OAuth 2.0 Dynamic Registration

Justin Richer for IETF87

Document History

- Originally, protocols defined their own standalone OAuth dynamic client registration process:
  - User Managed Access (UMA)
  - OpenID Connect (OIDC)
- Common elements abstracted and pulled into IETF draft
  - Along the way, we learned and uncovered a number of use cases
- IETF draft adopted by “parent” protocols
  - UMA has a direct normative reference
  - OIDC is compatible, will have normative reference when final

What's this for?

- How does the client talk to the authorization server?
  - Client software needs a **client identifer** to talk to an authorization server as per RFC6749
- How does the authorization server know about the client?
  - Authorization server needs to know **redirect URIs**, would also like to know things like **client name** and **client homepage** for UI

Use Cases

- Client software needs to register with an authorization server that the client has never talked to before
  - Client submits its metadata and gets a client identifier
- Same as above, but the authorization server requires an OAuth 2.0 token to call registration endpoint (this is explicitly allowed)
- Automated build system registers to obtain a client identifier, which is packaged into the deployed client

The spec defines:

- A method for a client to tell a server about its metadata (such as display name, homepage, redirect URIs, etc.)
- A method for the authorization server to issue a client id (and if applicable a client secret) to the client
  - Along with a copy of whatever information the client has been registered with
- A mechanism for the client's registered information to be managed over time (read/update/delete operations)
  - Uses RFC6750 Bearer Token to protect the endpoint

The spec does **not** define:

- Server discovery
  - Discovery needs its own robust specification
  - Discovery and registration may happen in concert but are orthogonal to each other
- Software metadata assurance
  - All metadata is self-asserted without some other mechanism for verification
  - Phil Hunt will submit a “software assertion” draft that can be a starting point to answering this question

Current draft status

Since IETF86 (Draft -08)

- Added internationalization support for human-readable client metadata fields
  - Thanks to Stephen for pointing out the need for this!
- Added discussions of client lifecycle and example use cases
  - Including discussion of client credential rotation
- Created an IANA registry for client authentication methods
  - Removed underdefined client authentication methods
- Added “software identifier” constructs to help auth servers tie together instances of clients
- Numerous editorial improvements
  - Much thanks to everyone who helped review!

Where we're at now

- 2.5 months into WGLC, I believe all known issues are addressed
  - Draft -14 (likely published a few minutes ago by the time you see this slide) is current status
- OpenID Connect and UMA are directly tracking with this draft
  - As are Blue Button+ and other OAuth-based protocol efforts
- We could use some help with:
  - Making security considerations section more robust
  - Implementation and interoperability testing

Moving forward

- I recommend we move Draft -14 out of WGLC and forward to IESG review now
- Phil Hunt's software assertions draft will extend this spec
- SCIM-based alternative registration (to be presented separately) will make its own way through IETF process
  - The data models are the same wherever possible
  - Specific goal of the SCIM-based registration spec should be to align the syntax and semantics as much as possible with the existing registration spec