Infrastructure - Story #2661

handling complicated identity-equivalencies for authorization

2012-04-28 14:31 - Rob Nahf

Status:	Closed	**Start date:**	2012-04-30
Priority:	Normal	**Due date:**	2012-10-06
Assignee:	Dave Vieglais	**% Done:**	100%
Category:	Authentication, Authorization	**Estimated time:**	0.00 hour
Target version:	Sprint-2012.39-Block.5.4		
Story Points:			

Description

Over time, dataone users will accumulate multiple institutional accounts that will be linked via equivalent-identity mappings, representable as a graph of identities linked with bidirectional equivalencies (each knows the other as an equivalent identity). These graphs are encapsulated and persisted in collection of Person objects the CN identityManager maintains, and need to be available to entities (CNs and MNs) responsible for authorizing a requestor's action.

Dataone needs to articulate both:
* the desired behavior of authorization with regards to the "transitivity" of permissions across these graphs (does group membership convey? does verified status? is either limited somehow? etc.)
* the responsibilities of each of the players involved in authorizing actions for users (the CNs, CILogon, and the MNs) to meet the desired behavior

Responsibilities for the CNs include:
* what Persons to return in the SubjectInfo, and are they guaranteed to be equivalent identities (can the authorizing agent trust that)?
* is that set of Persons the complete graph?

Responsibilities for CILogon:
* guarantee to return only unadulterated SubjectInfo from getSubjectInfo in the certificate?

For the authorizing entity:
* Does it need to recurse the set of Persons provided with the certificate SubjectInfo, or can it just include all of those subjects?
* Is it expected to call cn.listSubjects() to complete the graph if there are missing Persons from what was provided?
* Does it apply the appropriate restrictions on transitivity if that's the policy?

Subtasks:

Task # 2662: Decide transitivity behavior for verification, equivalent identities wrt/ ...	**Closed**
Task # 2663: what is returned by CN.getSubjectInfo()?	**Closed**
Task # 2665: Authorization consistency between language implementations	**Closed**
Task # 2666: create python tool for returning equivalent IDs from session	**Closed**
Task # 2667: create java tool for returning equivalent IDs from session	**Closed**
Task # 2668: create python tool for returning isAuthorized given AccessPolicy and list ...	**Closed**
Task # 2669: create java tool for returning isAuthorized given AccessPolicy and list of...	**Closed**
Task # 2670: check for character limit of customMessages within certificates	**Closed**

History

#1 - 2012-04-28 15:12 - Rob Nahf

- Category set to Authentication, Authorization

- Assignee set to Dave Vieglais

- Subject changed from handling complicated identity-equivalencies to handling complicated identity-equivalencies for authorization

#2 - 2012-04-28 18:14 - Rob Nahf

Three approaches were proposed with regards to what to put in the getSubjectInfo() response that's handed to CILogon and stuffed into a certificate:

put a "starter set" of Person objects in the SubjectInfo that represents the Person of the connecting subject and it's immediate (1st degree) equivalent

identities. This would be the full graph for the majority of users, and would save the cost of recursion for the cn in processing getSubjectInfo.

include the entire set of Person objects in the equivalant-identities graph.

Fully connect the graph so that approach 1 would equal approach 2, and save the recurring cost of recursion for an upfront hit to connect the graph.

Fully connecting the graph may be problematic to long term maintenance of identities - if an errant equivalency is made and needs to be retracted, it would be impossible to disconnect the two sub-graphs unless some sort of history of the datastore is maintained.

#3 - 2012-04-30 18:35 - Rob Nahf

adding tasks based on outcome of the standup discussion.

see also epad notes (the epad doesn't contain all decisions) : http://epad.dataone.org/Sprint-2012-17 towards the bottom

#4 - 2012-05-07 16:55 - Dave Vieglais

- Position deleted (2)

- Position set to 2

- Target version changed from Sprint-2012.17-Block.3.1 to Sprint-2012.19-Block.3.2

#5 - 2012-05-30 12:31 - Dave Vieglais

- Position set to 1

- Target version changed from Sprint-2012.19-Block.3.2 to Sprint-2012.21-Block.3.3

- Position changed from 1 to 450

- Position deleted (18)

#6 - 2012-06-12 16:31 - Rob Nahf

- Target version changed from Sprint-2012.21-Block.3.3 to Sprint-2012.23-Block.3.4

- Position set to 2

- Position deleted (452)

#7 - 2012-06-22 17:52 - Rob Nahf

- Status changed from New to In Progress

- Target version changed from Sprint-2012.23-Block.3.4 to Sprint-2012.25-Block.4.1

#8 - 2012-07-09 13:38 - Dave Vieglais

- Target version changed from Sprint-2012.25-Block.4.1 to Sprint-2012.33-Block.5.1

- Milestone changed from CCI-1.0.0 to CCI-1.0.4

#9 - 2012-08-31 13:52 - Dave Vieglais

- Position set to 1

- Position deleted (21)

- Target version changed from Sprint-2012.33-Block.5.1 to Sprint-2012.35-Block.5.2

#10 - 2012-09-05 18:08 - Dave Vieglais

- Milestone changed from CCI-1.0.4 to CCI-1.0.5

#11 - 2012-10-03 15:40 - Chris Jones

- Milestone changed from CCI-1.0.5 to CCI-1.1

- Target version changed from Sprint-2012.35-Block.5.2 to Sprint-2012.39-Block.5.4

- Due date set to 2012-10-06

#12 - 2012-10-08 16:53 - Dave Vieglais

- Status changed from In Progress to Closed

Work on this topic appears to be complete.