GSS Naming Extensions

Naming Extensions Recap

- Names (identity) as bag of attribute/values
  - Attribute source (authenticated vs. asserted)
  - Attribute criticality
  - Functions
    - get/set attr/value, list a name's attrs, composite name export/import, etc..

Issues Recap (generic)

- GSS_Inquire_name_attribute() is scary
- Negative ACL entries vs partial identity
- Attribute values as tagged blobs vs native types
- Concern that naming is too hard
- Anonymity/Pseudonymity
- Desire for credentials-related extensions

GSS_Inquire_name_attr()

- Allows for discovering if a given name attribute's values are useful for use as ACL entry subjects
  - w/o app a priori knowledge of the attr's semantics
    - Scares Sam
- Should have a NAME input parameter
  - For self-describing attrs
- Or should go away

Negative ACEs vs. Partial Identity

- Negative ACL entries work only if all information about a subject is known at ACL eval time
  - ID-as-bag-of-attribute-values allows for partial ID
  - So, partial ID and negative ACEs don't mix?
- Solutions?
  - Special attr to indicate that a set of attrs/vals is complete?
  - Do nothing, good luck to negative ACE users?
  - Mixed deny/allow ordering?

Tagged Blobs vs. Native Types

- Attributes are nice, but what do I do with octet string values?
- CAT WG had an I-D that dealt with attributes as {enum type id, union of native types}
  - Problem: new types, open types (think SAML)
- Proposal: keep blobs, add functions to convert to native types

Is Naming Too Hard?

- But original GSS names are too simple!
- Identity as-bag-of-attributes seems to be the way to go
  - See SAML

Anonymity

- Larry Zhu proposes anonymous Kerberos V principal names
- Sam Hartman proposes Kerberos V extensions for privacy
- GSS issues surrounding anonymity
  - Never mind pseudonymity!

Issues Recap (mech-specific)

- Kerberos V transited field is a set, not a list
  - Will fix in I-D
- PKIX text needs work