

Test Plan Questions and Answers

Test Plan Questions and Answers

The UMA WG chair met with one of the bounty program submitters who is working on an UMA test plan, to answer his questions. Here was the result.

Questions and answers

Q: What is the test assignment?

The goal in this conversation is to be as concrete as possible, within time constraints.

The overall goal of having a hosted validator is to test conformance and interop by implementations, against the protocol's testable assertions.

Since the spec documents to date are incomplete, we can't get 100% of the way in the time allotted for the bounty program.

So given all this, the suggestion is to develop a draft (likely incomplete) test plan that highlights wherever more information is needed from the WG.

A good model is the [SAML 2.0 test plan](#), but it had an advantage that UMA currently doesn't have: a conformance requirements document! This is something the group should put a priority on.

Q: Who gives the assignment to test?

Our assumption is that the WG is giving the assignment, and we will be as consensus-driven as possible (with the chair and vice-chair providing advice as necessary in between opportunities for the group to confer). So Eve and Maciej will be giving normative guidance, subject to group consensus in cases of controversy. We'll operate on an assumption of communications transparency; so, for example, we should copy the wg-uma list on answers.

We suspect that the SMART project has testing materials that haven't been shared yet. We will ask them to share this as soon as possible if they're able.

Q: Who takes the assignment to test?

The test plan is its own end, for now and for the current purposes. Ultimately when we get to the point of hosting validator software, the implementors of software that is intended to be UMA-compliant should be able to self-assess their compliance. Later, we may run F2F or virtual events where conformance and interoperability will be formally assessed by a third party.

There is some indication of interest by multiple parties in doing several independent implementations. We hope and anticipate that this validation activity will mutually reinforce this interest.

*Q: What kind of test do you want to do? A technical (system) or functional (user acceptance?) -> important for formation test cases and scope
What is the scope? What is out of scope? What are the acceptance criteria of test? When are the acceptance criteria met?*

The type of testing desired is conformance testing, which requires a software-independent view. We should stick to seeing this as testing whether implementations adhere to the "messaging contract" dictated by the spec.

What do we do in the case where UMA has statements other than MUSTs and MUST NOTs (SHOULDs, MAYs, OPTIONAL, etc.)? At a minimum we need to have a basic conformance section that states the scope of minimum conformance; this is a gating factor. If we also want to state other conformance levels or modes or profiles, we can do so. Note that the conformance section must make clear which non-core UMA specs would be included in minimum conformance.

The hope is for all of the user stories that are (a) of the "protocol" and "security" categories and (b) in the 1.0 backlog may potentially be sources of additional test cases, beyond the actual assertions in the protocol documents and the design principles and requirements document (these should ideally be turned into user stories, in fact). The test plan developer should analyze these sources of information to come up with unified test cases.

Q: What are possible risks associated with the UMA? Examples like security, performance etc. If a risk is not associated with UMA, this risk will not be tested: No risk, no test!

Noted. 😊 This is where the security-category user stories come into play (and we still need more of them), along with any security considerations in the protocol specs themselves. We're not too concerned with performance considerations for the moment; later functional testing might get into this (e.g., looking at denial-of-service considerations in implementations).

Q: Concerning the budget for executing the test (execution not part of submission!), how long will the people who give the test assignment allow the test to be executed? Or is this dependent on the result of the test plan? Under which conditions, laid upon by the people who assigned the test, should the test be executed, eg. functionality test plan, milestones, resources available, procedures?

We are going to defer much of this discussion for now, because it's slightly early days yet. So the draft of the test plan may not be able to provide these details.

Q: We could also discuss the events, not controllable by the test process, which must take place to run a successful test process, e.g., quality unit tests, quality test objects (validation software components), backup from development, delivery of test objects, reaction time development on defects. These things are very important for a well planned test execution, otherwise delays occur, resulting in a blow on the budget.

We don't yet have a very good issue-tracking system, although for now, we suspect we can just rely heavily on email for answering questions that arise. We can perhaps use Bugzilla or Jira (which we may be able to use easily through Kantara's Confluence licensing) for proper issue tracking.

Q: Who are the accepters of the test, who are the other stakeholders, next to the tester and the assigner?

Other major stakeholders are:

- The voting participants in the UMA WG
- Developers of software that is intended to be UMA-compliant

Since we're not doing user acceptance testing in this process (yet), end-users are not (yet) major stakeholders. Other kinds of testing and future testing might involve end-users more fully.

Q: Is there a list of all the available documentation on UMA: system development method, project documentation etc.?

We think these are the following relevant sources:

- Protocol specs, particularly security and conformance considerations sections) – see the Working Drafts page for a summary of these
 - Those developed by the UMA WG
 - Those referenced by the UMA specs but are developed elsewhere (which suggests that their own security and conformance considerations sections should be consulted)
- User Stories document
- Requirements document (ultimately to be more fully represented in the User Stories document)

There is a lot of other information available on the wiki, but it's largely non-normative and we won't count on it being found. It is possible for the test plan developer to "watch" (subscribe to) the relevant pages to track any changes to them.

Q: Are you acquainted with Requests for Change, here usable when wishes from acceptance for testing occur after formal agreement of the acceptance criteria; critical for a well defined tests cope and test risks.

We will consider this more fully when we get into a mode of formal issue tracking. This will help us figure out when a previously successful test gets invalidated by new changes.

Q: How do you want the UMA software be developed: agile or traditional, important for development regression tests and strategy.

The group would have no formal opinion on how implementers choose to develop their own software. However, we are using an analogue of agile methodology in developing the protocol itself. This may suggest a strategy to use in the test plan.

Since UMA is in draft 1.0 form and will remain so for quite some time (given that we intend to contribute it to IETF for final standardization), we need to develop a labeling system for major snapshots of the spec so that the test plan could be made unambiguous as to what it is testing.

Summary of action items

The following action items were suggested by the discussion recorded above.

Who	What
WG	Write a conformance section or document for the UMA protocol, indicating at least which protocol spec assertions are normatively required for minimum conformance and which specs are included in the conformance scope.
SMART project team	Share test plan and/or testing materials, if able.
UMA WG	Flesh out user stories from protocol spec and requirements document sources.