PKIX Naming and the GSS-API
PKIX Naming and the GSS-API

- Generic GSS-API name types
  - “user” and “hostbased service name”
  - A simple mapping to PKIX would be good

- Many PKIX name types
  - ’Twould be good to have GSS name types for them

- Certificates can have many names, but the GSS-API requires a single canonical name
  - Name-based authz w/ GSS exported name tokens
  - MUST smooth this over
  - Existing certs should be usable w/ GSS mech
Solutions: Generic GSS name types

- User names can map onto rfc822Name SAN
  - About as good as krb5 mech's mapping of this NT
- Host-based service names can map to dNSName SAN + EKU for service name
  - Between anyExtendedKeyUsage and local policy we can make existing certs usable for 'host' and 'nfs' services with a PKIX-based GSS mech
Solutions: New GSS name types

• New GSS name types corresponding to all `GeneralName` choices:
  – OID prefix + choice tag number as last OID arc
  – Except for `otherName` choice – there the OID of the `OtherName` will do as a GSS name type OID

• Import/display syntax for these is as for the PKIX name types themselves
Solutions: Exported name tokens

- The canonical representation of any given PKIX name for use in exported GSS name tokens can be very simple:
  - **DER** encoding of the corresponding GeneralName

- **Except** for hostbased service names
  - If we use EKU as mapping for service
  - We could/should define a SAN for this

- **NOTE**: CAs need not support this SAN as it need not be present in certs, so no CA deployment issue should arise
Solutions: Many names → 1 Name

• Follow the IKEv2 model
  – Peers can assert the name they want to be seen as
  – Nodes verify that their peers are allowed the names that they assert

• So initiators send \{Certificate, GeneralName\} to acceptors and vice versa [hold comments, wait 1 slide]
  – The asserted name has to be in the cert

• And then the exported name token for a peer is the DER encoding of the peer's asserted name
Solutions: Many names → 1 Name

- "So initiators send \{Certificate, GeneralName\} to acceptors and vice versa"
  - Actually, \{Certificate, index of name\} would be much easier to process
    - 0→DN, 1→1\textsuperscript{st} SAN, 2→2\textsuperscript{nd} SAN, .., N→last SAN
  - This conflicts with use of EKU to represent the service component of hostbased service names
    - Because EKU is not part of PKIX naming
  - So, send \{Certificate, index of name, EKU OID\}
Solutions: Many names → 1 Name

• Need a good default for $\text{GSS\_C\_NO\_NAME/GSS\_C\_NO\_CREDENTIAL}$
  – Username type for initiator creds
    • rfc822Name
  – Hostbased NT for acceptors
    • dNSName SAN + EKU