OAuth 2.0

Jan Hettich

LVRUG

1/29/2014

OAuth 2.0 Usage Patterns

- IETF RFC 6749: Proposed Standard
- Widely used for some familiar use cases
  - Sign-in via Twitter, Facebook, Meetup
  - Expose API's to 3rd party applications
  - Wrapped by OmniAuth
- How versatile for other use cases
  - Internal API's ?

Key Ideas

- Authorization framework for API's / HTTP
- Orchestration of authorization flow among entities with limited trust relationships
- Separation of Resource Server and Authorization Server roles
- Resource owner grants access to 3rd party applications to use API's that exposed protected resources
- Credentials presented once and replaced with opaque access tokens

Roles

- Resource Owner
    - entity that can grant access to a protected resource
- Resource Server
- Authorization Server
    - issues authorization grants and access tokens
- Client
    - web application
    - browser-based application (Ajax or SPA)
    - native application

Types of Authorization Grants

- Authorization Code ("server flow")
  - separate steps to obtain authorization and access tokens
- Implicit
  - browser-based client gets an access token in one step
- Resource Owner Password
  - high trust environment, legacy applications
- Client Credentials
  - based on client rather than resource owner directly

Applicability

- Broad scope supporting a wide range of use cases for securing both internal and external API's
- Core specification provides an architectural framework as well as specific HTTP bindings
- Some implementation issues are addressed in companion specifications that are under active development