Infrastructure - Bug #2583

Metacat CN-CN replication permOrder issue with EML-defined access rules

2012-04-05 22:32 - Ben Leinfelder

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Ben Leinfelder	**% Done:**	100%
Category:	Metacat	**Estimated time:**	0.00 hour
Target version:	Sprint-2012.13-Block.2.3		
Milestone:	CCI-1.0.0	**Story Points:**	
Product Version:	*		

Description

Related issues:

Related to Infrastructure - Task #2613: Convert existing deny/denyFirst rules...	**Closed**	**2012-04-17**

History

#1 - 2012-04-05 22:54 - Ben Leinfelder

- Category set to Metacat

- Assignee set to Ben Leinfelder

Here's the scenario for sample guid 'knb-lter-sbc.40.2':
- It is an EML 2.1.0 document with embedded access control rules that use the "denyFirst" permOrder are added the a MN instance of Metacat.
- System Metadata for access control does not know about nor care about the permOrder that Metacat uses to record access control.
- CN-unm gets this object from the MN and adds it to it's store - the System Metadata is immediately replicated to the other CNs (CN-ucsb and CN-orc) via the shared Hazelcast System Metadata map and then the EML file is replicated to them via Metacat replication.
- Upon EML replication to these other CNs we already have System Metadata (including access control rules with permOrder=allowFirst) recorded.
- The EML is parsed on the replica CNs and the access control rules embedded in it (permOrder=denyFirst) conflict with those that already exist from the prior System Metadata replication (where permOrder=allowFirst because that was what we chose as the default for System Metadata in Metacat).
- This makes Metacat's xml_access table inconsistent because we end up with a mix of denyFirst and allowFirst for the same guid (in fact this should not even be allowed to be inserted in cases like this, but that's a separate issue).

#2 - 2012-04-09 22:45 - Ben Leinfelder

Now we won't write EML-defined access control rules to the DB during:
- D1 API insert/update
- Metacat replication

This should shield us from polluting the AccessPolicy defined in SystemMetadata when dealing with EML objects.

We also decided to convert all deny/denyFirst access rules to use the allow/allowFirst approach assuming there is no loss of the semantics of the access control block.

#3 - 2012-04-17 22:34 - Ben Leinfelder

- Status changed from New to Closed

This "solved" by ignoring EML access control rules in the way described.