Audience & Goals

• Audience
  – System administrators who need manage users and groups
  – At centers, campuses, research groups

• Goals: At the end of this tutorial you will...
  – Create users
  – Create groups
  – Add users to groups
Agenda

• Background
• Creating user identity
• Create user home directory
• Create groups
• Add/remove user from groups
Background: Identity Mechanisms

• **WSI-Basic Security Profile**
  – Specifies where and how credentials are included in SOAP headers and how they are authenticated
  – May be multiple credentials in the SOAP header
    • Thus each web service call may present not just one identity, but a whole set of identities
      – A “Joe” identity, a “UVA-faculty” identity, a “group X member” identity

• **WS-Trust Secure Token Service**
  – A request arrives with a set of credentials
  – Based on those credentials and internal policy a new credential may be returned
    • For example, present “Joe” credential. Joe is a member of group “XSEDE”. Return “XSEDE” credential.

• **Calling Context Credentials**
  – In Genesis II, if X.509 public key of callee is known, then caller passed pre-delegated credentials to callee in calling context (in SOAP header)
Background: Logging In

- “Logging in” is a misnomer: Really just acquiring credentials to be used for access control and carried in SOAP header
- X.509 self-signed public/private keypairs are automatically generated for client sessions
  - Session data (such as keys) is stored in $GENII_USER_DIR
- Ways to acquire credentials
  - Create username/password token (**passwordLogin**)
  - Use existing X.509 on local disk (**keystoreLogin**)
    - This could include myProxy delegated certificate
  - Use WS-Trust STS (**IDPLogin**)
    - A “user” IDP that accepts username/password and returns a delegated and signed set of assertions.
    - A “group” IDP that checks for membership in the group and if the caller holds a credential that is authorized to be “in the group”, returns a delegated and signed set of assertions.
    - The assertions are delegated to the holder of the client session private key.
Background: Creating Users

• Anybody can create a “user”
  – Created user may not have authority to do much

• Newly created user can do nothing until identity is added to resource access control lists OR user is added to a group that already has privileges

• Adding a user to a group is simple if you control (have write access) of group
  • Simply give user read and execute permission on group
The User Creation Process

1. Create a user identity
2. Create home directory for user
3. Add user to desired groups
Creating a User Identity

• To create a new user:
  – Select user name and password
  – Select container to host user credential and name for user identity on container (service is X509AuthnPortType)
  – Command syntax
    ```
    create-user <container-service-path> <user-id>
    --login-name=<name>  --login-password=<passwd>
    [--valid-duration=<time-string>]
    ```
• Example:
  ```
  create-user
  /containers/myContainer/Services/X509AuthnPortType fritz
  --login-name=fritz  --login-password=th3c@t
  ```
• Link to user identity under /users (write permission in /users required)
  ```
  ln
  /containers/myContainer/Services/X509AuthnPortType/fritz
  /users/fritz
  ```
Creating a Home Directory for a User

• Create home directory for a user
  \texttt{mkdir /home/fritz}

• Give user rwx permissions on home directory
  \texttt{chmod /home/fritz +rwx /users/fritz}

• Check access control on directory (in GUI)
Creating a Group

- To create a new group
  - Choose container to host group

- Command syntax

  idp
  <container-path>/Services/X509AuthnPortType/<group-idp-name>

- Link to group identity under /groups (write permission in /groups required)

  ln
  <container-path>/Services/X509AuthnPortType/<group>/<group-path>
Adding a User to a Group

• To add user to a group
  – Must have write permission on group
  – Give user *execute* permission for group, then create link to group in user’s identity directory

```
chmod <group-path> +x <user-identity-path>
ln <group-path> <user-identity-path>/<group-name>
```
Remove User from a Group

• To remove user from a group
  – Must have write permission on group
  – Reverse creation steps: remove user’s *execute* permission for group, then unlink to group in user’s identity directory

• Command syntax

  ```bash
  chmod <group-path> -x <user-identity-path>
  unlink <group-path> <user-identity-path>/<group-name>
  ```

• Note: if user has already acquired a credential, it will be good until it times out (i.e. until valid duration expires)