OVAL Board Meeting (7/11/2011)

Attendees

Eric Walker – IBM
Dave Waltermire – NIST
Kent Landfield – McAfee, Inc.
Steve Grubb – Red Hat
Chandrashekhar B – SecPod Technologies
Rob Hollis – ThreatGuard, Inc.
Chris Wood - Assuria

Jonathan Baker – MITRE
Matt Hansbury – MITRE

Meeting Summary

Welcome

The group was welcomed to the 2011 3rd quarter OVAL Board Meeting.

Status Report

A status update of the OVAL project was delivered. The following items were covered:

OVAL Language & Interpreter

The OVAL Team has spent significant effort over the past few months on the 5.10 OVAL Language release and the OVAL Language Specification, and these items will be discussed in length later on the call.

The Interpreter team has been adding some additional 5.9 tests to support Red Hat Enterprise Linux 5. The tests for that platform should now be mostly implemented and a new release of the Interpreter will be made available, including these tests, ahead of the 5.10 release.

Additionally the team has spent time adding assorted items as part of the 5.10 release, and an implementation of the PowerShell proposal, as presented by the Microsoft team at the Summer Developer Days in June, has been produced to help demonstrate and confirm its content.

OVAL Adoption

The Adoption program has had 1 additional vendor declaration over the past few months (jOVAL.org).

With the emphasis on the 5.10 release and the OVAL Specification, the Adoption Program has been limited in the amount of effort expended over the past few months.

OVAL Repository

The OVAL Repository’s definition count at the time of the call was 10,805 definitions.

Additionally, it was mentioned that the new submission processing tools were being used and refined, as evidenced by several hiccups over the past few months. The tools have made the processing a great deal easier, and automate a number of things that were manually done in the past. With several bug fixes in place, the tools now seem to be working very smoothly.

With the emphasis on the 5.10 release and the OVAL Specification, the OVAL Repository tools have also not received a large amount of attention over the past stretch.

OVAL Language Specification Status

The team provided an update on the status of the OVAL Language Specification next, highlighting the fact that a first draft of the document would be released on 7/12/2011. This draft should be considered a working draft, containing all of the overall structure that is planned for the document. A thorough editorial review is needed, and there will be several place holders in the document where we are actively working. Having said that, there are a number of places where strong community review and feedback is essential.

Of particular importance, the OVAL Team has tried to address a number of documentation gaps found in the Language Schema documents while creating the Specification. In doing so, the team did its best to document these gaps in a way that followed accepted use of the Language, however community feedback here is quite important.

The discussion then focused on two Specification topics:

Transition Plan

In order to go from a Schema-based authoritative source for OVAL's documentation to a Specification-based one, a transition plan is required. Jon Baker opened this discussion by sharing the OVAL Team's vision for this transition, by suggested that we release the Specification with OVAL 5.10 as an informative document only, and then to make it the normative source with the 5.11 release. This would make for a transition over the period of effectively two releases and allow the community to mature the document to a stable state.

Discussion

Following this description, the board was asked what they thought about the transition plan.

Dave Waltermire shared some thoughts from the NIST team's experience in SCAP, while developing the first draft of NIST IR 800-126. In general he felt that the best approach here is to create a targeted initial Specification that focused first on the well agreed-upon topics and features and immediately transition to using that Specification. From there incremental additions could be made to address more complex or contentious issues.

Additionally it was suggested that by using MUST, MAY, SHALL, etc., as described in RFC 2119, we could create a somewhat relaxed first release of the Specification that was Language complete, but that left the more complex or contentious under-documented items optional initially.

Jon Baker shared some reservations about immediately transitioning to the Specification as the authoritative source for the OVAL Language. He was concerned that not documenting the full Language could perpetuate some of the ambiguous or under-documented features of the Language, and could result in a wasted opportunity to improve the clarity of the Language. He did agree that finding the right balance here was important.

Conclusion

At the conclusion of this discussion, Jon Baker suggested that in the email announcing the first draft of the Specification, he would propose that the transition plan will involve no lag between versions, and that the Specification would immediately become the normative source for the OVAL Language. To help ease this transition, any ambiguous or unclear requirements would be stated in an optional way using MAY or SHOULD or equivalent terms.

Additionally the team will try to bring attention to those parts of the Specification where under-documented or unclear features have been more formally described. These sections, in particular, will need the attention of the community.

Component Schema Documentation

Next the OVAL Team described how the specification will separate the core of the OVAL Language from the Component Schemas that contain the specific tests for the different platforms. Jon Baker explained the team's current plan is to fully document all of the Core OVAL Language constructs (such as OVAL Common, OVAL Definitions, OVAL Variables, OVAL System Characteristics, OVAL Results, and OVAL Directives) within the OVAL Specification, but to then leave separate the documentation for the Component Schemas. The specification will describe, in an appendix, how to create a component schema and the official recognized component schemas, but will not include the component schemas directly.

Once the Specification for the Core Language is complete, the team will begin to address the creation of specification documents for all of the Component Schemas in the Language and deliver them separately.

Discussion

There was some discussion following this description in order to achieve clarity on the proposed approach. Specifically, questions were asked to clarify that the Component Specifications would be delivered separate from and following the OVAL Core Language Specification. It was also made clear that the Core Specification would have sections that gave detailed instructions on how the Core Language can and should be extended to Component Schemas.

Additionally, it was made clear that this separation would more easily allow 3rd parties to extend the Language to create their own Component Schemas.

In general the community reacted favorably to the idea of better detailing how to extend the OVAL Language and to separate out the Component Schemas.

Conclusion

Given the positive response to the approach that the OVAL Team has proposed, they will continue to move forward by keeping the separation between the Core OVAL Constructs/Specification and the Component Schemas and constructs.

Release 5.10 Update

With some constraints on time, the 5.10 update was given in summary. The release is coming along nicely and should be ready for an on time release of August 16, 2011.

While working through some of the 5.10 issues, the OVAL team has chosen to defer a number of items from the 5.10 release.

Some of the folks on the call showed interest in further discussing these deferred items, and therefore it was decided that an email would be sent, detailing those deferred items, with a plan to possibly follow up with a community conference call, if needed.

Experimental Schemas

For some time, the idea of setting up a "sandbox" where experimental Schema ideas can be tested and vetted, has been discussed.   At this point, the support for such a thing does exist, and so consideration is under way for it. Examples of the types of things that could have been included in an experimental schema from the past include the OVAL TPM Component Schema and the OVAL Results Schema.

The OVAL Team would like to work with the community to develop a process for creating and collaborating on experimental or niche topics. This process would include defining the concept and the policies and procedures for operating a community sandbox. This must include clear guidance for determining when a capability should be added to the OVAL Language or to a sandbox, and for migrating capabilities from a sandbox to the OVAL Language. . In order for this community sandbox to be well used and supported by the community, this sandbox concept must be well defined and documented.

Discussion

During the subsequent discussion, folks generally liked the idea, but wanted to make sure that there was a clear process in place.   Specifically, it was commented that we would not want to make the sandboxing of a test required ahead of inclusion within the official OVAL Language.   Additionally, it was made clear that we need to very clearly document how a proposal to the Language can be included directly into the official Language without an experimentation phase, and in what cases a test or Schema needs to be put through the experimentation phase.

Other commenters voiced a strong opinion that we as a group need to step back and consider the overall plan for the OVAL Language and where it is headed, before committing to these types of things. It was agreed that following the 5.10 release, the community should come together to decide how best to evolve the Language.

Conclusion

The concept of allowing experimentation within the OVAL Language and formalizing to some degree how such a process would work was looked upon favorably. However, there was also a sense that we as a group need to step back a bit and ensure that we have an overall, long term vision for evolving the Language. A phone call, following the 5.10 release was proposed to begin this process.