The Future of CPE

Drew Buttner and Brant Cheikes

cpe:/h:mitre:abuttner, cpe:/h:mitre:bcheikes

29 October 2009

Agenda

- Status Update
- The Very Near Future: CPE 2.2, 2.3?
- The Near and Further Future: CPE 3.0
- Strategy
- Discussion Topics

The Situation Today

- **CPE has achieved some success**
  - V2.2 released 11 March 2009, included in SCAP 1.0 draft
  - Stewardship has evolved into a shared responsibility of MITRE and NIST
- **CPE's full potential has not been reached**
  - Technical and procedural issues
  - Unsatisfied use cases
- **MITRE and NIST working to clarify and streamline their roles and responsibilities**
- **In FY10 we will push to "move CPE to the next level" of capability and value**

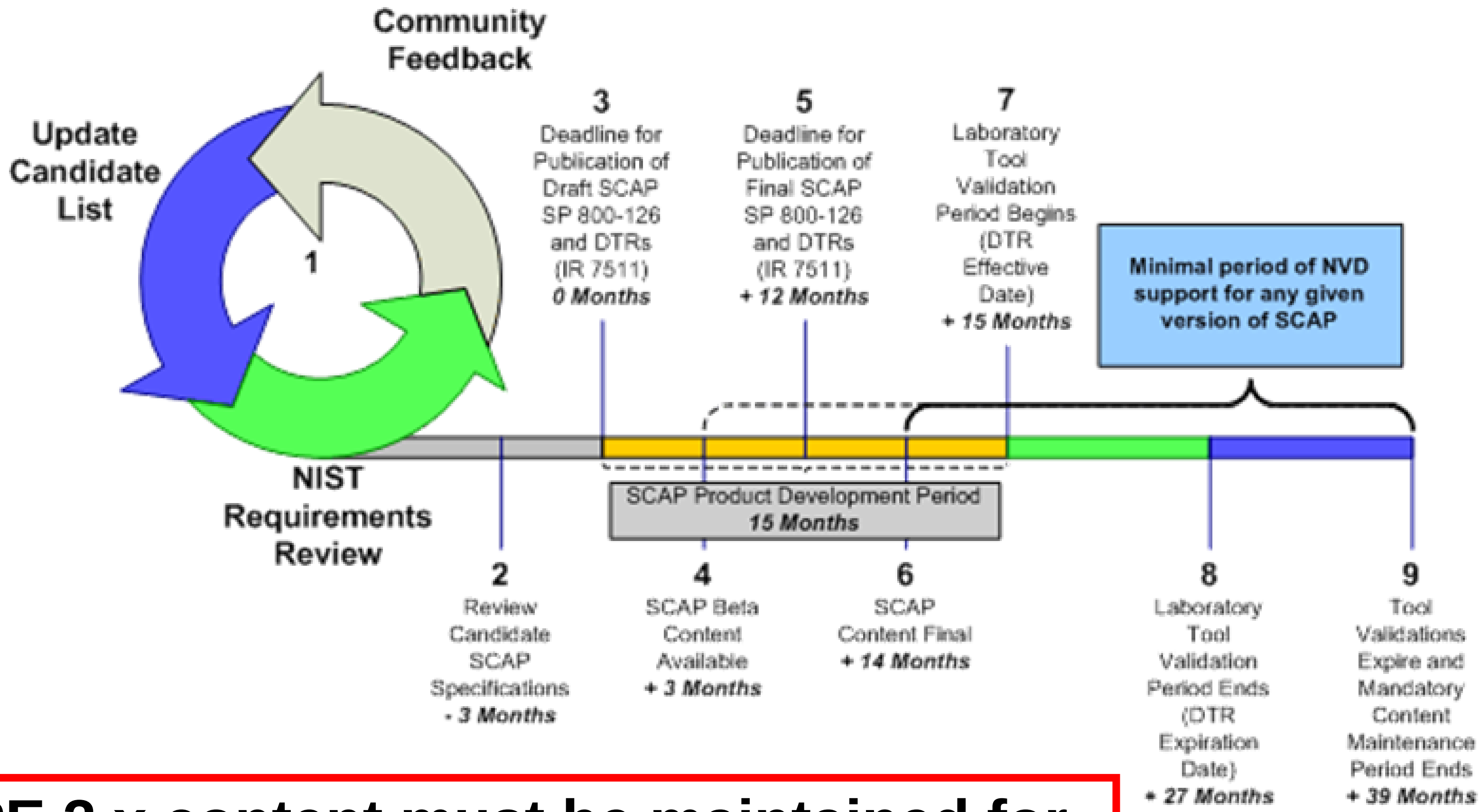
Roles in FY10

- **NIST**
  - Hosts the data, makes all content decisions
  - Represents SCAP interests

New!

- **MITRE**
  - Provides leadership on options and tradeoffs
  - Moderates community technical discussions
  - Balances competing interests
- **DoD**
  - Sponsors the work, provides oversight
  - Represents DoD interests

SCAP Lifecycle and CPE 2.x

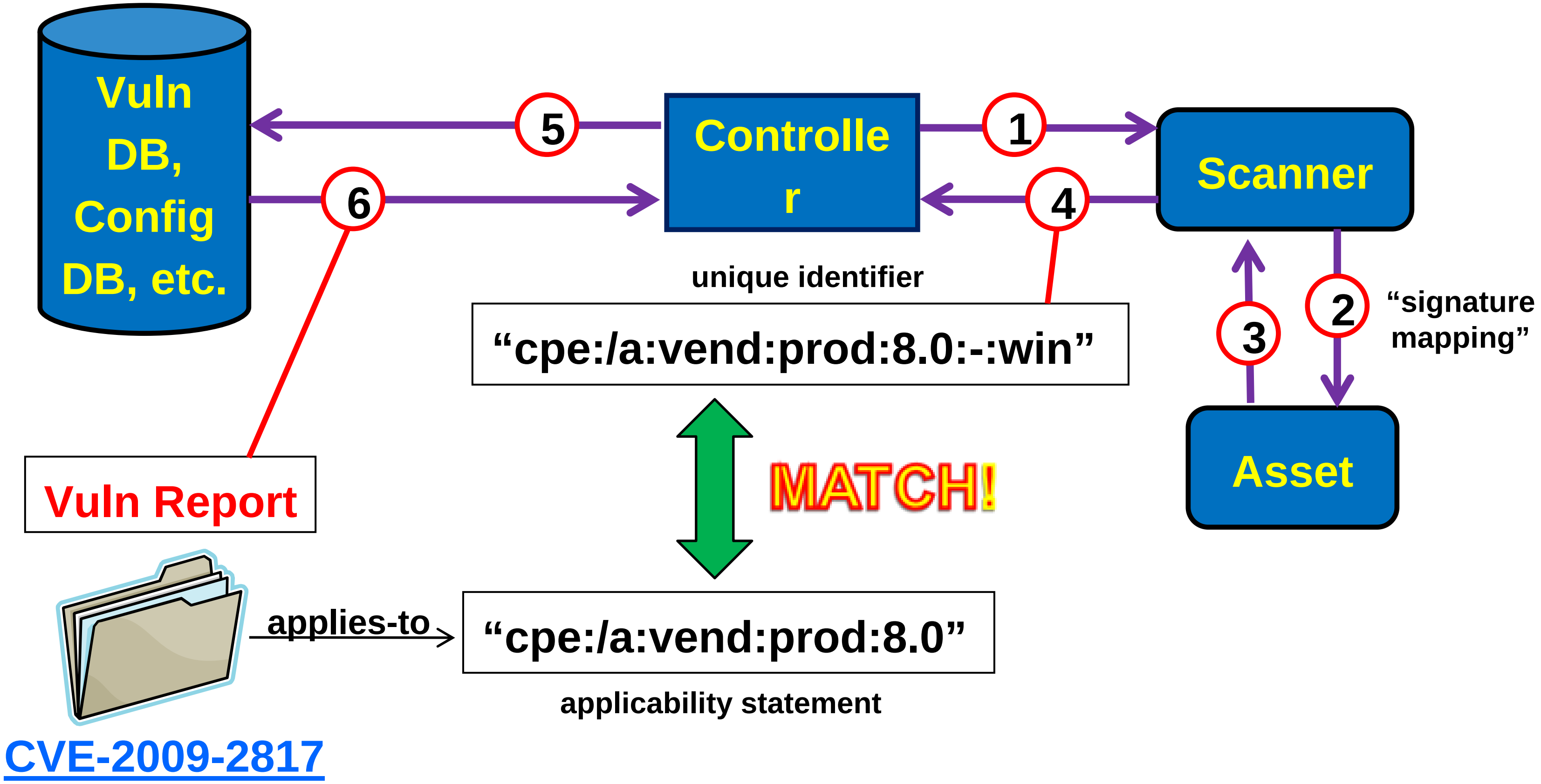

Community Feedback
Update Candidate List
1
NIST Requirements Review
3
Deadline for Publication of Draft SCAP SP 800-126 and DTRs (IR 7511)
0 Months
5
Deadline for Publication of Final SCAP SP 800-126 and DTRs (IR 7511)
+ 12 Months
7
Laboratory Tool Validation Period Begins (DTR Effective Date)
+ 15 Months
Minimal period of NVD support for any given version of SCAP
SCAP Product Development Period
15 Months
2
Review Candidate SCAP Specifications
- 3 Months
4
SCAP Beta Content Available
+ 3 Months
6
SCAP Content Final
+ 14 Months
8
Laboratory Tool Validation Period Ends (DTR Expiration Date)
+ 27 Months
9
Tool Validations Expire and Mandatory Content Maintenance Period Ends
+ 39 Months

CPE 2.x content must be maintained for 25 months after SCAP 1.0 becomes final

CPE 2.x Near-Term Plan: V2.3 Maintenance Release?

- **Goal: limited effort, with focus on keeping long-term O&M burden low**
- **Implement "editorial changes" only**
  - Clarify areas proven to be sources of confusion
  - Document content decision rationales
- **Possibly split the spec into three parts:**
  - Naming, Dictionary, Matching

CPE 2.x CONOP: Overview



Vuln DB, Config DB, etc.

Controller

Scanner

Asset

1

2

3

4

5

6

"signature mapping"

unique identifier

"cpe:/a:vend:prod:8.0:-:win"

MATCH!

"cpe:/a:vend:prod:8.0"

applicability statement

applies-to

Vuln Report

CVE-2009-2817

CPE 2.x CONOP: What Works

- **URIs have proven useful as names**
  - Unique, compact and human-readable
  - Not typically hard to create
- **Matching algorithm is uncomplicated**
  - No access to central Dictionary required
- **Seven components capture much of what's needed to distinguish among products**
  - Part, Vendor, Product, Version, Update, Edition, Language

CPE 2.x CONOP: What Doesn't Work

- **The core data model has shortcomings**
  - – The seven components don't capture all we need
    - Complex versioning schemes, "edition" overloading
    - Relations within and between product descriptions
  - – Naming and matching are entangled
    - Name-related decisions forced to consider matching reqts
- **Critical use cases not addressed**
  - – Full-spectrum discovery and reporting
  - – Community-curated value-added information
- **Dictionary hygiene has suffered**

Use Case: Full-Spectrum Discovery & Reporting

- **Requirement:**
  - Support non-credentialed & passive scanners
  - Handle “unlisted” product discovery
- **Methods for discovering software on devices and networks and either:**
  1. mapping them to curated CPE product descriptions as accurately as possible, or
  2. providing the maximum amount of data to allow an analyst to map them.

Use Case: Community-Curated Value-Added Information

- **Requirement:**
  - Enable vendors to provide and manage value-added information about discovered products
- **Methods to allow authorized providers to "own and operate" selected attribute-value pairs within existing CPE product descriptions, e.g.,**
  - Signatures associated with the product
  - Relationships to other products, or other entities outside the CPE product repository

Moving CPE Forward: Priority Challenges

- **Must enhance the core data model**
  - Consider abandoning URI-based naming scheme
- **Must support critical use cases**
  - Full-spectrum discovery and reporting
  - Community-managed value-added information
- **Must implement an efficient, sustainable content-management process**
  - Open to authorized providers

Strategy

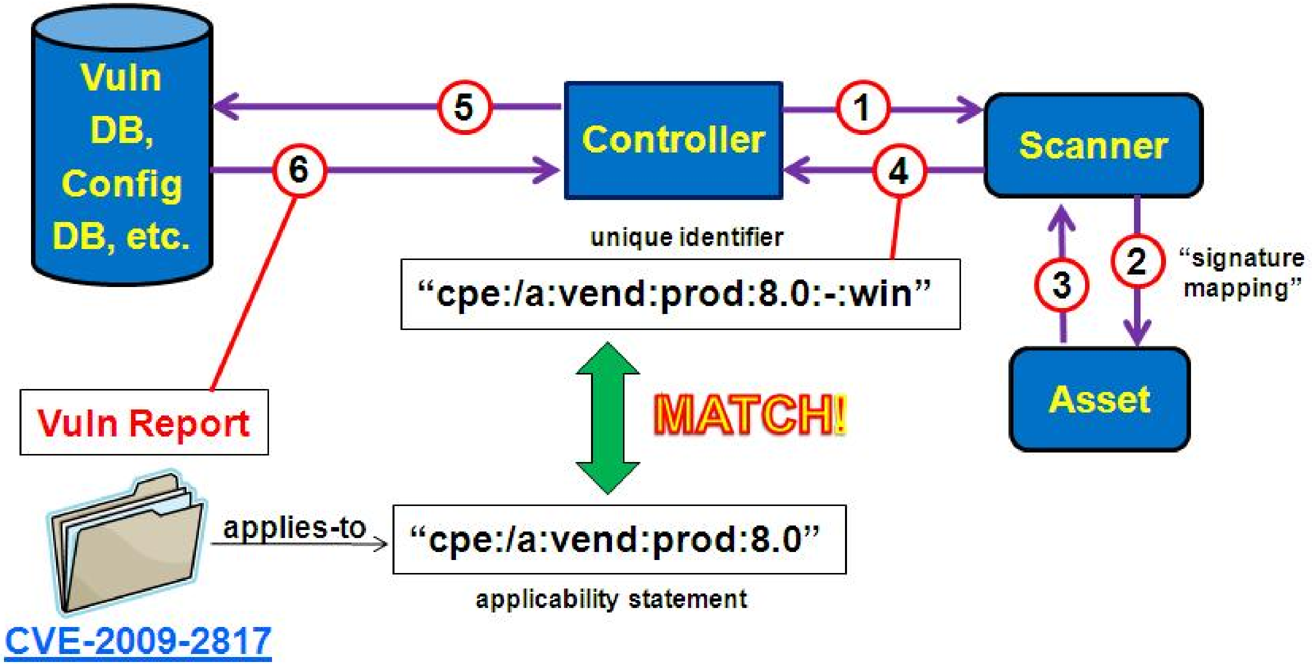
- **MITRE to initiate open discussion aiming to produce v3.0 by 1 Sep 2010**
  - Active community engagement will be critical!
- **Hold focused vendor meeting(s)?**
- **Collectively gather and vet requirements**
- **MITRE & NIST jointly propose solutions that satisfy requirements**
  - Proposed solutions welcome from community too
- **Community review**

Starting the Discussion: Topic Outline

1. **Extending the CONOP**
2. **Enhancing the core data model**
3. **Versioning schemes**
4. **“Unlisted” products**
5. **Name changes**
6. **Applicability statements**

Topic #1: Extending the CONOP

- 2.x CONOP assumes exchange of compact IDs
- New use cases imply requirement to exchange <u>structures</u>



Topic #2:
Enhancing the Core Model

- **What is the 2.x core data model?**

```
<cpe-item name=
 “cpe:/o:microsoft:windows_xp::sp1:professional”>
 <title xml:lang=“en-US”>Microsoft Windows XP</title>
 <notes> … </notes>
 <references> … </references>
 <check> … </check>
 <meta:item-metadata modification-date=
  “2007-09-14T13:36:49.090-04:00”
  status=“DRAFT” nvd-id=“58621”/>
</cpe-item>
```

Topic #2: Enhancing the Core Model

- **Two principal options:**
  - Keep the URI format, just add more components as the need arises
  - Abandon the URI as carrier of all product-description elements, convert to attribute-value structure
- **Keep or discard URI name format?**
  - Pros: Unique, compact, human-readable, easy to create
  - Cons: Not practical/scalable as attributes increase

Topic #2: Enhancing the Core Model

- **Possible approach to standardizing a set of required and optional attributes**
  - Required (examples):
    - “category”, Vendor, “core product name”, “market name”, “version scheme”, Update, Edition, Architecture, TargetSW, Language, Status, Owner
  - Optional (examples):
    - Supports-Role, Provides-Function, OS-Family
  - Curate attribute values in central repository
  - Support both XML and RDF/OWL models?

Topic #3:
Versioning Schemes

- **How to handle wide variety of vendor-specific versioning schemes?**
  - How much version-related information needs to be directly accessible for matching purposes?
- **Option 1: Coerce to <*maj*><*min*><sub><rest>**
  - Simple to represent
  - Not straightforward to coerce automatically
- **Option 2: Explicitly model each scheme**
  - The set of schemes is relatively small and stable

Topic #4:
“Unlisted” Products

- **How to handle “unlisted” products?**
  - By “unlisted”, we mean that the central repository does not contain a curated description
  - So there is no guarantee that a machine-generated description can be resolved without human assistance
  - But portions may be resolvable, e.g., known vendor but unknown product

Topic #5: Name Changes

- **How to handle name changes?**
  - Scenario 1: Vendor changes the market name of a product from one release to the next
  - Scenario 2: Vendor changes their own name
  - Scenario 3: Vendor A sells product line P to Vendor B
  - Scenario 4: Vendor A takes control of Vendor B (merger/acquisition)
- **In all (?) cases, on-disk signatures will not reflect change until next release is installed**

Topic #6: Applicability Statements

- **What are the requirements for applicability-statement expressivity?**
  - Range statements
    - Versions “prior to [and including]” <v>
    - Versions <v1> “through” <v2>
  - Temporal statements
    - Product releases “prior to” <date>
  - To what extent should applicability statements be “future proof”?
    - Should we allow the creation of applicability statements which could match products not yet on the market?

Open Discussion

- **Feedback on use cases and priorities?**
- **Feedback on what’s most needed to increase CPE value to community?**
- **Feedback on technical approaches?**
- **Should we schedule a CPE workshop soon?**

Backups

Definition:
Installable Software Product

- **A user can download or buy it**
- **There is a vendor/organization/person that produces it**
- **An enterprise IT administrator can push it out over the enterprise network and install it into their environment**
- **It is (or can be) recorded by an asset management tool**

Use Cases Currently Outside Scope

- **Network-based discovery**
  - Proprietary “fingerprinting” approaches
- **Forensics**
  - Need to represent relationships between installable products and, e.g., component DLLs and drivers
- **IT management**
  - Need to refer to non-standard categories of managed IT assets