UMA telecon 2010-09-16

UMA telecon 2010-09-16

- Date and Time
- Agenda
- Attendees
- Minutes
  - New AI summary
  - Roll call
  - Approve minutes of 2010-08-26, 2010-09-02, 2010-09-09 meetings
  - Action item review
  - Legal subteam update
  - Updates from the wider UMA world
  - Paris F2F planning
  - Scenarios and use cases
  - Discuss defining outsourced validation of short-lived tokens
  - Discuss the "scope flow" required throughout the protocol
- Next Meetings

Date and Time

- WG telecon on Thursday, 16 Sep 2010, at 9-10:30am PT (time chart)
  - Skype line "C": +9900827042954214
  - US: +1-201-783-9022 | Room Code: 295-4214

Agenda

- Administrative
  - Roll call
  - Approve minutes of 2010-08-26, 2010-09-02, 2010-09-09 meetings
  - Action item review
  - Legal subteam update
  - Updates from the wider UMA world
  - Hueniverse opportunity
  - Paris F2F planning
  - KI spam filter news
- Scenarios and use cases
  - Review Mario's "distinctive aspects" classifications
  - Review the location scenario as a canonical scenario
- Protocol work
  - Discuss defining outsourced validation of short-lived tokens (new spec text to review?)
  - Discuss the "scope flow" required throughout the protocol
- AOB

Attendees

As of 02 Sept 2010, quorum is 6 of 10.

1. Adams, Trent
2. Fletcher, George
3. Hardjono, Thomas
4. Hoffmann, Mario
5. Holodnik, Tom
6. Machulak, Maciej
7. Maler, Eve
8. Moren, Lukasz
9. Scholz, Christian

Non-voting participants:

- Anna Ticktin
- Herve Ganem
- Mark Lizar
- Frank Wray

Regrets:

- Kevin Cox
- Domenico Catalano

Minutes

New AI summary
Roll call

Quorum was reached. (In fact, every single voting participant was in attendance.)

Frank is a new participant – welcome! He’s an IDM architect, doing consulting. He’s especially interested in cloud initiatives and user-centric identity management.

Approve minutes of 2010-08-26, 2010-09-02, 2010-09-09 meetings

Minutes of 2010-08-26, 2010-09-02, 2010-09-09 meetings APPROVED.

Action item review

<table>
<thead>
<tr>
<th>Date</th>
<th>Name</th>
<th>Opn</th>
<th>Action Item</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010-08-26-1</td>
<td>Mark</td>
<td>Opn</td>
<td>Write up CCTV scenario.</td>
<td>Closed</td>
</tr>
<tr>
<td>2010-08-26-4</td>
<td>Eve, Domenico</td>
<td>Opn</td>
<td>Write up generic Alice/Bob trust framework scenario.</td>
<td>In progress.</td>
</tr>
<tr>
<td>2010-08-26-5</td>
<td>Eve, Iain</td>
<td>Opn</td>
<td>Eve to ask Iain to write up personal data store scenario.</td>
<td>Iain has a plan for producing some new scenario material based on his Mydex deployment work.</td>
</tr>
<tr>
<td>2010-08-26-6</td>
<td>Eve</td>
<td>Opn</td>
<td>Ping Denise Tayloe of Privo to see if she has interest in taking custodian scenario forward.</td>
<td>Waiting till Mario does some edits first.</td>
</tr>
<tr>
<td>2010-08-26-8</td>
<td>Christian</td>
<td>OBE</td>
<td>Extract spec issues from 2010-08-26 minutes.</td>
<td>We have since superseded the Aug 26 discussion points.</td>
</tr>
<tr>
<td>2010-09-02-1</td>
<td>Mario</td>
<td>Opn</td>
<td>Categorize all existing scenarios by their distinctive aspects by 2010-09-09.</td>
<td>Thomas is stepping in to help.</td>
</tr>
</tbody>
</table>

Legal subteam update

Mark sent Sep 8 meeting notes to the list. He comments: There are happenings in the world of regulation that seem to impact how we think about UMA. The subteam is excited to continue.

Updates from the wider UMA world

George attended IIW East, where there was some discussion about OAuth’s charter going forward once the core spec is out. Hannes has floated a list of proposed/known extension work on the OAuth list to gauge interest. Thomas observes that changing the list of deliverables of a working group in IETF usually requires changing the charter. Eve reported on the OAuth list that we (with Maciej as ringleader) are interested in carrying the dynamic registration draft forward, and she notes that we have interest from several quarters on the native-app part of this spec.

Mario is currently attending an Assisted Living conference, and suggests that scenarios from this environment may be valuable. He is mentioning the idea of using UMA at that conference.

Maciej reports that his team has gotten a paper accepted at MW4SOC (Middleware for Service-Oriented Computing) on their UMAj framework. He’ll prepare a version of the paper for general distribution over the next week or so. Most of the components of the UMAj framework will be open-sourced within the next two weeks, and will be presented at Devoxx as well.

Mario describes a project he has just launched in his team for doing mobile-based calendar and location sharing. The project involves Google. Eve observes that mobile projects seem especially hot right now; we’ll likely have to keep an eye on OAuth’s redelegation work to make sure to incorporate it for mobile cases.

Eve suggests that we take advantage of Eran’s offer to guest-post at Hueniverse once UMAj is out.

Paris F2F planning

Eve has proposed the following rough agenda:

- Review and approval of submitted scenarios
- Model for flow of scope information in protocol
- Protocol security considerations
- OAuth digital signature progress and its impact on UMA

(See also Eran’s recent blog post on OAuth and signatures.)

Scenarios and use cases

Eve hopes that once we start to see more standardized APIs, hopefully we’ll also see standardized scopes. We’ll probably need to pretend that there exist (or will need to invent) standardized location-sharing API scopes.
The **CCTV scenario** is about a person who wants to get access to some CCTV video (on which the person appears) to flesh out a police report. The essence of it is "rights-based access", which UK and EU citizens have (but US citizens don't). This is a "service-to-person" sharing scenario since the data subject is the requesting party, and the service that is holding the recorded video is the host. Christian asked some questions in email that essentially probe whether there are Flickr-like person-to-other versions of the scenario.

**Discuss defining outsourced validation of short-lived tokens**

We discussed the following proposed spec text. Comments made during the meeting are in *italic*, whereas interstitial notes in the original are in [brackets].

**Step 3: Requester access resource**

3a. Requester accesses a resource using the access token

the Requester uses normal OAuth, using the access token received from the AM to make the request. It's doing it as described in http://tools.ietf.org/html/draft-ietf-oauth-v2-10#page-29

In case the request is invalid, e.g. because it's missing an Access Token, an "invalid_request" error as describe in section 5.2.1. in the OAuth specification will be returned.

3b. Verification of Requester Access Tokens

The Host verifies the requester access token by sending it to the token verification endpoint of the authorization manager. The authorization manager validates the Requester access token and returns the result to the Host. The authorization manager MUST validate the access token and ensure it has not expired and that the Host it is used for is the one it was requested for. *Herve asks: Can't you have an MITM attack if only server-authenticated HTTPS is used between the AM and host?* *George says that since the AM can't be spoofed, only the host's side could be spoofed, and many more things could go wrong than just stealing their token and using it instead. We have to make the simplifying assumption that the host isn't itself malicious, which makes the proposition more tractable.* *Eve comments that we're borrowing all of OAuth's faults and solutions, including any signature-based solution, here.*

In case of a positive result the authorization manager MUST return the list of scopes this access token was issued for. The host MUST validate if one of the scopes is sufficient to access the requested resource.

[Should we mention briefly here, and in more extended fashion in a Security Considerations section, how a short token life can offer lightweight/weak requester authentication/correlation? Security Considerations I'd say.] *George thinks the short-life security technique is not a bad idea, but it does present performance challenges if a resource needs to be accessed really frequently, by one or many clients. Given this, Eve guesses that this will create "upward pressure" for solving the requester authentication problem more thoroughly through digital signatures, as an option on top. Christian anticipates that this will then come under the category of "OAuth problem". Eran has recently argued that interoperable APIs will need signatures. Webfinger presents an opportunity to do the necessary discovery to make global PKI in OAuth work.*

Requests made to the host to the authorization manager MUST be an OAuth protected request itself using the host access token obtained in step 1 (xref here).

Requests made to the token validation endpoint of the authorization manager are made using a POST request containing the requester access token and the ip address of the request from the requester using the "application/x-www-form-urlencoded" format as defined by W3C.REC-html401-19991224*. [Do we need to say if/how the AM takes this IP address into account? We were figuring that it's just for logging, not necessarily for decision-making, probably good idea.]

*[ Might want to consider referencing XFF processing as the peer IP is often not good enough (i.e. just the netscaler in front of the app server). Note also, that based on experience with AIM client, it's not possible (even for short lived tokens) to rely on the IP the AM sees from the requester being the same IP the Host will see when the requester access the protected resource] This was George's note in the etherpad. So maybe we should add a non-normative note suggesting trapping the `X-Forwarded-For: header`. Example of a request to the token verification endpoint using the access token in the header:

```plaintext
POST /token_verification HTTP/1.1
Host: am.example.com
Authorization: OAuth vF9dft4qmT
Content-Type: application/x-www-form-urlencoded
token=sbjshbs/SSJHBSUSSJHVjsgvghsvshgsv&ipaddr=192.168.1.1
```

3c. Token Verification Response

After receiving and verifying a valid requester access token verification request the authorization manager constructs a response containing the registered scopes for this particular requester access token using a JSON document inside the body of an HTTP response using the 200 OK status code. The scopes are a list of strings and must match the scopes registered previously with the authorization manager by the host. *George suggested that we simply say that the scopes come from a list of scope strings previously registered by the host, and that we xref the previous section where that's described.* *We'd like to treat it as out-of-band for UMA to complain about incorrect scope values.*

Example:
If the requester access token is invalid, the AM returns a response to the host according to section 5.2.1. of the OAuth specification including the reason for the failure. The host then returns this response back to the requester. Since we're adding to the regular OAuth picture by allowing the host to outsource the judgment of an invalid_token or expired_token error, we need to explain more clearly that this is what's going on, and that the host should pass it through using normal OAuth error responses. Also, Maciej notes that we need to clearly distinguish the AM's error responses in complaining about the host access token vs. the requester access token. George doesn't like overloading HTTP messages with application error codes – but we'll come up with something. It gets tricky around what the user agent sees or doesn't see. AOL looks at HTTP as the transport mechanism over which the requests are being made, so you might get 200 OK but an app-level error. It's not super-RESTful, but solves this conundrum. Maciej argues that it counts as RESTful since the error isn't on an HTTP level. Absolved!

If the requester access token is valid but the registered scopes do not match the resource it was used for then the host MUST return an OAuth error response to the Requester as described in section 5.2.1. of the OAuth specification using the "insufficient_scope" error code. This is just "normal OAuth", and we're explaining how it applies to an UMA environment.

Discuss the "scope flow" required throughout the protocol

Eve suspects we need to invent a security solution that allows the host to hand the requester the scope that the requester should ask for over at the AM, without letting the requester tamper with the scope information. (Or is tampering really a worry?)

We discussed the nature of the split in responsibility between host and AM that comes with our current scope proposal. In regular OAuth, a host (really a resource server plus authorization server) has 100% of the responsibility for both "what's possible to do" – scopes – and "what's granted to be done" – policy decisions. We're moving the needle roughly towards the middle, with the host responsible for "what's possible to do" and the AM responsible for "what's granted to be done".

George suggests that, in an environment where short-lived tokens are being used, at unauthorized-access time (in Step 2) the host can simply ask the AM to give it some (potentially hashed or otherwise protected) scope string that represents a particular scope, hand it to the requester, and then the requester can turn around and hand it to the AM (still in Step 2). Then at potentially-authorized-access time (now in Step 3 but rapidly following Step 2!), the requester can present the token that is associated with the correct scope, and everything unrolls.

Thomas observes that this sounds like a Kerberos authenticator! Indeed it's something like a "scope authenticator". Also, in order to avoid doing signature stuff, we're adding more and more round trips. But George points out the value of such "naive" implementations in getting started, and sometimes the optimizations possible in such a system are still better than having to handle signatures.

Next Meetings

- WG telecon on Thursday, 23 Sep 2010, at 9-10:30am PT (time chart) on line C