Security Automation Developer Days

June 14-17, 2011

Table of Contents

Introduction

Security Automation Developer Days was held on June 14 - 17, 2011 at The MITRE Corporation in Bedford, MA. This event was the most recent chapter in an ongoing series of workshops, beginning in June of 2009 at MITRE, Bedford and February, 2010 at NIST in Gaithersburg, MD.

Ninety-two people registered for the event, and roughly 60 people were present each day of the workshop. Over the four days, twelve sessions were held and this document contains a comprehensive summary of each of those sessions.

As you prepare to review these minutes, the authors would once again remind you that the standards cannot continue to advance without ongoing discussion of the issues throughout the year. This is accomplished through dialogue in the email discussion lists. A complete list of these email discussion lists can be found here: http://measurablesecurity.mitre.org/participation/index.html . Please sign up for those lists that interest you.

What follows is a detailed summary of the discussions from the event.

Attendee List

BAE	-	Dexter Kirkpatrick
Boeing	-	Kevin Auwae Sam Miles
Booz Allen Hamilton	-	Melody Berry Adam Halbardier Timothy Harrison
BTC,Inc	-	Bhesh Krishnappa
CIS	-	Steven Piliero
CygnaCom Solutions	-	Kelvin Desplanque
Dept of State	-	Wei Chen George Madzy George Moore Vu Nguyen
DISA	-	Brady Alleman Jim Govekar Jason Mackanick Pete Schmidt Joseph Wolfkiel
DTCC	-	Aharon Chernin
EMC/RSA	-	Matthew Coles
ESC FAB-T	-	Michael Rioux
EWA-Canada	-	Erin Connor
G2, Inc.	-	Melanie Cook Kurt Dillard Matthew Kerr Shane Shaffer Greg Witte
GDIT	-	Todd Messner
General Dynamics	-	Jeremy Wyant
Global Business Connection, Ltd	-	Richard Losier
HP Enterprise Services	-	Chris Johnson
IBM	-	Al Cooley
ManTech	-	Brian Hill Karl Brower

McAfee	-	Kent Landfield
	-	Dick Whitehurst
Microsoft	-	Jeffrey Snover
		Xiaoxi (Michael) Tan
MITRE	-	Jonathan Baker
		Steve Boczenowski
		Stephen Boyle
		Eitan Check
		Brant Cheikes
		Mark Davidson
		Matt Hansbury
		Danny Haynes
		Jasen Jacobsen
		Ivan Kirillov
		Robert Martin
		Gerry McGuire
		Daniel Mitchell
		Linda Morrison
		Lisa Nordman
		Emily Reid
		Joseph Sain
		Jon Salwen
		Robert Schmeichel
		Charles Schmidt
		Ingrid Skoog
		Bryan Worrell
		John Wunder
		Margie Zuk
Modulo	-	Marlon Gaspar
NASA	-	Gary Gapinski
		Richard Haas
NIST	-	Harold Booth
		Murugiah Souppaya
		David Waltermire
NSA	-	Mike Kinney
		Dan Schmidt
Planet Payment	-	Slim Zouaoui
Polycom	-	Erik Cockrell
Rapid7	-	Ryan Poppa
		Harish Srinivas
Red Hat	-	Jack Rieden
Rockport Systems	-	Carl Banzhof

Scarfone Cybersecurity	-	Karen Scarfone
SecPod Technologies	-	Chandrashekhar Basavanna Chandan Somashekaraswamy
SEI	-	Jeff Davenport Mary Popeck
Smithsonian Institution	-	Bruce Daniels
PAWAR	-	Richard Kelly Jack Vander Pol
Symantec	-	Dragos Prisaca Josh Turpin
Tenable Security	-	Jim Hanson Mehul Revankar
ThreatGuard, Inc.	-	Robert Hollis
Tripwire	-	Benjamin Jansen Adam Montville
USAF	-	Bob Holby
Varen Technologies (DoD)	-	Jim Ronayne

Tuesday June 14th

XCCDF 1.2 Community Review

Background and Introduction

The primary session was a review of the significant modifications made to the XCCDF specification and schema over the course of the last two years as a result of group discussion and consensus. This session was intended to provide a forum for participants to raise any final concerns ahead of submission of these materials as official NIST draft specifications.

Complex Values

The first topic covered support for complex values. This is a change to support the use of lists as well as arbitrary XML structures in XCCDF Values so they could be exported to checking systems. Previously, XCCDF only supported exporting of singleton values.

A number of people immediately raised questions about the use cases for supporting arbitrary export of XML structures. While OVAL accepts import of lists of items and could therefore make immediate use of XCCDF's new ability to export lists, the current version of OVAL does not have any structures that could be utilized by the XML export capability as outlined in the XCCDF proposal. (It was noted, however, that a future version of OVAL may include a "record" structure, which could be applicable to this kind of export.) Many people argued that, given the lack of a present use case, this aspect of complex values should be removed from the next version of the XCCDF specification. (The ability to export lists would still be added in the new XCCDF specification.) It was further noted that URI-encoded records could be encoded in a simple string, allowing something close to the proposed capability to be supported by current structures.

Dave Waltermire countered that, while there might not be a current use case, the proposed capability was a good way to make XCCDF more extensible. A number of cases, such as router policies or access control records, were noted as places where complex records would be necessary to describe a target value. In response to concerns about adding an implementation burden to tool developers, Dave noted that 800-126 would not require implementation of this feature in tools and would prohibit its use in content to ensure there was no incompatibility between vendors.

A concern was raised by Joe Wolfkiel that, if the feature was included in the specification but not all vendors implemented it, tools needed to fail gracefully if they didn't handle the new Value structures. He noted examples where lack of support for a given check system simply broke tools.

A vote was taken with three alternatives:

1) Keep support for arbitrary XML Value export structures in the new spec as a mandatory feature (but which 800-126 was then likely to countermand)

2) Keep support for arbitrary XML Value export structures in the new spec as an optional feature with a detailed description of how tools should fail gracefully if they did not support content that used this capability
3) Removal of support for arbitrary XML Value export structures from the new spec

The attendees were evenly split between options 1 and 3, with option 2 receiving only a few votes. Discussion closed without a final decision on which choice to take.

Finally, it was noted by Gary Gapinsky that the proposed schema only supported lists of at least one element. He stated that there may be a case where one wishes to export an empty list, as distinct from no list at all, and suggested allowing lists to contain zero elements. A poll of the audience agreed that this would be a beneficial modification.

Conclusion:

- Allow 0-length lists in the new schema structures
- The decision as to whether to support arbitrary XML Value export remains unresolved

Check Negation

This feature allows the result of a checking system to be negated before the XCCDF result for the corresponding Rule is computed. It was noted that this would allow OVAL inventory checks to be used within compliance policies as both positive and negative components.

Jim Ronayne supported the change, but noted that because OVAL also has multiple levels at which checks can be negated conventions should be spelled out as to the proper place to signal result negations for various situations.

Conclusion:

- The feature is included without additional changes
- Conventions should be set out on proper authoring of XCCDF and check content, particularly with regard as to when negations should be expressed

CPE 2.3

The proposed XCCDF specification uses CPE 2.3 rather than the older CPE 2.0.

Joe Wolfkiel noted that CPE 2.3 supports multiple bindings. (Bindings represent specific formats of a CPE pattern or name.) Currently the XCCDF specification simply references CPE 2.3 but does not identify any particular binding. It was agreed that it would be beneficial for there to be an agreement as to which binding to use so that a common expression of a CPE name was used across content. The group decided to defer discussion about the specific binding to employ in XCCDF until after the CPE presentation, since that would provide some necessary context for this discussion. Unfortunately, the CPE discussion did not leave enough time for this conversation to take place so this remains unresolved.

Conclusion:

- The community needs to agree as to which CPE 2.3 binding should be used in XCCDF content

Metadata

The proposed revision to XCCDF greatly expands the ability to include metadata, removing limits on which metadata schemas can be utilized and adding metadata fields in all major XCCDF structures.

Questions were raised about the lack of any explicit structure in the metadata information. Specifically, it was noted that, without some standardization, if a group wished to annotate their content (for example, a hospital annotating with information relevant to their regulatory needs), then any ability for tools to act upon these annotations would need to be worked out on a one-to-one basis with a given vendor. It was further noted that some metadata would be of common interest to all users, but without standardization, each author would end up with their own, individual encoding it in the metadata. Finally, it was noted that many data repositories can easily support metadata in simple name-value pairs, but when the metadata is expressed in a complex XML structure, repositories can have trouble recording this information in a useful way. The lack of any restrictions or conventions on the metadata field would not prevent the latter situation.

While not disagreeing with the previous points, some community members questioned whether the presence of commonly utilized fields might better be handled by direct inclusion in XCCDF itself rather than standardization in the metadata. Others questioned whether metadata standardization might be better handled through efforts outside of XCCDF. Ultimately, however, it was agreed that there was not enough time to establish standard procedures for metadata before the next release of XCCDF needed to be ready and that the proposal as it currently appears in the specification would meet community needs in the meantime.

Conclusion:

- The feature will be included without additional changes
- An action item is noted to explore the standardization of metadata information that might be common across communities

Check-Import

This proposal involves a clarification and enhancement to the existing check-import capability. It was noted that this feature is still not "complete" in that it requires some other entity to define the mapping between the names that appear in the check-import element and the checking structures that get returned. (In the same way, the "check" capabilities in XCCDF are also not complete for the same reasons, but this does not inhibit their use because 800-126 defines this mapping.) The community was polled as to whether this would be a problem but the community was satisfied leaving this mapping to 800-126.

Conclusion:

- The feature will be included without additional changes
- The maintainers of 800-126 will be contacted regarding the inclusion of a mapping between names and structures as used by the check-import element

Multi-check

The multi-check feature allows nameless check-content-refs in a Rule (that is, check-content-ref statements that identify a file, but do not identify any particular structure in that file) to be reported individually rather than being combined into a single result. This would allow entities such as OVAL patch files to generate one result per patch rather than a single result that failed if any patch was missing.

Gary Gapinsky asked why the multi-check property applied to Rules rather than checks. The answer was that it had been placed at the Rule level simply because that was the level at which the similar "multiple" property operated. It was noted that, if the multi-check property was moved to the check level, not only would it allow greater granularity of control, but that tailoring actions could then select whether nameless checks were combined or reported individually by selecting the appropriate. The community agreed that this change made sense and that the multi-check property would be associated with checks.

Multiple members of the community expressed concern about multi-check's effect on scoring. Specifically, by breaking out a single Rule into multiple rule-results, that Rule would gain a weight far beyond what it would have had if it was represented by a single combined result. It was noted that, in some scoring models, an encapsulating Group could have a governing effect on that Rule, but this would not be the case with any of the flat scoring models. In the end, the target score would need to be calculated based on a combination of the Profile selected and the number of individual checks run by nameless check references.

It was also noted that, in OVAL, one could have inventory definitions in a file whose only purpose are to be extended by other definitions and which do not, alone, indicate compliance or non-compliance. XCCDF would make no distinction between these inventory checks and other vulnerability, compliance, or patch checks. As such, if multi-check were true these inventory definitions would be given their own rule-result and contribute to a final score, and if multi-check were false, the lack of presence of a particular piece of software would lead to the general failure of the Rule as a whole (even if the lack of this software was a desired characteristic of the referencing definitions).

A vote was taken as to whether this feature should be deferred until the scoring implications could be better worked out or whether the feature should be included and the scoring worked out afterwards. The community was evenly split on this and no consensus was reached at the end of discussions.

Conclusion:

- The multi-check field will be moved to the check level from the Rule level
- The decision as to whether to include or remove this feature from the next version of XCCDF remains unresolved

XCCDF Document

It was announced to the community that a structural revision of the XCCDF specification is being undertaken by Karen Scarfone of NIST. This revision is intended to consolidate discussions of various features, provide better organization of the document as a whole, and to improve readability.

Version

It was noted that there was some question as to what version the new revision of XCCDF should be tagged with. Specifically, it was noted that if XCCDF was given a version of 1.2, that the schema's namespace would need to change, leading to a lack of transparent backwards compatibility between documents. Some had argued that changing to version 1.2 would be beneficial, not only because it would force an explicit selection of one schema over another for any given piece of content, but because, given that backwards compatibility was already broken, it would allow for additional clean-up of the XCCDF schema.

Due to imminent end of the XCCDF session there was limited discussion on this and a vote was taken with 3 options:

1) Use version 1.1.5
2) Use version 1.2, but otherwise make no changes to the specification
3) Use version 1.2 and make additional, non-backwards compatible changes to clean up the XCCDF schema

In the vote, option 3 had the overwhelming majority.

It was noted that an XSL stylesheets could be created to convert between old and new versions of XCCDF. It was requested that such stylesheets be part of the release.

Finally, it was requested that a document outlining the versioning policy for XCCDF be drafted to provide better guidance on this question in the future.

Conclusion:

- The new version of XCCDF will be XCCDF 1.2
- A list of clean-up actions will be drafted and sent to the community for review
- An XSL stylesheet to convert between XCCDF 1.1.4 and XCCDF 1.2 will be created
- A versioning guide for XCCDF will be drafted

Break-out sessions

Multiple break-out sessions were held following the XCCDF session. All of these sessions focused on the capability originally introduced to the community as "external Profiles".

Break-out session 1 (June 14, noon)

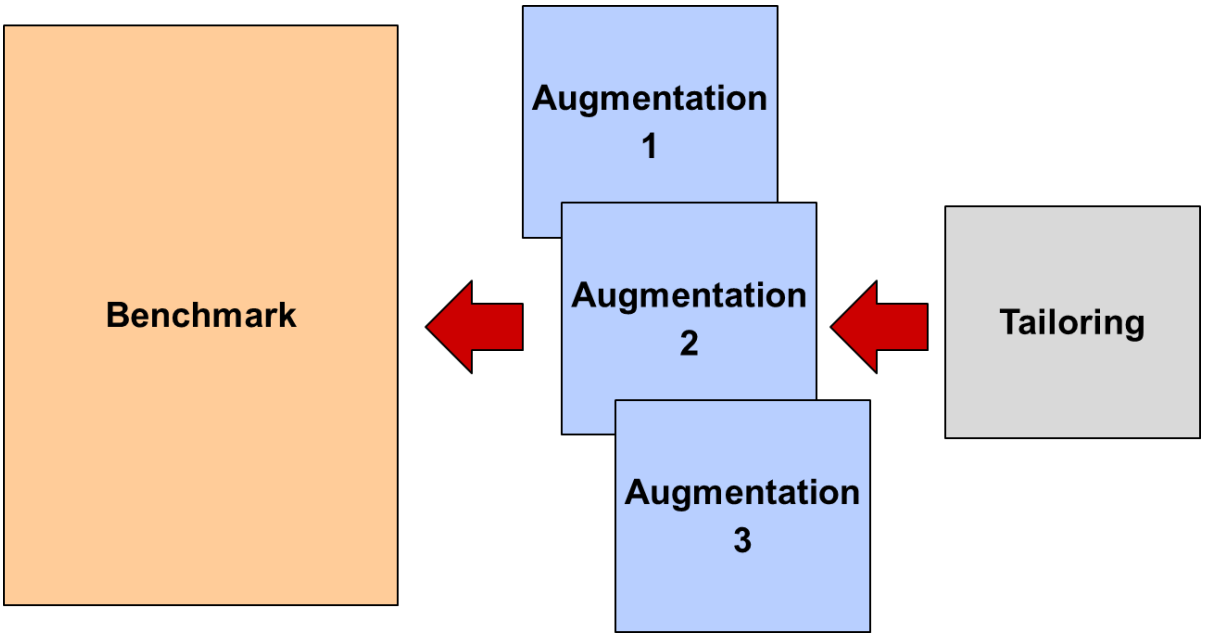
This session focused on review of the previous discussions and outlining of the major open questions. It noted the two primary use cases under discussion:

- tailoring – This uses a clear and minimal description of the tailoring activities made to a Benchmark. It is intended to be applied after all other tailoring (e.g. Profile selection) has completed. One group of beneficiaries of this use case would be auditors, who could use this file to quickly determine how an assessment deviated from some known, authoritative standard.
- augmenting – This involves not only tailoring of a Benchmark, as might be undertaken by a Profile, but the ability to add new Profiles, Rules, Groups, and Values as well. The result would be a new Benchmark and would be processed as such. One group of beneficiaries of this use case would be Benchmark authors, since it would allow them to add new Rules and Profiles to

an existing Benchmark without modifying the source Benchmark (breaking any signatures it had) or copying the whole Benchmark over (creating a maintenance challenge if the source Benchmark were to be revised).

The attendees had no objection to either use case and were largely divided by the mechanics necessary to support these cases and whether both cases could be handled by the same mechanic. Specifically, if a powerful, but complicated, augmenting mechanic was devised, its applicability to the tailoring use case was questionable because auditors might have trouble tracing the tailoring actions relative to the base Benchmark.

A 3-tier model was developed by Kent Landfield showing the logical relationship between a source Benchmark, augmenting files, and a tailoring file:



In this diagram, three augmenting files are applied, in sequence, to a source Benchmark. Each augmenting file adds to the effective data model of the Benchmark by adding Rules, Profiles, etc., that then are treated as normal components of a Benchmark with regard to extension, user selection, and processing. The tailoring file gets applied after all the augmentation files. Unlike the augmentation files it does not change the Benchmark's data model, but instead performs tailoring steps relative to the composite Benchmark's data model.

Several possible mechanics for augmentation were mentioned, including use of XInclude statements, use of XSLT stylesheet transformations, and the creation of new XCCDF elements to allow importation of other Benchmarks.

The issue of scoring and reporting came up multiple times. It was agreed that any solution for either of the use cases will need to leave some artifacts so that reviewers could know what was ultimately performed in a system assessment.

Break-out session 2 (June 14, PM)

The group discussed the idea of making composition easier by treating Rules as building blocks and Benchmarks simply as a means of bundling them together. Mechanically, there is little difference between this and the structure of XCCDF today. Conceptually, however, this represents a change in that, today, one usually thinks of a Rule as belonging to a Benchmark. (E.g., "This is a USGCB Rule.") Under a decomposed model, USGCB would simply be one particular collection of a given set of Rules and those Rules in other contexts would not retain any implicit association with USGCB as a whole. Others noted that decomposition, if pursued too energetically, could result in a content management nightmare where each Rule needed to be managed independently. It was noted that the focus on decomposition and reuse in OVAL, while useful in many aspects, has resulted in significant content management challenges there.

The group discussed some of the previously proposed mechanics. It was noted that XInclude already is a part of the XCCDF specification and that, if augmentation/tailoring was accomplished using XInclude, no changes would be needed in the schema or specification beyond noting the new use cases. Others noted that XInclude was a very awkward way of accomplishing these goals, that an external file that simply added one new Profile to an existing Benchmark would require the use of many XInclude statements, and that this mechanic would make things much more difficult for authors and tools alike and hamper the spread of XCCDF adoption.

Gary Gapinski provided an example of a schema that guided an XSL transform of an XCCDF Benchmark. The document was not, itself, XSLT but was a custom transformation language that then directed a transform. It was noted that such a language could take a variety of forms. Concerns were raised as to whether this would provide enough transparency and clarity to support auditors in the tailoring use case.

The 3-tier model from the previous break-out was brought forward and the community was asked whether it was necessary to have 3 separate tiers or if a tailoring file could simply be a very simple augmenting file that was applied last. (A 2-tier model.) This question was deferred as one of the parties who might be more knowledgeable on the implications of this was not in attendance.

It was noted that, if the purpose of the augmentation use case was to help authors, that the use of good tools could make the mechanic moot in that users could be presented with an intuitive interface and that, even if the resultant document was ugly, the user would not be affected. It was countered that authoring tools are still not ubiquitously used, and that ignoring the underlying complexity of the XML will hamper those who use XML editors, as well as the auditors who are actually reading the XML to determine behavior in the tailoring case.

With regard to reporting, it was noted that the proposed practice of including a set-value statement for every tailored Value in the TestResult element does not scale to enterprises that are running thousands of assessments. In these cases, the results need to identify the source of the tailoring actions (e.g., a Profile), rather than outline each and every change. To support this, all tailoring actions would need to be directly traceable to that named Profile (or equivalent structure).

The conversation concluded with a request to provide examples of the various mechanics to help the community better understand the implications of each method.

Break-out session 3 (June 15, noon)
Three examples of a simple Tailoring use were presented to the group: using XInclude, using a new benchmark-include element, and McAfee's tailoring schema.

The group discussed whether tailoring actions should be explicit (user driven) or implicit (automatic and without user knowledge). Some argued that being implicit was bad because it meant users were not aware of what assessment they were actually running and results might not provide enough information to determine what had been tailored after the fact. Others argued that, in a commercial world, making the assessment process more automated with fewer requirements on the user was necessary and that, if the tailoring is well written, user awareness of the tailoring in the assessment process is unnecessary. This seemed to be a point where the government and commercial interests diverged, with the former favoring explicit awareness and the latter preferring implicit automation.

It was noted that many of the recently raised concerns focused on traceability. To address this some proposed adding a simple Boolean flag to the TestResults to indicate whether there had been further tailoring actions (either manual or through an external tailoring file) of a Benchmark relative to the Profile named in the TestResults. If this was set to true, reviewers would know to consult other sources to determine how an assessment had been modified.

The session concluded without a solid proposal on the mechanics of supporting the tailoring/augmenting use cases.

Conclusion:

- Support for tailoring/augmenting Benchmarks using external files remains an open question

Other Discussions

In one of the other developer day sessions, the group discussed how to document deprecated items in SCAP standards. It was noted that, in the XCCDF specification, the approach taken had been to retain deprecated fields in lists, but to mark them as deprecated and to remove all text describing how these fields were used. In the discussions, community consensus was that descriptions of the use of deprecated fields should remain in the specifications. The argument was that, without these descriptions, users were given no clue as to how to understand deprecated fields that they did find and that authors (inadvisably) using deprecated fields might end up deviating from the original use and produce incompatibilities. It was further suggested that these descriptions also include the reasons for deprecating the given field and also note alternative features that could subsume any lost functionality.

Conclusion:

- Update the specification to retain descriptions of deprecated fields as well as justifications and alternative mechanisms

OCIL

Not yet available.

Situational Awareness and Incident Response (SAIR) Tier III

Not yet available.

Content Management

Not yet available.

Wednesday June 15th

SCAP 1.2

Not yet available.

Asset Reporting

During the Asset Reporting presentation, the following were discussed: Asset Inventory (AI), Asset Reporting Format (ARF), Asset Summary Reporting (ASR), and Tasking.

ARF/AI Update

ARF & AI are pending approval from NIST. Once NIST approves the specifications, they will be published on the NIST website.

Minutes

1. Refinements
    a. Refinements are a concept in the new ASR data model proposal. The discussion topic was "Where are refinements defined?" Should the requestor specify which refinements they want in the resultant ASR, or should the creator of the ASR create refinements as it sees fit to describe the data?
    b. Is it possible to provide enough context/scope within the ASR document?
    c. How should empty sets of data be represented?
2. DoD A[ssessment]SR creates one report per asset population.
3. Should the [SQL] "Where" clause be included in the ASR document? It could provide valuable context.
    a. Should the entire tasking request be included in the ASR document? What happens when the tasking is a verbal request?
4. What about digital signing and encryption?
5. Re: Should different reports be in different ASR documents?
    a. It is hard to determine which data & asset groups are 'related' in a single report. For example, you could have a single asset population and have completely unrelated metrics within that report.
6. Re: Should ASR require itself as a payload within ARF?
    a. One argument was made for ASR not requiring ARF, but certain use cases could define ASR in ARF outside of the ASR spec if necessary.
7. Re: Result Sets
    a. DoD ASR has seen COUNT satisfy almost all of their use cases
    b. The group was mute on the applicability of other result sets

8. Topics for consideration:
    a. Reporting against artifacts, findings, scoring
    b. Including the report request in the ASR
9. Re: FISMA Reporting
    a. FISMA reporting requires a broad scope of data sets, asset groups, and reporting results. If we were to somehow restrict reporting, it may cause a single FISMA report to be broken into multiple ASR documents.
    b. Pros/Cons?
        i. Allowing for a whole FISMA report will be complex
        ii. Not allowing for FISMA reporting causes the request/result structure to be complex.

Tasking

1. Tasking relates to data collection, but could possibly be used in Remediation among other specifications.
2. It was brought up that this work effort could be more accurately called “Report Request Language” instead of “Tasking”.
3. Tasking - Seems to specifically relate to Continuous Monitoring. Is this a stated goal?
4. Core fundamentals for tasking are core across domains, but the actual actions are domain specific
5. Joe W. brought up the concept of tasking as it related to auditing. The tasking would be, in his example, like windows scheduler; IE, give me XX on Friday 9am weekly.
6. The distinction that was made was this: The distinction between going out and getting the data (IE, run an assessment and give me the results) vs reporting on known data (IE, assessment data exists in a database, tell me about it).
7. Mike K. asks why you need to do tasking in order to do standard gov't reporting, since all those reports will already be created/defined and a special tasking language wouldn't be necessary.
8. Tasking structure slide did not generate discussion
9. The request composition slide did not generate discussion
10. Dave W. mentioned that results formats are very verbose, and that tasking will be able to reduce the verbosity of the results format.
11. Concepts came up like how do you target? How do you specify the temporal characteristics? Is 'tasking' really all that different than any other tasking that has already been defined? Should we just start from the ground up? Where do you specify response times (QoS items) among other items?
12. Concepts brought up: synchronous vs asynchronous, workflows, approvals.

Content Management Best Practices

Not yet available.

Thursday June 16th

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 occurred 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.

Note

One of the OVAL related discussions was led by CyberESI (OVAL for Artifact Hunting). CyberESI plans to provide minutes from their session, and therefore this document will cover only three of the four OVAL related discussions.

Why the OVAL Language Needs a Specification?

The OVAL Language needs a specification to clarify the ambiguities in the language. These ambiguities are caused by missing or conflicting documentation, specification by example, and overloaded terminology. Together these ambiguities make it difficult for those looking to learn the OVAL Language, write content, or develop tools that utilize the OVAL Language.

The primary goal of the OVAL Language specification is to make the language more accessible and easier to understand for members of the community. This involves separating the OVAL Language from its XML Schema representation which will remove any barriers caused by not knowing XML schema and allow for implementations in other representations (e.g. JSON, etc.). Furthermore, it will eventually serve as the authoritative documentation for the OVAL Language. It will also help to update and consolidate information about the OVAL Language which was previously distributed across the OVAL Language Schemas, OVAL Forum Archives, and OVAL Web Site.

Revised OVAL Language Use Cases and Requirements

As part of the development of the OVAL Language specification, the OVAL Language use cases[1] were revised. The primary change made to the use cases was the addition of use case scenarios which provide specific examples of how the OVAL Language can be used. The revised OVAL Language use cases were posted to the oval-developer-list for review on 4/29/2011[2]. The OVAL Language requirements were also revised to be more binding agnostic. Specifically, the references to OVAL Content being a document were removed. The updated requirements were posted to the oval-developer-list on 5/9/2011 for review[3].

OVAL Data Model Overview

The OVAL Language Data Model defines what OVAL Constructs and OVAL Enumerations are and what they are composed of. The OVAL Language Data Model contains models for the OVAL Language Core Schemas (oval-common-schema, oval-definitions-schema, oval-system-characteristics-schema, oval-results-schema, and oval-directives-schema). This distinction represents a shift away from the OVAL

Language being all of the OVAL Language Core and Component Schemas. With this approach, the OVAL Language Component Schemas are simply extensions of the OVAL Language Core schemas. In the OVAL Language Data Model, constructs will be defined using textual descriptions, UML diagrams, and a table of properties and enumerations will be defined using textual descriptions and a table with the valid values in the enumeration.

What If Something is Deprecated?

The OVAL Language contains constructs and enumeration values that have been deprecated. The OVAL Language Deprecation Policy[4] states:

"When an OVAL Language construct is marked as deprecated its usage becomes strongly discouraged and it may be removed in a later release."

In our current draft of the OVAL Language Data Model, we decided not to include the deprecated constructs or enumerations to make the specification cleaner. However, it is important to note that even if a construct is deprecated, it can be used until it is finally removed from the OVAL Language. The community was asked for thoughts on this decision.

The community stated that it would be beneficial to include the OVAL Language Deprecation Policy and the OVAL Language Versioning Methodology in the specification as appendices. Furthermore, the community mentioned that constructs should only be removed from the specification when they are removed from the OVAL Language. This is because the specification is needed to make people aware of the deprecated constructs, why they were deprecated, and what constructs, if any, can be used in place of the deprecated construct. Lastly, it was suggested that the OVAL Language Deprecation Policy definition should be modified such that it states "…constructs *will* be removed in a later release." rather than "…constructs *may* be removed in a later release.". By making this change, it will explicitly indicate that the deprecated constructs will be removed and that they should not be relied upon.

How to Handle Constructs Defined Outside of OVAL?

In the OVAL Language Requirements, OVAL Language Requirement 3.1.4.1 states that the language must provide a mechanism to ensure the integrity and authenticity of all content written in the language. The current XML Schema representation of the OVAL Language uses XML Signatures to fulfill this requirement. The community was asked if it made sense to include signature properties in the current constructs that utilize XML Signatures in the OVAL Language Data Model even though the specification was geared towards being binding agnostic.

One member of the community stated that if it is supported by the OVAL Language Schemas then it should be documented in the specification. It was also pointed out that the implementation of the OVAL Language must provide a mechanism to ensure the integrity and authenticity of the content, but, tools are not required to implement it.

The OVAL Language specification represents a big shift from the XML Schema representation of the OVAL Language to a more logical binding agnostic view of the OVAL Language.

One member of the community recommended that since XML Digital Signatures exist in a separate namespace they should be allowed anywhere in a document such that they can be used or not used as needed. This would make it such that the allowed locations for XML Digital Signatures would not need to be defined in the specification.

Another member of the community stated that the goal of a specification is to provide standardization, a common interpretation, and interoperability and if we do not define how specific elements work we will not achieve those goals. They also stated that we need to document the XML Schema binding because that is what we have and if we are going to get interoperability we need to understand what these things mean. That means if we are going to include external constructs in the specification we will need to reference the normative guidance for the constructs and we should not over specify the constructs on top of the normative guidance.

It was also pointed out that this is not a matter of referencing or not referencing, but, rather are the external constructs captured in the logical model of the specification or in the binding section of the specification.

It was also recommended that the use of XML Digital Signatures be deprecated as well as explicit metadata and instead just allow people to use metadata where they see fit rather than trying to predict what metadata will be useful. Over the years, additional metadata is added because you have chosen to limit the information that resides in other namespaces which would keep it separate from the OVAL Language and tools could just use this information if they wanted to.

Another member of the community disagreed with deprecating XML Digital Signatures be by defining it in specific locations tools will need to or at least should support it and know where to look rather than relying on some out of band negotiation. This is the opposite direction of where we want to go with similar things in this domain.

Since there were differing opinions on this topic, this discussion has been continued over the oval-developer-list and can found at the following link.

http://making-security-measurable.1364806.n2.nabble.com/OVAL-Specification-topics-td6521019.html

How to Handle Container Constructs?

The OVAL Language contains many container constructs that are an artifact of the current XML Schema representation. An example of this is that an OVAL Definitions Document, in current XML Schema representation, can contain zero or one DefinitionsType construct(s). The DefinitionsType construct can then contain one or more DefinitionType constructs. The community was asked if it made sense to include the DefinitionsType container construct or simply specify that OVAL Definitions Document can contain zero or more DefinitionType constructs.

The overwhelming response from the community was that the OVAL Language specification should align closely with the XML Schema representation of the language and should include the container constructs. The primary reasons that were given for taking this approach were that the community is

primarily concerned with the current XML Schema representation of the language and by abstracting it out to a more logical view it may result in a specification that is harder to understand as well as cause synchronization issues where you have to make sure everything is consistent between the logical view and the XML Schema representation which is challenging. Lastly, some members of the community expressed that, from experience, trying to separate the logical view from its biding representation was challenging and really ended up resulting in twice as much work. It was also recommended that the specification should initially focus on the XML Schema representation of the language to minimize any difficulties when transitioning from the OVAL Language Schemas to the OVAL Language specification.

It was also asked if there was a driver behind separating the logical representation of the OVAL Language from its current XML Schema representation. The primary driver here was to make the OVAL Language more accessible for those who may not have a background in XML Schema as well as open the door for future innovation such as using a different binding representation.

How to Handle the Datatypes?

The OVAL Language datatypes are currently defined by the DataTypeEnumeration which is a union of the SimpleDatatypeEnumeration and the ComplexDatatypeEnumeration. The SimpleDatatypeEnumeration contains all simple data which can be represented as a string value whereas the ComplexDatatypeEnumeration contains structured data such as the record datatype. In the OVAL Language specification, we decided not to maintain this separation of the datatypes into distinct groups (e.g. simple vs. complex), but rather just represent all datatypes in the DataTypeEnumeration because they are semantically the same thing. The community was asked if this decision made sense and there was consensus that this representation of datatypes was appropriate for the OVAL Language specification.

What to do with OVAL Language Metadata Constructs?

The ElementMapType and DeprecatedInfoType constructs from the oval-common-schema are defined to provide structured metadata about the OVAL Language. Specifically, the ElementMapType construct describes the relationship between an OVAL Test, OVAL Object, OVAL State, and OVAL Item and the DeprecatedInfoType construct describes why a construct was deprecated and in what version of the OVAL Language. The community was asked whether or not these constructs belonged in the OVAL Language.

One member of the community stated that since the metadata constructs are defined in oval-common-schema, which is part of the OVAL Language, they should be included somewhere in the specification.

Another member of the community considered metadata constructs as constructs of convenience in terms of describing the structure of a document using a particular schema language and are not part of the OVAL Language and should not be included as part of the specification unless they are included by reference.

A member of the community disagreed that they are simply a construct of convenience because they help schema authors and tool developers who may process or create OVAL content. As a result, the specification should provide clear guidance on how to use these constructs.

Another member of the community mentioned that it might be useful to simply define these constructs in the appendix that contains the OVAL Language Deprecation policy.

A member of the community also mentioned that these constructs add no value to an OVAL instance document and should not be included.

Lastly, another member of the community disagreed that they provide no value because an XML Schema document is an instance document and these constructs are used to describe it. If an XML Schema is considered as an instance document having these constructs defined apply to use cases such as in the case of tools.

Due to the differing opinions regarding this topic, this discussion has been continued over the oval-developer-list and can be found at the following link

http://making-security-measurable.1364806.n2.nabble.com/OVAL-Specification-topics-td6521019.html

How to Represent xsd:choice Constructs?

The current XML Schema representation of the OVAL Language utilizes the xsd:choice construct to allow for one element, from a collection of elements, to be present in an XML instance document. An example of when the xsd:choice construct is used is in the ComponentGroup construct which can be an object_component, variable_component, literal_component, or any one of the OVAL Functions. In a UML diagram, the ComponentGroup could be represented as a composition of zero or more of these constructs or it could be represented as an inheritance relationship where each construct is derived from the ComponentGroup construct. In the OVAL Language specification, they are currently represented as a composition relationship in the specification. The community was asked if there was a preference in how they are represented. A member of the community stated that treating the xsd:choice construct as a composition relationship seems to align with the XML Schema representation.

Other Discussion Topics

Due to time constraints, the OVAL Language Specification Review session ended after the discussion of the "How to Represent xsd:choice Constructs?" topic. The discussion, for the remaining topics, has been continued over the oval-developer-list and can be found at the following link.

http://making-security-measurable.1364806.n2.nabble.com/OVAL-Specification-topics-td6521019.html

References

[1] OVAL Language Use Cases http://oval.mitre.org/adoption/usecasesguide.html

[2] OVAL Language Use Cases Discussion http://making-security-measurable.1364806.n2.nabble.com/FOR-REVIEW-Revised-OVAL-Language-Use-Cases-td6316500.html

[3] OVAL Language Requirements Discussion http://making-security-measurable.1364806.n2.nabble.com/FOR-REVIEW-Revised-OVAL-Language-Requirements-td6345171.html

[4] OVAL Language Deprecation Policy http://oval.mitre.org/language/about/deprecation.html

PowerShell Proposal

This session was led by Kelly Hengesteg, Jeffrey Snover, and Michael Tan from Microsoft.

Introduction

By Kelly Hengesteg

Microsoft began writing OVAL content for Exchange and SQL Server and realized PowerShell was the only means to examine some of the configuration items for those products. It was also determined that more products from Microsoft will use PowerShell as the basis for configuration in the future. However, OVAL does not support PowerShell so a proposal to extend OVAL to support PowerShell was devised.

The presentation is intended to provide an overview to PowerShell, and to introduce the proposed design for integrating PowerShell into OVAL.

PowerShell Overview

By Jeffrey Snover

Historically, UNIX had a very powerful shell and Microsoft had a very poor shell. The PowerShell team's intent was to create a powerful new shell incorporating interaction with common Windows constructs – WMI, registry, et al.

With the Common Engineering Criteria (http://www.microsoft.com/cec/en/us/cec-overview.aspx), Microsoft is specifying PowerShell as the mechanism for the configuration management of their products and systems moving forward. PowerShell Cmdlets will support configuration, operation verification tests, lifecycle, diagnostics, and data management for their products and systems.

PowerShell has been embraced by the virtual machine community, e.g. VMWare, because the number of machines people need to manage has exploded with the wide adoption of virtualization.

At its core, PowerShell has an automation engine DLL that gets hosted and accessed in various ways – interactive shell, hosted inside programs like Exchange. The PowerShell engine operates on either a string (which is parsed) or a known data structure. In the OVAL proposal, a schema that invokes the automation engine as an API is used.

Jeffrey then presented an overview of the execution architecture of PowerShell comparing it with UNIX and VMS DCL. He showed a few demonstrations of PowerShell commands.

He also explained some of the restrictions that can be placed on the execution environment of Cmdlets. A "constrained runspace" allows detailed configuration of what can and cannot be executed.

Discussion

Question: On what versions of Windows is PowerShell available?
Answer: PowerShell 2.0 is available on Windows 7 & 2008 R2. XP and above can run PowerShell 2.0.

Question: In a constrained runspace, is it possible to discover the available commands? Answer: Yes.

Question: How do Cmdlets map to the messy storage mechanisms at the lower layers? Answer: Execution units supply a contract to PowerShell engine and what they do, PowerShell does not care.

Proposal

By Michael Tan

NOTICE: Unfortunately, the microphone was off during this portion of the presentation undermining the development of detailed minutes. Please email oval@mitre.org if you have any corrections or additions to this section.

Michael led a detailed review of the PowerShell cmdlet_test proposal that he sent to the oval-developer-list (http://making-security-measurable.1364806.n2.nabble.com/Windows-PowerShell-Proposal-for-SCAP-tp6464505p6464505.html).

Discussion

NOTICE: There were several questions, but again, the microphone was off and the answers cannot be heard in the recording.

Question: Is piping of commands supported, or can I use more than one select? Answer: We are not modeling a full PowerShell pipeline. Generally, you will only need the Gets (get-<*noun*> cmdlets). We allow you to use the Select to identify the fields in a response from a Get, but we do not support a full pipeline. If that capability is needed we can enhance the proposal to support the capability. However, doing so increases the risk of malicious use of PowerShell and we do not think the capability is needed at this time.

Question: Is there anything that will prevent the assessment of the cmdlet_test from a remote host? Answer: No, PowerShell cmdlets can be remotely executed.

Question: Will the cmdlet_object support variables on its entities, like other OVAL Objects do? Answer: Yes, the entities in the cmdlet_object proposal are just like all other entities in the OVAL Language. As standard OVAL entities the var_ref attribute will be supported.

Question: Could OVAL be used to, at run time, get the module name and supply that value as a variable to the module name entity?

Answer: Yes, there is a PowerShell cmdlet, Get-Module, that can be used to get the module information. This information could then be used to as a value for a variable on another cmdlet_object?

Question: Is it possible to inject commands via abuse of the ';' character or other characters? Answer: One of the goals in developing PowerShell was to specifically avoid such a possibility. There is only one place that this can be done, and it is called Invoke-Expression. This cmdlet has been reoved from the allowed set of cmdlets.

Question: Is the OVAL cmdlet_object proposal intended to provide input to the PowerShell DLL via a data structure or a string?

Answer: The proposal is essentially populating a data structure that is supplied to the DLL. However, tools can really take either approach to interfacing with PowerShell.

Remark: Notice that this is the first occurrence of the record data structure in an OVAL Object data structure. This structure is currently used in OVAL States only.

Remark: The list of allowed cmdlet verbs has been specifically whitelisted to include only those that do not impact the system state. We filtered out all cmdlets that are known to have side effects. We have limited the allowed verbs via OVAL Language Schema restrictions.

Remark: The Microsoft proposal includes sample code that demonstrates the usage of the proposed cmdlet_object.

Question: Can a cmdlet be signed?

Answer: Yes, cmdlets can be signed to allow them to run in a restricted runspace and you can check the signatures of cmdlets before execution.

Question: What is the import verb?

Answer: import is a way of getting data from one state and pulling it into a system. This verb will allow you to read from various files.

Question: A goal of OVAL has been to look at the primary source where data is stored not secondary interfaces that mirror the data. In PowerShell, is there anything that might lead to differing results based upon how the data was retrieved?

Answer: Yes, that is possible with any access layer. If you don't get your response from the single source of truth then there is a possibility of caching causing problems if you query other interfaces. However, the fact is that the underlying configuration data is inconsistently stored. PowerShell cmdlets are intended to provide a single interface to this configuration data.

Remark: There are several known configuration items that can only be checked via PowerShell cmdlets because the data is held in a proprietary data store.

Remark: As Windows evolves there will be new cmdlets added, but it is unlikely that existing administration capabilities will be removed. This means that there will still be registry access, and API access to password policy information. We will simply be adding in more new capability with PowerShell cmdlets, not replacing capability.

Question: Will Microsoft produced baselines use the cmdlet_test exclusively?

Answer: No, the plan is to continue using legacy OVAL Tests whenever possible and only use the new cmdlet_test when a legacy OVAL Test is not available. If the community wants a 100% PowerShell baseline, that is possible, but we would need to hear from the community that it is needed or preferred.

Question: Will Microsoft's security tool support the export of baselines using cmdlet_objects in a hybrid or exclusive mode?

Answer: For now, the tool will export hybrid content. If there is significant demand we could consider exporting cmdlet only content but some settings are not available through cmdlets and there is significant vendor investment in the legacy OVAL Objects so we feel we should continue to support those legacy objects.

Remark: PowerShell does not replace WMI. WMI is a standards based approach to management. It is not being replaced. WMI and PowerShell are complementary.

Question: What does a query look like when no select statement is present?

Answer: As a best practice, the select statement in the cmdlet_object should always be used to specify which fields should be represented in the OVAL Item. Without a selection, various engines might represent the data differently.

Remark: Outputting consistent results is key.

Question: Will the OVAL Interpreter prototype code be made available?

Answer: Yes, as soon as it is completed we will post the prototype on the OVALDI sourceforge.net project. There is also a code sample in the Microsoft proposal.

Remark: We would like to see this proposal folded into version 5.10.

Question: Can you show an example of getting a low level configuration setting?
Answer: To clarify, the applications that run on Windows are much further along in the adoption of PowerShell than Windows itself. Therefore, for most Windows configuration items the legacy API is likely to be the only interface to the configuration data.

Remark: the PowerShell proposal includes examples and references to complete listings of cmdlets for several applications.

SQL Test Overview

The ind-def:sql57_test and ind-def:sql_test have been in existence for some time, but have not been widely implemented. This discussion provided some real world experiences and challenges with the test, specifically the ind-def:sql57_test.

Matt Hansbury began by providing an overview for the IRS/SCAP content development, including history, plans, and challenges as background for the discussion. Rob Hollis, from ThreatGuard, followed up with an in depth conversation regarding the ind-def:sql57_test's unique challenges and lessons learned. He presented a series of challenges, designed to spark discussion and re-evaluation of some aspects of the test for possible future versions of OVAL.

Database Applicability

Once of the challenges faced in using the ind-def:sql57_test was determining which database(s) a given OVAL Test applied to. While talking about how to determine which checks to run on specific databases (such as Oracle, MSSQL, etc.), Rob pointed out that ThreatGuard used the affected element at the OVAL

Definition level to provide this hint. This effectively uses the metadata in the definition to determine applicability. Additionally, it was mentioned that the applicability capabilities that have been discussed within the OVAL Community of late would also be a potential solution here (see discussion on "applicability_check" http://making-security-measurable.1364806.n2.nabble.com/Proposal-for-extending-Oval-criteria-criterion-and-extend-definition-to-specify-applicabilityChecks-tp6271556p6271556.html).

Database Connection String

Another point of discussion was the connection string. Proper use of the connection string has never really been fully fleshed out. There are long standing concerns about the open ended nature of the entity leading content developers to expose database credentials. However, there is a need for great flexibility in defining connection parameters because the access level and how a tool connects to a database can greatly affect the assessment results. One suggestion was to parameterize the connection string entity to allow more flexible connectivity. While this could be useful in some cases, it wasn't applicable to the IRS case, since the target connection is made before the OVAL Definition is encountered. This experience leads to another option which be to simply define the connection information separately from the OVAL Object.

Additionally, it was asked if tasking could help with the connection string issue. Tasking is seen as an emerging area that will be aggressed by the enterprise reporting group as they work on specifications like ARF and AI. However, it was pointed out that if you add too much of this to the tasking, then you lose the opportunity to tailor things on a per OVAL Definition basis and you introduce a dependency upon tasking in OVAL.

Other Topics

It also came up that there were use cases where the content author would need to be able to specify not only the target system to connect to, but also the specific database instance.

The idea that we might need separate tests for Oracle, MSSQL, etc. was also raised. This would allow for variations of SQL since nearly all major databases have their own extension of SQL.

Another commenter mentioned the distinction between assessing the configuration settings of the database server, vs. assessing an application's settings that happen to reside in a database instance. This served to highlight the fact that there were a number of use cases under consideration for this test.

During the discussion it was pointed out that anonymous database field names should not be used. For example, SELECT COUNT(*), should be replaced with SELECT COUNT(*) AS total. This is explicitly documented in the schema documentation for the ind-def:sql_test. If fields are not named there will be inconsistencies across database server implementations. This worked for the set of databases that have been tested, but it is not clear if it will work across all database implementations. However, a follow up point was made that we currently write most OVAL Definitions to specific platforms, so this wasn't viewed as a major issue. We simply need to acknowledge which platforms we are targeting when writing a given test.

Later, while discussing the discovery issue, it was mentioned that in many cases, authors need more control over which database instances are assessed. Sometimes one needs to assess all instances of a database, in other cases, a more granular approach is required. Tasking was brought up as a possible solution here, as well. It was observed that the optimal solution here may ultimately involve a hybrid solution involving both formal and informal constructs to accommodate the appropriate use cases.

While discussing the challenges of assessing multiple databases within a database server, it was pointed out that here is no documented convention for handling the result of each instance assessment. This is an area that requires more community discussion. One solution that was offered was to leverage the OVAL System Characteristics system_info element to distinguish instances. The system_info element is intended to identify the asset that was assessed. Perhaps we need to consider the asset being assessed to be the specific database instance instead of the operating system which was running the database server. This could be implemented today leveraging the AI specification in the xsd:any space of the system_info element. This would allow tools to create a single OVAL Results document that represented the full assessment of each database instance.

Finally, the question was asked by Matt Hansbury if we could agree to make the connection_string entity optional. It was pointed out that to do so would make it such that the information regarding the connection string would be entirely removed from OVAL. Therefore it would be better to keep the connection string information as mandatory for now.

Conclusion

At the end of the discussion is became clear that while it was possible to implement the ind-def:sql57_test in a real world environment, it was necessary to make a number of assumptions in order to properly execute the checks.

The community needs to continue to gather use cases for the tests to try to fully understand all of the required components and options that is required of the test. Once this information is truly understood, one or more revisions should be made to the test, in order for it to better serve all of the community's needs.

Remediation Panel Discussion

Introduction

Gerry McGuire, of MITRE, served as moderator for this discussion.

NIST published a draft of IR-7670 – Open Specifications for an Enterprise Remediation Automation Framework – This lays out a roadmap for inter-related specifications.

- Lays out eight interrelated components
- Short term goal – start writing the first pair of these specifications:
  - CRE – Common Remediation Enumeration
  - ERI – Extended Remediation Information

These industry experts will answer questions that are a determining factor to these specifications' production:

- **Joe Wolfkiel** (DISA)
- **Mike Kinney** (NSA)
- **Chris Johnson** (HP)
- **Kent Landfield** (McAfee)

Question 1: Shall CRE parameters be human or machine orientated?

Examples:

Machine Oriented: A timeout for screen saver activation would be passed as a value "300" and the ERI would supply the human units "seconds".

Human Oriented: A timeout for screen saver activation would be passed as the descriptive string "5 minutes" and the ERI would provide information to convert this to a machine usable value.

Mike: CRE is just an identifier with no other data. The ERI is the definition of what the CRE is and a parameter would be passed in addition to it to the interpreting device. ERI could contain the parameter or not. I see the CRE, ERI, and parameters as a triplet which triggers a remediation. CRE is a standalone identifier that can be paired up with other identifiers (ex: CCE, CVE) to give you remediation. The ERI is the human readable part that explains what the CRE really means.

Kent: I would like to see some semblance of both. We're too early in the process with these types of standards to make a decision though and the more automation we can have, the better. Too much human readable text makes it harder to work with when automating things. We shouldn't have to choose one or the other. Should be human readable while still easily parsed by automation tools.

Chris: I agree with Kent and see that there is a need for both. In addition, I recommend keeping the CRE as light as possible. It should just be an identifier ideally while using the ERI to capture all of the additional information which can then be referenced by any tools that need it.

Joe: I believe it should be machine based because it is most beneficial when automating things. Focus on making it machine readable. However, the need for good comments is a must in order for it to be understandable by humans. This way, a remediation can be sent to a human and still be understood. We need to avoid making this ambiguous. But at the end of the day if it will be sent to a machine then it needs to be machine readable.

Jon Baker (MITRE): For the team working on the DoD remediation pilot, what have you needed, machine or human oriented values?

Jack Vander Pol (SPAWAR): In the demo, CREs were just identifiers with no concept of parameters.

Richard Kelly (SPAWAR): Current setup doesn't do anything with ERI but does look at the CRE ID and then looks that up in a database to learn what it has to do. Only one parameter is usable in the demo to allow for some customization, but there is no ERI capacity.

Joe: It helps to think through the different stages of the workflow. If I get a policy from the DoD that says "if you have this finding, then this is the remediation that you should be applying," and you have some repository of all known remediations, if you decide you can't use that one, that you want to apply a different one, and make an annotation as to why you chose it. When sending the tasking, all you need to do is send the ID for it to be understood. If it is to be done by humans, they must fully understand what it means to apply that remediation.

Mike: The manager function needs the ERI so someone can make an intelligent decision as to which CRE to apply. This is the decision made by the human. The requirements are the CRE, which is an ID, and ERI which explains what the CRE is, and in case they need to change the remediation method, they need to know what each CRE & ERI means. The tool doesn't care though, and as such only needs some way of mapping the two. The endpoint doesn't need an ERI unless you have a user there that has to make a decision, and this is not a prevalent use case.

The Enterprise use case requires more pieces to compare what was scanned to policy to find out what the directed course of action is. Someone then needs to know that they can apply that in their infrastructure

Question 2: How should we approach defining the prioritization of remedies?

Not all vulnerabilities have the same priority. A proposal is to add a priority (scalar value) as an element in the ERI. The scoring algorithm is external to the CRE specification.

Mike: Depends on how you define remediation. If remediation is just fix for a CCE. If it's a fix for a CCE or a CVE, that's another thing. If it includes mitigations (workarounds), that is another one.

I believe that remediation is defined as bringing systems into compliance based on the CCEs that are found, not CVEs or mitigations or anything else. We need to at least get this first part done (a fix for CCEs), otherwise nothing may get accomplished.

Kent: So you do agree with this definition, but you want it phased in so that we focus on CCEs first, CVEs second, and everything else third.

Mike: I don't know that CVEs will ever be fixed by a manager due to the way we distribute software and patches. When you have a client go to a patch manager, you can't push anything. If the manager doesn't have a patch, then you can't apply it. You could enable it and cause a reboot in the middle of the day.

Kent: I believe remediation constitutes anything that needs to be corrected on a network that has any sort of security ramifications. Implementation is a different issue. Regarding the prioritization of remediations, it comes down to how critical the issues are and the criticality of the systems. We are focusing on changing settings/applying patches/etc..., when in reality we need to take risk into account. We have to focus on putting in the scoring system that we are developing and incorporating that into a management environment that allows for deterministic directed actions based on the severity of the event they are trying to correct.

We need to recognize the severity that is put on the issues that we're trying to address from the primary source vendors. These vendors have the greatest awareness as to what the real issues are and how it effects their operating environments. Priority and severity have to be addressed, but they are bigger than just configurations. I do agree that whatever we do does need to be phased into development; otherwise we will never get to where we need to be.

We would really like to see a more integrated patching system so that it can be more automated and more easily validated than it is today.

Chris: I see priority as not being "hardwired" into the remediation. We will be looking at a number of different factors when doing this risk calculus and determining what actions to take when facing a certain deficiency. We need to be looking at the environment and all other factors that come into play outside of the actual remediation action we are performing. We already capture many of the metrics needed to prioritize remediations.

Joe: Agrees with Kent and Chris. In addition: Prioritization in a network environment is a function of aggregation. If you are thinking about a patch that would fix multiple vulnerabilities then CVSS isn't good enough. Applying a service pack that fixes hundreds of vulnerabilities should probably take priority over patches that fix one or two.

When driving down risk, fixing medium vulnerabilities on many systems before a single instance of high vulnerability might be more beneficial. Fixing vulnerabilities on devices directly facing the Internet first may also be of importance. We're hoping to drive prioritization by assigning/aggregating risk as weighted averages across populations to best drive down risk.

Prioritization as a function of risk is new, and we are just getting started on it. Prioritization working together with remediation is a function of looking at what a remediation does in terms of what risk it drives down, the population that drives it down, and the threat that is actually targeting that in real time. I would not tightly couple remediation with risk management prioritization.

You can't combine scoring systems or even aggregate within scoring systems. The main issue is the effectiveness of driving down risk

Chris: There are multiple scales for prioritization. Prioritize within a host relative to which CREs you want to run, across a group, and across the enterprise, and we need to keep all of these in mind.

Kent: I agree: We do not need to calculate risk. Risk needs to be its own item that takes information from a lot of different sources. There are also environment-specific variables that cannot be detailed in a standard.

Mike: Right now we have CVE and CVSS. This is fine for what patches should be applied for a specific system without looking at total risk. We will only get a good picture of how to score things when we know what is on our network, and what each devices situation is (patched/unpatched, how they are configured, how close they are to the internet, etc…). For now we need to use what we have until we have a better way to look at everything on the network.

Question 3: How is the version field used?

There are two fields defined in CRE; version (an integer) and deprecated (a boolean). When a new version of a CRE is added, is the existing version deprecated? That is, should a new version replace and deprecate the current CRE?

Mike: If you have a CRE that states how to fix an issue and you replace it with a new fix, wouldn't you just add a new CRE rather than deprecating?

Kent: We don't know enough at this point to effectively answer this question from a policy perspective. We need to determine what the value may be in deprecating a CRE vs. creating a new one.

Chris: When covering deprecation, think about it across the other enumerations we currently support. Make sure our policies regarding deprecation are consistent, make sense, and work well together.

Joe: I hope you would only deprecate a remediation if you find out after the fact that it doesn't do what you thought it did. The goal is to get people to stop using the deprecated version, so the newer version should be the correct method. I'm not sure if this matches the deprecation definition across the board.

Dave Waltermire (NIST)**:** What if the definition of the CRE is ambiguous or confusing and you want to clarify it? Would revising the CRE be advantageous or would it be better to just create a new one?

Mike: Why not just redefine the ERI instead of the CRE?

Dave Waltermire (NIST)**:** You can't define metadata about the remediation unless you understand what the remediation is. This is why I was suggesting that the change would have to be more fundamental. If the CRE is ambiguous then you need to disambiguate it before you make any changes downstream.

Jim Ronayne (NSA)**:** What was the version attribute supposed to do?

Chris: We need to think about where parameters are going to live. We talked about literal and human readable parameters and keeping those two in sync if, per chance, one would be in the CRE and the other in ERI. There is some risk there and we have to make sure that the deprecation actions and versioning is consistent and kept in sync between the two.

Matthew Wojcik (MITRE)**:** (via email) A CRE would be deprecated if and only if it is found to be fundamentally incorrect. It describes a remediation method that simply does not exist on the relevant platform or the ID was assigned at the wrong level of abstraction, so the CRE needs to be split or merged. The version number is intended to indicate that some change was made to that CRE item so people can tell that a change occurred.

participant: How do you know if there is a later release about a specific CRE?

Jon Baker (MITRE): Like we mentioned the other day, even if you deprecate a CRE, you probably want to change the version number to indicate change.

participant: This problem isn't unique to CRE.

Question 4: What structure is needed in the CRE Reference field?

A recent comment mentioned that the reference field in CRE should be structured. What does this requirement mean to you? Is there a disadvantage to this approach?

Mike: No comment

Kent: Can we include more than one reference? If so, what is the structure to make them easily consumable (both visibly and for automation purposes)? However we structure multiple references if we are to allow it, we need to keep it consistent.

Chris: How will the references be used? We need to look more at use cases to see how we are going to be using references. This will help determine how much structure needs to be present. Looking at other current specs and how they handle references might help us learn and see what has or has not worked in those situations.

Joe: I'm in favor of some form of structured, intelligent extensibility. If we don't know what goes into a field, but we are pretty sure there is more than one reference, and we have reasonable belief that people would want to structure their references into separate labeled parts, doing something extensible would give you some flexibility.

Mark Davidson (MITRE)**:** What kind of content would go in the reference section? Would it be the CVE that it fixes, or a technical bulletin?

Mike: The ERI explains what the CRE is, so you might want a variety of things there. For example, a local certificate article, Microsoft patch information, or any reference that helps you determine what that means.

Mark Davidson (MITRE)**:** Would a reference point to an ERI, or is the ERI meant to be local?

participant: The ERI would be local but it would have the references in it

Jon Baker (MITRE): I think the reference in the CRE is there to disambiguate it from the other CREs and that that is its main purpose, similar to CVE and CCE.

Jim Ronayne (NSA)**:** The spec should be very clear about the intent of both the reference sections in the CRE and in the ERI so it is obvious where one goes.

Question 5: What processes and methods should we use to manage, coordinate, and disseminate content decisions for CRE?

Mike: Decisions are based on policy, so policy has to tell you what they think you should do, so that would have to come from an authoritative source that is the keeper of policy. We would likely need some form of tiered policy and whoever is the keeper of the policy should be responsible for distribution.

John Wunder (MITRE)**:** In the past this question has come up during split/merge decisions. To clarify the question, I think that this is during CRE creation and management.

Mike: So there has to be some form of central repository so they would be assigned by one entity, but you need the ability to affect things that haven't been assigned yet, so you also need to have something on the tool to manage your own CREs.

Jon Baker (MITRE): CVE and CCE have evolved over time. CRE has to make similar content decisions (split/merge, include/exclude) to these two and it is challenging with CRE because CREs will be namespaced so they may not reside globally just because they reside in one organization. We need some way to convey these split/merge decisions to have some form of consistency.

Kent: I agree. Having one authority won't scale well commercially. They will have to deal with things that will never get registered as a CRE at a global level because they are local applications, issues and needs. We need to support a model where local organizations can create their own suites of CREs so they can use them internally. Most remediation tools allow this and this customization functionality is heavily used.

Chris: Today's activities are a good example for how we might handle best practices for CRE creation and maintenance. We're putting out publications that steer others towards best practices that will ensure that the content can be evaluated, that it performs well, and we can identify issues as we go along and use the specs. This is what content creation for CREs should be like.

Joe: The DoD implementation of CPE has given up on having a central repository like the one CVE uses. When scaling CRE, we might find the same thing, having separate companies making their own CREs in their own CRE namespaces. I think we will need a hierarchy-friendly design with no central decision-maker.

Mike: So are we wasting time assigning CREs in a central repository?

Joe: I'm not saying that. We are going to have to map locally until CREs are assigned at a higher level, otherwise development will take too long. We will probably need federated system of repositories with aliasing back and forth based on namespace.

Mike: So if you buy a tool with preset CREs, and you add your own fix for something. If everyone does this, then it is extremely important how this information gets sent back.

Joe: If you can find a consistent way to tag metadata. The more structured, the more limited the enumerations in the metadata are, the easier it is to match it. If you have several systems with several remediations that are really the same thing and when that's reported, you really want to deprecate the policy that required these separate remediations and replace it with a new policy with an enterprise wide remediation. We need to be able to react when we see a new fix getting sent back.

Mike: How long would it take to get policy changed?

Joe: Not that long. It depends on the level of policy.

Gary Gapinski (NASA) Federation will not occur. The only type of federation might be to use separate namespaces or identifiers for content. There are also plenty of excellent contemporary examples of distributed version control systems being used for software that could be equally useful for SCAP protocol specs. For example github, which allows you to grab anything you need or want and then potentially merge back if you update. Another example is Mercurial.

"github" is not federated with anything in particular; it is where you go if you want something that has chosen to host itself there. The projects themselves are separate in that they have different names. These solutions work well and I wouldn't expect us to be able to outperform a scheme such as this.

participant: It might be interesting to use something like github; maybe an automatic portal that gives others their own identifier numbers that is then managed by an authority.

Kent: You could start simple with something like creating an SVN and giving access to it to all vendors so they can update it with all CREs they want to publish and then give customers ability to pull this data down. It is inexpensive and easy to manage. Give each publisher a folder.

Question 6: Define remediation and how does that relate to mitigation?

Consider:

- Does remediation cover patching?
- Does remediation cover partial fixes?
- Does remediation cover uninstallation?
- Does remediation cover disabling a service?
- Should attack vectors be mapped to CREs? (e.g., CAPEC)

The draft document NIST-IR-7670 contains:

"…remediation is defined as "a security-related set of actions that results in a change to a computer's state" and may consist of changes motivated by the need to enforce organizational security policies, address discovered vulnerabilities, or correct misconfigurations. Remediations can include changes to operating system and application software configuration settings, the installation of patches, and the installation or removal of applications, software components or libraries."

Mike: I make a distinction between mitigation and remediation and incident handling. I have no problem with CRE being an identifier for all of those, but to me, remediation is bringing a system back into compliance with a policy, no more and no less.

Kent: Remediation covers just about anything that fixes a security issue in your environment. Mitigations are just a form of remediation. There are different potential ways to fix a problem in the lifecycle of vulnerability. Some are permanent patches or solutions or removal of software. There are times when mitigations allow you to put a small barrier between you and the problem at hand without putting a permanent solution in place because there is no permanent solution at the time or the solution isn't suitable for your environment needs. I don't see mitigation as something different; I see it as a step in the lifecycle of remediation.

Chris: Remediation is something that effects a desired state change, allowing you to achieve some desired end state on that system. This could be a policy, as Mike mentioned. Mitigation may not completely address the core issue you are going after and may only last temporarily. The difference is based on effectiveness.

Joe: I don't like it because we haven't clearly defined the terminology. In my mind, remediation is something that completely reduces the risk of successful exploitation to zero. Mitigation exists on a continuum of reducing the risk. I don't think this definition goes far enough. If remediation is meant to encompass mitigation then I don't think this definition goes far enough. There are changes external to a system that can mitigate the risk of that system as well. If there is an accompanying definition for mitigation then I think we are going to constrain remediation only to the target system, thereby disregarding other possible factors (firewall policy, web filtering, administrative things).

Chris: Some people may argue over whether or not one solution has been mitigated or if it has been remediated (someone may prefer a stopped service vs. uninstallation).

Gerry: What do you consider a remediation? Is patch installation? Uninstallation? Stopping a service?

Mike: Regarding the first question, I think remediation is a subset of mitigation. Remediation is a 100% solution for mitigation where other mitigations may not be. If I uninstall software, that would be a remediation. If I patch a firewall on the outside and do nothing on the vulnerable system, that is a mitigation.

Do patches fit into remediation? We have patch management to deal with CVEs but not with CCEs. We need a solution to fix CCEs. When we get that done then we can combine CCEs, CVEs, and perhaps CWEs together and have a complete solution. Whatever you call it, it doesn't matter as long as all three issues are fixed.

Kent: It is definitely a terms issue that we need to address (how people look at this). I see mitigation as short term "blockers" and remediation as the overall action of fixing problems within the network. Patches are a remediation action, as well as uninstallations and stopping services.

Joe: I agree. A remediation is the 100% mitigation of a risk. It would be easier to differentiate a remediation from mitigation if you said that remediation is something that completely eliminates the underlying weakness as opposed to reducing the exposures. A patch that fixes a software flaw is a remediation. Disabling a service just limits access to the still existing weakness.

Chris: Following that example, we are looking at the intent. The same service could be disabled as a configuration policy. You could be referencing the same CRE to a different purpose and I don't know how you would label it at that point.

Joe: So if I change my password policy from ten to 12, have I remediated intrusions from 80% to 100%? We won't know. Configuration is more of a mitigation than remediation; if remediation is a 100% risk reduction.

Mike: There are over 80,000 CPEs, all of which have configuration items. If you want a CRE for every configuration item and piece of software you have, you will have an unmanageable database. If you then add a CRE for every vulnerability, CWE, and mitigation possible, it will be even harder to select from.

Jim Ronayne (NSA)**:** Since the IDs are namespaced, there is no good way to control content. Each vendor will do what they think should be a CRE, and we have to deal with that later when processing results.

Dave Waltermire (NIST)**:** This whole definition discussion seems like a derailing exercise. Measuring effectiveness and risk are the critical factors when determining if something is a remediation or mitigation. I think the important distinctions are the different types of remedy actions that we want to be able to standardize around (managing patches, software installation, etc…) and this is what we should be sorting out today.

Gary Gapinski (NASA) Would an improvement be precluded from a consideration from under the as yet to be defined concept of remediation?

Mike: I like Dave's statement that it is a remedy, not any of these other terms. We are looking at remedies, but we need to corral what those remedies are composed of or we will have an unmanageable number of remedies, and we won't know until we are already working on it what other problems we have to address.

Gary Gapinski (NASA) While we are arguing over what the right word for this is, what I am looking at is a means by which to perturb the posture of a system, regardless of whether it is to correct a fault or to affect something beneficial, and it is likely to be used for both. Is there anything that is excluded from consideration under CRE?

Beneficial meaning a configuration change is found to increase performance by 20% or to decrease system resource consumption by 20%

Mike: That is not a security issue, and the original intent was to deal with security issues.

Gary Gapinski (NASA) Understood, but the process of making such a change is just as complicated

Jon Baker (MITRE): Over time, the definition of what gets a CVE or CCE changes over time. Bearing that in mind, maybe that definition is good enough for now and we will immediately be faced with this question again, but we will be able to more succinctly answer this in context.

Kent: (in response to Gary) It is a valid question. Being a remediation vendor, the reality is from what we have seen is that when you build a product that can be used for multiple things beyond security, customers will use it for that. If we do the standards right, we shouldn't limit the ability of tools to do some innovative things just because we are trying to maintain a specific definition that will most likely change at some point in the future. Limiting functionality is not necessary, so long as we achieve our main goal.

participant: I agree with Dave that the definition is somewhat pointless. In the end we are really talking about a configuration instruction, so talking about semantics of remediation or mitigation just limits scope and intent of the standard.

Joe: My original suggestion for this standards name was the Open Remediation Modification Language because within the DoD, we bought a patch management tool for our network defenders and later found out everyone that did patch management already had their own management tools with different groups of people making uncoordinated changes on the network.

Matthew Wojcik (MITRE)**:** (via email) Perhaps a better question to get peoples thoughts on is what should be assigned a CRE. Regardless of how you define remediation, workaround, mitigation or fix, what kind of thing do you want to have CREs for and therefore be usable in this model?

Chris: I look at having it having a more general purpose capability. We shouldn't be creating this definition to constrain how it might be used.

Joe: If we write a spec that says we will support system modifications only if they are security related vs one that covers non-security related modifications as well, someone will more quickly buy one that covers both.

Kent: With other enumerations we have learned and evolved the meanings of how those are addressed and used. We still haven't put any real meaning into this spec yet, and I think we need to experiment. I think we need a lot of experience in this area of network change because it is very different from just reporting. I used to have nightmares about blue screening the missile defense system and with remediation we can to this. Assessments don't have this risk.

Mike: Having a CRE for each CCE would be very useful, and we will eventually want them for CVEs as well. We do address CVEs with patches whereas we do not currently address CCEs with a fix.

Dave Waltermire (NIST)**:** Regarding the CWE issue, there have been studies that have shown that configurations can reduce the risk of undiscovered vulnerabilities. From that perspective, if we are able to identify what some of those relationships are between configurations and weaknesses, I think we could accomplish a lot using CREs associated with CCEs.

Kent: So CCEs first, CVEs second, and CWEs third.

Open Questioning

Jon Baker (MITRE): (to Kent) With CVEs we are enjoying wide industry adoption. Since you have a commercial solution, would you go back through your data set and map in CRE IDs?

Kent: We did it with CVEs, but I can't give you an answer just yet. I'm hoping this community can move faster with standards-based remediation because our current product is too proprietary. I would like to start developing the next generation of remediation product using these new standards. It may be more beneficial to work forwards to a new product rather than mapping into an old product.

On another note, CVE is on an opt-in participation basis from an international standpoint at this time. We need to try and address some international issues we are facing if we want these standards to work worldwide and not have the rest of the world go in a different direction than where we are going.

CPE

Overview

Brant Cheikes (MITRE) presented a high-level update on recent CPE events. Specifically:

- All four CPE 2.3 specifications have been released by NIST for public comment. They are posted here: **http://csrc.nist.gov/publications/PubsNISTIRs.html**
- Public comment period for all CPE 2.3 specifications closes 24 June 2011.
- CPE 2.3 will be included in SCAP 1.2
- Reference implementations for CPE 2.3 are under development and will be announced and posted when ready.
- The CPE Official Dictionary has been wholly operated and maintained by NIST for well over a year. The dictionary currently lists ~33K CPE names, and receives hundreds of new entries every month. A dictionary search capability is available here: http://web.nvd.nist.gov/view/cpe/search. Growth statistics are available here: http://nvd.nist.gov/cpe-stats.cfm

Brant briefly reviewed the CPE 2.3 development history. Discussion started at ITSAC 2009, requirements were developed during the Feb 2010 Developer Days session, the CPE Core Team formed in March 2010, and the bulk of the development work was done during the period March thru July 2010, with ongoing refinement since then.

The most significant changes to CPE 2.3 include:

- CPE 2.3 takes the form of four (4) separate but related specification documents. The specs are conceptually organized in a stack: Naming on the bottom, Matching building on Naming, and Dictionary and Language building on both Naming and Matching.
- The Naming specification introduces the concept of the Well Formed Name, and defines two name "bindings" for machine exchange: the 2.2-style URI binding, and the new "formatted string" binding.
- Four new name attributes were introduced in CPE 2.3: software_edition, target_sw, target_hw, and other.
- The Naming specification provides the infrastructure for single- and multi-character wildcard characters, and the Matching specification uses this infrastructure to define how source names that include wildcard characters may be matched against target names without wildcards.
- The Matching specification breaks the name matching functionality apart into a set of functions.
- The Dictionary specification calls for the Official Dictionary to contain only so-called "identifier" names, i.e., concrete names that are as fully specified as possible. In addition, there is an enhanced deprecation system, support for recording name provenance, documentation requirements, and a distinction between the "Official" dictionary (maintained by NIST) and "extended" dictionaries which may be maintained independently of NIST.

After the summary updates, Brant brought up three specific topics related to the new Naming specification:

- Name bindings: URIs and Formatted Strings
- "Packing" new attributes in a v2.2 URI
- Usage of special characters

Topic 1: Name Bindings

The discussion of Name bindings focused on the similarities and differences between the two binding forms (URI and formatted string). The objective of this part of the session was to review these forms with the attendees and solicit any comments or feedback. During this segment, the following Q&A occurred:

> Q: In the formatted string binding, is it the case that all special characters have to be escaped except '-' (hyphen) and '.' (period)?
>
> A: In the formatted string binding, all printable non-alphanumeric characters must be escaped when embedded in the formatted string except the hyphen, period, and underscore. Special characters may appear without quoting when it is intended that they retain their special meanings.

This exchange prompted a number of comments and suggestions about quoting in the formatted string binding. Some suggested that all printable non-alphanumeric characters be allowed to appear without quoting except for a handful like ':' (colon, the field separator), asterisk and question mark. It was noted that we need a quoting mechanism so we can distinguish special characters intended to have special meaning (e.g., single and multi-character wildcards) from characters that just represent themselves. Brant agreed to take another look at the use of quoting in the formatted string before the Naming specification goes final.

Aside from the comments about quoting printable non-alphanumerics in the formatted string binding, the general consensus was that the name bindings as defined in the CPE 2.3 specification were acceptable.

Topic 2: "Packing" new attributes in a v2.2 URI

This segment of the session focused on how the five edition-related attributes (legacy edition, software_edition, target_sw, target_hw, other) will be "packed" into the edition component of the 2.2-style URI binding. The essence of the approach involves using the tilde character as an internal field separator. No concerns with this approach were expressed. However, it did provoke the following question:

> Q: If we go to say, 2.4, any concept of how to expand formatted string?
>
> A: We acknowledge that a weakness with the 2.x naming approach is that we need to expand the names (and change the specifications accordingly) whenever we want to add new attributes. This cannot be corrected within the 2.x line. NIST has been discussing the creation of a "metadata

repository" as a way of associating an open-ended set of attribute-value pairs with, e.g., CPE identifiers.

Topic 3: Handling special characters in the URI

This segment focused on how the two newly-introduced special characters would be encoded in the 2.2-style URI. The current approach involves mapping the special-character '?' (question mark) to the "%00" percent-encoding form, and the '*' (asterisk) to the %01 percent-encoding form. By introducing these two percent-encoded forms, we are able to ensure that there is a way to encode special characters in both URIs and formatted strings. The choice of the two forms is almost arbitrary within the otherwise unused ASCII characters.

One of the attendees pointed out that %00 maps to the null character, which could erroneously be interpreted as the C-language string terminator. There was some agreement that this could be a problem, so we provisionally agreed to shift the choice of encoding forms to %01 and %02.

This concluded discussion of the three Naming-related topics. We then turned to three Dictionary-related topics:

- Elimination of non-identifier names
- Elimination of abbreviations
- Use of new attributes

Topic 4: Elimination of non-identifier names

Brant explained that the CPE 2.3 Dictionary specification restricts the Official Dictionary to so-called "identifier" names only. This represents a significant change in dictionary practice. The v2.2 dictionary contains a significant number of name "prefixes", i.e., shorter versions of other names in the dictionary, e.g.,

cpe:/o:microsoft:windows_xp
cpe:/o:microsoft:windows_xp::sp1
cpe:/o:/microsoft:windows_xp::sp1:pro

According to the 2.3 Dictionary specification, only the third name would be included, and/or the previous names would have to be revised to include appropriate hyphens, e.g.,

cpe:/o:microsoft:windows_xp::-:-
cpe:/o:microsoft:windows_xp::sp1:-
cpe:/o:/microsoft:windows_xp::sp1:pro

Brant noted that NIST and MITRE have not yet fleshed out plans for rolling out the CPE 2.3 dictionary based on the 2.2 dictionary. In principle, this process could involve deleting or deprecating shorter names which match other longer names already in the dictionary. This discussion prompted quite a bit of debate:

Q: What do we deprecate the shorter name to?

A: We can deprecate either to nothing (i.e., when a name is deleted) or to many other names.

A: One option is to not deprecate, but add a new name and remove the short one. Name is still valid, just longer.

Q: Too many things will break. Content authors group things by platform. We need to look at content, this could have an adverse effect.

A: Why? There should be no change in content.

Q: Because if you take it out of the Dictionary, it is no longer there.

Q: The CPE name is still a valid abstraction of a thing that matches something in the dictionary.

Q: We need to look at this proposal more carefully or we risk breaking existing content.

Q: What about a single winxp with no version? That is valid until there are more versions, would you remove prior entry?

Q: CPE needs to provide common names for interoperability. We're on a slippery slope here if you want other products to consume what you represent. Be careful when using official dictionary compared to internal dictionary.

The consensus of the group was that there are some concerns with the proposed 2.3 Dictionary policy, and that we need to review it carefully before implementing it to ensure that we don't break existing content.

Topic 5: Elimination of abbreviations

The 2.2 specification recommends use of abbreviations in CPE names. For example, we use "pro" rather than "professional", "std" instead of "standard", "ie" instead of "internet_explorer". While this approach makes names much shorter and possibly more readable, it potentially thwarts efforts to partly automate mapping of product "signatures" discovered on endpoints to their CPE names. So we're considering reviewing the existing CPE dictionary to find all those names which contain embedded abbreviations, and then convert (deprecate and add revised names) those names into names with the abbreviations spelled out. The main cost to end users is that names become longer, and there is some effort required to deal with a potentially large number of name deprecations. The deprecation mechanism ensures that there will always be pointers from the old (deprecated) names to the new (longer, without abbreviations) name.

There wasn't too much discussion around this issue—in general there was a sense that there's a potential to do some good here without much impact on end users. However…

Q: Have we reviewed SCAP content that uses CPE in benchmarks? Will we have to change that content?

A: Before submitting new names let's see if there will be impact to existing benchmarks.

Q: If for some reason, a particular abbreviation like "ie" is not expanded for a given name because existing content would be impacted, would we try to maintain consistency across that set of abbreviated names?

A: Possible solution is to use the longer form of the name in future names, but retain the abbreviation in existing names.

Topic 6: Use of new attributes

We've added several new edition-related attributes: software_edition, target_sw, target_hw. We've given them bare-bones definitions. Now we need to take advantage of these in the dictionary. Rather like the case with expanding abbreviations, this work entails going thru the dictionary, finding all the names that provide values for the edition component, then developing proposals for how to split that information usefully across the new attributes. This could result in another batch of name deprecations/additions.

Luckily, we've determined that out of ~33K CPE names in the existing dictionary, only ~1650 names have any value at all specified for the edition component, and of those, there are only ~150 unique values. So this shouldn't be a hard task to execute. MITRE would develop the proposals and put them before the community for review.

Comments in response:

C1: If you end up developing "valid values" lists, could you make those "should" rather than "shall"? That is, not require that conforming implementations use those specific values. That could preclude auto-generation of interim CPE names.

C2: Same concern—review names for potential impact on existing benchmarks.

In summary, it shouldn't be hard to look over ~150 names to come up with a good way of regularizing values for several different edition-related fields. Once this is done, about 1650 names could be affected.

Friday June 17th

OVAL for Artifact Hunting

Not yet available.

CCE

Not yet available.