Federating Non-Web Applications: A SAML Perspective

Scott Cantor
cantor.2@osu.edu
The Ohio State University /
Shibboleth Project
ECP: A Brief History

- Dates back to a Liberty Alliance profile for mobile phones
- Dormant during the Infocard “craze”*
- Server-side delegation in Shibboleth
- Implemented in Shibboleth for HTTP client usage

(*) term used loosely
ECP (V1.0)

- Original profile in SAML 2.0 standard
- Limitations inherited from browser SSO:
  - HTTP applications
  - bearer tokens and cookies
  - RP MITM detection limited to SSL (oops)
- Advantages:
  - Avoids HTML forms
  - IdP discovery left to client; can be pre-provisioned, based on “identity selectors”, etc.
ECP (V2.0)

• Compatible superset with negotiated extensions:
  • Channel binding concept borrowed from GSS-API
  • Holder of key subject confirmation requires client to prove possession of a key
  • Clarified conformance requirements for interoperability, especially client authentication methods
Implementation

- Shibboleth V2 SP, V2.3 IdP
  - Original profile
  - Authentication via container (HTTP Basic, TLS, SPNEGO, etc.)
  - Separate authentication setup for most IdPs
- Future releases
  - ECP V2.0
  - IdP V3 authentication changes
Client Requirements

- Simple XML processing
- Basic HTTP client including redirects, cookies
- Mechanism to provision and verify IdPs
- Desired form of user authentication

Non-requirements:
- HTML parsing or rendering
- Significant SAML processing
- Key management

ECP V2.0 additions:
- Optional use of private or secret keys
Wasn't this a non-web panel...?

- ECP message exchange pattern strongly resembles SASL/GSS-API mechanisms
- Significant overlap with Moonshot
  - Client UI
  - Service identification
  - Identity Provisioning / Selection
  - IdP Trust Management
  - Attribute release management