Common Platform Enumeration

Summary of Recent Developments

Brant Cheikes
Task Overview

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

CPE encompasses:
- A prescribed name format
- A language for describing complex platforms
- A methodology for assigning canonical names
- An algorithm for comparing names
What Problem Does CPE Solve?

Interoperable IT Product Names

Controller

Vendor 1 Scanner

Asset 1

Vendor 2 Scanner

Asset 2

1

windows NT 5.0.2195

2

windows 2000

These are the same product!
CPE Concept of Operations

1. Controller
2. “signature mapping”
3. Asset
4. Scanner
5. unique identifier
   “cpe:/a:vend:prod:8.0::win”
6. “applicability statement”
   “cpe:/a:vend:prod:8.0”

Vuln DB, Config DB, etc.

Vuln Report

CVE-2009-2817
Technical Use Case Analysis

- Study performed in November 2008
  - To better understand the technical use cases
  - Interviewed members of the CPE Community
  - See: http://cpe.mitre.org/about/use_cases.html

- Four technical use cases were identified:
  - Software Inventory
  - Network-Based Discovery
  - Forensic Analysis/System Architecture
  - IT Management

- Software Inventory identified as a “must have”
Elements of the CPE Standard

- Part 1: Specification
  - Development moderated by MITRE
  - Specification v2.2 released 11 Mar 2009
  - Development of v2.3 began March 2010

- Part 2: Official Dictionary
  - Maintained and managed by NIST
  - Contained 32,057 approved names on 25 Apr 11
  - Hundreds of new/modified entries every month
Format of a CPE 2.2 Name

cpe:/ <part> : application, O/S, hardware
   <vendor> : vendor name
   <product> : product name
   <version> : product version
   <update> : update level of the product
   <edition> : edition of the product
   <language> : internationalization
Examples of CPE 2.2 Names

- cpe:/a:zonelabs:zonealarm_internet_security_suite:7.0
- cpe:/o:redhat:enterprise_linux:4:update5:ws
- cpe:/h:intel
- cpe:/a:jon_smith:tool_name:1.2.3
- cpe:/a:adobe:reader
CPE Official Dictionary

- Maintained by NIST
  - Part of the National Vulnerability Database
  - See: http://nvd.nist.gov/cpe.cfm

- New entries accepted by e-mail
  - cpe_dictionary@nist.gov

- ~ 33K entries as of 6/1/2011
  - Hundreds of new entries per month
Example Dictionary Entries

<cpe-item name="cpe:/a:adobe:acrobat:9.3.3">
  <title xml:lang="en-US">Adobe Acrobat 9.3.3</title>
</cpe-item>

<cpe-item name="cpe:/o:microsoft:windows_7:-:-:x64">
  <title xml:lang="en-US">Microsoft Windows 7 64-bit</title>
  <notes xml:lang="en-US">
    <note>This CPE Name represents version 6.1.7600 of the Windows OS</note>
  </notes>
</cpe-item>
Brief History of CPE 2.3

- First proposed during CPE session at ITSAC 2009
  - "Goal: Enhance near-term usability while working on a comprehensive solution"

- Requirements collected during February 2010 "Developer Day" CPE workshop

- CPE Core Team formed in March 2010
  - MITRE, NIST, DOD, Cisco, McAfee, nCircle

- CPE v2.3 developed on short timeline (March thru July)
  - Fundamental changes to the "architecture" of CPE
  - Minimal changes to the functionality of CPE
CPE Specification v2.3

- CPE v2.3 intended as a “maintenance release”
- Development of v2.3 started 15 Mar 2010 with formation of CPE Core Team (MITRE, NIST, DoD, Cisco, McAfee, nCircle)
  - Implemented as four separate specifications organized in a “specification stack”
    - Naming, Matching, Dictionary, Language
    - MITRE lead author for Naming and Matching
    - NIST lead author for Dictionary and Language
- New drafts being released for 2\textsuperscript{nd} public comment
  - Naming - NIST IR 7695 – Published 28 Apr 2011
  - Matching – NIST IR 7696 – Published 28 Apr 2011
  - Dictionary – NIST IR 7697 – Awaiting publication
  - Language – NIST IR 7698 – Awaiting publication
CPE 2.3 Specification Stack

- Modular
- Easier to maintain
- Easier to extend
- More flexible w/r/t specifying conformance requirements
CPE v2.3: Summary of New Features

- It’s four “real specifications”
  - Detailed, precise
  - Fully backward-compatible w/ v2.2

- New Naming features:
  - Well-Formed Name (WFN): an abstract common form
  - Two WFN bindings: URI and formatted string
  - Four new attributes: softwareEdition, targetSw, targetHw, other
  - Support for single (?) and multi (*) wildcards

- New Matching features:
  - Limited implementation of single- and multi-character wildcards
  - Separate functions for name-level and attribute-level matching
Naming (1 of 5):
The Well-Formed Name (WFN)

NOTATION

```wfn: [part=\"a\", vendor=\"microsoft\", product=\"internet_explorer\", version=\"8.0.6001\", update=\"beta\", edition=NA]```

- A WFN is:
  - an abstraction, not intended for machine interchange
  - an unordered list of attribute-value pairs

- Eleven (11) allowed attributes are specified

- Attribute values are:
  - Logical values (ANY or NA), or
  - Character strings obeying certain requirements
NOTATION

```
wfn: [part="a", vendor="microsoft",
    product="internet_explorer",
    version="8.0.6001",
    update="beta", edition=NA]
```

IMPORTANT NOTE!!
WFNs by themselves do not solve the interoperable-name problem!
Naming (3 of 5): Binding WFN to URI

WFN

```
wnf:[part="a",vendor="microsoft",
    product="internet_explorer",
    version="8.0.6001",
    update="beta",edition=NA]
```

bind_to_URI(w)

unbind_URI(u)

```
cpe:/a:microsoft:internet_explorer:
    8.0.6001:beta:-
```

CPE v2.2-style URI binding
Naming (4 of 5): Binding WFN to Formatted String

WFN

\[
\text{wfn:} \{\text{part}="a", \text{vendor}="microsoft", \\
\text{product}="internet\_explorer", \text{version}="8.0.6001", \\
\text{update}="beta", \text{edition}=\text{NA}\}
\]

\[
\text{bind\_to\_fs}(w)
\]

\[
\text{unbind\_fs}(fs)
\]

cpe23:a:microsoft:internet\_explorer:
8.0.6001:beta:-:*:*:*:*:*:*:

Formatted string binding
Naming (5 of 5): Allowed Attributes

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

Carried over from CPE 2.2

- swEdition
- targetSw
- targetHw
- other

New in CPE 2.3
WFNs, URIs, Formatted Strings (1 of 5)

Formatted string binding

cpe:2.3:o:micro\$oft:windows_?::*:*:en-us:home*::x64:-

WFN (notation only)

wfn:[part="o", vendor="micro\$oft", product="windows_?", version=ANY, update=ANY, edition=ANY, language="en\-us", software_edition="home"", target_sw=NA, target_hw="x64", other=NA]

URI binding

cpe:o:micro%24oft:windows_%00:::~~home%01~~x64~~:en-us
WFNs, URIs, Formatted Strings (2 of 5)

Formatted string binding

```
cpe:2.3:o:micro\$oft:windows_?:*:*:*:en-us:home*::*:x64:-
```

Distinctive prefix with CPE version

URI binding

```
cpe:/o:micro\%24oft:windows_\%00:::~~home\%01~~x64~~:en-us
```
WFNs, URIs, Formatted Strings (3 of 5)

**Formatted string binding**
```
cpe:2.3:o:micro$oft:windows_?:*:*:*:en-us:home*:-:x64:-
```

**Backslash escape character**
```
cpe:2.3:o:micro\$oft:windows_?:*:*:*:en-us:home*:-:x64:-
```

**Percent encoding**
```
cpe:/o:micro%24oft:windows_%00::~~home%01~~x64~~:en-us
```

**URI binding**
```
cpe:/o:micro$oft:windows_%00::~~home%01~~x64~~:en-us
```
WFNs, URIs, Formatted Strings (4 of 5)

Formatted string binding

cpe:2.3:o:micro\$oft:windows_?:*:*:*:en-us:home*:--:x64:--

Unquoted single-character wildcard

Unquoted multi-character wildcard

URI binding

cpe:/o:micro%24oft:windows_%00:~~home%01~~x64~~:en-us
WFNs, URIs, Formatted Strings (5 of 5)

Formatted string binding

cpe:2.3:o:micro\$oft:windows_?:*:*:*:en-us:home*:--x64:--

New attributes “packed” into v2.2 edition component

URI binding

cpe:/o:micro%24oft:windows_%00::~~home%01~~x64~~:en-us
Matching: Overview

- All matching algorithms specified in terms of WFNs
  - So matching is agnostic to binding

- Specified functions:
  - \texttt{CPE\_Name\_Compare(source, target)}
    - Pairwise compares source attribute values to target attribute values
    - Returns a table of results
  - \texttt{CPE\_Attribute\_Compare(source, target)}
    - Compares a source attribute value to a target attribute value
    - Returns a result
  - \texttt{CPE\_x(source, target)}
    - \( x \) one of \texttt{DISJOINT, SUBSET, SUPERSET, EQUAL, INTERSECT}
    - Compares a source WFN to a target WFN and returns \texttt{TRUE} if the set-theoretic relation holds
Matching (1 of 5):  
Step 1 – Unbinding to WFNs

Source (formatted string)

cpe:2.3:o:micro\$oft:windows_?-*:*:*:en-us:home*-::x64:-

Target (URI)

cpe:/o:micro%24oft:windows_7:6.1:sp1:~~home_basic~~-~x32~:en-us

Source WFN

wfn:[part="o", vendor="micro\$oft", product="windows_?", version=ANY, update=ANY, edition=ANY, language="en\-us", software_edition="home**", target_sw=NA, target_hw="x64", other=NA]

Target WFN

wfn:[part="o", vendor="micro\$oft", product="windows_7", version="6.1", update="sp1", edition=ANY, language="en\-us", software_edition="home_basic", target_sw=NA, target_hw="x32", other=ANY]
Matching (2 of 5): Step 2 – Attribute-Level Comparison

**Source WFN**

```
wfnn:[part="o", vendor="micro\$oft", product="windows_?",
version=ANY, update=ANY, edition=ANY, language="en\-us",
software_edition="home*", target_sw=NA, target_hw="x64",
other=NA]
```

**Target WFN**

```
wfnn:[part="o", vendor="micro\$oft", product="windows_7",
version="6\-1", update="sp1", edition=ANY, language="en\-us",
software_edition="home_basic", target_sw=NA, target_hw="x32",
other=ANY]
```

**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>micro$oft</td>
<td>windows_?</td>
<td>ANY</td>
<td>home*</td>
<td>NA</td>
<td>x64</td>
<td>NA</td>
</tr>
<tr>
<td>Tgt</td>
<td>o</td>
<td>micro$oft</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>

© 2011 The MITRE Corporation. All rights reserved.
## Matching (3 of 5):
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>
Matching (5 of 5): 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>micro$oft</td>
<td>windows_?</td>
<td>ANY</td>
<td>home*</td>
<td>NA</td>
<td>x64</td>
<td>NA</td>
</tr>
<tr>
<td>Tgt</td>
<td>o</td>
<td>micro$oft</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>micro$oft</td>
<td>windows_?</td>
<td>ANY</td>
<td>home*</td>
<td>NA</td>
<td>x64</td>
<td>NA</td>
</tr>
<tr>
<td>Tgt</td>
<td>o</td>
<td>micro$oft</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>
CPE Dictionary: Quick Summary

- Draft NIST IR 7697 defines the concept of a Common Platform Enumeration (CPE) Dictionary, the rules associated with CPE Dictionary creation and management, and the data model for representing a CPE Dictionary
  - Acceptance criteria
  - Deprecation process
  - Identifier lookup and dictionary searching
  - Management documents
  - Official and extended dictionaries

- NIST will continue to maintain the CPE Official Dictionary
Open Issues

- **Issues with dictionary quality:**
  - **Fixable:**
    - Full of naming inconsistencies
    - NVD entries tagged with CPEs that aren’t in the dictionary
  - **Not fixable:**
    - Not an up-to-date enumeration of all existing products
    - Doesn’t solve the signature mapping problem

- **Confounds two functions: identifying and describing**
  - A name can be either an identifier of a specific product
    - `cpe:2.3:o:microsoft:windows_7:6.1.7600:sp1:-:en\-us:home_premium:x64:--`
  - Or a description of a set of products
    - `cpe:2.3:o:microsoft:windows_7:*:*:*:*:home*:x64:*:*`

- Can’t gracefully handle vendor/product name changes
- Can’t represent useful relationships, e.g., part-of, next-version, ...
- Can’t represent roles, e.g., server, client, domain-controller, ...
- Doesn’t support needs of non-credentialed scanners
Status and Next Steps

- CPE v2.3 essentially done
  - Still to do: webinar and review/feedback session at Developer Days in June

- MITRE and NIST met on 2 May to discuss CPE future/plans

- Identified three tasks, in this priority order:
  1. Prepare technical proposal to transition v2.2 Official Dictionary to v2.3, taking advantage of new name attributes
  2. Collaborate with TagVault.org to establish appropriate ties between CPE names and ISO/IEC 19770-2 software ID tags
  3. Collaborate on an enterprise name-management framework, based on a DOD design proposal
Task 1: Transition Dictionary from v2.2 to v2.3

- Over time, the v2.2 dictionary has inconsistently recorded edition-related components of a product name, ex:
  - cpe:/o:microsoft:windows:vista::x32-enterprise
  - cpe:/o:microsoft:windows-nt:vista::x64-home_premium
  - cpe:/a:hp:insight_diagnostics::online_windows_2003_x64
  - cpe:/a:businessobjects:crystalEnterprise_ras_for_unix
  - cpe:/a:ca:brightstor_arcserve_backup:::oracle
  - cpe:/h:lexmark:x646

- The CPE v2.3 Naming specification defines separate attributes to hold “software edition”, “target hardware”, “target software” data
  - “Unrealized gain” until the dictionary is updated
  - Requires careful analysis, proposal preparation, vetting

- Recommended next steps:
  - Get this done in FY11
Task 2: Collaborate with TagVault.org

- ISO/IEC 19770-2 (2009) is an international standard for “software identification tags” to facilitate software asset management
  - Broad value proposition across many use cases
  - Small but growing industry adoption (e.g., Symantec, Adobe)
- TagVault is a non-profit formed under IEEE-ISTO
  - Trusted registration/certification authority for software identification tags (aka SWID tags)
- Does not displace CPE, but offers strong opportunity to collaborate for mutual benefit
- MITRE attended SWID Summit on 4 May 2011
- Recommended next steps:
  - Join TagVault as Corporate End-User ($1000 annual)
  - Join TagVault working group to define path for integration of CPE names and SWID tags
Task 3: DOD Enterprise CPE Management Architecture (1 of 2)
Task 3: DOD Enterprise CPE Management Architecture (2 of 2)

- DISA is funding prototype development to meet a real operational need
- NIST wants to explore whether this work could form the foundation of a new SCAP specification
- MITRE has done a quick study to understand the key technical issues associated with the proposed design
  - Short summary:
    - Many technical hurdles associated with automatic generation of CPE names at endpoints
    - A major engineering effort that, while valuable, does not address highest-priority CPE community needs
- Recommended next steps:
  - MITRE supports DISA and NIST on request
This design requires “draft” CPE names computed at a lower level.
“Computable” CPE Names for Enterprise Name Management

- Problem: Dictionary maintenance is labor intensive
  - Growth driven by community submissions
  - Human review necessary to validate submissions
  - Existing dictionary full of gaps, inconsistencies

- Some in CPE community have suggested that CPE names could be “computed” from information obtained using standard APIs on endpoints
  - If possible, could significantly enhance value of CPE...
  - ... but there is conflicting information on feasibility
  - DOD exploring this as part of multi-tier CPE name management prototype

- So we decided to do a quick feasibility study
  - Windows 7, Linux (Debian and Fedora), Mac OS X
“Computable” CPE Names: Results (1/4)

Windows:

- MS Installer is standard interface for application installation
- Interface records three attributes: product name, product publisher, and version
- Many challenges to overcome in order to compute a well-formed CPE name using these attributes:

  - Inconsistent recording of publisher:
  - Inconsistent embedding of product-related information in the product name string
    - Microsoft Office Excel MUI (English) 2010
    - Microsoft Visual C++ 2008 ATL Update kb973924 - x64 9.0.30729.4148

- No methods found to reliably extract CPE “update”, “edition”, “target_hw”, “target_sw” name attributes
“Computable” CPE Names: Results (2/4)

**Linux:**

- Package management standards vary by linux distribution
  - Popular management tools include RPM, dpkg, and pkgutil
- These provide reports about “packages”, which are not necessarily the same as applications
- Packages are described in terms of “package maintainer” and “package identifier” attributes
- Challenges:
  - The “package maintainer” is not necessarily the “vendor” or “publisher”
  - Package identifiers are not straightforwardly parsable into CPE name attributes

**Mac:**

- Mac OS X uses linux-style package manager
- But many Mac apps are installed using drag/drop, bypassing the package manager
“Computable” CPE Names: Linux Package Manager Examples (3/4)

- **Firefox on Debian Linux:**
  - Package: firefox
  - Maintainer: Ubuntu Mozilla Team <ubuntu-mozilla-team@lists.ubuntu.com>
  - Version: 3.6.15+build1+nobinonly-0ubuntu0.10.10.1

- **MySQL on Fedora:**
  - Name: mysql
  - Version: 5.1.52
  - Packager: Fedora Project
  - Vendor: Fedora Project

- **Python on Mac OS X**
  - version: 2.6.2
“Computable” CPE Names: Findings (4/4)

- Cannot directly compute most CPE name elements
  - Only “version” seems relatively easy to obtain

- To do: document and post findings to CPE discussion list
Q&A