OVAL Board Meeting (4/11/2011)

Attendees

Eric Walker – IBM
Steven Piliero – Center for Internet Security (CIS)
Luis Nunez – CISCO
Aharon Chernin – DTCC
Morey Haber – EEYE
Dave Waltermire – NIST
Carl Banzhof – Rockport Systems
Kent Landfield – McAfee, Inc.
Steve Grubb – Red Hat
Chandrashekhar B – SecPod Technologies
Rob Hollis – ThreatGuard, Inc.
Alberto Bastos – Modulo

Jonathan Baker – MITRE
Matt Hansbury – MITRE
Danny Haynes – MITRE

Meeting Summary

Welcome

The group was welcomed to the 2011 2nd quarter OVAL Board Meeting.

Status Report

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

OVAL Repository

The OVAL Repository's definition count at the time of the call was 10,663 definitions. The Q1 2011 Top Contributor Awards were announced. The winners were Hewlett-Packard, SecPod Technologies, & Symantec Corporation.

Additionally, it was noted that the significant submission processing upgrades were approaching completion any day now.

OVAL Language/Interpreter

The OVAL Interpreter team has recently released a 64-bit version of the OVAL Interpreter. This build addresses the issues surrounding registry and file system redirection on 64-bit versions of Windows. The OVAL Interpreter team has also developed a 32-bit version of the OVAL Interpreter that is capable of collecting both 32-bit and 64-bit settings on 64-bit versions of Windows. This build of the OVAL Interpreter is available in the 5.8.3 branch. Additionally, the OVAL Interpreter now supports the linux-def:rpmverify_test and the linux-def:partition_test. The current focus of the team has been to fully support and vet the tests necessary for the successful implementation of the Red Hat Enterprise Linux 5 USGCB content. Lastly, the OVAL Interpreter team is working to support the release of RPMs for 64-bit versions of Red Hat Enterprise Linux 5.

OVAL Adoption

The OVAL Adoption Program currently has 28 organizations that have filled out the OVAL Adoption declarations, encompassing 41 products & services and there are several additional requests queued up.

OVAL Language Specification Status

The group was reminded that since the IT Security Automation Conference in September 2010, the OVAL team at MITRE has been working to develop a specification for the OVAL Language. The current status is that the team is working on the Data Model for the specification and is diligently working to complete an initial draft ahead of the 5.10 release (currently expected in late July).

Also, it was reiterated that once an initial draft of the OVAL Language Specification is complete, it will be sent out to the community for comment. The community review will initially be sent out to the OVAL Board, and following OVAL Board feedback, will be disseminated to the oval-developer-list as well.

Release 5.10 Update

In order to better share the motivations and plans for the 5.10 release, the team gave an update on where it is headed. For the release (currently planned for late July), the team is focused on driving the language support for both Red Hat Enterprise Linux as well as MacOS. It was also suggested that the team would like to be able to provide a full set of SCAP content to support the DoD Mac OS X 10.6 baseline, which is currently under development, in order to help the community see how content of that nature could be created.

[Rob Hollis] Can you comment on the macos-def:plist_test, specifically how can work be done on the Mac OS baseline without solidifying the macos-def:plist_test?

[Jon Baker] Around half of the baseline content requires the use of other tests, which allow work to be done on the baseline, before the macos-def:plist_test is clarified, if required. The team is currently working with Apple to determine if any changes are required.

[Rob Hollis] On Mac OS uname cannot be reliably used to detect the operating system version plists are required to accurately detect the OS version. CPE checks associated with an XCCDF Benchmark need to use the macos-def:plist_test in order to accurately identify the version of Mac OS running on a system.

At this point, some additional information was shared regarding the 64-bit Windows platform, and the question was asked to the group if there are any other anticipated issues with other 64-bit platforms. None were shared at this time.

Developer Day Follow Up

The team acknowledged that there remained some follow up from the Spring Security Automation Developer Days hosted by NIST. Follow up messages will be sent out to the oval-developer-list to resolve any issues that require further discussion. This will allow the community to provide additional feedback. The minutes for the OVAL sessions were sent out last week.

OVAL Session Slides

http://oval.mitre.org/community/docs/OVAL_Slides_Spring_2011_Dev_Days.zip

OVAL Session Minutes

http://oval.mitre.org/community/docs/OVAL_Spring_2011_Developer_Days_Minutes.pdf

Additional Comments/Questions

Following the formal update, the team asked the group for any additional comments or concerns. There were several comments mentioned:

[Aaron Chernin] What is the outcome from the one test discussions?

[Jon Baker] The minutes were posted last week, and given the mixed response from the audience, more discussion will be required.

[Kent Landfield] Are we happy with the direction of OVAL? Is it sufficient to simply add new tests to the language as needed or should be looking into other areas such as network device assessment?

[Jon Baker] The team has spent the past couple years focused on incrementally adding tests. The idea of network devices as a direction to push into has been discussed for some time. In general, it had not been felt that it had wide community support. What do folks think? What about un-authenticated checking? We have previously focused on authenticated checking for desktops and servers? Does anyone have thoughts on this?

[Aharon Chernin] We do unauthenticated scans using proprietary tools with proprietary checks and it's hard to reconcile the results from proprietary scans conducted with different tools. We would be interested in standardizing a format like OVAL for unauthenticated assessments. Additionally, we would be interested in the adding to OVAL the ability to allow more arbitrary commands to be run in definitions. Do vendors make use of proprietary ways to execute scripts?

[Kent Landfield] We use proprietary scripts to fill a gap in OVAL, in order to provide our customers with features that they request. We would prefer to have done this in OVAL, but it is not supported.

[Jon Baker] There is a concern that by allowing open scripts in the Language, people would default to using the supported scripting capabilities rather than the more traditional test structures that already exist in the Language, even where possible. This would move in a direction of less openness, not more. Does anyone share this concern?

[Kent Landfield] We do not, because our team is familiar with the existing tests and will make use of them. However, we can see Jon's concern. It seems that if a developer is forced to write one or more scripts to repeatedly provide the same function, a new OVAL test is in order.

[Jon Baker] That is a fair issue. Would it be worth considering adding some infrastructure to support this kind of 'any' test? This would allow OVAL to continue to be used for making the assertions and provide support for parameters to the scripts. We need a way to ensure that we are identifying the places where scripting is being used frequently enough as a solution to warrant the development of a specific test in the Language.

[Morey Haber] We should investigate network devices because infrastructure needs work.

[Jon Baker] A real example of this is with continuous monitoring, an agency recently did a tool comparison between two vulnerability scanners and found dramatically different results for the same set of hosts. Analyzing the results to understand what was actually checked by the tow products and why the products reporting conflicting results was very challenging because the checking logic was buried in proprietary formats. One reason to have OVAL, CVE, and the other standards is that they provide transparency into these problems. In a continuous monitoring architecture where vulnerability scan data is aggregated across the enterprise for rise scoring purposes it is crucial that the same issue are being checked and accurately reported on. OVAL provides a real benefit here, but, it is somewhat disappointing that we are not further along. With continuous monitoring, we have chance to work on this.

[Kent Landfield] One problem with continuous monitoring is that the federal government has many products deployed. Agencies will not buy single monolithic installations from one vendor. We need to work together to provide better interoperability between tools. Vendors need to work together to provide the government with the ability to integrate solutions for their environments.

[Dave Waltermire] This is why we have more need for integrating scripting into OVAL. Scripting provides mechanisms to expand out to new capabilities quicker because we don't need to go through the whole standards process. Allowing open scripting would not help with the parameters and results, but would allow for quicker expansion to new platforms and use cases.

[Kent Landfield] From my perspective, this isn't going to happen overnight whether it's content, data, or connection model.

[Stephen Piliero] I also see these issues while supporting Red Hat & MacOS. I would prefer to use OVAL tests, if available, but have resulted to other technologies for checking due to the lack of support for scripting.

[Jon Baker] An open scripting ability in OVAL would help with the macos-def:plist_test concerns brought up earlier. We would have already written content for Mac OS X that used scripts to check information stored in plists. However, it this flexibility would have come at the cost of some transparency, in that it could hide the need to alter an OVAL test or creating a new one. Currently when people need to check for something new in OVAL they ask for a new test. There is a risk that people would use a script and not ever ask for the new test.

[Kent Landfield] I believe a call for information would help alleviate that gap.

[Aharon Chernin] This open scripting test doesn’t necessarily have to take away that transparency.

[Dave Waltermire] There are two types of transparencies here. One is the transparency of being able to see and understand the content being executed. The other is how to know what capabilities are missing from the language. The latter can be addressed by communication within the community.

[????] We need to address the security concerns of an open scripting test in OVAL.

[Dave Waltermire] These concerns are already there in OVAL, with sensitive data being collected.

[Kent Landfield] These security issues should be taken care of at the connection level, not the data model level. We can layer security in architecture, but, it is not a data model issue.

[Dave Waltermire] We already have some scripting capability in OVAL with the ind-def:sql_test.

[Jon Baker] The win-def:wmi_test also offers the ability to change the system state.

[Dave Waltermire] This makes it all the more important for us to better support digital signatures to protect the data.

[Aharon Chernin] Thinking about one test, why not one test, one object, one state, and one variable? Each of these elements could be differentiated by a simple type attribute.

[Jon Baker] I think we are too far down the road to switch now. We have taken the approach differentiating tests, objects, states, and items based upon element names ad have several years of work invested in that model. A change now is probably not worth the effort.

[Jon Baker] We will hold a conference call in the next week or two with the community to discuss this ‘any’/open scripting test and its implications.

Action Items

- Set up a conference call, with the community, to discuss scripting and the execution of commands in OVAL.