Common Platform Enumeration (CPE)
Overview of Release 2.3

Brant A. Cheikes
bcheikes@mitre.org
Session Objectives

- High-level tutorial on CPE
- Focus on latest release: CPE 2.3
- Describe the problem that CPE solves
- Provide examples of CPE names, CPE applicability language statements
- Illustrate the name matching procedure
- Discuss key challenges and next steps
What is CPE?

- CPE is:
  - A MITRE-led open standard
  - A structured naming scheme for IT products
  - Enabling technology for security automation

- CPE encompasses:
  - Two prescribed name formats
  - An authoritative dictionary of vetted, approved names
  - Algorithms for comparing names
  - A language for describing complex platforms
What Problem Does CPE Solve?

Interoperable IT Product Names

These are the same product!
How Does CPE Solve the Problem?

NIST NVD

CPE Dictionary

Approved names

Published names

35K+ CPE Names
Oct 2011

Adoption Drives Success

CPE Community

CPE Adopters

Proposed new names

Products
CPE Use Case Example

**Diagram**

1. **Scanner**
   - Unique identifier: "cpe:/a:vend:prod:8.0:-:win"
   - "signature mapping" to Asset

2. **Asset**
   - "cpe:/a:vend:prod:8.0"
   - Applicability statement

3. **Controller**
   - 6 unique identifier: "cpe:/a:vend:prod:8.0:-:win"

4. **Controller**
   - 4 unique identifier: "cpe:/a:vend:prod:8.0"

5. **Vuln DB, Config DB, etc.**

6. **Vuln Report**

**CVE-2009-2817**
State of the Standard

- CPE is in a transition period

- CPE 2.3 is the current version
  - Specified by four NIST Interagency Reports (August 2011)
    - NIST IR 7695—Naming
    - NIST IR 7696—Matching
    - NIST IR 7697—Dictionary
    - NIST IR 7698—Applicability Language
  - Required in SCAP 1.2

- CPE 2.2 continues to be supported
  - Specification published in March 2009
  - Required in SCAP 1.0, 1.1
CPE 2.3 Specification Stack

- Modular
- Easier to maintain
- Easier to extend
- More flexible w/r/t specifying conformance requirements
NIST IR 7695 specifies the basic concepts and syntax of CPE names

- Defines the Well-Formed Name (WFN) and two allowed bindings
  - URI binding
  - Formatted string binding

- Specifies mechanical procedures for translating between binding forms
NIST IR 7696 specifies the procedures for comparing two Well-Formed Names and determining the relationship between them:

- EQUAL
- SUBSET
- SUPERSET
- DISJOINT
NIST IR 7697 specifies the Dictionary data model, requirements for Dictionary creation and maintenance, and basic concepts for Dictionary operation.
NIST IR 7698 specifies a language for creating and using “applicability language statements”.

Enables the definition of a “platform” as a structure of ANDs, ORs and NOTs of CPE names.
CPE 2.3 Significant Changes (1/2)

Naming:
- Adds four new name attributes
- Defines the Well-Formed Name and two allowed "bindings"
  - URI (v2.2-style) and Formatted String
- Specifies procedures for binding and unbinding
- Lays foundation for limited use of wildcards for name matching

Matching:
- Defines attribute- and name-level matching separately
- Allows unordered comparison of attribute-value pairs
  - Eliminates version 2.2 “prefix property”
- Enables limited matching with single- and multi-character wildcards
CPE 2.3 Significant Changes (2/2)

- Dictionary:
  - Extends the data model to allow both URI and formatted string name bindings
  - Defines name acceptability criteria, including name completeness and uniqueness
  - Defines dictionary entry provenance and the deprecation process
  - Defines required dictionary management documents
  - Opens the door to “extended CPE dictionaries”

- Applicability Language:
  - Adds support for formatted string binding
  - Adds the cpe:check-fact-ref element which allows calls to external checking systems such as OVAL
CPE 2.3 Name Attributes

- part
- vendor
- product
- version
- update
- edition
- language

Carried over from CPE 2.2

- sw_edition
- target_sw
- target_hw
- other

New in CPE 2.3
CPE 2.3 Name Examples

- **(Application) Microsoft Office 2007 Professional Service Pack 2**

- **(Operating System) Microsoft Windows 7 64-bit Service Pack 1**
  - URI: cpe:/o:microsoft:windows_7:-:sp1:x64
  - FS: cpe:2.3:o:microsoft:windows_7:-:sp1:-:*:*:*:x64:*

- **(Hardware) 3Com Router 3012**
  - URI: cpe:/h:3com:3c13612
  - FS: cpe:2.3:h:3com:3c13612:-:*:*:*:*:*:*:*
<cpe:platform id="789">
  <cpe:title>
    Microsoft Windows XP with Internet Explorer 7.x or 8.x
  </cpe:title>
  <cpe:logical-test operator="AND" negate="FALSE">
    <cpe:fact-ref
      name="cpe:2.3:o:microsoft:windows_xp:*:*:*:*:*:*:*:*"/>
  </cpe:logical-test>
  <cpe:logical-test operator="OR" negate="FALSE">
    <cpe:fact-ref
      name="cpe:2.3:a:microsoft:internet_explorer:7.*:*:*:*:*:*:*:*"/>
    <cpe:fact-ref
      name="cpe:2.3:a:microsoft:internet_explorer:8.*:*:*:*:*:*:*:*"/>
  </cpe:logical-test>
</cpe:platform>
All matching algorithms specified in terms of WFNs
  - So matching is agnostic to binding

Specified functions:
  - Compare_WFNs(source, target)
    - Pairwise compares source attribute values to target attribute values
    - Returns a table of results
  - CPE_x(source, target)
    - x one of EQUAL, DISJOINT, SUBSET, SUPERSET
    - Compares a source WFN to a target WFN and returns TRUE if the
      set-theoretic relation holds between source and target
CPE 2.3 Name Matching: Attribute-Level Comparison

Source WFN


Target WFN


Compare_WFNs(source, target)

<table>
<thead>
<tr>
<th>Attrib</th>
<th>Part</th>
<th>Vendor</th>
<th>Product</th>
<th>Version</th>
<th>Sw_ed</th>
<th>Tgt_sw</th>
<th>Tgt_hw</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Src</td>
<td>o</td>
<td>microsoft</td>
<td>windows_?</td>
<td>ANY</td>
<td>home*</td>
<td>NA</td>
<td>x64</td>
<td>NA</td>
</tr>
<tr>
<td>Tgt</td>
<td>o</td>
<td>microsoft</td>
<td>windows_7</td>
<td>6\1.1</td>
<td>home_basic</td>
<td>NA</td>
<td>x32</td>
<td>ANY</td>
</tr>
<tr>
<td>Result</td>
<td>=</td>
<td>=</td>
<td>⊈</td>
<td>⊈</td>
<td>⊈</td>
<td>⊈</td>
<td>≠</td>
<td>⊆</td>
</tr>
</tbody>
</table>

© 2011 The MITRE Corporation. All rights reserved.
CPE 2.3 Name Matching: Name Comparison Table

<table>
<thead>
<tr>
<th>No.</th>
<th>If Attribute Relation Set =</th>
<th>Then Name Comparison Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>If any attribute relation is DISJOINT (≠)</td>
<td>Then CPE name relation is DISJOINT(≠)</td>
</tr>
<tr>
<td>2</td>
<td>If all attribute relations are EQUAL (=)</td>
<td>Then CPE name relation is EQUAL (=)</td>
</tr>
<tr>
<td>3</td>
<td>If all attribute relations are SUBSET (⊂) or EQUAL (=)</td>
<td>Then CPE name relation is SUBSET(⊂)</td>
</tr>
<tr>
<td>4</td>
<td>If all attribute relations are SUPERSET (⊃) or EQUAL (=)</td>
<td>Then CPE name relation is SUPERSET (⊃)</td>
</tr>
</tbody>
</table>
### CPE 2.3 Name Matching: Name-Level Results

#### CPE_DISJOINT=TRUE, CPE_EQUAL=FALSE

<table>
<thead>
<tr>
<th>Attrib</th>
<th>Part</th>
<th>Vendor</th>
<th>Product</th>
<th>Version</th>
<th>Sw_ed</th>
<th>Tgt_sw</th>
<th>Tgt_hw</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Src</td>
<td>o</td>
<td>microsoft</td>
<td>windows_?</td>
<td>ANY</td>
<td>home*</td>
<td>NA</td>
<td>x64</td>
<td>NA</td>
</tr>
<tr>
<td>Tgt</td>
<td>o</td>
<td>microsoft</td>
<td>windows_7</td>
<td>6.1</td>
<td>home_basic</td>
<td>NA</td>
<td>x32</td>
<td>ANY</td>
</tr>
<tr>
<td>Result</td>
<td>=</td>
<td>=</td>
<td>⊃</td>
<td>⊃</td>
<td>⊃</td>
<td>=</td>
<td>≠</td>
<td></td>
</tr>
</tbody>
</table>

#### CPE_SUPERSET=TRUE (equivalent to v2.2 CPE_NAME_MATCH)

<table>
<thead>
<tr>
<th>Attrib</th>
<th>Part</th>
<th>Vendor</th>
<th>Product</th>
<th>Version</th>
<th>Sw_ed</th>
<th>Tgt_sw</th>
<th>Tgt_hw</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Src</td>
<td>o</td>
<td>microsoft</td>
<td>windows_?</td>
<td>ANY</td>
<td>home*</td>
<td>NA</td>
<td>x64</td>
<td>NA</td>
</tr>
<tr>
<td>Tgt</td>
<td>o</td>
<td>microsoft</td>
<td>windows_7</td>
<td>6.1</td>
<td>home_basic</td>
<td>NA</td>
<td>x64</td>
<td>NA</td>
</tr>
<tr>
<td>Result</td>
<td>=</td>
<td>=</td>
<td>⊃</td>
<td>⊃</td>
<td>⊃</td>
<td>=</td>
<td>=</td>
<td></td>
</tr>
</tbody>
</table>
Open Issues and Challenges

- CPE does not solve the “signature mapping problem”
  - Left to vendors to determine which CPEs are installed on a given computing asset
  - A serious concern for asset inventory tool vendors

- CPE Dictionary maintenance is costly and error prone
  - A need-driven human-in-the-loop process driven by community submissions of candidate names

- Many community needs cannot be addressed without a major release which may break backwards compatibility
  - Representing relationships, e.g., part-of, next-version, …
  - Representing roles, e.g., server, client, domain-controller, …
  - Supporting needs of non-credentialed scanners
What’s Next?

- Support roll-out of CPE 2.3 Dictionary at NIST
  - Document all dictionary management procedures and naming guidelines
  - Convert all 2.2 URI names to 2.3 formatted strings
  - Build infrastructure to provide simultaneous support for CPE 2.2 and 2.3 dictionaries

- MITRE working with Tag Vault.org to explore use of “software identification tags” to link installed applications to their CPE names
  - See ISO/IEC 19770-2 for further information on software ID tagging
To Learn More

- **CPE home page at MITRE:**
  - [http://cpe.mitre.org](http://cpe.mitre.org)

- **CPE home page at NIST:**
  - [http://nvd.nist.gov/cpe.cfm](http://nvd.nist.gov/cpe.cfm)

- **CPE 2.3 Specifications:**
Q&A