Infrastructure - Task #2106

Feature # 1764 (Closed): Finalize dataoneTypes schema for public release

Types.AccessRule: Uploader accidentally getting locked out

2011-12-07 05:56 - Roger Dahl

Status:	Closed	**Start date:**	2011-12-07
Priority:	Normal	**Due date:**	
Assignee:	Matthew Jones	**% Done:**	100%
Category:	Documentation	**Estimated time:**	0.00 hour
Target version:	Sprint-2011.51-Block.6		
Milestone:	CCI-1.0.0	**Story Points:**	
Product Version:	*		

Description

Someone that creates a new object without providing an access policy never has the opportunity to modify or create the access policy on the new object if they put someone other than themselves as the rights holder. In that case, they're entirely locked out from the new object. It would be better if we could set up something where the uploader would have to perform some more explicit action to remove his own rights to the object.

History

#1 - 2011-12-22 06:40 - Matthew Jones

- Status changed from New to Closed

- % Done changed from 0 to 100

Discussion on 12/21/2011 indicated that this might indeed be a problem, but for now we are going to wait and see. Client code should check for this and generally warn the user that it is a bad idea to not designate the submitter as the RightsHolder, or at least grant the submitter at least write access via the accessPolicy. If it is proves a problem with a lot of stranded objects, then one potential solution is to modify MNs to reject create() or update() calls on which the submitter is not listed as a RightsHolder or in the AccessPolicy -- then removing that person later would become a separate step. Updated the schema documentation to clarify that, for now, the submitter has no special rights.

For now, the decision is to not make any changes. Conversation leading to this decision is below for reference purposes.

IRC #dataone conversation from 21 Dec 2011:

[3:58pm] matt: roger has pointed out that if the Submitter is not listed as RightsHolder, a person can immediately lock themselves out:
[3:58pm] matt: https://redmine.dataone.org/issues/2106
[3:58pm] matt: I have encountered this myself while testing
[3:58pm] roger: Is that a thank you for giving you lots of work, or for not adding any more issues?
[3:59pm] matt: right!
[3:59pm] roger: hehe
[3:59pm] matt: but right now, that's how its defined. submitter is informational only, and immutable.
[4:00pm] matt: RightHolder is now required, and the person submitting needs to be very careful to set either rightsHolder or accessPolicy properly so they continue to have access to their own document
[4:00pm] roger: In the discussion in the dev meeting, it didn't seem like there were any good solutions.
[4:00pm] matt: soo... options: 1) leave as is, accept that lockouts will occur, authoritative MN will have to intervene,
[4:01pm] matt: 2) change so submitter always has changePerm permission; but because submitter is immutable, that permission can never be revoked
[4:01pm] matt: so, if a PI has a student submit their lab data, the student will forever have total access
[4:02pm] matt: I'm torn as to which way to go
[4:02pm] vieglais: option (1) with appropriate checks in client and MN code
[4:02pm] matt: metacat implements (1) right now I think
[4:02pm] roger: Add an UploadedOnBehalfOf field?
[4:02pm] matt: there's no check in MN code - its too late then
[4:03pm] vieglais: the MN could reject on create, no?
[4:03pm] matt: ie, this is a side effect of create() with system metadata set a certain way
[4:03pm] matt: it could, but then it would eliminate a perfectly valid submission
[4:04pm] matt: (e.g., Joe is submitting for Charlie, but only Charlie should have subsequent rights)
[4:04pm] matt: the client is the right place to check, as that's where communicaiton wth the submitter occurs
[4:04pm] roger: Put a blurb into the create() call that clients have to check for this condition and warn the submitter?
[4:05pm] vieglais: create with a switch. like sudo create - yes I really want to make this potential mistake
[4:05pm] matt: :0 you really want to change our REST methods?
[4:06pm] vieglais: nope.
[4:06pm] vieglais: so then the only real option is in the client code

[4:07pm] matt: I can put a cautionary note in the Submitter and rightsHolder field docs so it is clear to people the implications
[4:07pm] roger: How about implicitly including the Submitter in the list of Rights Holders on the MN side? That way, the submitter has initial rights and someone must perform an action to remove him.
[4:08pm] roger: So the MN would add the submitter into the list of rights holders if not already there.
[4:08pm] matt: hm. what if the client doesn't want that?
[4:09pm] roger: They have to run a second SetAccessPolicy call to explicitly remove him.
[4:10pm] matt: so Joe uploads a doc that says Charlie is the RH, and then Joe is added automatically with changePerm in an access policy (supplementing any additional access policies already in the sysmeta)
[4:11pm] matt: then Joe would do an immediate set access call to remove himself?
[4:11pm] roger: Right.
[4:11pm] matt: seems complicated, and unlikely people would understand.
[4:11pm] roger: It makes the removal of his rights into an explicit action.
[4:12pm] matt: yep. but it also makes his explicit request contain implicit additions
[4:12pm] roger: True.
[4:12pm] matt: which might lead to confusion; I said to do X, you did X+Y
[4:12pm] roger: Closely related alternative is to reject create() if the submitter is not in the rights holders list.
[4:13pm] vieglais: my preference would be to not modify content, but find a way to allow for programmatic checks to reject if necessary
[4:13pm] matt: if we do that, there is no reason to have a separate field -- just need one field then
[4:14pm] roger: So if we do the reject, there is no MN modification of content.
[4:14pm] matt: dave -- that's my preference too. I have an aversion to automatically injecting content or changing behavior on a user
[4:15pm] matt: roger -- you lost me on that one
[4:16pm] roger: So, we can require that the client add the subject to the rights holders. In the normal case where they want to retain the rights, that verifies that they do what they want to. If they want to give up the rights immediately after create, they follow the create with a setAccessPolicy that removes themselves. That way, removing is explicit and there are no implicit modifications of what they...
[4:16pm] roger: ...submitted.
[4:19pm] matt: does setAccessPlicy() make any changes to rightsHolder? I thought it only changed accessPolicy?
[4:20pm] roger: Checking... I thought rightsHolder was part of the access policy?
[4:20pm] matt: nope. its a separate field
[4:21pm] roger: I see. So instead of setAccessPolicy, the second call would whatever we use for chaning the list of rights holders.
[4:22pm] roger: It would be setRightsHolder().
[4:23pm] matt: ah right. I forgot about that one.
[4:24pm] matt: reading the schema docs here, they are actually saying the sumitter is the default rights holder if one is not provided
[4:24pm] matt: but interestingly, rightsHolder is required, whereas submitter is currently optional
[4:24pm] roger: Submitter is optional? That is kind of interesting.
[4:25pm] vieglais: submitter is gleaned from the auth
[4:25pm] matt: (which I guess is ok because the MN takes submitter from the certificate that is logged in anyways)
[4:25pm] roger: Oh, of course.
[4:25pm] roger: Yes, that was a recent change.
[4:26pm] matt: ok, so now we have 3 optons: the first two I listed above as (1) and (2), plus roger's suggestion (3): require on create() and update() that submitter be the rightsHolder, and any changes to that must occur on subsequent calls to setRightsHolder()
[4:27pm] matt: all MN implementations would need to enforce (3)
[4:28pm] roger: And if they didn't, behavior would fall back to (1).
[4:28pm] matt: yep
[4:29pm] matt: what exception would be thrown on create() or update() if the submitter wasn't the rightsHolder?
[4:29pm] vieglais: invalidRequest?
[4:30pm] vieglais: kind of generic though
[4:30pm] matt: InvalidSystemMetadata?
[4:31pm] matt: I think we should punt on this one, and go with (1)
[4:31pm] roger: I think InvalidSystemMetadata, with a note about why it was rejected.
[4:32pm] matt: lets see if its an issue, and if it is, we can get MNs to implement the more complicated behavior in (3)
[4:32pm] matt: either way, there should be no signature change
[4:32pm] vieglais: yeah. I think that seems reasonable.
[4:32pm] roger: Ok.
[4:33pm] matt: I'm going to paste this IRC conversation into the ticket if that's ok with you both?
[4:33pm] roger: Sure.
[4:33pm] vieglais: np
[4:36pm] matt: ok thanks!