

Metacat - Bug #92

need access control tracking for metadata documents

08/20/2000 10:49 PM - Matt Jones

Status:	Resolved	Start date:	08/20/2000
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:	92		
Description			
<p>Need to create a mechanism for tracking access control information for metadata documents. This might be done by creating a new set of attributes and tables to store access control lists (acl), or we might try to store this information in the SDSC MCAT along with the existing ACL info for datasets. The latter would likely be more complicated, so initial attempts will probably create local storage for the ACLs.</p> <p>Also, need an easy-to-use API for the MetaCatServlet through which clients (like dmanclient) can query and change the access control lists (either the owner of document is only person who can change acls, or there needs to be a specific permission granting the ability to change acls).</p>			
Related issues:			
Blocked by Morpho - Bug #88: need ability to manipulate access control lists		Resolved	08/20/2000

History

#1 - 09/22/2000 03:43 PM - Matt Jones

Changed target milestone to Beta2

#2 - 01/12/2001 11:20 AM - Jivka Bojilova

DONE

1. ACL info for resources (data and metadata docs) is applied through access files only.
2. Access files are submitted to Metacat in the way like other metadata docs with the difference that in background records are created in xml_access and xml_relation tables for use.
3. Access files could be submitted to Metacat only after submission of the resources specified within them by <resourceIdentifier> tags. Resources are specified with their whole metacat URLs. docids are extracted from the URL by parsing the URL query string using MetaCatUtil.parseQuery(URL murl.getQuery()) routine.
4. Access file can specify acl info for resources only owned by the user submitting the access file or having "all" permissions on all off them. In other case submission of the access file is rejected.
5. Access files itself get by default public read access as all metadata docs which is convenient for now during development by probably should be made optional in order to be specified by the client. (?)
6. It is possible same permission for a user on given resource to be specified from different access files. In this case the permission that is the most (by time duration, by perm_order: "allowFirst" or "denyFirst") is used (simple algorithm is implemented).
7. "accessfileid" attr in xml_access table stores docid of the access file. Also "docid" attr in xml_relation table is again the docid of the access file that brings the relationships (<accessfile, 'isaclfor', resource>). These attrs are convenient when access file is updated or deleted to delete the related records from xml_access and xml_relation tables first (these 2 tables keep only the current/last version of the access file)
8. "read" action checks the user for having "read" permission on the found docs (if not publicly readable or not his/her own). Only docs on which user have "read" permission are extracted (along with his/her own and publicly readable).
9. "update" and "delete" actions check for "write" permission on the manipulated doc (owned docs are permitted again).

#3 - 03/27/2013 02:13 PM - Redmine Admin

Original Bugzilla ID was 92