Persistent Identifiers

SWITCH
Serving Swiss Universities

Chad La Joie
chad.lajoie@switch.ch

Identifier Properties

- Transient/Persistent - whether the ID lasts for an extended period of time
- Transparent/Opaque - whether the ID clearly identifies the user (e.g. email address)
- Scope - the security domain in which the ID exists
- Targeted - an ID meant for one (group) relying party(ies)
- Revokable - whether the ID can be revoked
- Resuable - whether the ID can, if revoked, be reused

Ways to Transmit an Identifier: Attribute

- The identifier is transmitted as a SAML attribute
  - current identifier attributes: swissEduPersonUniqueID, uid, swissEduPersonMatriculationNumber, employeeNumber, mail
- Pros
  - Very easy for SAML products to work with
  - Better expresses the concept of users idenitifed by bags of attributes not unqiue keys
- Cons
  - Identifiers from attributes may not be usable as the identifier in an attribute query
  - The SAML assertion may also carry a name identifier in addition to identifier-attributes which can lead to some confusion
  - Not especially good for expressing scoped identifiers, the syntax can become confusing

Ways to Transmit Identifiers: Name Identifier

- The identifier is transmitted as a SAML name identifier of some particular format
  - SAML 1: <NameIdentifier>, SAML 2: <NameID>
  - commonly used format: transient
- Pros
  - Can be used to make attribute queries
  - Usually clearer to people what exactly the value means
  - Automatically scoped
- Cons
  - Not all SAML products expose this information to applications

Two Standard Persistent Identifiers

- eduPersonTargetedID
  - An attribute that is persistent, opaque, targeted, scoped, and non-reusable
  - Two encoding styles:
    - [deprecated] as a scoped attribute
    - an attribute whose value is a SAML 2 name identifier
- Persistent Name Identifier
  - A name identifier of type *persistent*
    - properties: persistent, opaque, scoped, and non-reusable
  - Shibboleth's implementation also adds the targeted property

Persistent IDs in IdP 2

- The identifier is generated by one of two data connectors:
  - *ComputedID* hashes relying party ID, IdP ID, some other value (a salt, the value of an attribute, etc.)
    - IDs **are not** revokable
    - IDs that employ the value of another attribute change with that attribute
  - *StoredID* generates the ID as follows:
    - the **first** ID for a given relying uses the same method as *ComputedID*
    - subsequent IDs are UUIDs (type 5)
    - this approach allows a seamless transition from Shib 1.3 or 2.0 using *ComputedID*
    - information about the generated ID is stored in a database
    - IDs **are** revokable
- The identifier is then converted in to either an attribute or name identifier using the standard attribute defintion/ encoder model

Recommend Deployment Model

- Use the *StoredID* data connector
- Store the information in a database with high-availability support
- **Or** deploy a simple database (e.g. HSQL) on two or more of the IdP cluster nodes and use Sequoia

  http://community.continuent.com/community/sequoia