Putting Access Control in the Hands of the Users

@I2 Global Summit
9 April 2014
Program

- Introduction to User Managed Access (UMA)
- Use case 1 (Maciej)
- Use case 2 (Roland)
User Managed Access

• Kantara project

".. address the harmonization and interoperability challenges that exist between enterprise identity systems, Web 2.0 applications and services, and Web-based initiatives."

• UMA WG

".. to develop specs that let an individual control the authorization of data sharing and service access made between online services on the individual’s behalf"
OAuth is a three-entity protocol for securing API calls in a user context.

1.2. Protocol Flow

---(A)--- Authorization Request -> Resource Owner

<= (B)--- Authorization Grant ---

---(C)--- Authorization Grant --> Authorization Server

<= (D)----- Access Token -----+ 

---(E)----- Access Token -----> Resource Server

<= (F)--- Protected Resource ---

Figure 1: Abstract Protocol Flow

UMA’s original goal: apply privacy-by-design to OAuth data sharing

Outsources protection to a centralized “digital footprint control console”

The “user” in User-Managed Access (UMA)

Standardized APIs for privacy and “selective sharing”

Some guy not accounted for in OAuth…

Further reading: tinyurl.com/umapbd
Introduction
Register resources
Resource set description

• name
  • Human readable name

• icon_uri

• scopes
  • A bounded extent of access that is possible to perform on a resource set

• type
  • A string uniquely identifying the semantics of the resource set

• subsets
Register rules
Resource request

- Resource Server
- Authorization Server
- Client
Initial registration of requester
Permission registration
Permitted access
AM1.0 vs AM2.0

- Complex and feature-rich
- Usually proprietary
- Mobile/API-unfriendly
- Brittle deployment architecture
- Not agnostic to authn method
- Hard to source distributed policies
- Usually coarse-grained

- RESTful and simpler
- Standard interop baseline
- Mobile/API-friendly
- Just call authz endpoints vs. deploying an agent
- Agnostic to authn method and federation usage
- Flexible in policy expression and sourcing
- Leverages API’s “scope-grained authorization”
Use case 2

- SAML + UMA
The pieces

AS=UMA Authorization Server
RS=UMA Resource Server
UC=UMA client
SP=SAML2 Service provider
IdP=SAML2 Identity Provider

Alice
The configuration

AS=UMA Authorization Server
RS=UMA Resource Server
UC=UMA client
SP=SAML2 Service provider
IdP=SAML2 Identity Provider
1. AS->RS intro
2. Create rules

Alice

AS=UMA Authorization Server
3. Attribut release
demo
Next steps for the WG...and you

• Get involved!
  – Become an “UMAnitarian” (it’s free)
  – Participate in the interop and our implementation discussions
  – Follow and engage with @UMAWG on Twitter

• Current work:
  – Technical: claim profiling to allow claim-gathering using SAML, OpenID Connect, LDAP...
  – Business: Binding Obligations spec to tie “terms of authorization” to multi-party state changes