2013-08-30 FF AuthN-Z Mtg

Attendees

- Andrew Woods ⭐
- Greg Jansen
- Scott Prater
- A. Soroka
- Jonathan Green

General

- Indicates who took minutes - ⭐
- ReadyTalk:
  - U.S.A/Canada toll free: 866-740-1260, participant code: 2257295
  - International toll free:
    - http://www.readytalk.com/intl
    - Use the above link and input 2257295 and the country you are calling from to get your country's toll-free dial-in number
    - Once on the call, enter participant code 2257295
- IRC:
  - Join the #duraspace-ff chat room via Freenode Web IRC (enter a unique nick)
  - Or point your IRC client to #duraspace-ff on irc.freenode.net

Agenda

1. AuthN/Z Design and Use cases

Minutes

Topics

1. Transparency in security in REST interface
2. Clearly defined extension points, and prioritization
3. We are not securing URLs, we are securing objects/datastreams... the model
4. Short-term goals
5. Persistence should be unified?

Transparency in security in REST interface

1. In f3 it is difficult for Islandora to use security
2. Desire for something that is easy to use from the API level
3. Would like to not have two security layers
4. Is there an example that demonstrates these principles
   - ??
5. Suggestion that security may not be appropriate at the Fedora level
6. Interest in the ability to create a responsive UI
   - Must render quickly
   - Does not mislead the user by providing deadend buttons
7. Would like to introspect objects, or sets of content
   - What is available?
8. Enforcing security on f4 objects will require multiple calls within the app
9. We need to seriously consider performance
10. Reflecting on unix and DBs
    - You request a resource and get something back or not

- Action: Jonathan, create usecases

Clearly defined extension points, and prioritization

1. Want to avoid requiring users to learn new tools
2. Need unified, simple, consistent tooling
3. Get agreement on what frameworks will be used

We are not securing URLs, we are securing objects/datastreams... the model

1. If we secure the model, we are securing the URLs
2. Wisc is unable to support securing obfuscated URLs
3. The question is, how to do it efficiently

Short-term goals

1. PEP can be made effective
    - ensure correct response codes
    - early work on filtering search/triplestore results
    - mock PEP
2. First cut, store policies within f4
3. Goal
    - Three users, three rights r/w, r-o, no-access
4. Two questions
    a. What permissions does this principal have?
    b. What can I do?

Short-term Goals

1. Define a Policy Enforcement Point (PEP) interface (done)
2. Make sure Fedora REST calls honor PEP decisions with proper response codes (via mocked or stubbed PEP implementation) (in progress)
3. Create a simple ACL model for persistence on Fedora objects. (with read, write and acl-write/admin roles assigned to usernames)
4. Create and test a simple, non-XACML PEP that is driven by this ACL.

Help welcomed with the following:

- Creating an extension point for retrieving user details and marshaling these into security principals we can serialize as strings (for ACLs). (E.g. LDAP groups, named IP ranges (On Campus), age, whatever)
- Outline app developer needs with regards to no-op security checks
- Outline app developer security needs for UI development generally
- More thoughts on what we can do to support PEPs that delegate to external PDPs.

Persistence should be unified?

- Holding policies close to the content is a matter of durability

Actions

- Jonathan Green to document use cases for responsive UI scenarios
- Greg Jansen to email goals for remainder of Sprint B3