
[11:19] <sid> i thought it was possible - but i am not able to locate access permissions for datatable or any other entity
[11:19] <sid> ok
[11:19] <matt> this is an ongoing issue
[11:19] <jing> yes.
[11:19] <matt> see the bug on access control in EML for a description of the issues
[11:20] <matt> bug 1132
[11:20] <sid> ok matt .. this is regarding the email that rick sent yesterday regarding specnet - i will have to backtrack on the reply that i sent to him
[11:21] <matt> or we can fix it in metacat
[11:21] <matt> which we need to do anyways
[11:21] <matt> it would be a nice fix to have in for the metacat 1.4 release
[11:23] <sid> so access control for entities will remain in additional metadata in eml?
[11:23] <jing> I think so.
[11:23] <sid> actually i think i will read bug#1132 first
[11:24] <matt> yeah
[11:24] <matt> read that, then lets discuss it
[11:32] <sid> matt - ignoring access inside dataset would be a big change
[11:42] <sid> but i do agree that it will take care of a lot of complexity that we will have to face otherwise
[11:44] <sid> maybe a rule can be specified that if no overall access if described then dataset access is considered for assigning access to the document - but no future document can have access inside dataset
[11:45] <matt> you lost me there
[11:45] <matt> i thought the proposal in 1132 was to have 2 levels of access: metadata, and data
[11:45] <sid> yes
[11:46] <matt> if data access is unspecified, it defaults to the same as metadata

[11:46] <sid> yup
[11:46] <matt> you see a problem with this approach?
[11:46] <sid> thats what i meant
[11:46] <matt> ok
[11:46] <matt> i guess i misinterpreted what you meant by "overall access"
[11:47] <matt> the question is how to specify these rules in the context of EML2.0.0 and EML2.0.1
[11:47] <sid> only problem will be when dataset has different access and citation has different access and protocol has different access
[11:47] <matt> technically metacat is not following the rules for EML200 now because they are unimplementable as specified
[11:48] <matt> access and citation and protocol are all metadata, so only the topmost access block would apply there
[11:49] <matt> there shouldn't be any opportunity for conflicting access specs
[11:49] <sid> and if topmost is missing which access block do you default to?
[11:49] <matt> if topmost is missing, only owner of metadata/data has any access at all
[11:50] <sid> can you default to dataset access for the sake of backward compatibility
[11:52] <sid> even if we just default to dataset access, i think it will take care of 99%(if not 100%) eml documents we have
[11:52] <matt> what do you mean by "dataset access"?
[11:52] <sid> access block inside dataset block
[11:53] <matt> right now, if eml/dataset/access is missing, nobody but owner has access to anything
[11:53] <matt> i think this should be maintained
[11:53] <sid> yeah i think so too - but what if it is specified
[11:54] <matt> if /eml/access (in 2.0.1), or /eml/dataset/access (in 2.0.0) is present, we follow the rules laid out there for the whole package
[11:54] <sid> ok
[11:54] <sid> all fine and clear
[11:54] <matt> if there is an additional rule set for the data only (location to be determined), then that can override the spec from the top level
[11:55] <matt> that's it I think
[11:55] <sid> i agree
[11:58] <sid> will this mean once metacat changes it wont accept any eml 2.0.0 documents?
[11:58] <matt> no, we'll need to maintain backwards compatibility
[11:58] <matt> although the access rules may be interpreted differently
[11:58] <matt> as we just discussed
[11:58] <sid> so new documents being inserted can be eml 2.0.0
[11:59] <matt> yes

#5 - 05/05/2004 11:17 AM - Saurabh Garg

From what I have understood till now (with Jing's help)

1. The document inserted to Metacat is first validated in MetacatServlet.java using the EMLParser.class from eml module. I am not sure about one thing here - when eml-2.0.1 is released, will EMLParser.class take care of parsing the version 2.0.1 also or will there be a seperate class to take care of parsing the new version?
2. Once this step is taken care of - from DocumentImpl.java, different classes are called on basis of eml version type. So for eml2, EmlSAXHandler class is called. EmlSAXHandler takes care of how access is handled. So I could think of two options of doing this:
~~> Put code in EmlSAXHandler to distiguish between eml 2.0.0 and eml 2.0.1~~
and handle access accordingly.
-> Rename EmlSAXHandler to Eml2_0_0_SAXHandler. And make a new class Eml2_0_1_SAXHandler to take care of version 2.0.1. So from DocumentImpl code, we find out which version the current eml document belongs to and call Eml2.0.X_SAXHandler accordingly. Eml2_0_1_SAXHandler will be same as Eml2_0_0_SAXHandler accept for the way in which access is handled.

I am more inclined towards the second idea as it would scale well.

#6 - 09/10/2004 02:42 PM - Saurabh Garg

Changing target_milestone to 1.5 as this bug covers eml-2.1.0 also.

#7 - 09/10/2004 02:57 PM - Saurabh Garg

I am closing this bug as access control for eml version 2.0.1 is implemented now. A new bug will be opened for eml version 2.1.0

#8 - 03/27/2013 02:15 PM - Redmine Admin

Original Bugzilla ID was 968