CPE Developer Days Winter 2010 Workshop Minutes

Date: February 22, 2010
Location: NIST, Gaithersburg, MD

Attendees

First Name	Last Name	Organization
Scott	Armstrong	Symantec
Jonathan	Baker	MITRE
Ross	Barrett	Rapid7 LLC
Steve	Boczenowski	MITRE
Harold	Booth	NIST
Andrew	Bove	Secure Acuity
Andrew	Buttner	MITRE
Brant	Cheikes	MITRE
Paul	Cichonski	NIST/Booz Allen Hamilton
Jeff	Cockerill	G2, Inc./SCAP
Sudhir	Gandhe	Telos
Vladimir	Giszpenc	DSCI
James	Goodier	L3
Katherine	Goodier	L3
James	Hanson	Tenable Network Security
Rob	Hollis	ThreatGuard, Inc.
Christopher	Johnson	NIST
Tim	Keanini	nCircle
Joshua	Keich	Redseal Systems
Matthew	Kerr	G2 Inc.
Mike	Kinney	DoD
Verda	Kosnett	VPC Solutions Inc
Jason	Mackanick	DISA FSO
Rafael	Marquez	IRS
Christopher	McCormick	SCAP/NIST
David Michael	McKinney	Symantec
Adam	Montville	Tripwire
Joe	Nardone	Symantec
Gary	Newman	Belarc
Mary	Parmelee	MITRE
Alan	Peltzman	DISA
Marius	Popescu	SNC-LAVALIN ECS
Ryan	Poppa	Rapid7 LLC
Dragos	Prisaca	Symantec
Jim	Ronayne	Sparta
Charles	Schmidt	MITRE
Shane	Shaffer	Security Automation Technical Director
Douglas	Taylor	Federal Reserve Information Technology
John	Tebbutt	NIST
Blake	Thomas	Federal Reserve
Peter	Tracy	Belarc
Josh	Turpin	Symantec
Rich	Walchuck	Tenable Network Security
Eric	Walker	Bigfix, Inc.
Matthew	Wojcik	MITRE
Joseph	Wolfkiel	DoD

Segment 1: Introduction and Overview

Part 1: Mr. David Waltermire gave an introduction and welcome to the overall event

Part 2: Mr. Brant Cheikes - CPE Introduction and Overview
Mr. Cheikes explained today's goal as an interactive experience which will help us understand what it will take to position CPE to better serve the community. Mr. Cheikes summarized what the group will cover in the Introduction and Overview segment:

- describing target milestones;
- discussing process issues;
- describing how the workshop activities fit into longer-range plans;
- discussing survey results;
- setting participant expectations.

Mr. Jim Henson requested that Mr. Cheikes give a two or three sentence overview of the purpose of CPE. Mr. Cheikes responded with his personal perspective that CPE was created to solve an interoperability problem. CPEs, he explained, were intended to provide a common ID for products; to provide a means for scanning (asset inventory) tools to describe what they found; and to allow that to be correlated with vulnerabilities, etc. We have come to understand that the community is using CPE as a solution to problems that we did not originally envisage.

Mr. Cheikes explained the CPE Core Team's perspective: compared to other security automation standards, CPE is not succeeding as well as hoped. The goal today is not to propose solutions, but rather to listen to the community's views about the shortcomings of CPE in practical applications and how to improve it. That is why the group is here.

Mr. Cheikes summarized what we want to accomplish today:

- defining the near and longer term needs for the CPE standard;
- identifying and prioritizing potential changes;
- determining next steps for moving the standard forward.

Describing target milestones: Mr. Cheikes outlined two major milestones for CPE development.

1. We have a near-term objective of updating the spec to version 2.3, which fixes as much as we can without breaking backward compatibility. Maintaining backward compatibility means that version 2.2 content would still be considered valid in version 2.3. However, we still want 2.3 to make CPE more useful. We hope to identify things that fall into that category.
2. We suspect to achieve complete success we will need to revise to 3.0; however, this will require major changes that will break backward compatibility. We need to have a better understanding of the problems that will present for the CPE community. These are conceptual milestones, to explore the possibilities for version 2.3 or 3.0. This group might ultimately decide it is not productive to update to version 2.3, which means that everything is on the table.

Discussing process issues: Mr. Cheikes explained the timeline constraints associated with CPE changes. Whatever we may do with version 2.3 needs to be stable by July 2010, which is a very short timeframe. The next major release, 3.0, will not be possible until at least a year later. These deadlines are imposed by the SCAP lifecycle.

Describing how the workshop activities fit into longer-range plans: Mr. Cheikes gave an overview of how the workshop fits in to the CPE development process. The CPE Core Team distributed a survey to the CPE community prior to the workshop. Participants will review the results of the survey as a starting point for workshop discussion. The group will evaluate, revise and add to the results. After the workshop the Core Team will provide a report on survey and workshop outcomes. Then the Core Team will draft the next CPE specification. This may include a minor version revision to 2.3 or a major version revision to 3.0. The community will review and give feedback on the draft via the CPE discussion list.

Mr. Cheikes went on to explain that presently the Core Team is comprised of MITRE and NIST with help from the DoD; but extends an invitation to anyone who wishes to become part of Core Team. We understand that developing standards is not the community's primary function -- there is a Core Team in place to move this forward on its behalf. However, we believe that in order for CPE to succeed, we need more direct community involvement to define the standard and set its technical direction for the future. MITRE and NIST cannot accomplish this alone.

Mr. Cheikes further explained the interactive nature of the workshop. Going on the assumption that most everyone in the room has some familiarity with CPE, attendees are here to provide assessments of what has worked and what has not. This benefits the Core Team, which did not come to merely offer solutions and listen to feedback; rather, the Core Team used the survey to guide plans for today, but are not limited to the survey. There will be new ideas that arise during the course of this workshop.

Discussing survey results: Mr. Cheikes gave an overview of the CPE Stakeholder Survey results. The top challenges for implementing CPE focused chiefly upon Implementation and Best Practices, while the top Critical Issues consisted of CPE Specification Changes, Best Practices, and Dictionary Content issues.

Mr. Waltermire asked if there is there a severity difference between "challenges" and "critical issues". Mr. Cheikes explained that critical issues were intended to represent "show-stoppers". Challenges are important but not necessarily identified as "do or die".

Setting participant expectations: Mr. Cheikes made a distinction between "consensus" and "agreement". Consensus is determining what you can live with vs. agreement on everything that needs to be done to improve CPE. Today the group should think about aiming for consensus.

Mr. Cheikes requested that participants make their recommendations as specific & feasible as possible. He noted the group should keep Occam's Razor close at hand. The path to success may involve simplifying CPE, rather than adding multiple layers of new features. Improving the standard may involve altering or even disposing of portions of it. For instance, one can achieve backward compatibility in a version 2.3 by relaxing/eliminating requirements that were imposed in prior versions. Is there a simpler notion of CPE that can serve as a core for a broader range of use cases?

Mr. Cheikes opened the floor for questions and comments: Mr. Keanini suggested that the group keep in mind there is always the issue of intellectual property. The common practice with all security automation standards is to resist submitting something within one's patent space. Mr. Cheikes agreed and added the group needs to focus on cross-platform standardization, leaving to the vendors all those design choices which should be sorted out in the marketplace.

Dr. Katherine Goodier asked if the standard is open to international users. Mr. Cheikes replied it is completely open and public. Mr. Waltermire added we are not trying to solve these issues for just one community (i.e., the Federal Government); we are trying to solve this for all communities which share a common set of use cases.

Mr. Jon Baker asked what distinction is being made between "quality assurance" and "dictionary content" (referring to a displayed slide describing ten categories of survey responses). Ms. Parmelee responded that Quality Assurance is related to practices, measures, and common processes, while dictionary content is more specifically about what should be included in the dictionary.

Mr. Waltermire asked if QA includes validation. Ms. Parmelee explained that this is generally true; however, these categories are an attempt to group the survey results into common themes. The Core Team made a decision not to address specifics of dictionary management today since the topic was too complex and time-consuming.

Mr. Cheikes invited CPE's primary sponsor, Lt Col Joseph Wolfkiel, to say a few words. Lt Col Wolfkiel reported the DoD has a critical interest in the success of CPE. He is hoping for a set of Developer Days workshops every four months. He is concerned because DoD is trying to do enterprise-wide deployments using CPE. Much of DoD's use of CPE is failing. Standards cannot survive with ambiguity and there is ambiguity in the current CPE specification. CPE implementation means different things to different factions. Vendors say "we use CPE" but that has many interpretations. For the DoD, CPE fails because the standard has not become usable. DoD needs a standard that supports compliance, network inventory, federated querying, and is scalable — and something is needed in the near term. If DoD cannot make CPE/SCAP succeed, then funding may disappear and the group may not get another opportunity to do this for a while.

Part 3: Ms. Mary Parmelee - CPE Workshop Activities Overview
Ms. Parmelee described a high-level process diagram of the day's activities, including everyone's role in the process. She chose moderators to help keep the conversation on track and on time. Finally, she reviewed the workshop ground rules. Ms. Parmelee went on to explain that even if participants have not read the specification, they can still give useful input. We need help from the community in order to drive the standard in the right direction. For each major segment we will start with a seed list of survey results that came directly from the CPE Stakeholder Survey.

Segment 2: Describing CPE Community Needs

Ms. Parmelee began the discussion (Handout circulated to the participants, summarizing critical issues and challenges for CPE.) The handout comes directly from the survey. These are the top needs -- apart from dictionary content management -- that we are hoping to address today. First we want to go through each of these and make sure we all agree on what they mean and then take a vote for priority. Based on survey inputs the following list emerged:

1. Improve methods for aligning (e.g. matching and mapping) CPEs with platform-related data. E.g. software names, software versions, internal data structures, host ID, registry signatures, matching unauthenticated scanner results.
2. Improve specification change management. E.g, Ensure that adopters are not expected to make radical changes in short periods of time. Ensure that changes are based on balanced community needs rather than a few special interests.

3. Improve the breadth of community participation. Especially non-Security automation software manufacturer participation.
4. Improve compatibility across SCAP-related standards. E.g. Standards are currently being developed in silos, currently cannot use CPE to restrict OVAL checks.
5. Clearly define dictionary scope and intended usage. E.g. contain all software, contain software with known vulnerabilities, a check (OVAL or other) for every CPE, a list of product names.
6. Improved implementation support for CPE. E.g. improved documentation, more examples, shared proof of concept code
7. Improved naming convention. E.g. how to handle vendor for open source, automatable conventions, CPE for historic releases, reserved names for product, vendor, centralized vs. shared distributed conventions
8. Make the CPE more extensible. E.g. extensible, language schema, enable inclusion of additional information in a CPE, enable support for web applications, vulnerability management.
9. Make the CPE spec more flexible. E.g. separate business logic from enumeration, remove format related constraints, support more than just exact matches.
10. Improved version handling. E.g. ensure cross-platform version consistency, standardize on version granularity, centralized standard vs. shared distributed standard, defining clearer method for defining versions in CPEs.

Mr. Keanini asked how much overlap there is between these points and the last few years of discussions on the mailing list. Ms. Parmelee responded that the Core Team didn’t conduct a detailed comparison. However, the Core Team has a good idea of what the major issues are from more than just the survey results. The survey is intended to start conversation.

An unidentified participant remarked there are lots of participants from lots of places here, and that not all voices seem to be equal. The voices that really need to be heard are the product’s end users. The participant suggested a discussion of CPE’s uses should take place before a participant forum to address those needs.

Ms. Parmelee responded that the engineers who create software that use CPE are end users, and the people using that software are indirect CPE users. There is no assumption that a representative sample of the community is present. The outcome of this workshop and the survey is a straw man list of community needs and potential target capabilities. The next draft of the specification will yield broader community feedback. The Core Team will ensure it has a representative opinion before they change the standard.

The Core Team reviewed the list of critical needs collected in the survey and identified the most important issues. These were:

- dictionary scope and usage
- improving the naming conventions
- improving version handling
- improving compatibility across SCAP-related standards.

Ms. Parmelee began the conversation by explaining that the group needs to clearly define the future scope of the dictionary. A lively discussion ensued which touched upon the critical issues identified as being important for discussion. Details for Segment 2 are recorded in Appendix A of this document.

The outcome of the group's analysis during Segment 2 identified the following list of CPE-related critical needs:

1. The CPE dictionary should be comprised of a canonical list of product names/IDs
2. The CPE Dictionary should include a "Comprehensive" product enumeration
3. There should be a way to represent/describe products before entering them in to the curated repository
4. The process for curating CPE product names should be a federated process
5. The CPE Specification should focus on the "what" of product names, and not the "how" of implementing them for a specific use case?
6. The CPE name should contain a namespace component to enable federated curation
7. There needs to be a method for mapping between namespaces to connect federated CPE collections
8. The required URI format is a barrier to adoption the percent encoding problem is an example of why the URI format is a problem

Segment 3: Describing the CPE User's Greatest Barriers to Use

Ms. Parmelee asked for a show of hands of participants who are currently using CPE. She then asked each of these people to briefly and succinctly describe the single greatest barrier to using CPE effectively. The group identified the following CPE implementation barriers.

1. There is no authoritative name for software -- e.g. there, are thirteen different names of Opera. Which CPE do I need? The group needs a defined way for people to name things. This problem leads to an inability to generate a name that is correct.
2. Percent encoding in the URI. There is too much meaning being put into the identifier in general.
3. When different vendors version differently we need different ways to encapsulate this information. We should pull version information out of the identifier. The strict version numbering requirements are a problem. There are many ways to describe the same version of a product in some cases. Maybe the solution is to alias version numbers. Maybe break version numbers into several fields. We need better support for matching of versions in particular. The URI, largely because of the difficulty of encoding varying version formats. We need more granular capabilities for versions. One participant suggested that we may need around six version fields.
4. Human labor. Direct mappings are rare; many times lots of people manually figuring out mappings and this is labor intensive.
5. The mixing of naming and matching functionality in the URI.
6. The observed complexity of encoding versions extends to other fields. For example, the edition field. Overall, there is a lack of ability to express complex identifiers today. The prefix property is part of this problem.
7. CPE names are hierarchical but we are modeling a non hierarchical world.
8. Distinguishing the version of a piece of software vs. an update of that software. Currently we don't need all the detail in CPE fields.
9. Mapping what you are able to discover to an actual CPE name. Another way to ask this is, What is a feasible way to compute the CPE name? If we could make them more computable (repeatedly), we can eliminate some of the need for a centralized authority. We should think about this. We need to map what we are finding (software or paperwork) to an identifier.
    a. There may be things we can do to improve the situation. For example, when you discover a new product, don't map the version field into the CPE dictionary.

Instead, focus on vendor/product names. The spec doesn't say what part of the string you need to include. Versioning can be handled outside of the CPE name as a workaround.

10. The guidance in the document offers no help on selecting a vendor name. What happens when a name changes? Previous decisions produced a convention for this but this was never documented.
11. Much of what CPE is about today is mapping between data sets. Maybe we need to think about structuring data sets so we don't need an ID for mapping. In other words, tagging. Develop a decomposed set of tags for mapping data together. If everyone agreed on tags, we could do the mapping. Everyone could use their own IDs.
12. CCE doesn't use CPE because if a name is not fully specified it is unclear if it is meant to be inclusive or exclusive. I get controls and know that they are applicable to some version of a piece of software but not all. I cannot safely map to CPE because people might make the wrong interpretation.

Segment 4: Proposing Specific CPE Changes

Mr. Cheikes pointed out this part of the workshop will focus on the specification and how to change it in order to meet the needs identified in Segments 2 and 3. However, this is not intended as an exercise in group engineering – the group will only go to a certain depth. After this meeting, the Core Team will turn these into specific proposals. In looking over the issues, some common themes emerged. One theme is concern related to matching -- versions, primarily. Another theme is whether CPE names should be URIs or something else. Which should the group address first?

Lt Col Wolfkiel remarked that the point of CPE is matching, not identification. The only thing we do is match an installed product against vulnerabilities, licenses, etc. We want to find things on boxes and match back to what we know about them. URIs prohibits effective matching, so both of these topics are related.

Mr. Cheikes requested specific proposals for addressing each issue. The following were introduced:

1. Drop the prefix property, which doesn't break backward compatibility
2. Drop matching, which would break backward compatibility in terms of SCAP capabilities
3. Replace the matching algorithm, which would break all kinds of backward compatibility
4. If we drop the prefix property, we can make the URI an extensible string identifier
5. Replace the URI with an XML tagging approach. Would break backward tool compatibility, but not backward content compatibility. Product identification would be a product of matching tags to support set-based operations without the constraints that we have today.
6. Flatten the imposed hierarchy by decomposing the URI components and standardizing each component as a standalone module of content. This standardizes what the parts are, but not how those parts are packaged. This could be implemented in the form of making CPE a controlled vocabulary of product information e.g. product names, vendor names, etc..The group is not clear on how that would affect backward compatibility.
7. Break the specification into three or four companion specifications that address different kinds of functionality. The pieces of CPE are naming, matching, language, dictionary encoding and transport encoding. A stack-based approach was discussed that is similar to the way the W3C semantic Web Stack is structured. Mr. Waltermire believes that the rules for creating the CPE name should be separate from the transport rules. There would be a stack where higher level specs imposed requirements on lower level specs.

However, you could take out and replace any layer. The group is not clear on how that would affect backward compatibility.

8. Make some of the specification components optional. For example, if you are SCAP validated, currently you need prefixes and URIs and matching. This is not useful for some people. If you make it optional, that's fine. If you say you must do it this way, that's a problem. This could be done in a way that would not break backward compatibility.
9. Unpack the edition field to determine what should be included in that field, and where the extraneous values that are currently in that field belong. elevate target architecture (bitness), target software, and true edition to first class elements.

Discussion highlights: (see Appendix B on page 13 of this document for the detailed discussion)

Mr. Keanini noted there have been three proposals to move towards a tagging-based approach. Going this way accomplishes our goal of decoupling the functionality currently wrapped in CPE names. A CPE name should be just a name. Some prior proposals went to CPE mailing list and nothing happened. Ms. Parmelee responded that we are trying to apply a more structured process for addressing these concerns than was previously applied.

Mr. Cheikes observed that one approach is to enumerate a controlled vocabulary of components, such as vendors and products. Mr. Waltermire added that a controlled vocabulary is a common model people can map to so that we can use the same model to converse instead of the relying on having same identifier. Additionally, we could start implementing this using existing CPE infrastructure tags today.

Mr. Waltermire had prepared slides on NIST's tagging approach and he offered to present them to the group. Several participants expressed interest and so Mr. Waltermire presented a summary of the NIST tagging proposal. The key ideas behind this approach are:

- Separate identification from matching. ID is a string and the tags become the basis for matching. Eliminates the prefix property and replaces it with structured metadata. This allows decomposition of existing metadata and federated creation.
- Tags are name-value pairs. We aren't changing the core information in CPE. CPE names become lists of tags. Tags are applied to CPEs or to other data structures. We use a tag dictionary for tag creation and maintenance.
- Tag definitions have a name and source. The source is a namespace. Tag definitions have metadata to track revision history. The "Applicable to" field – defines relations between tags.
- Revised matching algorithm. Matching is now done by tags.

Mr. Cheikes noted that when this was originally proposed it was early in the CPE 2.0 lifecycle. The community consensus was to let the existing specification settle and see how it works out.

The group agreed that the tagging approach was worth further pursuit.

Summary of Proposals for changes to the CPE specification:

1. Drop prefix property
    a. Dependency: drop matching algorithm This causes no standard way to match
2. Drop the URI format requirement
3. Separate the enumeration from the language

    a. Make all but names optional
    b. Separate core naming, matching and language specifications
    c. Complexity – the language is in use today
4. Move to a stack-based design of specifications
5. Break up CPE names into a controlled vocabulary
    a. Enables computable relations between components
6. Allow component-based matching
7. Adopt a tagging approach to CPE
    a. Dependency – drop prefix property
    b. Requires new tools (but content is backwards compatible)
    c. Requires tools to parse components of a CPE name and transform components into tags
    d. Add new facts to current CPE schemas
    e. Stand up a sand box for community contributed tags
    f. Propose a technical working group to explore application of tagging proposal

Segment 5: Devising a Plan of Action

Mr. Cheikes asked the group how to best capitalize on the day's accomplishments. The Core Team planned to capture the notes and major issues and document today's activities, which will be disseminated in a few days/weeks. Today we want to talk about how to get the community to move the standard forward. If we want changes in 2.3 we only have a few months and we'll need more involvement. Usually the way to do this is technical working groups who can turn these ideas into actionable changes. The question then arises: will technical working groups be able to meet our timetables? Mr. Cheikes then asked for participants in a working group; approximately 12 attendees voiced their interest. Mr. Cheikes reinforced the concept of need for community involvement.

A participant asked how we will resolve the voting question. Mr. Keanini noted that the only SCAP standard that has successfully implemented a governance process was CVSS. In this case a committee would propose and the community would decide. Lt Col Wolfkiel noted that we did this before and sent a proposal to the list and no one voted. Mr. Cheikes suggested that we could have a working group to decide how voting worked. Lt Col Wolfkiel responded that we should pick an existing governance process now and use it. A participant asked why all the SCAP communities aren't using the same governance process.

The group identified three urgent areas for working group participation: governance, tagging approach, and versioning. Implementation support was also of great interest, but was of less urgency for helping support CPE version 2.3 decisions

Mr. Cheikes reminded the group of the goal to hold another workshop in four months. He asked if there should be intervening meetings, web conferences, or other communication in the interim? The group decided to have a web conference in two months. It was agreed, to begin forming the working groups as soon as possible.

Appendix A. Detailed "Describing Community Needs" Discussion (Segment 2)

Ms. Parmelee began the conversation by explaining that we need to clearly define the scope of the dictionary. There were people who think that the dictionary should contain an entry for every piece of software on the planet. This wasn't the original intent for CPE. We never intended CPE to have a comprehensive list of every software product.

Lt Col Wolfkiel – Whose original intent does this reflect?

Ms. Parmelee – The specification states that if there is not a CPE name then people can propose it. This implies that there is no need for the list to be comprehensive since it can always be extended when needed.

Lt Col Wolfkiel – So there could eventually be a comprehensive list.

Mr. Cheikes – When spec was developed, this wasn't clearly articulated.

Ms. Parmelee – Who in this audience thought this was supposed to be a comprehensive list of vendor products?

Mr. Hollis – Intent may be irrelevant. The issue becomes the process of getting things into the dictionary. If this is good, then the comprehensiveness of the dictionary at any given time isn't as much of an issue.

Ms. Parmelee – Our objective here is to scope the dictionary and determine what should be in it.

Mr. Cheikes – Is there a need for a comprehensive (within the bounds of reality) dictionary? The alternative is a dictionary that only contains software with reported vulnerabilities.

... – I disagree that this is the only alternative.

... – Going back to Rob's point: if there was a good process to create new CPEs this would lead to making the list more comprehensive naturally.

Mr. Hollis – No one is going to say that some things shouldn't go into dictionary.

Mr. Neuman – I don't understand where this question is coming from. The goal was to come up with a common name to organize what are currently disparate names. CPE isn't about what should or shouldn't be named. I would suggest this is the only criteria: If there is a need for a common name, give it a CPE.

Ms. Parmelee – How many people think the way that names are contributed now are useful?

Lt Col Wolfkiel – It would be useful for people to explain the process by which names come in.

Mr. Neuman – The process looks really broken. If you look at the NVD and the CPE dictionary this doesn't match.

Ms. Parmelee – We aren't discussing the management of the dictionary today because it will take all day. We are just discussing what should be in the dictionary.

Lt Col Wolfkiel – One criterion for inclusion would be a list of software that we need to share across products for vulnerabilities.

... – Why would you need to wait for a vulnerability to be reported?

Lt Col Wolfkiel – We don't. This is just a start.

Mr. Wojcik – It is also relevant for compliance and inventory. These could also feed the list.

Ms. Parmelee – It looks like we are saying the CPE needs to be as comprehensive as possible.

... – If I used an English dictionary with the same coverage as CPE we couldn't use it. Dictionary should be comprehensive enough to cover all currently known software with a system to add new ones.

Mr. Waltermire – Given a set of all products, any given communities of interest will only have an interest in a subset of those products. Some products only exist within a given organizational boundary. Should these be in a dictionary? Or should these exist only in the context in which they are meaningful.

Lt Col Wolfkiel – One way to phrase: CPE should include all commercial products vs. all software.

Mr. Waltermire – Commercially available is a trap: what about open source.

... – "common software"

... – "public software"

Mr. Cheikes – There is a need to describe a product found on an endpoint regardless of whether a canonical name exists. We still need to be able to correlate with other information. CPE today has confounded naming with correlation. We may be hearing that discovery is separate from the issue of having a standard name.

Lt Col Wolfkiel – This is off topic. We still are talking about what is in the CPE dictionary. What is the dictionary? All software products or just the products within a use case?

...- Does there need to be a list. If people can independently come up with same name there is no need for a list.

Lt Col Wolfkiel – If we only had one way to discover products this might work, but I don't think this is the case.

Ms. Parmelee – CPE is not just the name of a product, it is a description.

Mr. Keanini – Are you asking what metadata should be with the ID?

Ms. Parmelee – Should there be a check for every CPE name? What belongs in the dictionary?

... – Is this a CPE meeting or dictionary meeting?

Ms. Parmelee – All together.

... - Why are we not discussing the dictionary management process?

Mr. Cheikes – Because NIST is taking this on. The spec sets expectations: try to allow correlation with the dictionary. If community decides we need a comprehensive list, the job of accomplishing this goal (and decisions of how to do this) is levied on NIST.

... – Why are we even talking about what should be excluded? Is anyone saying there are things we should not allow in?

Vladimir G. (Vlad) – Should we just focus on the authenticated scanner?

... – Are people proposing that we should knowingly leave things out?

... – This discussion shows why we cannot separate dictionary from spec. We are unable to comprehensively map data to CPE dictionary. We are unable to use the spec to handle the ambiguity of what we are discovering. If the dictionary was exhaustive then we don't need the spec to handle ambiguity. If the spec handled the ambiguity, I don't need the dictionary.

Dr. Katherine Goodier – There seems to be a bias to English in the CPE dictionary.

Mr. Waltermire – That is not true. The dictionary is multilingual. CPE titles may be specified in any language.

Mr. Cheikes - Is there a need for the dictionary to be comprehensive?

...- Should anything be left out?

...- Organization-internal applications should not be included. However, there might be a need to support discovery of these applications. In this regard, the scope of CPE is not necessarily the same as the scope of the dictionary.

Lt Col Wolfkiel – We see that. We have DoD internal apps. If, however, we want to publish, we can't have people reject the publication on the ground that the software is organization-specific. Things get out. What is organization-internal today might not be so tomorrow.

Mr. Cheikes – Is there a need to express a name for things that don't have a curated dictionary name?

Mr. Waltermire – There is also a need to promote something from a local (organization-specific) state to a curated state.

Lt Col Wolfkiel – And allow deprecation. If there was something in the spec on how to create one's own list of ids we could keep all names internal until the first reported vulnerability and then submit to central repository for public use and deprecate our local name. This would allow scaling.

... – What about something like DNS to create federated repositories?

... - DNS is useful because others can refer to local instances. When we talk about local CPE repositories we are talking about things no one else will refer to.

Mr. Waltermire – There are many communities of interest for product names that we need to deal with. We need volunteers to populate the dictionary and that only gets us so far. We're trying to work with vendors, but not many are willing to contribute. We need a means to rely on third parties.

Ms. Parmelee –: Should there be names for vendors? Should there be a centralized list of names? How many people think it would be useful to have an (OVAL) check for every CPE?

... – What level of specificity? A CPE name is like a set. Not even a finite set. The OVAL def would need to detect set membership. It seems like this would be a huge amount of effort.

Mr. Wojcik – It wouldn't be practical to create a single OVAL for a CPE. Think of cross-platform apps. You would need many OVAL checks even for a single, fully qualified CPE.

...- The intention of CPE is that my inventory can be mapped to CPE labels. OVAL won't do that.

Mr. Keanini – I need an inverse functional property for a match.

Mr. Wojcik – Checks will be useful for some situations but not others.

Mr. Harrison – We need a way to create OVAL checks when necessary.

Mr. Waltermire – There are many ways to find product information: some open and some proprietary.

... – Previously when we have started talking about OVAL, people have said it was a separate problem.

Mr. Keanini – It is separate. The sensing of the system artifacts of a piece of software is not part of CPE.

Mr. Baker – CVE has a set of references to minimally identify the CVE entry. OVAL was added later to help identify the artifacts of the given vulnerability.

Mr. Keanini – CPE has a spectrum where sometimes specificity is useful and sometimes ambiguity is useful. OVAL is sometimes useful, but not as part of CPE spec.

... – CPE is just the common name. How to detect the software is separate.

Mr. Waltermire – This conversation is more like item 1 from the handout: – how to associate a CPE with other standards.

Mr. Wojcik – One purpose of associating a check with a CPE is not to provide a canonical detection, but to provide additional metadata to ensure we are talking about the same thing.

... – Is there any OVAL currently associated with CPE names?

Ms. Parmelee – Yes. They are optional. Should they be removed?

Mr. Keanini – They are in CPE as part of the spec dealing with matching.

DB – That's there say the CPE is present if the OVAL returns true. They provide a mechanism to give some way to say if the CPE is applicable to a system. They are meant as additional information to know what the CPE is talking about.

... – CPE is just a standard name. I don't want to see a requirement that OVAL accompany CPE submissions.

Mike Kinney (MIKE) – Until the list is complete, people have to either roll their own CPE names or not use CPE at all.

Mr. Hollis – Whenever you have a CVE without an OVAL you get an industry full of false positives.

Vote – Should CPE just focus on a flat enumeration or does the scope include matching? [2 people said it should include matching. Most everyone else said flat enumeration.]

Mr. Waltermire – Do we need a standardized process for defining how to match a CPE to a system entity?

... – Probably, but not part of CPE.

Lt Col Wolfkiel – Let's focus on the enumeration. The enumeration is the necessary first step.

Mr. Waltermire – We cannot eliminate functionality that is being used. CPE with OVAL information is used in SCAP.

Ms. Parmelee – We are just going to focus on the “what” (the enumeration) not the “how”.

Lt Col Wolfkiel – Part of the "what" of CVE is references to other sites. Do we need things like that in CPE? Maybe OVAL isn't the best way to do this, but it is a way.

Mr. Wojcik – The title and description of a CPE may not be sufficient in some cases. What else can we use.

... – The problem at hand is the creation of a common name, not what it applies to.

Mr. Wojcik – If we don't agree on what the name applies to, it defeats the purpose.

... – If we intend to identify the same piece of software, even if there is an error in that intent (that isn't the piece of software that we thought it was), we should say the same thing.

... – I would like my vendors to map discoveries to the name list. There is no need to tell vendors how to do this mapping.

Ms. Parmelee – But vendors often don't know how to map and they don't have resources to figure it out.

Mr. Keanini – We want a way to compute the membership of a name.

... – Should that be a separate spec?

Mr. Keanini – Possibly.

Ms. Parmelee – Method of detecting pieces of software gets into intellectual property issues.

Mr. Keanini – This is not a problem because we are mapping to a public namespace. As long as we are trying to talk about the same thing, how we get there is immaterial.

How can we federate CPE creation?

Multiple people – Use namespaces to indicate authorship. Entries without a namespace default to the NVD.

Lt Col Wolfkiel – If you have a local namespace but need to go globally, send to NIST and they can publish.

... – Won't every scanner have own namespace?

... – Yes. That is the way things are now.

Mr. Keanini – Libraries that interpret the URI vary. Escaping the URI may lead to problems of inconsistent interpretation. We need to treat the CPE identifier as a string. Processing it as a URI can lead to problems.

Mr. Cheikes - How many have found that the requirement that the name be URI is a problem? [A few people raised their hands.] There may be a need to change the name.

Lt Col Wolfkiel – It is not just %-encoding issue. The URI isn't extensible and creates problems in many other ways.

Mr. Waltermire – The fact that we are trying to store data in the identifier is part of the problem. Instead we should associate data with the identifier.

Appendix B: Detailed “CPE Specification Changes” Discussion (Segment 4)

Mr. Cheikes – We are going to change plans a bit in light of previous conversations. There is a demand for some focus on specific changes that we might make to the CPE specification to address the burning concerns. All conversations should be focused on the specification and how to change it. However, don't want to turn this into group engineering – we will only go to a certain depth. After this meeting, the core team will turn these into specific proposals. In looking over the issues, a few themes emerged. One theme is concern related to matching (versions, primarily). Another theme is whether CPE names should be URIs or something else. Which do we want to take up first?

Lt Col Wolfkiel – Observation: the point of CPE is matching, not identification. The only thing we do is match an installed product against vulnerabilities, licenses, etc. We want to find things on boxes and match back to what we know about them. URIs prohibits effective matching, so both of these topics are related.

Mr. Cheikes – I want to go around the room and looking for specific proposals.

Mr. Keanini – Some people think that a syntactical match is a way to compute equivalence. However, there are cases where the syntax is different but you must compute equivalence. This is semantic equivalence. We need to include both of these in the conversation. For example, semantics are relevant in aliasing and deprecation.

Mr. Neuman – I propose we drop the prefix property. Who would be impacted?

Mr. Cheikes – One of the things that the prefix property is intended to enable is the concept of sets of things.

Mr. Wojcik – Currently, the prefix property is how matching works. How would you do matching without it?

Mr. Neuman – We should get rid of the prefix property because products are not hierarchical.

Lt Col Wolfkiel – Building a data structure that mimics the prefix property is difficult. E.g. language doesn't depend on other fields.

Mr. Wojcik – Are you suggesting we find a new matching property?

Mr. Neuman – Sure.

Mr. Cheikes – Dropping the prefix property breaks matching as it is currently done.

Mr. Waltermire – Component matching relies on prefix property. I.e. a more specified component is a subset of a less specified component.

Lt Col Wolfkiel – Remove prefix matching and don't replace. If I want to match I can just do a search based on field. This lets me search without implying a hierarchy.

...- That works if you are looking for a single entity, but not if you are trying to define a set.

Mr. Cheikes - To summarize the proposals so far:

- Drop prefix property. Doesn't break backward compatibility.
- Drop matching. Doesn't break backward compatibility.
  - Mr. Waltermire – It would break backwards compatibility in terms of SCAP capabilities.
  - Mr. Keanini – This is dangerous because we exist in an open world. Something may show up tomorrow that shouldn't be part of the set.
  - Lt Col Wolfkiel – We can drop the prefix property and just match on fields and get same result. We just are no longer hierarchical.

- Mr. Wojcik – There are two places where matching is important: 1) I have info on stuff on network and I want information about that. 2) I have information about software and I want to tag it. You need both so that both workflows can be linked. There are different reactions to the open world problem. One option: if I want to tag information with a software name, regardless of how I indicate the set, have a way to say whether you mean all names, some names, or no names that match the name. Also have a date so that names created afterwards are marked as unknowns with regard to membership in that set.
- Lt Col Wolfkiel – That would be a new matching algorithm. That's big enough that it doesn't fit in 2.x. Instead, take matching out and let people do what they want. My other proposal is get rid of the URI format except to say it is only required in the dictionary submission process. Only the names would matter rather than their format.
- Mr. Cheikes – Maybe what you are suggesting that in the transition from 2.2 to 2.3 we allow alternative representations beyond the URI. Version 2.3 will not say that CPE is a URI. Instead, it is just a string.
- Lt Col Wolfkiel – Yes. There would be no obligation within a product to use URI.

Mr. Cheikes – To enumerate the current proposals (again):
- Drop prefix property
- Change from "URI" to formatted string
- Drop matching algorithm
  - Is there a proposed replacement match

Lt Col Wolfkiel – I want to drop the URI period. I want to be able to include information that doesn't fit in the URI.
Mr. Cheikes – Maybe in 2.3 you can add additional colon-delimited fields because you no longer need to worry about hierarchies.
Lt Col Wolfkiel – What is the value of using a colon as a delimiter?
... – I support using URIs because you can see at a glance what you have.
Lt Col Wolfkiel – I'm not saying the format shouldn't be human readable. Use XML or something like that.
Mr. Keanini – Escaping the colons in the URI will lead to problems. Dropping the URI, or interpreting a name as something other than URI fixes.
Mr. Cheikes – so proposal is just to stop calling it a URI and just call it a string. It may look the same but there is no implied interpretation.

Mr. Waltermire – I see 4 categories of backwards compatibility: content, tool implementation, functional relative to other spec, implementation issues related to inter-standard functionality. We need to keep all of these in mind.

Mr. Cheikes – Other piece from what Lt Col Wolfkiel was saying was a suggestion for an alternative, XML-based format.
Lt Col Wolfkiel – My proposal was even less than that. The only time you need to use the URI is when you are submitting to NIST. For validation, format should become irrelevant. I just want the names to perform matching and identify products.
Mr. Cheikes – How does that affect tool compatibility?
Lt Col Wolfkiel – In ARF I can pull information out of a CPE. It is still CPE, but I can use any transport format I want. Requiring that I transport CPE info as a URI limits me with regards to what I can build on top of CPE. I want CPE to be about component names and not about URI

format. CPE should define the component names and publish them and be completely unspecified on how they are transported.

Ms. Parmelee – So you are suggesting we decompose and standardize the parts of what is currently in the URI.

Lt Col Wolfkiel – Yes. We should say what the parts are but don't say how those parts are packaged.

Mr. Waltermire – So now you no longer have an identifier but instead you have a composite key.

Lt Col Wolfkiel – Yeah.

... – I have an idea: We have all these different concerns that are getting mashed together. One way to separate them is to identify a set of products via a set of attributes. These attributes say nothing about how to discover the product. This gives us something like a comma delimited list of attributes. That is the canonical identifier. You can think of it as a query. The query then becomes the primary key. The query (i.e. set of attributes) can map to one or more discovery mechanisms. The mapping to the discovery profiles is fuzzy. It just narrows you down to a set of attributes. You can have more than one discovery profile for the same set of attributes.

Mr. Cheikes – We want to be careful about reengineering CPE completely. I'm not sure how we would take that proposal and move it forward, at least in this room.

Mr. Waltermire- Would attributes be name-value pairs?

... – could be.

Mr. Waltermire – We've been thinking about a tag-based approach. Product identification would be a product of matching tags. Getting to a place where you could do set-based operations is the idea. Currently we are doing a poor job of that.

Mr. Cheikes – What happens if we drop matching. Right now we lose the ability to refer to sets of things. Right now the CPE language has the ability to represent complex collections. Is that being used?

Mr. Keanini – Way back in 2006 we thought everything should be logically separated. The name was just the name. On that same mailing list there have been three proposals to move towards a tagging-based approach. Going this way accomplishes our goal of decoupling the functionality currently wrapped in CPE names. A CPE name should just a name. Some prior proposals went to CPE mailing list and nothing happened.

Ms. Parmelee – We are trying to apply a more structured process for addressing these concerns than was previously applied.

Mr. Waltermire – I'm all for breaking up the spec to be just about naming. The problem is that the cast-offs still need to be resolved.

Mr. Cheikes- Yes. Matching is there for a reason.

Mr. Waltermire – From an SCAP perspective, we cannot lose CPE functionality that isn't picked up somewhere else.

Lt Col Wolfkiel – Agree. But if we could solve the naming problem we can fix those. Until the name is fixed, no solution is possible. We need to get back to the core issue that has prevented success in all prior situations. The only stick we have is the validation program. If validation focused on correct naming, then we could build on this.

Mr. Hollis – To strip the spec down and remove functionality would hurt because some of that stuff is being used.

Mr. Waltermire – Could we have separate specs for the various functionality? For example, a naming spec 2.3, a matching 2.3, etc. for all the different pieces of functionality? We end up with the same functionality, but allow it to rev independently.

Mr. Cheikes – How would SCAP validation treat that?

Mr. Hollis – The issue is the checking facility where you use OVAL to see if a CPE is present. That is being used. If we define a spec that removes all but naming then that is a case where something used today won't be in spec. I'm not saying this has to be OVAL, but there needs to be functionality that is carried through.

... – Agreed, but this is not the same as saying every feature must stay.

Mr. Waltermire – I'd say every feature is being used by someone, somewhere.

.... – The question is whether support of the feature is optional or not to be valid. For example, if you are SCAP validated, currently you need prefixes and URIs and matching. This is not useful for some people. If you make it optional, that's fine. If you say you must do it this way, that's a problem. The issue is what gets to be mandatory.

... – I just care about it generating the name my customers need. Talking about making some of the other capabilities optional is in my interest.

Lt Col Wolfkiel – If you break into separate specs, different sets of sub-specs can be required for different types of validation. If it is all in one spec but all parts of optional then it is harder to control.

Mr. Cheikes – What are the pieces of CPE?

Mr. Cheikes - Core naming, matching, language/encoding/transport.

Mr. Waltermire- There are two forms of encoding: dictionary and transport.

Mr. Cheikes – There is a format for naming information. There is a goal to relax or eliminate the URI format.

Mr. Waltermire – I would assume the rules for creating the CPE name should be separate from rules on transport.

Ms. Parmelee – If we split the spec into language, naming, and mapping, the URI format just applies to the naming.

Lt Col Wolfkiel – No. The URI format is not naming. It is transport.

Mr. Waltermire – There would be a stack where higher level specs imposed requirements on lower level specs. You could, however, take out and replace any layer.

Mr. Cheikes – The current spec where compliance is all or nothing is a problem. We need a way to break it up into independent but related elements. CPE compliance would be measured against a subset of these specs.

Mr. Keanini – Stack based organization of the specifications is a good point. The more we are thinking of stacks the more granularity we will have.

Mr. Waltermire – Basically we are talking about turning a 2.2 CPE spec into a 2.3 CPE specification stack.

.... – We got there by saying we wanted to get rid of the URI.

Mr. Waltermire – The stack would allow some communities to drop the URI.

... – The more we compartmentalize, the more we are avoiding rather than solving the issues.

Lt Col Wolfkiel – If we compartmentalize now, others can propose fixes to individual components.

Lt Col Wolfkiel – What if we stopped thinking about ripping everything out of CPE but instead base CPE on something else? What if CPE is based on something with enumerated vendor names and CPE becomes the actual logical parings of vendor and product.

Mr. Baker – So CPE continues to support tracking relationship?

Lt Col Wolfkiel – Yes.

Mr. Waltermire – So create a vendor enumeration and a product enumeration.

Ms. Parmelee – yes.

Mr. Cheikes – One approach to is to enumerate a controlled vocabulary of components, such as vendors and products.

... – What are you gaining by this? If you take away the prefix property then this is what you have anyway. Why do we need a structure for a list of vendors? What's different?

Mr. Waltermire – Computability. CPE provides a specific means to talk about a relationship. Having a controlled vocabulary of vendors and products (these being the two broadest groups) gives us a basis to do that. We might want to standardize on versions too.

Lt Col Wolfkiel – That would be cool, but it would be human intensive.

Mr. Waltermire – We're paying a lot right now not having that. We need to teach computers to do that: use a regex to break out different types of version structures.

Mr. Keanini – The problem is that 4-5 vendors are already doing this but not in a compatible way. If a way to standardize versions was created, this would let vendors come together.

... – You are talking about understanding the meaning rather than just parsing.

Mr. Keanini – String parsing is already being done. The interoperability problem comes in when we want to infer a set of potentially non-contiguous versions. We need to model this.

... – So you are saying we need to understand how version nonmembers relate to each other?

Lt Col Wolfkiel – If everyone had a common format then everyone understands version information in the same way.

Mr. Waltermire – It is much easier to centralize that understanding than to centralize a CPE repository. (Assuming a finite number of vendors).

Lt Col Wolfkiel – We are heading down a rat hole. We should take this offline and see if it works.

Mr. Keanini – The reason we started down this path was to determine why a controlled vocabulary is useful.

Mr. Waltermire – A controlled vocabulary is a common model people can map to.

... – If we had a standardized version structure, is this a controlled vocabulary?

Mr. Waltermire – Yes. That way we can use the same model to converse instead of the relying on having same identifier. And we could start implementing this using existing CPE infrastructure using tags today.

Mr. Cheikes – We definitely a need a better way to describe versioning in CPE. Maybe we should get a group together to look at this.

Mr. Baker – CPE could still be based on the underlying controlled vocabulary with no changes to the spec.

Ms. Parmelee – Dublin Core is a good example: it is a list of core terms.

Lt Col Wolfkiel – If we say controlled vocabulary is content and others are saying list of tags are content, we are not having a conversation. We need to agree on what we are talking about.

<break>

Mr. Cheikes – Are there any proposals related to "unpacking the edition component"? It has become something of a grab-bag field.

Mr. Neuman – Originally this was "what manufacturer calls the edition". We need to track 32 vs. 64 bit, which was previously shoved in the edition field. We should split out a field for that.

Lt Col Wolfkiel- Another item that gets shoved into the edition field is the target operating environment. For example, MS office "for Mac".

... – Can't you just you use two CPEs to handle this?

Lt Col Wolfkiel – Sometimes a vulnerability is just in Office for Mac but not in Office in general. Another example is Win XP on VMWare.

Mr. Waltermire – Another piece of information we might want to have a field for is if the host is functioning as a workstation.

Lt Col Wolfkiel – That is not a property of the software.

Mr. Waltermire – We could decompose edition the same way we are talking about decomposing version.

Lt Col Wolfkiel – Agreed. If we tagged values then we can separate them out. Right now it is a pain to find which CPEs apply to 64 bit software.

... – The intent was that that people could see "Home Edition". Why can't we just use CPE to say "& the OS is Mac."

Mr. Cheikes – The CPE doesn't let you make statements about the endpoint. ("On endpoint is X and Y.") It is just a list.

... – NVD does this.

Mr. Cheikes – NVD isn't part of CPE.

Mr. Cheikes – Do we want to explicitly break out bitness?

... – Bitness is a function of software not hardware.

Lt Col Wolfkiel – Correct.

Mr. Keanini – This is the heterogeneous viewpoint problem. It is hard to decide this is the only way we do it.

... – The issue is that when 64 bit gets tagged we now also have to go back and tag 32 bit.

Lt Col Wolfkiel – I'm just looking for where the software box says 64 bit. If the box doesn't say, I don't put it in. I just want a place to put this if I'm told the information.

Mr. Cheikes – Can we just sub-delimit the edition field?

Mr. Waltermire – Then we need to add consistent structure which isn't there now.

... – That would lead to lots of implementation challenges.

Mr. Cheikes – What are the proposals for handling bitness and architecture.

Dr. Katherine Goodier – You'll need to distinguish semantically between combined components such as you get with NetTop.

Mr. Waltermire – You need something like CPE to specify combinations.

Lt Col Wolfkiel – Are the edition, bitness, target software parts of a CPE name that should be separate or are they not important? In the latter case, stop tracking them.

Ms. Parmelee – Of people who are using CPE, how many use the edition component? How many don't because of its problems?

Lt Col Wolfkiel – I'd like to be able to use the components independently. I'm using it now as it is. I'd like to be able to track licenses when I buy different editions of software. Automated tools cannot break out the details in the current structure.

Ms. Parmelee – It sounds like many people are still using the field despite problems. Is the proposal to break out the components? [There are some affirmative votes]

Mr. Waltermire – How about tagging?

... – We need to separate the metadata from the name. Otherwise there will always be something that comes up later that needs to be split out and then we need to have this discussion over again.

... – When you create these things, a tagging approach would simplify. Everything comes from the fact that we need normalized names to do matching in the first place. We need consistency, but that makes it very difficult to manage. If we have a model to describe products this will go a long way to solve issues.

Mr. Cheikes – The point of the version 2.3 discussion is to look at small steps we can take.

... – We proposed tagging a year and a half ago.

... – How long would it take to implement tagging? Does it run parallel to what we have now?

Mr. Waltermire presents slides prepared by NIST which summarize a “tagging in CPE” proposal. Key ideas:

- Goals of proposals: separate identification from matching. ID is a string and the tags become the basis for matching. Eliminates through prefix property and replaces it with structured metadata. This allows decomposition of existing metadata and federated creation.
- Adding tagging: there is a tag repository with tag definitions. People created tagged CPE content and that goes into the CPE dictionary.
    - Mr. Cheikes – So we keep same CPE names but pull out tags?
    - Mr. Waltermire – Correct. Tags are name-value pairs. We aren't changing the core information.
- CPE names become lists of tags. Tags are applied to CPEs or to other data structures. We use a tag dictionary for tag creation and maintenance. We can impose constraints on tag values using similar methods to XML facets. We can also add tag matching constructs (e.g. range-based matching)
- Tag definitions have a name and source. The source is a namespace. Tag definitionss have metadata to track revision history. The "Applicable to" field – defines relations between tags. We use the specification of facets to constrain the tag. (E.g. an enumeration of allowed values.)
- Revised matching algorithm. Matching is now done by tags. A CPE name becomes a list of tags.

Mr. Cheikes – When this was proposed it was early in the CPE 2.0 lifecycle. The community consensus was to hold onto the existing spec and see how it works out.

Lt Col Wolfkiel – NIST's proposal is to create a controlled vocabulary with any content. We still need to understand what the software is. The XML Dave presented was a transport model – it doesn't solve the naming issue.

Mr. Waltermire – In the default tag we broke out architecture and edition.

Lt Col Wolfkiel – We do that in ARF.

Mr. Waltermire – The advantage of tags is that we can break things up at any level.

Lt Col Wolfkiel – Developers hate it when the tag field can have any value.

Mr. Waltermire – Field contents were allowed to be dynamic to allow faster reving of content.

Lt Col Wolfkiel – The fact that new tags can be added means vendors have to go back and understand each new tag.

Mr. Waltermire – Only tag originators need to understand the tag.

Ms. Parmelee – ...and those people that need a shared understanding.

Mr. Waltermire – Only tools producing instances of tags need to understand the semantic meaning. Tools that are evaluating data don't need to understand the semantics of what they are evaluating. There is a class of tools that will need to hard code tag values or match them to API queries. But reporting tools will not need deep semantic knowledge.

Lt Col Wolfkiel – Basically this is allowing a list of words to be controlled outside the spec. There needs to be a common understanding of what needs to be present to be compliant. How can you validate when you are dynamically adjusting the language?

Mr. Waltermire – That is why we include history data to track a tag's evolution.

... – Isn't our goal here to say that bitness and architecture should be first class citizens. The mechanism that accomplishes this is orthogonal.

Ms. Parmelee – Edition content should contain only edition data. Other data that is currently in the edition component should be removed into a new component.

Lt Col Wolfkiel – If people call it edition but it really is target hardware (bitness) it should be in the bitness field. For example, if someone says "64 bit edition".

Mr. Cheikes – We have talked about wanting these things being computable. Some of these are human judgment. If we elevate these to 1st class citizens are we dooming ourselves.

Lt Col Wolfkiel – We are already dealing with this. It is no worse or better in terms of complexity.

Mr. Cheikes – Proposal: elevate target architecture (bitness), target software, and true edition to first class elements.

Ms. Parmelee – We should set up a working group to discuss the details of the tagging approach.

Lt Col Wolfkiel – The current proposal is too heavyweight for me.

Mr. Waltermire – The proposal is not the only transport option. This is still an open question.

Notes collected from the whiteboard at the conclusion of the work session:

Proposals for changes to the CPE specification:

- Drop prefix property
    - Dependency: drop matching algorithm This causes no standard way to match
- Drop URI requirement for compliance
    - Name becomes a string
- Separate the enumeration from the language
    - Make all but names optional
    - separate core naming, matching and language specs
    - Complexity – the language is in use today
- Move to a stack-based design of specs
- Break up CPE names into a controlled vocabulary
    - Enables computable relations between components
- Allow component-based matching
- Adopt a tagging approach to CPE
    - Dependency – drop prefix property
    - Requires ne tools (but content is backwards compatible)
    - Requires tools to parse components of a CPE name and transform components into tags
    - Add new facts to current CPE schemas
    - Stand up a sand box for community contributed tags
    - Propose a technical working group to explore application of tagging proposal

Edition content should contain only edition data. Other data that is currently in the edition component needs to be evaluated as new component values (e.g. architecture (bitness), target software)