CPE™
cpe.mitre.org

Common Platform Enumeration

Summary of Recent Developments

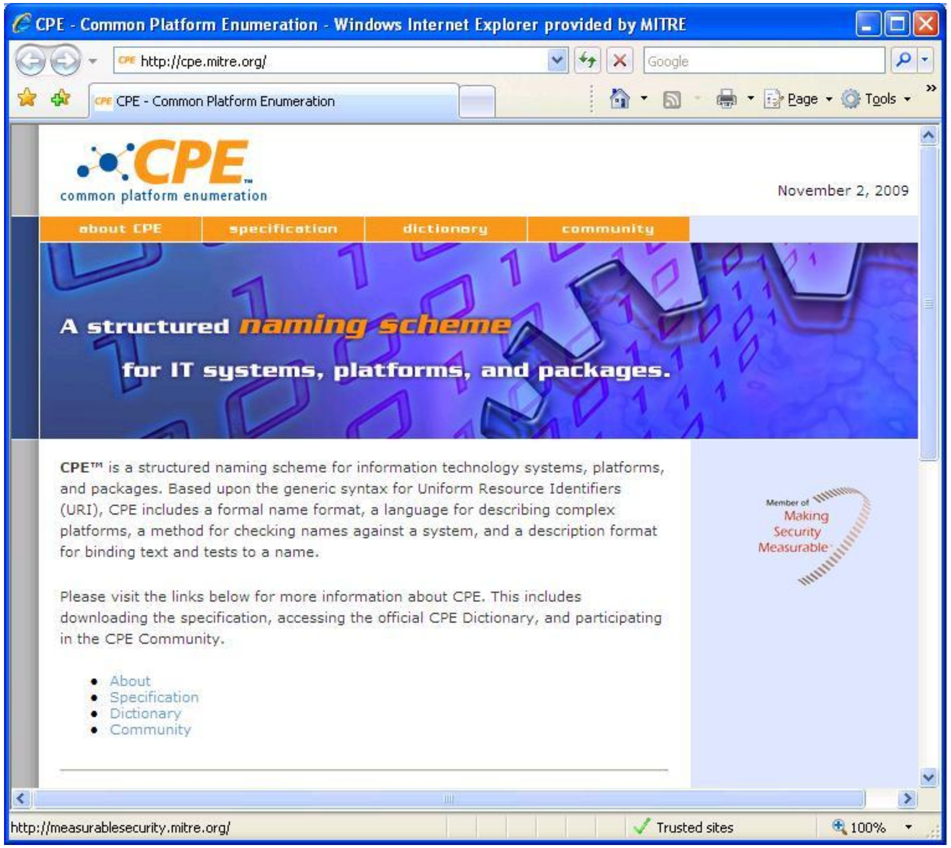
Brant Cheikes

MITRE

Approved for Public Release (11-2789); Distribution Unlimited

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**



CPE - Common Platform Enumeration - Windows Internet Explorer provided by MITRE
http://cpe.mitre.org/
Google
CPE - Common Platform Enumeration
Page
Tools
CPE
common platform enumeration
November 2, 2009
about CPE
specification
dictionary
community
A structured naming scheme
for IT systems, platforms, and packages.
Member of Making Security Measurable
http://measurablesecurity.mitre.org/
Trusted sites
100%

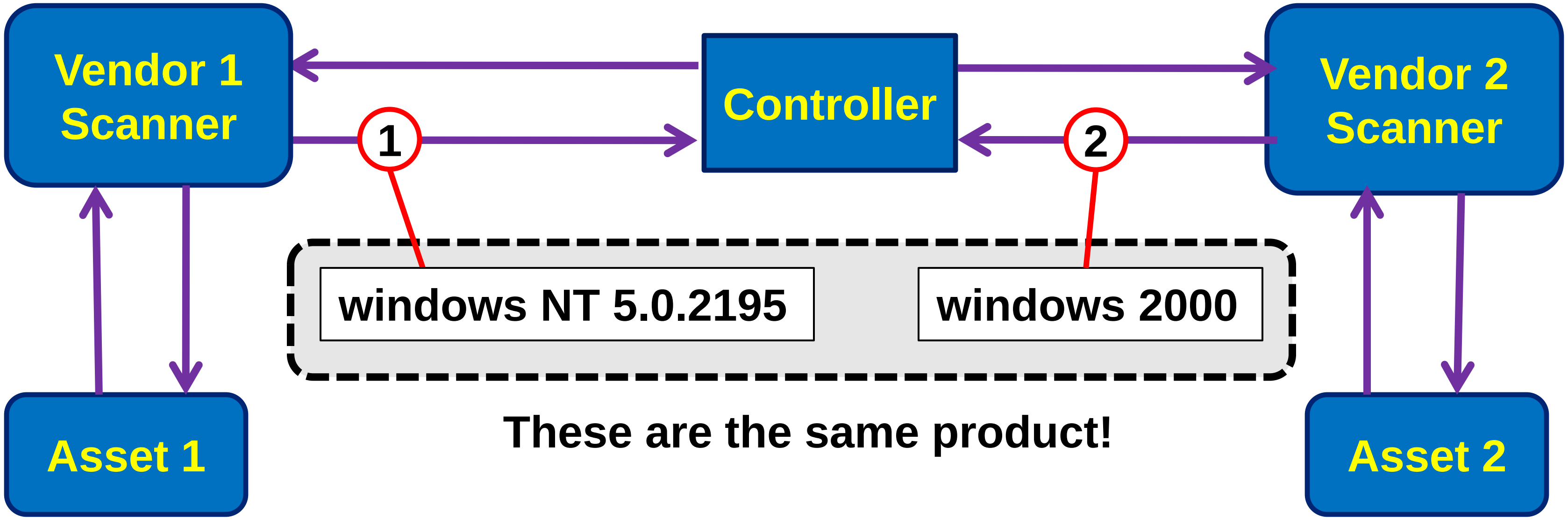
CPE™ is a structured naming scheme for information technology systems, platforms, and packages. Based upon the generic syntax for Uniform Resource Identifiers (URI), CPE includes a formal name format, a language for describing complex platforms, a method for checking names against a system, and a description format for binding text and tests to a name.

Please visit the links below for more information about CPE. This includes downloading the specification, accessing the official CPE Dictionary, and participating in the CPE Community.

- About
- Specification
- Dictionary
- Community

Approved for Public Release (11-2789); Distribution Unlimited

What Problem Does CPE Solve?

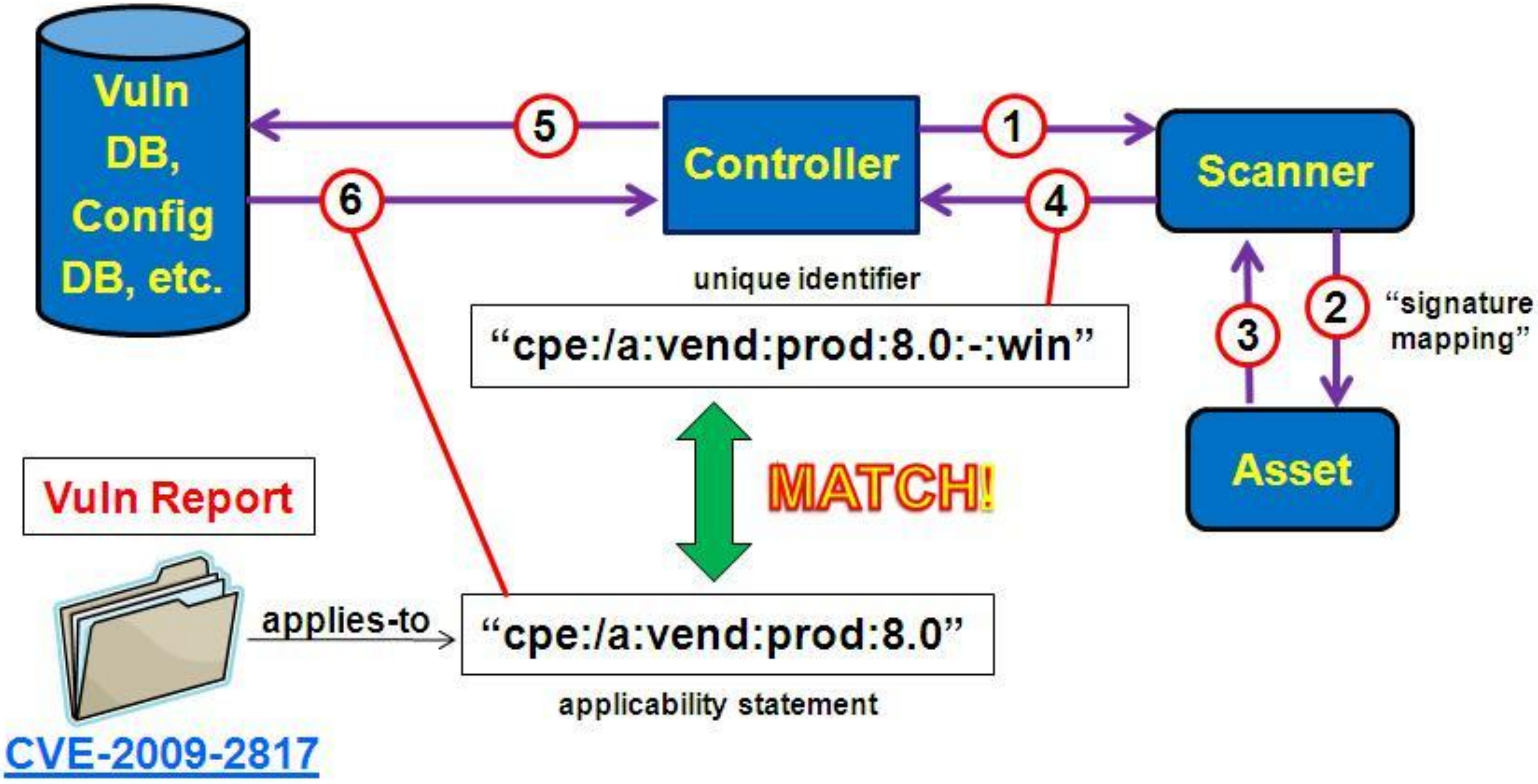


These are the same product!

Interoperable IT Product Names

Approved for Public Release (11-2789); Distribution Unlimited

CPE Concept of Operations



Approved for Public Release (11-2789); Distribution Unlimited

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”

Approved for Public Release (11-2789); Distribution Unlimited

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
    - See: http://nvd.nist.gov/cpe-stats.cfm

Approved for Public Release (11-2789); Distribution Unlimited

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

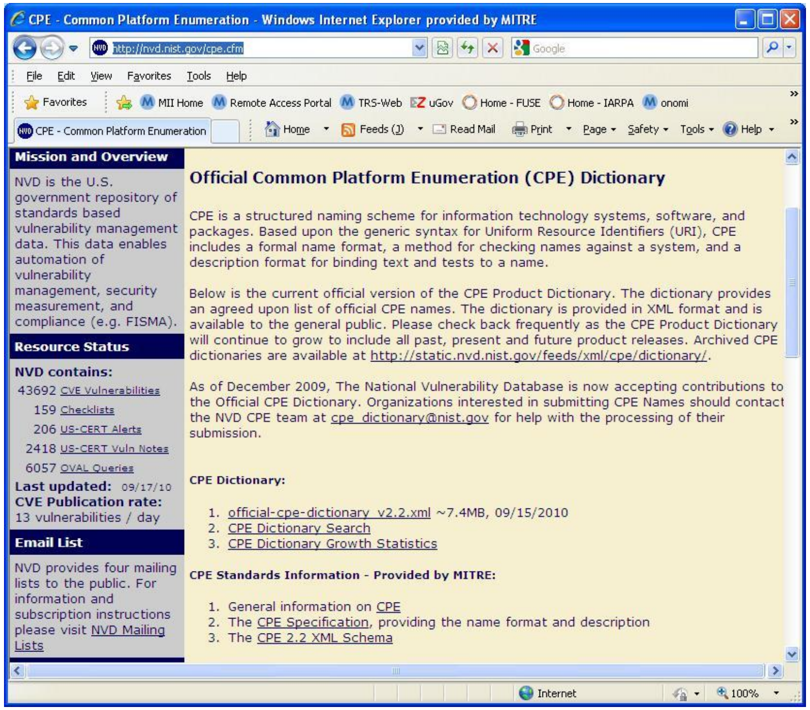
Approved for Public Release (11-2789); Distribution Unlimited

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


CPE - Common Platform Enumeration - Windows Internet Explorer provided by MITRE
http://nvd.nist.gov/cpe.cfm
Google
File Edit View Favorites Tools Help
Favorites
MII Home
Remote Access Portal
TRS-Web
uGov
Home - FUSE
Home - IARPA
onomi
CPE - Common Platform Enumeration
Home
Feeds (J)
Read Mail
Print
Page
Safety
Tools
Help
Internet
100%

Mission and Overview

NVD is the U.S. government repository of standards based vulnerability management data. This data enables automation of vulnerability management, security measurement, and compliance (e.g. FISMA).

Resource Status

NVD contains:

43692 CVE Vulnerabilities
159 Checklists
206 US-CERT Alerts
2418 US-CERT Vuln Notes
6057 OVAL Queries

Last updated: 09/17/10
CVE Publication rate:
13 vulnerabilities / day

Email List

NVD provides four mailing lists to the public. For information and subscription instructions please visit NVD Mailing Lists

Official Common Platform Enumeration (CPE) Dictionary

CPE is a structured naming scheme for information technology systems, software, and packages. Based upon the generic syntax for Uniform Resource Identifiers (URI), CPE includes a formal name format, a method for checking names against a system, and a description format for binding text and tests to a name.

Below is the current official version of the CPE Product Dictionary. The dictionary provides an agreed upon list of official CPE names. The dictionary is provided in XML format and is available to the general public. Please check back frequently as the CPE Product Dictionary will continue to grow to include all past, present and future product releases. Archived CPE dictionaries are available at http://static.nvd.nist.gov/feeds/xml/cpe/dictionary/.

As of December 2009, The National Vulnerability Database is now accepting contributions to the Official CPE Dictionary. Organizations interested in submitting CPE Names should contact the NVD CPE team at cpe_dictionary@nist.gov for help with the processing of their submission.

CPE Dictionary:

1. official-cpe-dictionary_v2.2.xml ~7.4MB, 09/15/2010
2. CPE Dictionary Search
3. CPE Dictionary Growth Statistics

CPE Standards Information - Provided by MITRE:

1. General information on CPE
2. The CPE Specification, providing the name format and description
3. The CPE 2.2 XML Schema

MITRE

Approved for Public Release (11-2789); Distribution Unlimited

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

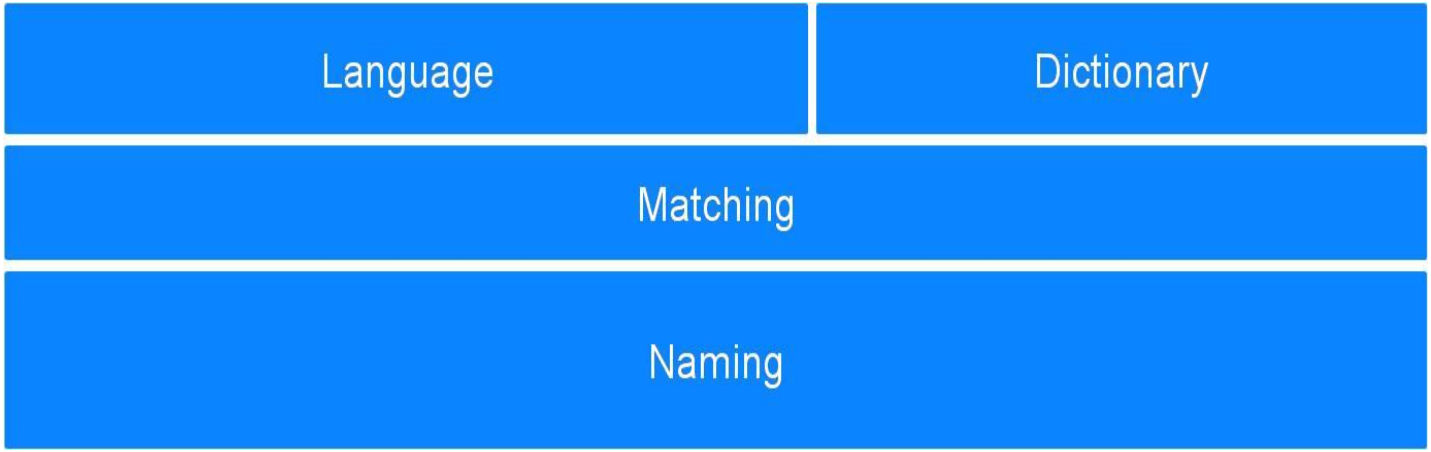
- CPE v2.3 developed on short timeline (March thru July)
  - Fundamental changes to the "architecture" of CPE
  - Minimal changes to the functionality of CPE

Approved for Public Release (11-2789); Distribution Unlimited

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 2nd 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



Language

Dictionary

Matching

Naming

- 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: software_edition, target_sw, target_hw, 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 <u>abstraction</u>, not intended for machine interchange**
  - **an <u>unordered list</u> of attribute-value pairs**
- **Eleven (11) allowed attributes are specified**
- **Attribute values are:**
  - **Logical values (ANY or NA), or**
  - **Character strings obeying certain requirements**

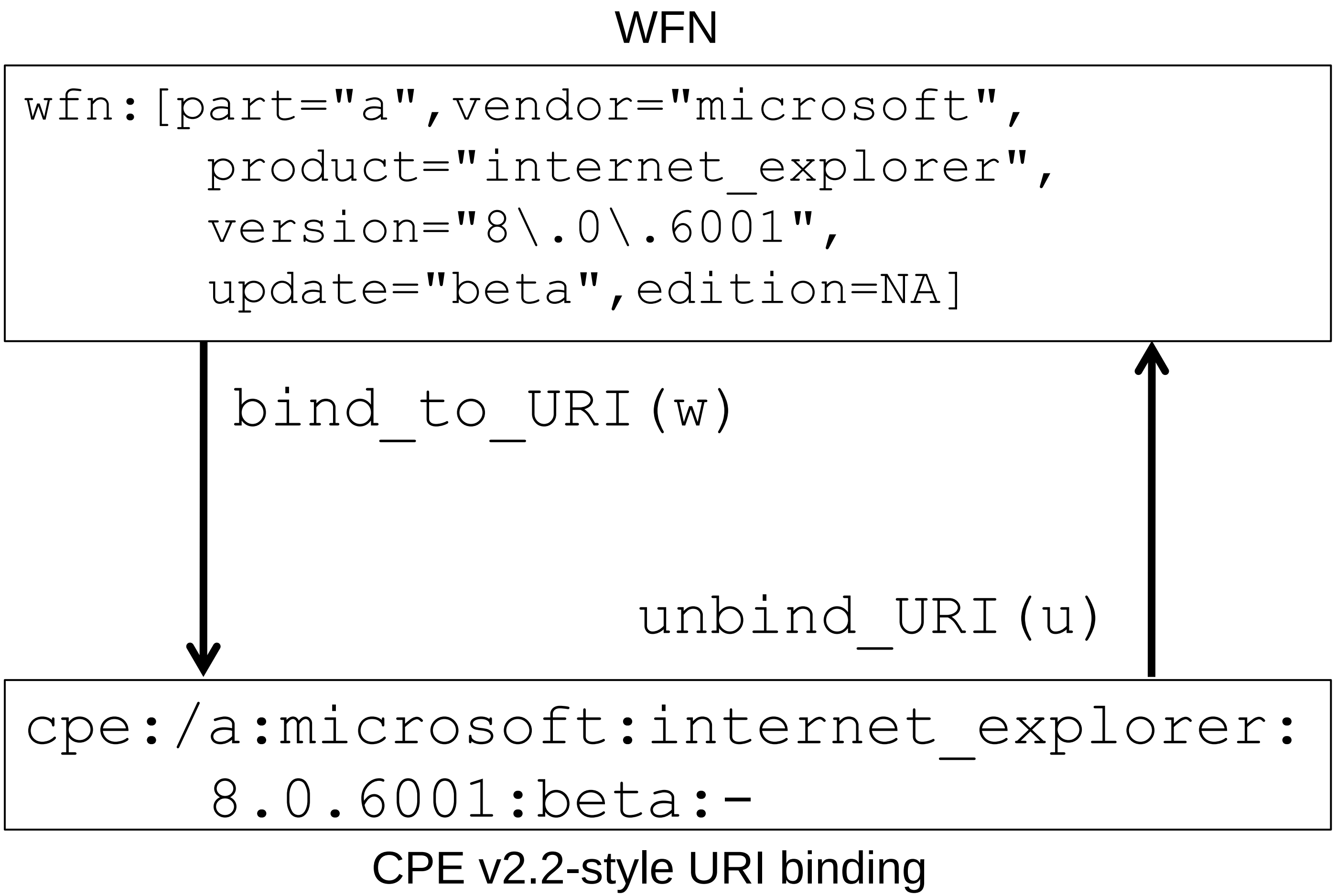
Naming (2 of 5):
The Well-Formed Name (WFN)

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



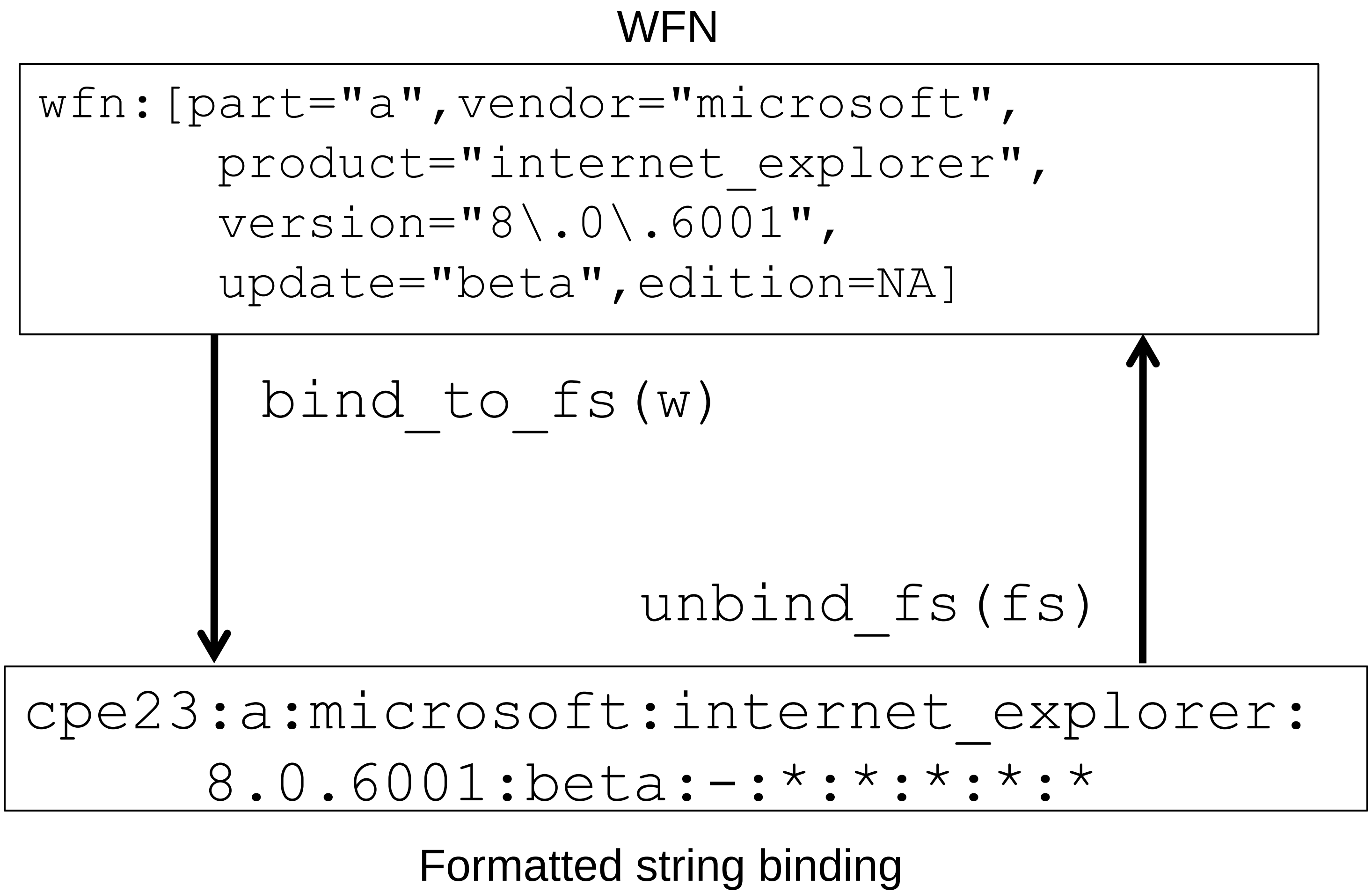
WFN

```
wfn:[part="a",vendor="microsoft",
      product="internet_explorer",
      version="8\.0\.6001",
      update="beta",edition=NA]
```

```
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

```
wfn:[part="a",vendor="microsoft",
      product="internet_explorer",
      version="8\.0\.6001",
      update="beta",edition=NA]
```

`bind_to_fs(w)`

`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

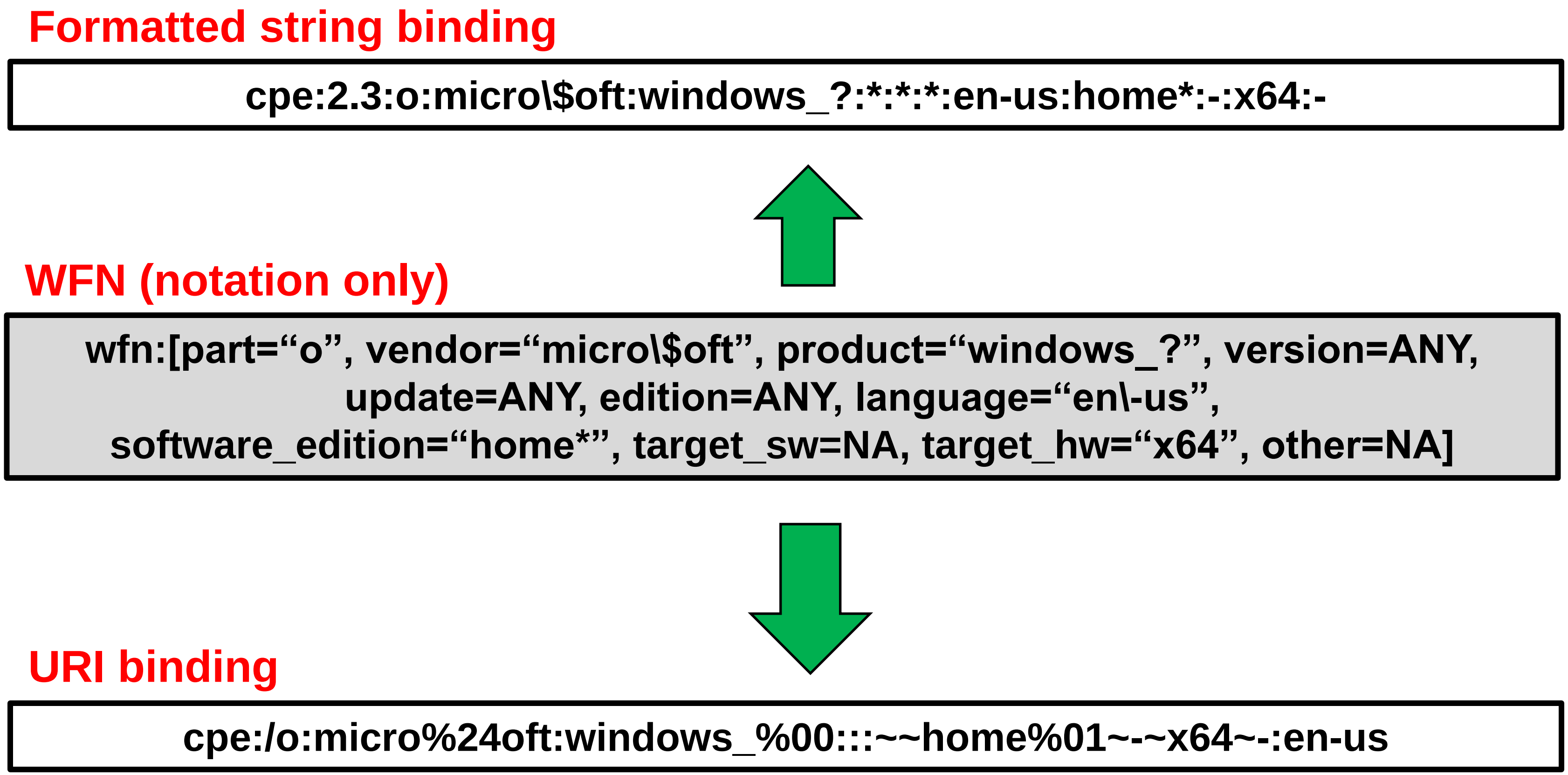
Carried over from CPE 2.2

- sw_edition
- target_sw
- target_hw
- other

New in CPE 2.3

Approved for Public Release (11-2789); Distribution Unlimited

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:-
```

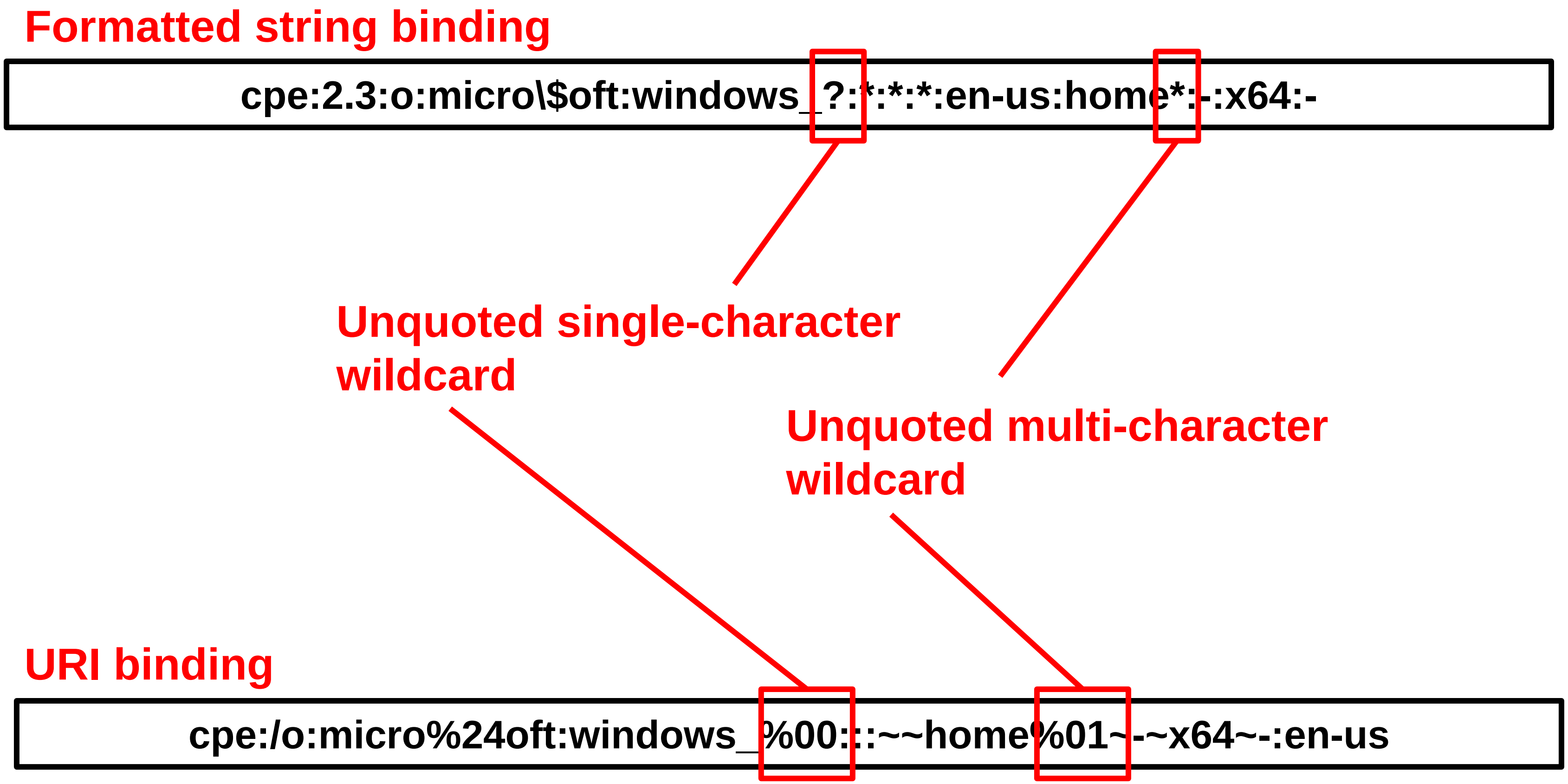
Backslash escape character

Percent encoding

URI binding

```
cpe:/o:micro%24oft: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:-

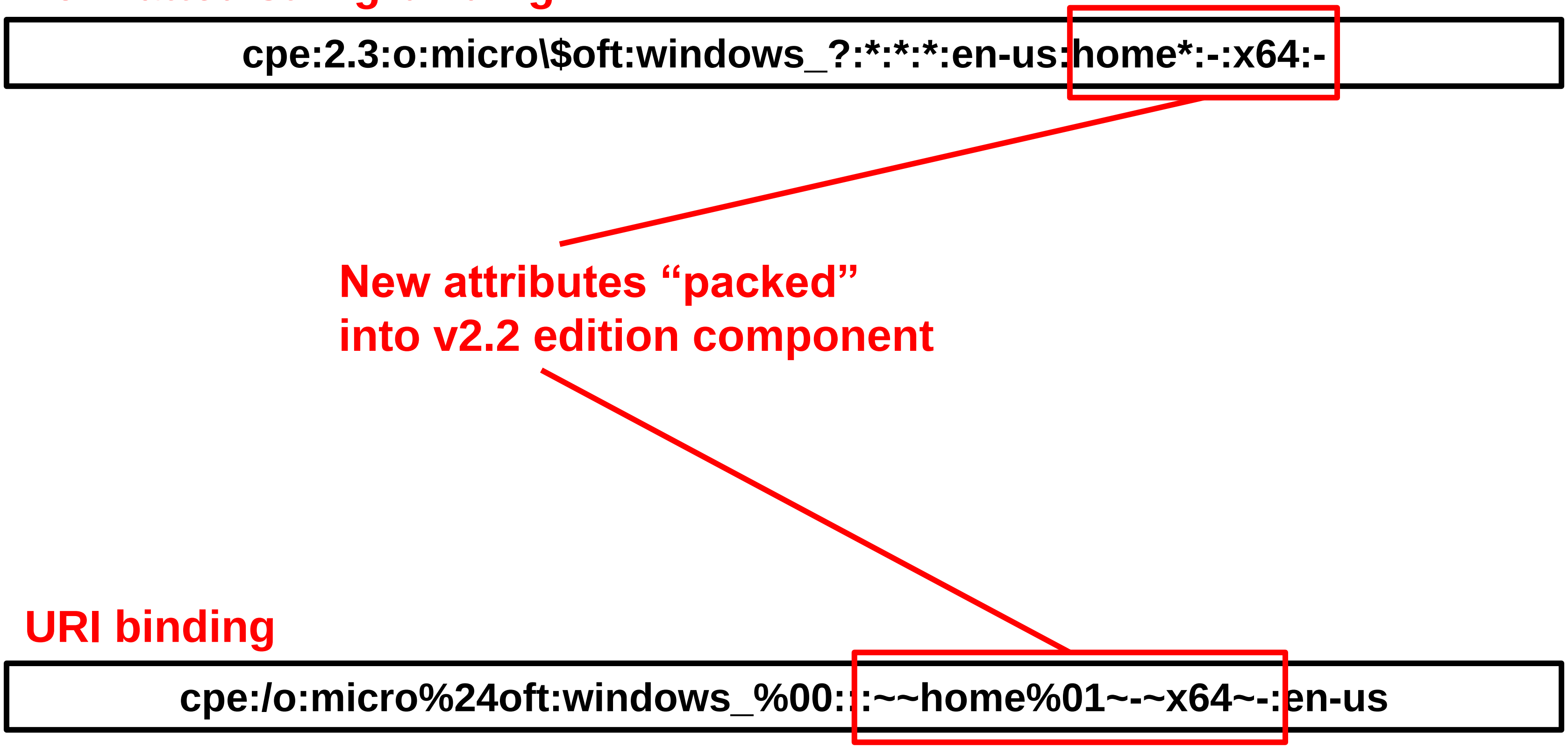
Unquoted single-character wildcard

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

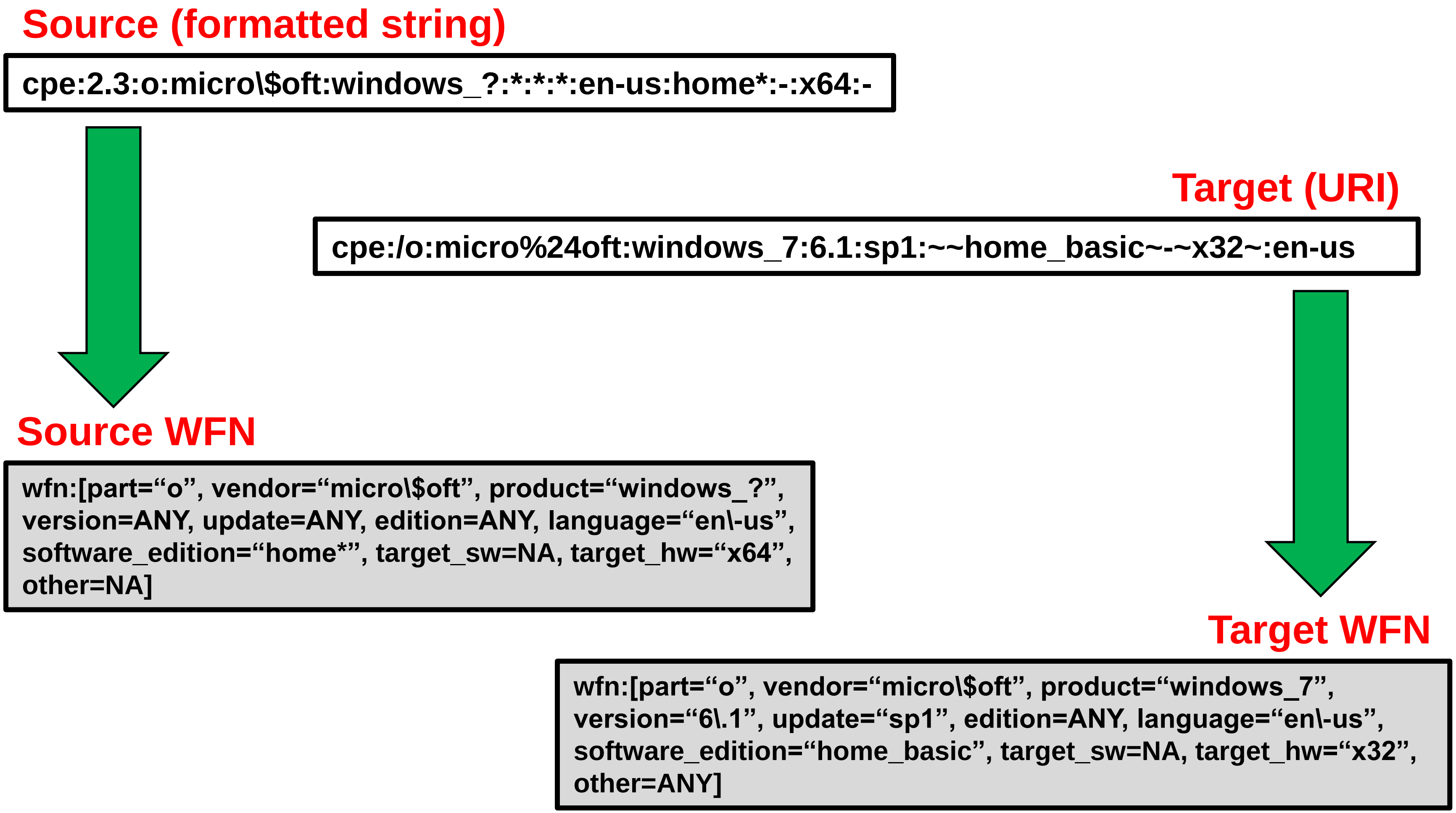
Approved for Public Release (11-2789); Distribution Unlimited

Matching: Overview

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

- Specified functions:
  - `CPE_Name_Compare(source, target)`
    - Pairwise compares source attribute values to target attribute values
    - Returns a table of results
  - `CPE_Attribute_Compare(source, target)`
    - Compares a source attribute value to a target attribute value
    - Returns a result
  - `CPE_x(source, target)`
    - x one of DISJOINT, SUBSET, SUPERSET, EQUAL, INTERSECT
    - Compares a source WFN to a target WFN and returns 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

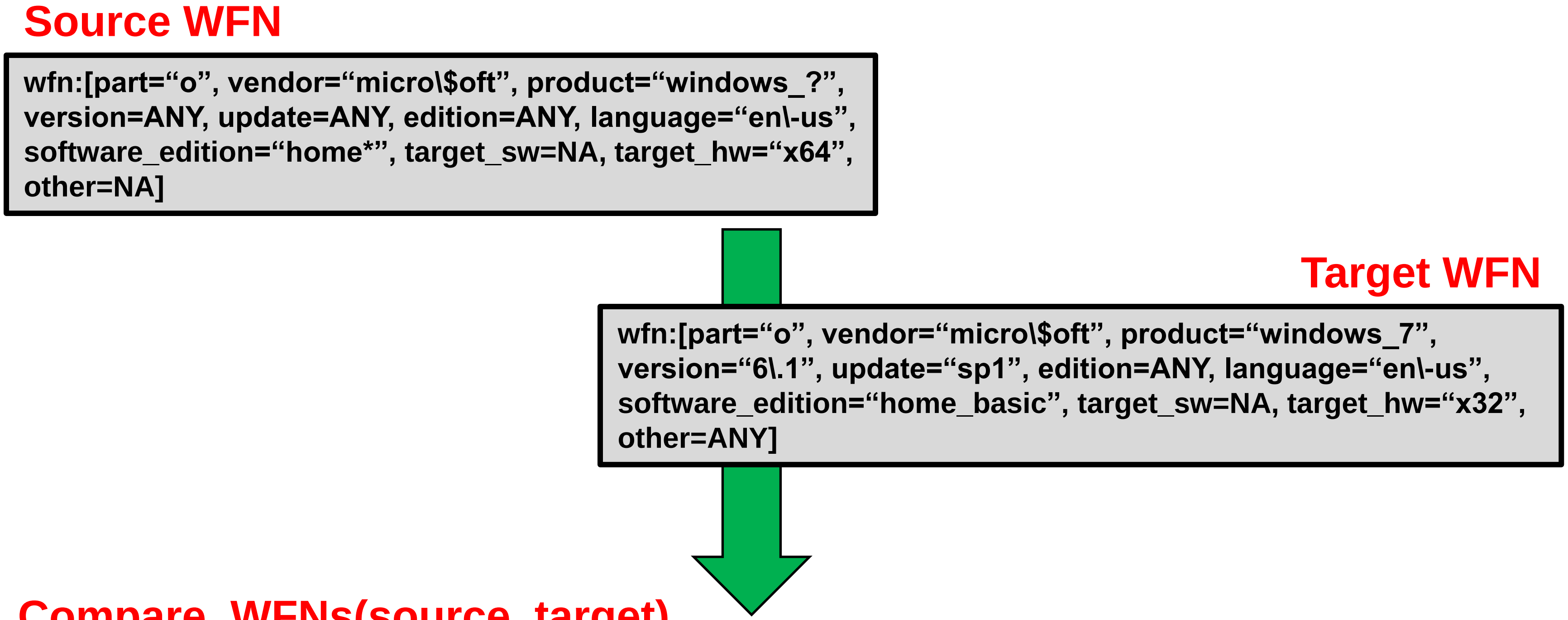
wfn:[part=“o”, vendor=“micro\$oft”, product=“windows_?”, version=ANY, update=ANY, edition=ANY, language=“en\-us”, software_edition=“home*”, target_sw=NA, target_hw=“x64”, other=NA]

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]

Approved for Public Release (11-2789); Distribution Unlimited

Matching (2 of 5):
Step 2 – Attribute-Level Comparison



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]

Compare_WFNs(source, target)

Attrib	Part	Vendor	Product	Version	Sw_ed	Tgt_sw	Tgt_hw	Other
Src	o	micro\$oft	windows_?	ANY	home*	NA	x64	NA
Tgt	o	micro\$oft	windows_7	6\.1	home_basic	NA	x32	ANY
Result	=	=	⊃	⊃	⊃	=	≠	⊂

Approved for Public Release (11-2789); Distribution Unlimited

Matching (3 of 5): Name Comparison Table

No.	If Attribute Relation Set =	Then Name Comparison Relation
1	If any attribute relation is DISJOINT (≠)	Then CPE name relation is DISJOINT(≠)
2	If all attribute relations are EQUAL (=)	Then CPE name relation is EQUAL (=)
3	If all attribute relations are SUBSET (⊂) or EQUAL (=)	Then CPE name relation is SUBSET(⊂)
4	If all attribute relations are SUPERSET (⊃) or EQUAL (=)	Then CPE name relation is SUPERSET (⊃)

Approved for Public Release (11-2789); Distribution Unlimited

Matching (5 of 5): Name-Level Results

CPE_Disjoint=TRUE, CPE_Equal=FALSE

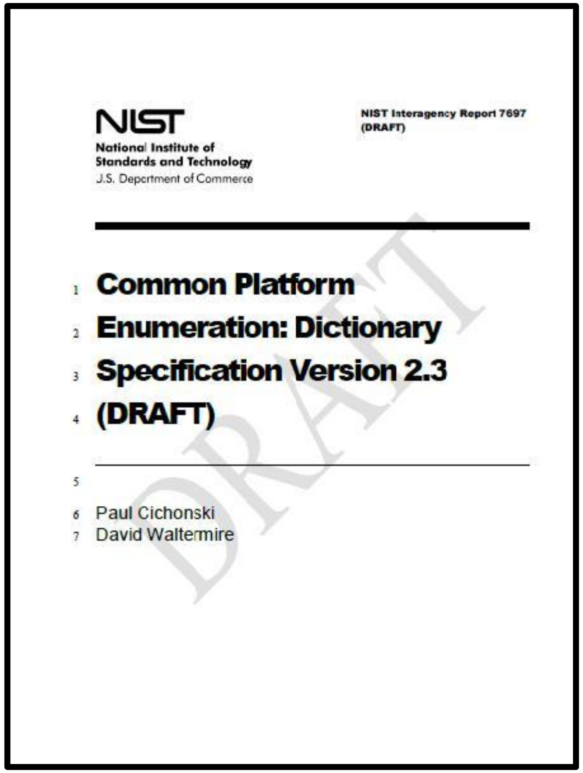
Attrib	Part	Vendor	Product	Version	Sw_ed	Tgt_sw	Tgt_hw	Other
Src	o	micro\$oft	windows_?	ANY	home*	NA	x64	NA
Tgt	o	micro\$oft	windows_7	6\.1	home_basic	NA	x32	ANY
Result	=	=	⊃	⊃	⊃	=	≠	⊂

CPE_Superset=TRUE (equivalent to v2.2 CPE_NAME_MATCH)

Attrib	Part	Vendor	Product	Version	Sw_ed	Tgt_sw	Tgt_hw	Other
Src	o	micro\$oft	windows_?	ANY	home*	NA	x64	NA
Tgt	o	micro\$oft	windows_7	6\.1	home_basic	NA	x64	NA
Result	=	=	⊃	⊃	⊃	=	=	=

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



NIST
National Institute of
Standards and Technology
U.S. Department of Commerce

NIST Interagency Report 7697
(DRAFT)

Common Platform Enumeration: Dictionary Specification Version 2.3 (DRAFT)

Paul Cichonski
David Waltermire

Approved for Public Release (11-2789); Distribution Unlimited

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

Approved for Public Release (11-2789); Distribution Unlimited

Task 1: Transition Dictionary from v2.2 to v2.3

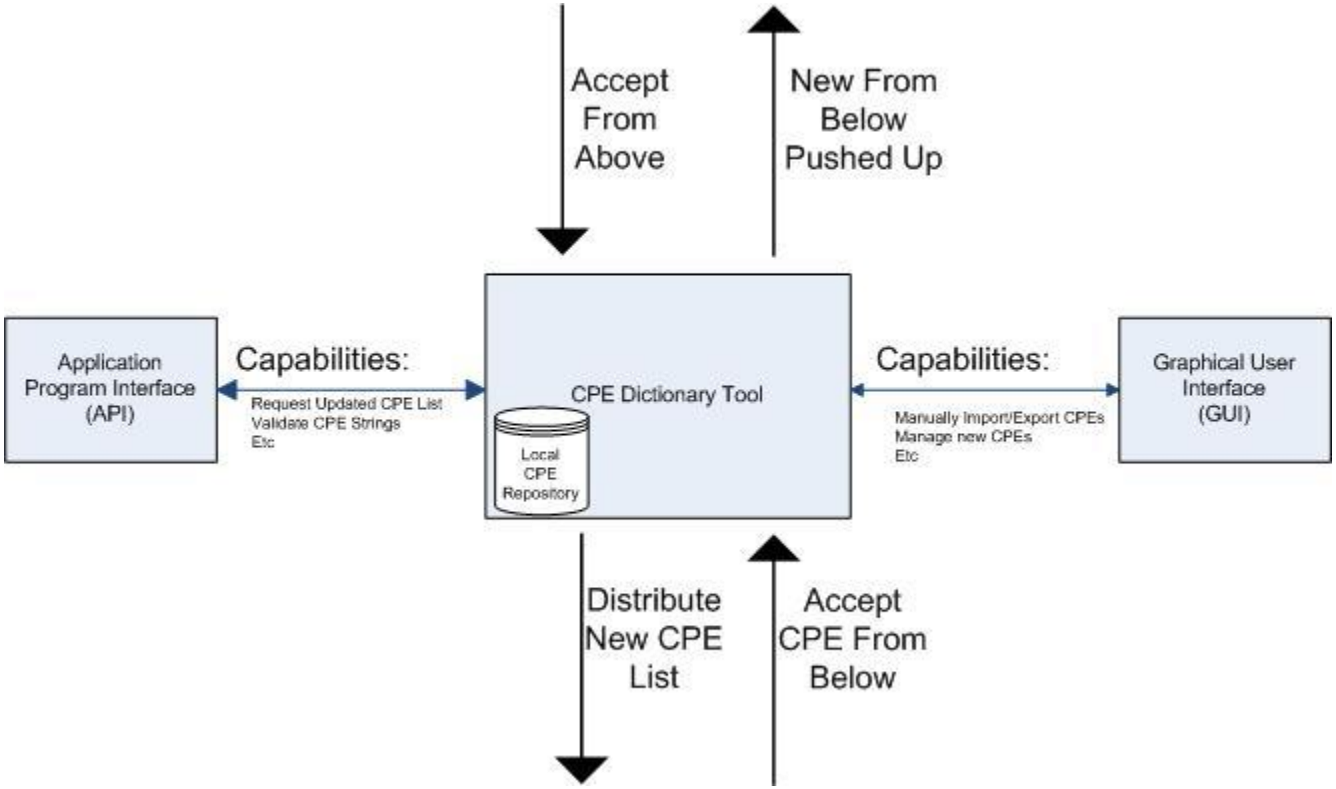
- Over time, the v2.2 dictionary has inconsistently recorded edition-related components of a product name, ex:
  - cpe:/o:microsoft:windows:vista::x32-enterprise
  - cpe:/o:microsoft:windows-nt:vista::x64-home_premium
  - cpe:/a:hp:insight_diagnostics::online_windows_2003_x64
  - cpe:/a:businessobjects:crystal_enterprise_ras_for_unix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

Approved for Public Release (11-2789); Distribution Unlimited

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)



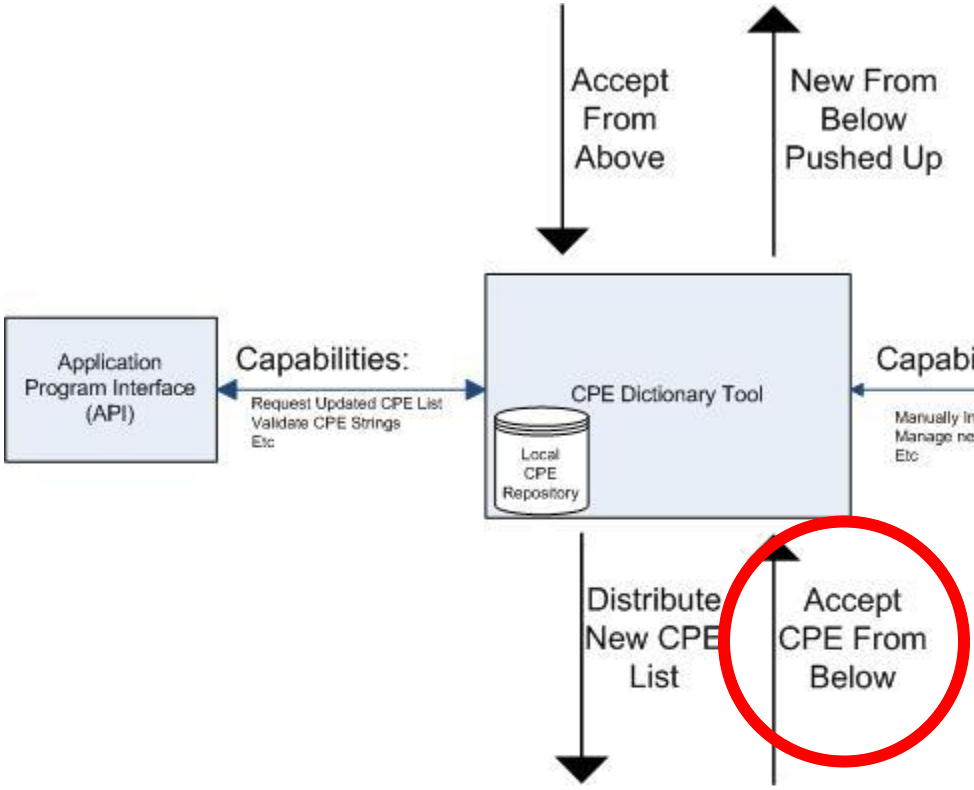
Approved for Public Release (11-2789); Distribution Unlimited

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

Approved for Public Release (11-2789); Distribution Unlimited

DOD Enterprise CPE Management Vision



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:
      - Adobe, Adobe Systems, Adobe Inc., Adobe Systems Incorporated, …
    - 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

Approved for Public Release (11-2789); Distribution Unlimited

“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-mozillateam@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
  - package-id: org.python.Python.PythonApplications-2.6
  - 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