User Managed Access Core Protocol

Logical testcases based on User-Managed Access (UMA) 1.0 Core Protocol[i]

Step 1: Authorizing user introduces host to AM

Preconditions

host has the role Oauth resource server

AM has the role Oauth Authorization Server

Authorizing User has the role of Oauth resource owner

Substeps

The following substeps are performed in order to achieve step1:

1. The host looks up the AM's metadata and learns about its API endpoints and supported formats.
2. If the host has not yet obtained an OAuth client identifier and optional secret from the AM (NOT STABLE YET??, CONFORMANCEISSUE), it registers with and binds to the AM dynamically, for example via description in Scholz, C.[ii].
3. The host obtains an access token (format not yet known,)from the AM with the authorizing user's consent, by following the OAuth 2.0 web server profile.
4. The host optionally registers scopes with the AM that are intended to be protected, via UMA resource registration[iii].

Substep1 The host looks up the AM's metadata

Step1Branch1

preconditions:
authorizing user types a URL in web form field and by clicking the SAVE-button this URL is saved in the hostmeta document (this step is configurable by implementor)
Description retrieval hostmeta document: see section 2 hostmeta in Hammer-Lahav, E.[iv]

Step1Substep1Branch1 Happy flow

IF	"AM.example.com" is the AM's domain in hostmeta doc.
AND	hostmeta document consists of elements as given in User-Managed Access (UMA) 1.0 Core Protocol
THEN	host create URL (ex.: "https://am.example.com/.well-known/host-meta")
AND	host submits GET-request (Format, see *Hammer-Lahav, E.*[v]) on URL

Step1Substep1Branch2 ***(still in progress??)***

IF "AM.example.com" is the AM's domain in hostmeta doc.

AND hostmeta document consists of elements NOT as given in User-Managed Access (UMA) 1.0 Core Protocol[vi]

THEN host DO NOT create URL ("https://am.example.com/.well-known/host-meta")

AND host DO NOT submit GET-request ((Format, see *Hammer-Lahav, E.*[vii]) on URL

Failing elements: *AM*

Substep2 host dynamicallly registers with AM

For reference see description in Scholz.

Step1SubStep2Branch1

*In progress??***, Conformance issue**

IF host already obtained client identifier and optional secret from AM previously

THEN NO dynamic registering host with AM necessary

ELSE

Step1SubStep2Branch2

Reqs: Scholz,p.5 prg. 3 requirements

Reg. Flow1: *Client Registration with pushed metadata*

see Scholz, p.9, prg. 6

IF host not obtained client identifier and optional secret from AM previously

THEN dynamic registering host with AM necessary

AND

IF CLIENT MUST do discovery Client Registration endpoint **(IN DEV., Conformance ISSUE)**

AND CLIENT sends metadata (JSON, reqs see par 6.1) to client registration endpoint

THEN Authorization Server checks data, verifies JSON Token issuer signature

Step1SubStep2Branch(Happy flow)

AND returns HTTP response with 200 status OK (Scholz, par. 6.2)

ELSE

Step1SubStep2Branch (error)

IF client registration request is invalid OR client registration request is unauthorized

THEN Authorization Server sends HTTP response with 400 status code (Scholz, par 6.3)

Failing element: client (registration request) ? requester

RegFlow2: *Client Registration with pushed URL and Pulled metadata*

see Scholz, p.12 prg. 7

Step1SubStep2step1: Registration request with URL

IF host not obtained client identifier and optional secret from AM previously

THEN dynamic registering host with AM necessary

AND

IF CLIENT MUST do discovery Client Registration endpoint **(IN DEV., CONFORMANCE ISSUE)**

AND CLIENT sends metadata URI(JSON, reqs see Scholz,prg. 7.1: Type and client_url) to client registration endpoint

THEN authorization server MUST check data, MAY perform **[RFC5785[viii]]**, [hostmeta] disc. Mech. (see Scholz,p.14 par. 7.2), further dev. Necessary)

Step1SubStep2branch1 (Happy flow):

AND returns HTTP response with 200 status OK

AND IF entity body contains 'client-ID' AND 'client_secret'

THEN HTTP " Cache-control" response header field value 'No-store'

Step1SubStep2branch2

Else IF entity body contains 'client-ID' (MUST)

THEN NO HTTP " Cache-control" response header field value 'No-store'

Else

Step1SubStep2branch3 (error)

IF client registration request is invalid OR client registration request is unauthorized

THEN authorization server sends HTTP response with 400 status code (Scholz,prg 7.4)

Else

Step1SubStep2branch4(error)

IF host-meta discovery NOT successfull

THEN authorization server sends HTTP response with 400 status code with error code 'hostmeta_error'(reqs Scholz, prg 7.4)

Failing element: can be authorization server or AM

Reg. Flow3: Native application *Client Registration*

see Scholz, p.17 prg 8

Reqs not detailed enough for testcase

SubStep 3: host obtains host access token

Preconditions:

Substep 1 ,2 of Core-Step 1 succeeded

host MUST use Oauth2 web server profile

host = Oauth Client

Authorizing user = Oauth-end user resource owner

AM = Oauth Authorization user

(Req. still in DEV.)

SubStep4: host registers resources to be protected

Conformance Issue: Shouldn't resourcing be required, and NOT optional?

Preconditions:

Substep 1 ,2,3 of Core-Step 1 succeeded

host has received acces token

Description

---

i https://github.com/mrtopf/UMA-Specifications; version-date: 4-12-2010

ii Scholz, C., "OAuth Dynamic Client Registration Protocol," 2010.

iii ?????
iv Hammer-Lahav, E., "Web host Metadata, : http://tools.ietf.org/html/draft-hammer-hostmeta-13" 2010.
v Hammer-Lahav, E., "Web host Metadata, : http://tools.ietf.org/html/draft-hammer-hostmeta-13" 2010.
vi https://github.com/mrtopf/UMA-Specifications; version-date: 4-12-2010; paragraph 2.1
vii Hammer-Lahav, E., "Web host Metadata, : http://tools.ietf.org/html/draft-hammer-hostmeta-13" 2010.
viii Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)," RFC 5785, April 2010

Once the host has received an access token, it MAY, immediately or at any time until user authorization is revoked, wield the token at the AM's host_resource_details_uri endpoint to POST an XRD (**How, conformance-issue??**) structure to the AM describing the authorizing user's resources currently managed at that host in order to assist the AM in letting the authorizing user configure policies specific to those resources.

Step1SubStep4Branch1:

host immediately after substep2 is finished, wields the token at AM's host_scope_reg_uri endpoint.

IF host has recieved accestoken FROM AM AND (host immediately after substep2 is finished, wields the token at AM's host_scope_reg_uri endpoint.)

THEN host sends XRD via POST-message to AM with description managed auth. User's sources at host **(DESCRIBED HOW?, conformance issue, see above)**

Step1SubStep4Branch2:

host wields the token at AM's host_scope_reg_uri endpoint, just before **(time limit??, conformance issue)**user authorization is revoked.

IF host has recieved accestoken FROM AM AND (host wields the token at AM's host_scope_reg_uri endpoint, just before (time limit??)user authorization is revoked.)

THEN host sends XRD via POST-message to AM with description managed auth. User's sources at host **(DESCRIBED HOW?, conformance issue)**

Step1SubStep4Branch3

host wields the token at AM's host_scope_reg_uri endpoint, at exact time user authorization is revoked.

IF host has recieved accestoken FROM AM AND (host wields the token at AM's host_scope_reg_uri endpoint, just before **(time limit??, conformance issue)**user authorization is revoked.)

THEN host sends XRD via POST-message to AM with description managed auth. User's sources at host **(DESCRIBED HOW?, conformance-issue)**

SubStep4Branch4 (error)

host wields the token at AM's host_scope_reg_uri endpoint, just after (timelimit??)user authorization is revoked.

IF host has recieved accestoken FROM AM AND (host wields the token at AM's

host_scope_reg_uri endpoint, just before **(time limit??, conformance issue)**user authorization is revoked.)

THEN host NOT sends XRD via POST-message to AM with description managed auth. User's sources at host **(errordescription DESCRIBED HOW, conformanceissue?)**

Failing element: Host, wielding was too late

IN DEV.