Security Automation Developer Days – Spring 2011

Table of Contents

Introduction

The Security Automation Developer Days - Spring 2011 conference was held at NIST in Gaithersburg, MD. There were ten presentations related to OVAL. The slides for these presentations are available on the OVAL website at http://oval.mitre.org/community/developer_days.html. The notes below cover the discussions that happened during and after the presentations, but do not provide thorough coverage of the presentations themselves. Please refer to the slides to gain a better understanding of the material that was presented.

Version 5.10 Goals

A timeline for the 5.10 release was presented. 5.10 is planned to be part of SCAP 1.2. Attendees were encouraged to get involved with the 5.10 development by participating in the various OVAL mailing lists.

Proposals

Several new tests are being added and need to be vetted by the community. Several items have been identified as candidates to be deprecated; need community feedback on those. Several of these proposed changes were presented in the following briefings.

Discussion

The OVAL team was thanked for this update. It was noted that the OVAL Board is focused on getting the next version out and not what the next version will be.

A couple of people mentioned a desire to have more face-to-face discussions concerning new features and goals for upcoming versions of the language. Jon noted that the OVAL team is working with NIST to set up more meetings.

Question: What is the status of the OVAL specification?
Response: The OVAL team has been working on one internally and that it should be coming out “soon”. The specification will be developed to align with version 5.10.

Conclusion

OVAL 5.10 is planned to be officially released at the end of July 2011. The community will have one more opportunity for a face-to-face discussion at OVAL Developer Days in June.

Integrating Asset Identification in OVAL

The fledgling Asset Identification (AI) specification provides a standard format for representing asset identification information. AI is likely to be incorporated into XCCDF and other SCAP standards.

Proposals

This presentation demonstrated one approach to incorporating the AI into the OVAL system characteristics schema. It also raised several questions regarding the nature and extent of AI integration with OVAL.

Discussion

It was quickly brought up that the various SCAP standards should coordinate in how they use AI. It was also mentioned that since the AI specification is so new, we should be careful in how it is integrated.

Several other possible use cases were brought up, including using AI in the generator section to identify the tool or user that created a definition, and using AI to detail user roles in system data.

The presented proposal incorporates AI's "computing-device" element. There was some discussion on what exactly constitutes a "computing device" and whether this is the right level of integration considering that OVAL works for things like switches and routers. The AI development team plans to make sure that the definition of "computing device" is clarified.

Conclusion

Incorporating AI in an optional manner was agreed to be a good idea, especially since AI will be used by other SCAP standards. However, the various standard groups (XCCDF, OVAL, ARF, etc.) should coordinate in how it is integrated and ensure that integration is at the correct level in the AI schema.

One Test

Currently, OVAL is structured as unique test, object, and state groupings, e.g. registry_test, registry_object, registry_state. It has been suggested that this current structure is bloated and adds unnecessary complexity to OVAL. All of the xxx_test elements share the same structure of containing an object reference and optional state references. Since all of the tests are so similar, it has been proposed that they be replaced by a single “test”. Adopting a single “test” element could make content easier to create and make OVAL more maintainable.

Proposals

This proposal would deprecate all of the existing xxx_test elements and replace them with a single “test” element. Two schema implementation options were discussed. If a single test is adopted, OVAL must still ensure that the object and state element types referenced by tests are consistent. Two Schematron approaches to this problem were presented.

Discussion

The discussion of this proposal was long and spirited.

The initial reaction was negative. The point that one test would make implementation easier was doubtful to some. Others pointed out that they have a large body of existing content, so deprecating all existing tests and rewriting them with the new one test would be prohibitively expensive. Also, debugging content would be harder. Others stated that they have XSLT that relies on the test names for processing. Another participant noted that the one test approach appears more elegant, but doesn’t add any features.

Questions were raised about this change’s effect on validation performance. Would it be slower or faster? Currently the impact is unknown because there is no significant content using a single test.

Others voiced that they like the concept, but they would like to see a test type attribute so you know what you are doing when parsing the XML.

Others brought up a previously discussed idea of doing away with states and moving their information into tests. It was pointed out that such a change would break backward compatibility and would have to wait until a major release; the current “one test” proposal can be done without breaking existing content.

On the issue of deprecating all existing tests, people were concerned that users of their tools would call and complain about deprecation warnings. It was pointed out that being deprecated doesn’t make the content invalid; it only indicates that someday the item may be removed from the language. Adding one test indicates an eventual move to removing the existing test structure and encourages use of the new construct. A suggestion was then made to not deprecate existing tests, but just add this one test as another element in the language.

Discussion continued regarding various Schematron approaches to ensure proper object and state references.

One participant said, “I don't think this is a bad idea, rather an idea ahead of its time.”

Conclusion

In the end, the community voiced concern about many perceived negative, deep impacts to OVAL with this change and no overwhelming positives to encourage its implementation.

OVAL & TPM Demo and Discussion

A demonstration of an extension of the OVAL Language to utilize the TPM was presented.

Proposals

Consider adding the proposed TPM component schema to OVAL to support the demonstrated capabilities.

Discussion

It was pointed out that TPM is not always enabled at the end points.

There was quite a bit of discussion of the TPM infrastructure. There was discussion about assuring the authenticity of OVAL Results and how TPM may be used in the signing process.

Conclusion

While the community seemed very interested in the capability, they quickly observed that this is a capability that is a very long way from operational deployment. The TPM extensions are in the current OVAL 5.10 draft and TPM support code will be posted to the OVAL reference interpreter’s SourceForge repository.

OVAL Test Content

The OVAL Test Content is a set of definitions that will eventually cover all tests and capabilities of the OVAL Language, similar to unit tests. We hope that the content will help:

- Developers — use to help guide the development of new tools.
- Users — use as part of your evaluation of competing products.
- Content Authors — use as a reference for writing new content.

Discussion

Thanks were given to the OVAL team for providing this content. Then there were some questions about the distribution format. A couple of people asked that the test content be made available as a single file. In response, the OVAL team announced that they were publishing a tool for merging OVAL Definitions into a single file as part of the OVAL Utilities project on SourceForge.net. This tool could be used to consolidate the test content into a single file and address the needs of the community.

Someone mentioned the desire for a content generating test harness to get some randomness for more dynamic test content.

variable_instance Attribute - Deprecate or Fix?

The documentation for the variable_instance is inconsistent in the OVAL schemas. There has been traffic on the OVAL mailing lists indicating people are having some trouble understanding variable_instance. This presentation explained the correct meaning and usage of variable_instance and solicited feedback from the audience on their usage of variable_instance.

Proposals

The documentation for variable_instance needs to be cleaned up. It was proposed that variable_instance could be deprecated if it is not being used.

Discussion

Several people indicated they use variable_instance. At least one of the tool vendors said their interpreter handles variable_instance.

There was some discussion concerning the interplay of XCCDF and variable_instance. There are ways to craft XCCDF files to avoid generating results with variable_instance attributes. There are also cases where the author wants to generate variable_instances.

A suggestion was made and agreed on to clarify the documentation around what gets assigned a certain variable_instance id - that is what set of values get a certain variable_instance id.

Conclusion

The documentation will be updated to more clearly reflect the usage of the variable_instance attribute.

MAEC and OVAL

Malware Attribute Enumeration and Characterization (MAEC) is a language for sharing structured information about malware. This presentation provided a brief overview of the relevant aspects of MAEC followed by a discussion of MAEC observables and current need for new OVAL Tests.

Proposals

This presentation introduced the connection between MAEC and OVAL in order to help the OVAL community better understand why there are open feature requests for mutex and file signature checking and prepare the community for other similar feature requests to support the MAEC observables use case.

Discussion

There were some questions about signature checking, would a white list vs. black list approach be used? Answer: that would be implementation dependent; MAEC does not enforce one or the other.

Concern was voiced that the tests requested and the usage model presented could send OVAL down the path of competing with anti-virus software – which would be bad. The intent is to only use OVAL for system state checking as it currently is designed for.

Conclusion

The OVAL team will plan on adding the new tests to the 5.10 Schema, per the proposal presented.

New Functions

Discuss proposals for a count and unique function in the oval-definitions-schema and consider other possible additions for version 5.10.

Proposals

The way functions work was reviewed, and then proposals and examples for the two new functions (count & unique) were presented. The opportunity to discuss other possible function additions was given.

Discussion

Some questions regarding the origin of these function requests came up. The OVAL team responded that there were two concrete requests via the OVAL Developer list.

No specific objections were raised to the new proposed functions, and no additional new functions were discussed.

Conclusion

The two new functions (count & unique) will be added to the 5.10 Schema.

Mask Attribute

This discussion focused on the mask attribute as it is defined in the oval-definitions-schema and its shortcomings. The implications of either fixing the mask attribute or simply deprecating it and working to remove it from the language were then considered.

Proposals

During the presentation, it was noted that either the mask attribute should be removed or fixed to properly handle all cases, as in some cases invalid results could be created with the way the attribute is currently defined.

Discussion

Vendors were asked whether the mask attribute was something that they made use of. Several vendors noted that they did not use the attribute, but at least one vendor representative said that he has a customer that would find the attribute useful.

Some folks noted that the more information about the data collection made available, the better, which the mask attribute helps hide.

Conclusion

Since the current implementation of the mask attribute is incomplete, and can cause invalid results, it should be fixed or removed. Since the usage of the attribute is limited, deprecation could be considered, but at least some part of the community finds value in it. More discussion needs to occur here.

Error Handling

The discussion began with a review of how error handing in variables is currently defined in the OVAL Language. The shortcomings and implications of this approach were discussed in greater detail. Once understood, the conversation focused on reviewing a proposal to address the problem as well as describing the implications associated with the proposal.

Proposals

The issues involving variable error handling were presented and the proposal to provide more clarity in the documentation was made. Additionally, a proposal to document and make use of flags throughout variables, functions, and components was made to provide more clarity as to how errors should be handled.

Discussion

The question was posed by Matt Wojcik whether folks used constant_variables or not. The consensus was that folks are using this construct.

The community agreed that casting was important, and that in the absence of specific guidelines or features, inconsistent results could occur.

Matt Wojcik asked if this issue was one of pure documentation or if additional features are required to fix the issue. Jon Baker believes that the documentation needs updating for certain, while there may or may not be a need for additional features (specifically how casting works, and early vs. late binding).

Conclusion

The group acknowledged that more documentation was required to provide clarity as to how errors should be handled in variables. Additionally, a proposal will be made to the discussion forum to make use of flags to better detail error handling in the Schema.