# GSS-API Naming Extensions

IETF66

nicolas.williams@sun.com

# GSS Naming Recap

- The GSS-API has an opaque type for representing “principals”
  - The entities that GSS-API mechanisms can authenticate
  - The abstract type for this is “NAME”
    - The C bindings type for this is gss_name_t
- NAME objects can be obtained from:
  - Importing human-readable strings
  - Inquiring on credentials
  - Authentication

# GSS Naming Recap

- Portable GSS-API applications then are expected to get by with name based authorization
  - NAME objects can be compared
  - Or they can be “exported” as an octet string that can be compared octet-wise
  - Either way we’re talking about essentially comparing things derived from human-readable names
- Inherent in this model: names can be canonical

# Desired Extensions Recap

- Support for portable authorization based on things other than human-readable names
  - Name-based authorization is problematic
  - We want attribute based authorization, where attribute can be:
    - Platform-specific internal identifiers (e.g., POSIX UIDs/GIDs, Windows SIDs)
    - Many other things (think SAML, PKIX cert extensions, like EKUs, and many other things besides)
- Support for mechanisms that lack canonical principal names

# draft-ietf-kitten-gssapi-naming-exts

- Primarily deals with adding functions that treat the opaque NAME type as a bag of attributes
  - Query what attributes are present/available, are they “authenticated,” etc...
  - Get attribute values
  - Set attribute values
  - Export composite names (for inter-process communication)

# draft-ietf-kitten-gssapi-naming-exts

- Provides general mappings for Kerberos V authorization-data and PKIX certificate extensions onto the new NAME object attributes concept
- Mappings of specific authorization-data and certificate extensions types to be done elsewhere
  - But for some things, like EKUs

# draft-ietf-kitten-gssapi-naming-exts

- Also
  - Display NAME in specified name type syntax (if possible)
  - Map NAME to platform-specific types

# Not Yet Addressed

- Credentials extensions
- Initiator identity selection

# Next Steps

- Reach consensus on this approach
- WGLC
- I-Ds for credentials and identity selection extensions?