Mike’s Fast Feds Analysis

5/31/17

Discovery Document

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  “identity_provider”:
    {“handshake_endpoint”: “https://idp.example.com/fastfed/handshake/start”},

  “service_provider”:
    {“handshake_endpoint”: “https://sp.example.com/fastfed/handshake/receive”,
     “auth_protocols_supported”: [“OIDC”,”SAML”]}
}
```

- This document is superfluous--it would be better to return one document with all the needed information, services, schema, and organization.
- The blocks are closely tied to authentication, but there are other services: SCIM, UMA, FIDO, Introspection, Token Revocation, ACE. Because these keys are hard coded, it is not possible to extend in a standard way.

IDP Trust Model

- Create an initial_access_token and nonce that will be given to the SP. These values will allow the SP to call an OAuth endpoint and attain an access_tokens and refresh_tokens, so they can interact with the IdP programmatically.

- Arguably, this degrades the security of both SAML and OpenID Connect. If you can update the keys with an access token, just attack the token.
- To control access to this federation management API, UMA's ticket based grant would allow more flexible policy management. The RS would always be the counterparty (i.e. if you are the IDP calling the SP's fastfed endpoints, the IDP would be the UMA client and the SP would be the UMA RS).

Over-reliance on SCIM

To remove this friction, FastFed requires a *lingua franca* for user attributes. SCIM is the chosen language.

- DISCLAIMER: Gluu has been an early and ardent supporter of SCIM. We implemented both version 1 and version 2. We love SCIM! It's fantastic! But...
- Not widely adopted--few of our customers use it (or even know about it).
- Not good at describing schema such as ACR, AMR, and Scopes
- Bad at describing things like FIDO U2F tokens, or devices that are associated with a person.
- Very foreign to most.. will turn off many users.
- Net-Net: publish SCIM info only to those who want it--don't force it on everyone..

Auto-Configuration API

- Are there just two endpoints?
- POST / PUT / GET / PATCH / DELETE ?
- Parameters not defined
- Errors not defined
- Security not well defined
- Needs sequence diagram--requests and responses are unclear
- Support for approval workflow? Especially when SP initiates the transaction?
- Missing key rotation sequence / workflow / security
- Ability to query metadata?
- Basically this is needs a ton of clarification...

Data Model

- Two entities defined: SP and IDP
- Very complex object model--lots of keys and internal data structures.
- Not referenceable or reusable--entities have no canonical id.
- “auth_protocols” ? Is that authentication or authorization? Also, how do you add additional values other then “SAML” or “OIDC” How about FIDO, UMA or SCIM?
- Assumes SAML metadata (despite saying in the spec not all SAML entities have it.
- Schema mapping section is very verbose, hard to read and SCIM centric--especially the use of variables like $user.userName--assumes a lot of processing--not plain strings.

Why Linked Data (JSON-LD) ?

- Allows resources to be connected
- Large schema already exists on https://schema.org
- Object model: a Class can have a subclass.
- Easy to use--looks like JSON, except for two keys
  - @id defines unique identifier
  - @context defines schema for that Class

Federation Support

- Trust management is one of the biggest challenges--which websites or IDPs do you actually trust without manual approval process
- OTTO supports SAML and OpenID Connect federation
- Answers questions about “software statements”--how does an OP know if it wants to let a client dynamically register, especially if it’s releasing PII
- Possibility of Fastfed being compliant with not just 1:1 federation scenarios--potential for re-use.

OTTO Standards

Core Vocabulary http://gluu.co/otto-vocab

OpenID Connect Vocabulary http://gluu.co/otto-openid

OTTO JSON-LD Context Files:
https://github.com/KantaraInitiative/wg-otto/tree/master/context

API http://gluu.co/otto-api

IN PROGRESS: SAML Vocabulary http://coming.soon

SCIM, FIDO, UMA Vocabulary http://coming.soon

Extensibility

- Yes, you can just add unsupported keys and JSON stuff--but not interoperable.
- Linked data @context URL enables automatic retrieval of JSON schema
- Re-use of base classes enables software tools to process data and understand what kind of thing it is.
- Standard way to express Enumerations (for values)

OTTO Research

We had numerous guest speakers, looked at many federation standards and software, to ensure we had a thorough understanding of what out there, including:

- SAML federation metadata
- OpenID Connect federation metadata
- MDQ Protocol: https://spaces.internet2.edu/display/InCFederation/Metadata+Query+Protocol
- JISC Federation API https://api.dev.ja.net/api.html
- Jagger: http://jagger.heanet.ie/
- Federation Registry: http://wiki.aaf.edu.au/federationregistry2/home

Other advantages of using OTTO

- Possibly could use the schema and entity endpoint as a starting point
- Much simpler top level data structure--easy for devs to understand
- Not so closely tied to IDP/SP--supports new services that might want to be FastFed'ed
- Formal schema definition
- Support for query / search
- Potential for Gluu Server integration--providing an early open source implementation