Per-user host-based services; S4U2Proxy problems

Viktor Dukhovni, Two Sigma
<ietf-dane@dukhovni.org>

Nico Williams
<nico@cryptonector.org>

Per-user host-based services – PUHBSs

- Users sometimes need to run services
- They should get to (subject to policy)

1. Publish wild-card A/AAAA RRsets

- For each host in scope:

*.host.domain.example. IN A 192.0.2.1

2. Provision server credentials

- Modify existing server certificate and/or “keytab” protocol servers to detect and authorize requests for PUHBS credentials
  - For example, modify kpasswdd to permit host-based principals corresponding to real hosts to set the keys for hosts that are a one-label sub-domain of them
- So host.domain.example can get credentials for joeuser.host.domain.example and place them in a keytab file readable by joeuser.
- HTTP+SPNEGO requires service name = HTTP!

S4U2Proxy problems

- Impersonated credentials have impersonated principal's cname/crealm
  - If the impersonator's realm != impersonated's then the impersonator can't impersonate to services in the impersonated principal's home realm
    - Transit loop detected!
  - S4U2Proxy-unaware services can fail unsafe, not noticing that a client is being impersonated

S4U2Proxy solutions

- Don't use S4U2Proxy
- Instead use naming attributes [RFC6680]
- Define an pair of attributes for:
  - Impersonated principal's name (exported name token)
  - SASL-like authz-ID
- Put these in authorization data
- Impersonation aware acceptor apps will check these and apply local policy
  - KDCs can contribute policy ***just*** as with transit path policy