DRAFT-ABFAB-GSS-EAP

Sam Hartman

Painless Security, LLC

IETF 79

November 10, 2010

Current Status

- ➜ Initial draft sketches most issues
- ➜ Token formats, OID, key derivation intentionally missing
- ➜ One implementation based on initial draft

Missing Details to fill in

- OID, SASL name and UUID
- Mapping MSK to RFC 4121 key (discussed on list)
- Channel binding token
- Naming

Missing Parts Discovered

- AAA binding for acceptor names
- Format of error token
- Format of exported name
- Format of composite name
- Extension tokens

OID and Mechanism Naming

- One oid for each cipher suite supported
- SASL GS2 should have a mechanism name; Microsoft Negoex needs a GUID
- Recommendation: registry to track these values
- Question: what OID arc to use?

GSS CHANNEL BINDING

- ➜ Channel bindings are transported in a wrap-style token
- ➜ Large channel bindings are large on the net: no hashing
- ➜ Explicit support for the Kerberos behavior where null acceptor channel bindings means ignore. Good idea?
- ➜ Confirm authentication interruption cannot result in no CB verification.

Naming Proposal

Ignoring the hard parts, naming looks like:

- Initiators are named by NAIs
- Acceptors have:
  - Service name such as "imap"
  - Realm they belong to
  - Host name
  - Other stuff useful to specific service types

Naming: Not that Simple

- GSS requires acceptor names and initiator names in the same namespace
- Privacy: initiator names that only expose the realm, initiator names that expose nothing
- Initiators often don't know acceptor realms
- Decomposition for AAA transport

Naming: Where we Are

- Proposed name format combining acceptor and initiator names
- Need to specify privacy names (borrow from Kerberos?) and consider interactions with GSS anonymous flag
- Proposed AAA decomposition
- Proposed proxy behavior for unknown realms

Error Tokens

- error tokens reports authentication failures from acceptor to initiator
- Currently not integrity protected
- Major status, mechanism specific code and text string for developer debugging
- Proposed for inclusion

Proposed Extension Tokens

- ➜ Proposal to include an extension token in the last round trip for each party
- ➜ Channel binding is carried in this token

Input Needed

PROT_READY

- ➜ prot_ready permits an application to use security services before authentication is done; round-trip optimization
- ➜ Currently we do not support prot_ready
- ➜ Our ability to use it would be limited by EAP
- ➜ Should we add the complexity?

IANA Token Registry

- Several mechanisms are based on RFC 4121: this mechanism, PKU2U, IAkerb
- They tend to define token types
- We probably need an IANA registry for this

Discussion