2015-07-22 - WebAccessControl Authorization Delegate Planning Meeting

Time/Place

- Time: 3:00pm Eastern Daylight Time US (UTC-4)
- Call-in:
  - U.S.A/Canada toll free: 866-740-1260, participant code: 2257295
  - International toll free: http://www.readytalk.com/intl

Attendees

- David Wilcox
- Andrew Woods
- Nick Ruest
- Joshua Westgard
- Stefano Cossu
- Ben Wallberg
- Peter Eichman
- Karen Estlund

Agenda

1. Review WebAC fundamentals
2. Establish minimum Phase1 scope/use-cases
   a. Allow admin agent to always have full access to resources and ACLs
   b. Allow admin agent to CRUD ACLs
   c. Allow admin agent to assign ACLs to resources
   d. Allow a specific agent to READ a resource
   e. Allow a specific agent to READ and WRITE a resource
   f. Allow a specific agent to CREATE a resource, but not update it
   g. Allow a specific agent to assign an ACL
   h. Allow a class of agent to do the above (d - g)
   i. Allow a specific agent to do the above over a class of resources (d - g)
   j. Allow a class of agent to do the above over a class of resources (d - g)
   k. When access is denied return a 403 and a body (or link header) with cause
3. Reconfirm commitments
4. Schedule initial two sprints
5. Address questions (can also happen offline)
   a. ACL resource is its own ACL?
   b. What is the algorithm for finding an ACL on a resource?
      i. if is ACL (rdf:type Authorization), use itself
      ii. if incoming reference from ACL, use it
      iii. else traverse up ldp:contains or pcdm:hasMember or custom? relationships
   c. How should conflicting policies be handled? e.g...
      i. (userA=WRITE, public=READ) => result of WRITE request from userA?
      ii. (userA=READ, groupB=WRITE) => result of WRITE request from userA, assuming userA is member of groupB?
6. Discuss Phase2 scope/use-cases
   a. Allow a request from a specific I.P. address (or range?) to do the above for a resource and a class of resources (2.d - g)
   b. Enforce authorization policy on a resource (or class of resources) based on that resource's association to a licenses (or tag)
   c. Enforce datetime sensitive authorization polices (i.e. embargos / leases)
   d. Allow authorization decisions based on nested ACLs (i.e. acl:include)
   e. Demonstrate pattern for enforcing the same authorization decisions as found in the repository in the context of Solr queries

Related Documents

- https://www.w3.org/wiki/WebAccessControl
- https://github.com/duraspace/pcdm/wiki#webacl
- Authorization Delegates
- http://www.w3.org/ns/auth/acl

Minutes

- WebAC Spec
  - What is the agent/agent class being authorized?
  - What is that agent being authorized against?
    - Specific resource or class of resources
    - RDF type

    - What mode is the agent in with respect to the resource
        - Read, Write, Append, Control
    - Web IDs
        - Not going to implement this
        - Probably not using URIs for agents at least at first
    - W3C LDP working group is working on access control requirements
        - Nothing here looks very divergent from what we're talking about
- MVP
    - Minimum set of initial requirements
    - Question: How can we allow a user to have read/write access to anything they themselves create
        - By default, after creating a resource, its creator has read/write access to that resource
        - This might not be desirable in all cases
    - Does every resource get a default ACL on creation?
        - Or should the resource inherit whatever ACL is determined by the algorithm?
        - General agreement that a default ACL can be created that defines owners permissions for objects they create
    - How do we define an owner?
        - Ontology includes namespace acl:owner
        - The owner may or may not be the creator
        - Do all resources have an owner? Do we need a default owner?
        - The concept of an owner is not necessary for the MVP (based on feedback from those on the call - please let us know if you disagree)
    - Do we need a separate permission for deleting a resource?
        - Currently this falls under Write permission
        - There is a use case for allowing a user to edit a resource without deleting it
            - Islandora and Hydra have these use cases
        - This would be a divergence from the spec
            - Delete and Update would be subclasses of Write
    - What is the class of an agent?
        - Does this map to a group?
        - For Islandora: Drupal role
        - URI to a list of agents in a particular class
            - Appealing but may not be practical initially
- Scheduling sprints (tentative)
    - Aug. 24
    - Sept. 28
- We will have another meeting at the same time next week