Common Platform Enumeration

Developer Days – 9 July 2012

Brant Cheikes
bcheikes@mitre.org

Steve Klos
stevek@tagvault.org

MITRE

Session Objectives

- Raise awareness of:
  - Software Identification (SWID) tags
  - Relationship between CPE and SWID tags
- Discuss and solicit feedback on:
  - “Secure Asset Management” SWID tag certification plan
  - Embedding of CPE information in certified SWID tags
  - Requirements for package_footprint in certified SWID tags
- Encourage others to get involved

Background

- ISO/IEC 19770-2 SWID standard published in 2009
- TagVault.org formed under IEEE-ISTO in Feb 2009 to evangelize tag use
- MITRE joined TagVault.org as a Corporate member in 2011
- SWID BoF at ITSAC 2011 (Nov)
- MITRE and TagVault collaborated on *Certified SWID Tag Integration with CPE names* (proposal V1: 6 Jan 2012)
  - Defined method to embed CPE name information in SWID tags at time of tag creation
  - Circulated on cpe-discussion-list on 1 Feb 2012
  - Posted here: http://tagvault.org/sites/default/files/Certified%20SWID%20Tags%20integration%20with%20CPE%20names%20V1.0.pdf
- Continued to evolve approach within TagVault community (V2: 9 Apr 2012)
  - Posted here: https://register.mitre.org/devdays/certified_swid_tags_integration_cpe_names.pdf

Defining the Software Ecosystem

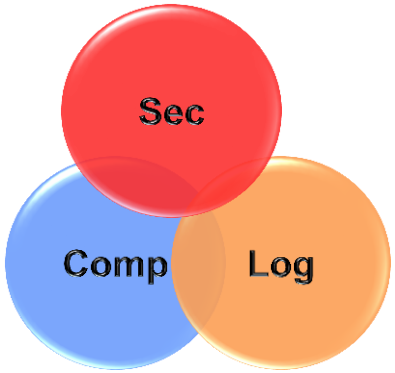
Keystones in the Market

MITRE

Realities of Software Identification

Software identification is more critical than ever

- Compliance is only one use case
- Security
  - Validation of software
  - Vulnerability assessment
  - Executable and patching identification
- Logistics
  - Order & deployment catalogs
  - Invoice reconciliation
  - Disaster Recovery support
- Compliance
  - License reconciliation
  - License optimization
  - Software governance (internal & external)



Realities of Software Identification

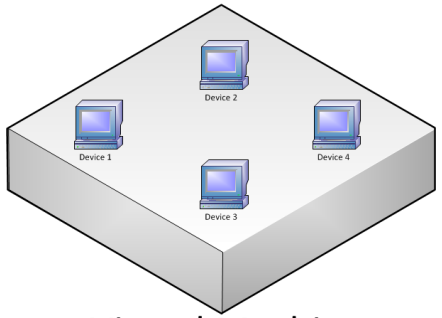
Software identification is more critical than ever

- Issues with 3rd party discovery tools
  - Vary in accuracy and platform support
  - No guarantee of accuracy
  - Training inventory details locks customers in
  - Multiple tools in use with inability to reconcile
- Software Publisher provided discovery tool
  - Each tool requires resources (servers, engineers, IT configurations)
  - Data integration with other IT operations may be limited
  - Ends up with bloated infrastructure and high resource utilization
- Too many publishers, too many configurations, too many releases, too many variables, not nearly enough time or resources!

Software Discovery Tool Analysis

Do we really have a problem?

4 Test Devices
- One system has base OS + MSIE 8.0
- Other systems built off base
- No “tricks” used during installation



Virtual Machine Test Devices

25 current products from **9** different publishers
- Mix and match installations
- Primary focus on current Microsoft and Adobe products

6 Discovery products tested
- 5 Commercial, 1 Open source
- 1 Product included raw and “interpreted” inventory

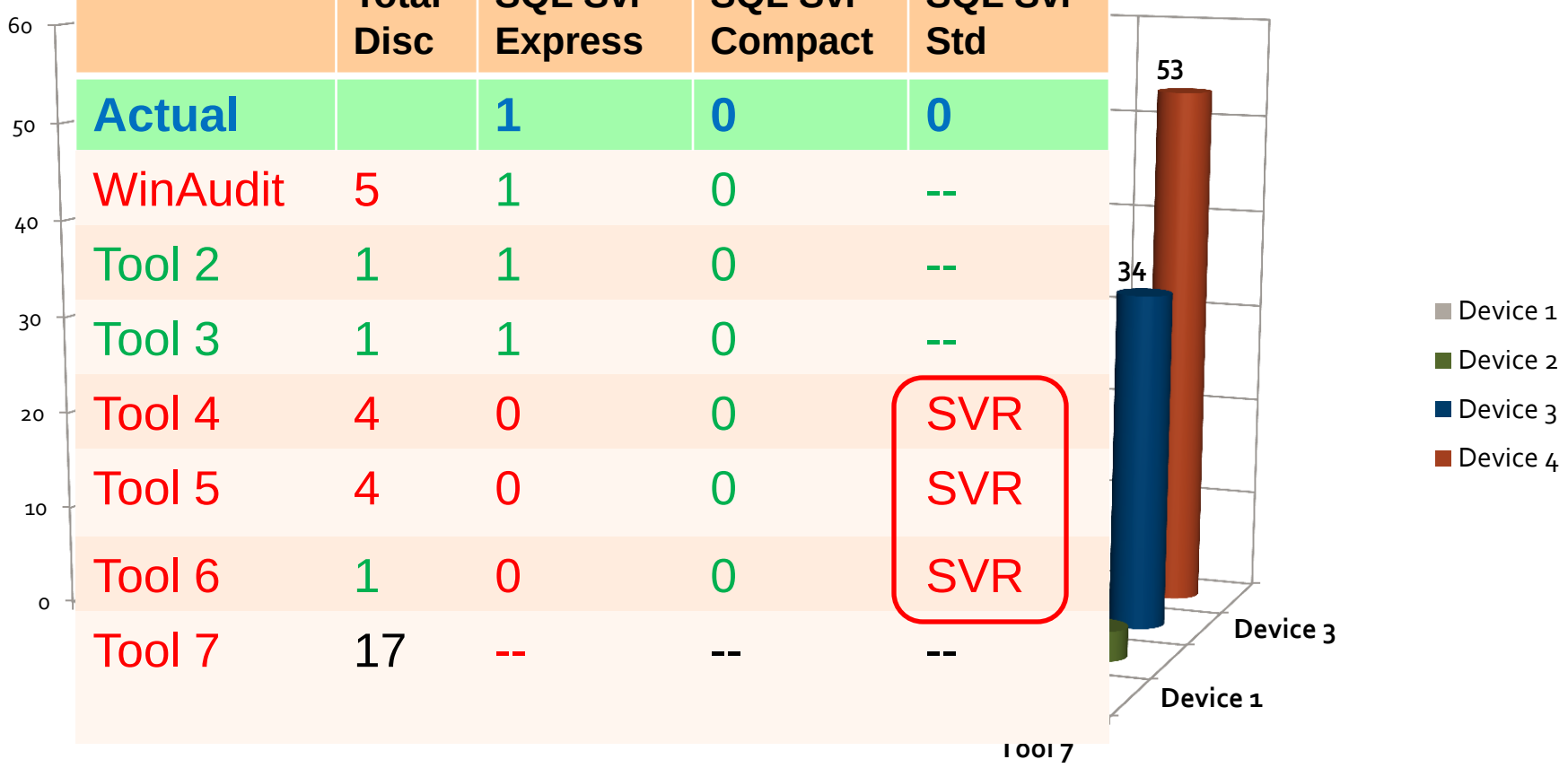
- All Inventory data normalized into one database
- Tool zero (‘0’) – results you would get with SWID tag data

6 products resulted in **7 different answers**

SQL Items Discovered

Device 3 result

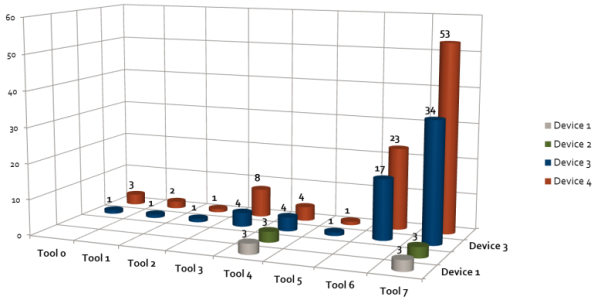
	Total Disc	SQL Svr Express	SQL Svr Compact	SQL Svr Std
Actual		**1**	**0**	**0**
WinAudit	5	1	0	--
Tool 2	1	1	0	--
Tool 3	1	1	0	--
Tool 4	4	0	0	SVR
Tool 5	4	0	0	SVR
Tool 6	1	0	0	SVR
Tool 7	17	--	--	--



SQL Items Discovered

Device 4 results

	Total Disc	SQL Svr Express	SQL Svr Compact	SQL Svr Std
Actual		**2**	**1**	**0**
WinAudit	11	2	1	--
Tool 2	12	2	1	--
Tool 3	1	2	0	SVR
Tool 4	8	0	1	--
Tool 5	4	1	0	--
Tool 6	1	1	0	--
Tool 7	23	--	--	--



- **How much trust do you have in your reports?**

MITRE

IEC/ISO 19770-2:2009

"Software Identification Data Tag"

- **Objective:**
  - **To establish a software asset management (SAM) data standard for software identification through creation of tags that contain consistent and reliable information conveying exactly which products are installed on which computing devices**
- **Status:**
  - **Published November 2009**
  - **Update starts in August 2012**
- **Implications:**
  - **Trustworthy software inventory resulting from advanced identification, installation and usage data**
  - **Links to other data sets as required through extended elements**
  - **Defining and linking tag elements to the entitlement elements proposed in 19770-3**

Software Identification (SWID) Tags

What are they

- **XML files installed with S/W**

7 Mandatory Elements

- Entitlement Required
- Product Title
- Product Version
- Software Creator
- Software Licensor
- Software Unique ID
- Tag Creator

30 Optional Elements

- Product Category (UNSPSC)
- Components of a suite
- Previous product or company names
- License Linkage Details (activation status, channel, cust type)
- …

MITRE

How Software ID Tags Help

Improve Accuracy

Consistent naming conventions ensure all installed software is identified by the inventory tool and reported.

Reduce Noise

Tag data enables customers to quickly and easily identify and reconcile items that require licenses, and skip counting other items like service packs, DLLs, etc.

Lower Costs

Tags reduce the amount of time customers need to spend on today's highly manual and complex reconciliation process and reduce costly errors.

Source: Microsoft, Heather Young
From: 2012 SWID Summit

Tags are easy and low cost

- **Microsoft recently announced support for SWID tags and TagVault.org**
  - **http://www.microsoft.com/sam/en/us/softwareid.aspx**
  - **http://blogs.technet.com/b/volume-licensing/archive/2012/04/20/microsoft-adopts-iso-software-identification-swid-tags-to-help-customers-manage-it-inventory.aspx**
  - **http://www.microsoft.com/global/sam/en/us/RichMedia/Software_ID_Tagging_640x480.asx**
- **Windows 8 Preview includes SWID tags**
  - **http://www.itassetmanagement.net/2012/06/14/windows8-iso-tag/**

TagVault.org

- Defining interoperability requirements
  - Similar in concept to Wi-Fi Alliance
- Defining certification requirements
  - Users and publishers work to same requirements
- Defining structural definitions
  - Tag_type allows more contextual representation
- Providing integration support
  - Supporting automation within SCAP
- Providing tools and documentation, services
  - Tag creation, build validation, signing, verification, etc
- Providing a public repository for SWID tag data
  - Genealogical data, community tag data for normalization

SWID Tags – what's needed

What are they

IBM
I.B.M.
IBM Corp.
IBM Corporation
International Business Machines

Regid.1986-04.com.ibm, …

Mergers, acquisitions & divestitures

- Previous company names
- Previous product names
- Product family identification

Other Examples

- **Customer type**
- **Activation status**
- **Distribution channel**
- **Package footprint**

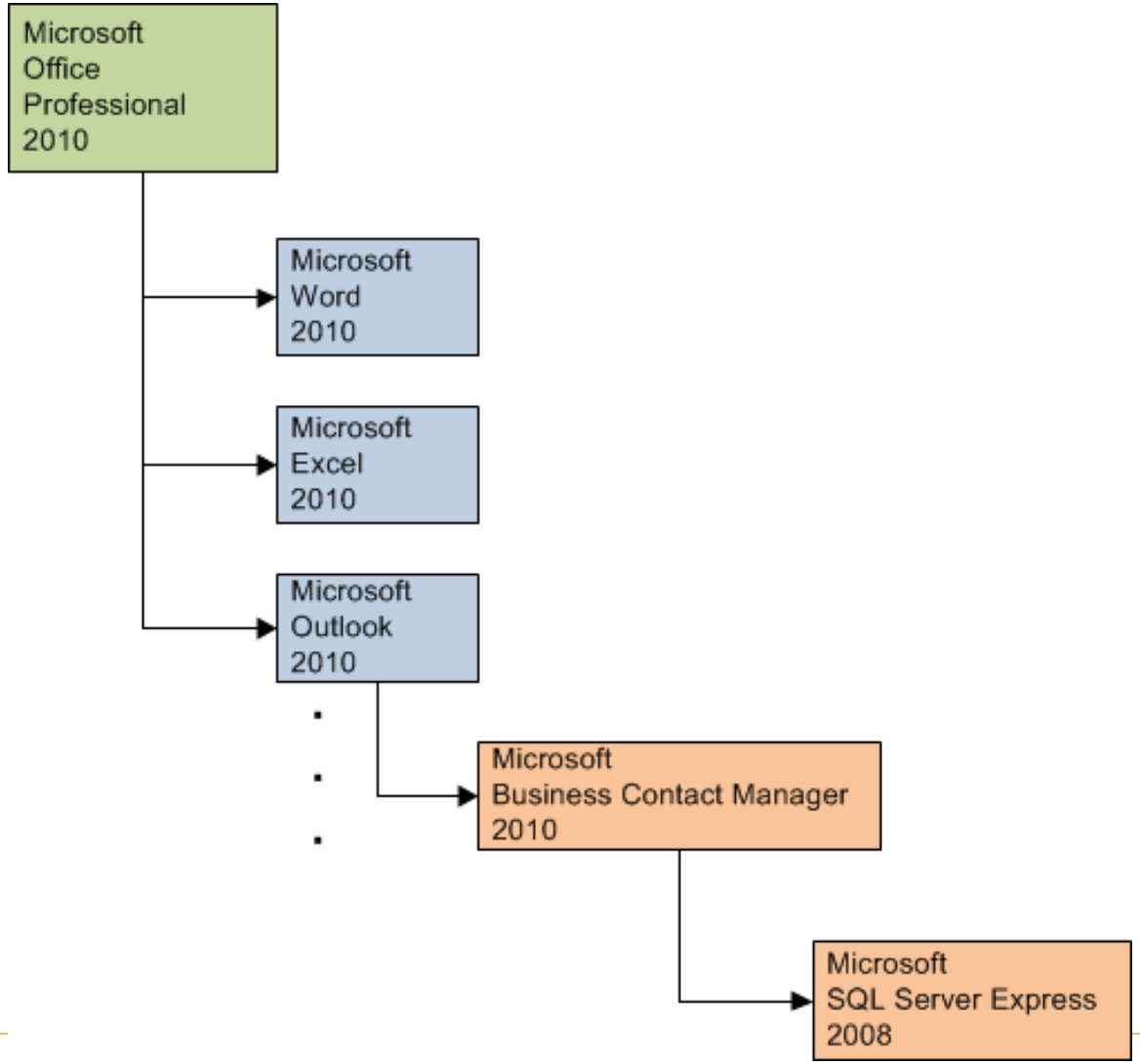
Consistency – Impact on Security

What you don't know CAN hurt you

- **Manually reviewing programs and features**
  - **Quicktime – Ver 7.69.80.9 installed on device**
  - **Does vulnerability apply (Minor Version 69 > 7)?**
- **Consistency & Normalization matter!**

SWID Tags – what's needed

Application Structural Data



MITRE

Certification

Tag Data

- **Digitally signed authoritative entity**
- **Provide package footprint details for an application**
- **Support suites, bundles and other product structures**
- **All tag elements normalized, such as:**
  - **data source**
  - **channel_type**
  - **customer_type**

Example elements of SWID Tag

```
<software_creator>
   <name>Microsoft Corporation</name>
   <regid>regid.1991-06.com.microsoft</regid>
</software_creator>

<swid:product_version>
        <swid:name>14.0.6029.1000</swid:name>
<swid:numeric>
            <swid:major>14</swid:major>
            <swid:minor>0</swid:minor>
            <swid:build>6029</swid:build>
            <swid:review>1000</swid:review>
        </swid:numeric>
</swid:product_version>

<swid:product_name>Office</swid:product_name>
<swid:product_edition>Professional</swid:product_edition>
<swid:licensing_version>2010</swid:licensing_version>

<swid:tag_type>Group</swid:tag_type>
```

MITRE

TagVault.org Certification Efforts

Certification Levels

Current Efforts

- Tag Certification moving to federated model
  - Validation processing done locally at publisher locale
  - Tools and process are very low impact
- SWID Tag Certification Levels
  - Base Certification - mandatory elements, normalized data
  - Asset Management Certification – 5 additional elements + suite/bundle definition
  - Secure Asset Management Certification – includes package_footprint

Future Efforts

- Community Support
  - Normalization and even signing for community S/W
- Tool Provider Certification
  - Reports, exports, data feeds use normalized data

MITRE

Current Uptake - Use

Publishers with SWID tags

- Commercial organizations publishing with SWID tags today
  - Adobe
  - Caphyon (Advanced Installer)
  - Flexera
  - Hewlett Packard
  - Microsoft
  - Symantec

- Platforms on which SWID tags can currently be found
  - Linux
  - Macintosh
  - OpenVMS
  - UNIX
  - Windows

Current Uptake - Tools

Tools Supporting SWID Tags

- **Tag Creation Tools (All three major Windows installer tools)**
  - **Caphyon (Advanced Installer)**
  - **Flexera (Installshield)**
  - **Opensource/Microsoft (WiX)**

- **Discovery/Compliance Tools**

  - **Aspera**
  - **Asset Metrics**
  - **CA Technology**
  - **Eracent**
  - **Hewlett Packard**
  - **Symantec**
  - **Magnicomp**
  - **Microsoft committed to support**
  - **Software Management.org**

MITRE

Key Takeaway

SWID Tags are here and essentially free

- Major software publishers are already supporting tags
- TagVault.org enjoying increased support (Microsoft joined)
- Software publishers can integrate and automate

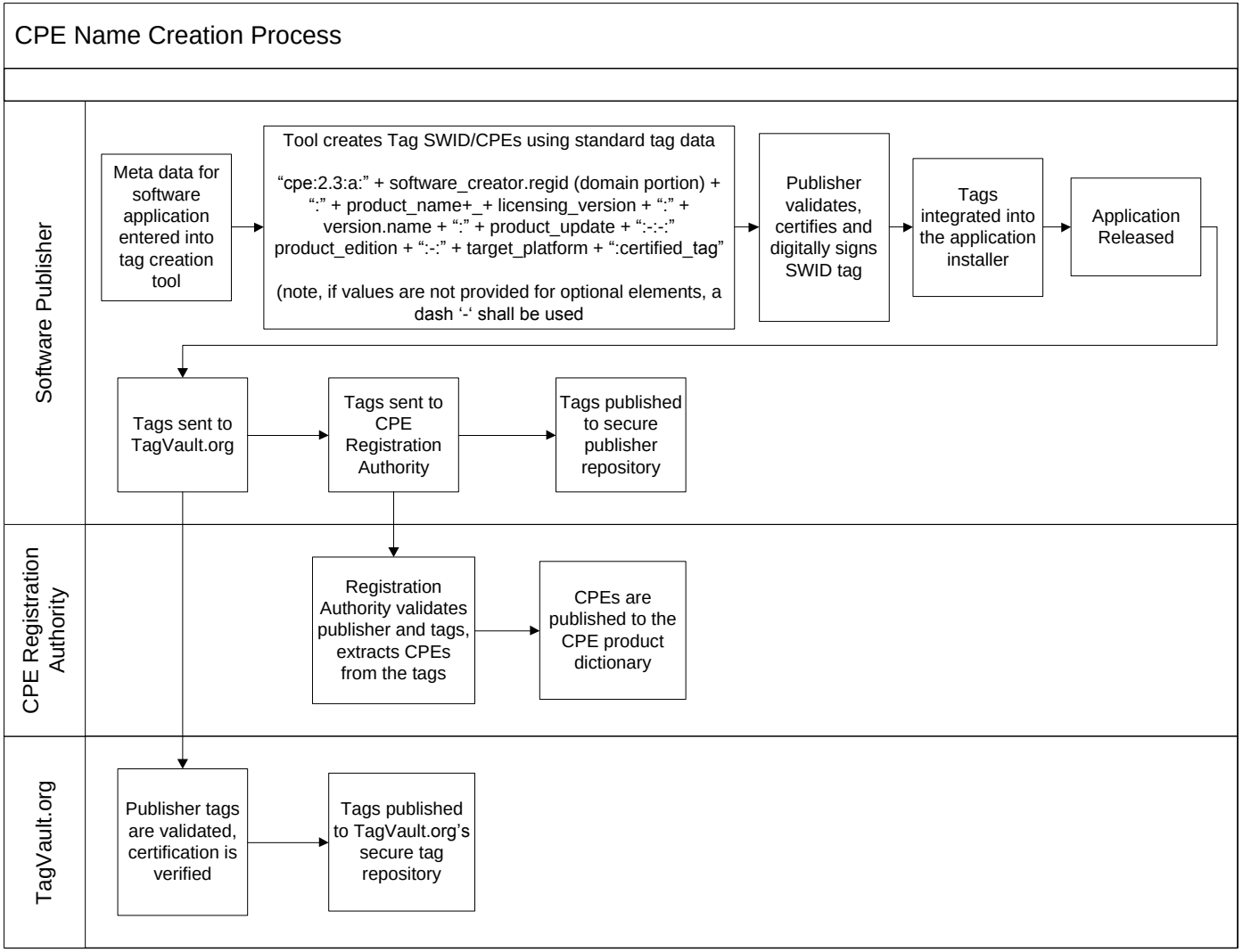
- SWID tags provide value to
  - Publishers
  - Tool providers
  - Software purchasing organizations

- SWID tags are here to stay and with the TagVault.org 3.0 certification requirements, CPE's can be automatically and repeatedly created
- All tools will use exactly the same data set

SWID/CPE Integration Proposal: Overview

- Proposal still under review by TagVault community
  - Must balance desire to integrate w/ CPE with requirement to manage additional burden on software publishers
- Next revision of ISO/IEC 19770-2 will specify five new optional elements of SWID tags:
  - Product_name
  - Licensing_version
  - Product_edition
  - Target_platform
  - Product_update
- These will be required for "Secure Asset Management" tag certification (expected - require for Asset Mgmt and above)
- Elements are combined to construct a valid CPE 2.3-conformant name
  - Not addressed: target_sw, language

Name Creation Process



Building a CPE Name from a SWID Tag

"cpe:2.3:a:"
+ **software_creator.regid** (domain portion) + ":"
+ **product_name** + "_" + **licensing_version** + ":"
+ **version.name** + ":"
+ **product_update** + ":-:-:"
+ **product_edition** + ":-:"
+ **target_platform**
+ ":certified_tag"

Examples:

cpe:2.3:a:tagvault.org:Tag_Creation_Utility:1.0.0.0:-:-:-:-:-:-:certified_tag

cpe:2.3:a:symantec.com:Enterprise_Vault:10.0.1.0:-:-:-:-:-:-:certified_tag

cpe:2.3:a:microsoft.com:Office_2007:12.0.6607.1000:service_pack_3:-:-:
Professional:-:-:certified_tag

CPE "Part": SWID Equivalent

CPE Attribute	SWID Element
Part	None – incorporated as option in the SWID tag verification and registration utility.
	Default option is to specify 'a' for application with options to change to 'o' for operating system when required.

- **Open question:**
  - **How can this information be recovered from the tag?**

CPE "Vendor": SWID Equivalent

CPE Attribute	SWID Element
Vendor	Software_creator.regid (fully qualified domain name in proper order – with the top level domain on the right hand side).
Example: **SWID Tag: software_creator.regid = regid.1992-12.com.symantec** **CPE: Vendor = symantec.com**	

- **SWID tag includes a regid based on the domain name**
- **Given reasonable potential that two different publishers may share the same organization-specific label but have a different top level domain, this proposal recommends that the fully qualified domain name be used for the vendor value**

MITRE

CPE "Product": SWID Equivalent

CPE Attribute	SWID Element
Product	product_name(new)_licensing_version(new)
	Product_name: New SWID tag element defined as an extended element in the TagVault.org certified SWID tag document. product_name value will be modified if necessary to meet the requirements of a CPE attribute-value string.
	Product_name will be registered with all registered values being linked to the regid for the owner. This ensures that the validation steps used for a specified vendor are only for that vendor's products.
	licensing_version: New SWID tag element defined as an extended element in the TagVault.org certified SWID tag document. licensing_version value will be modified if necessary to meet the requirements of a CPE attribute-value string.
	Licensing_version will be registered with all registered values being linked to the regid for the owner. This ensures that the validation steps used for a specified vendor are only for that vendor's products.

Example:

SWID Tag: **product_title = Microsoft Office Professional 2010**
Product_name = Office
Licensing_version = 2010

CPE: product = Office_2010

MITRE

CPE “Update”: SWID Equivalent

CPE Attribute	SWID Element
Update	product_update (new)
	New SWID tag element defined as an extended element in the TagVault.org certified SWID tag document.
	Product_update will be registered with all registered values being linked to the regid for the owner. This ensures that the validation steps used for a specified vendor.
Example: **SWID Tag: product_update = service pack 2** **CPE: Update = service_pack_2**	

- **NB: <u>no normalization across products</u>**
- **Deference to vendor choice**

CPE “SW_Edition”: SWID Equivalent

CPE Attribute	SWID Element
SW_Edition	product_edition (new)
	New SWID tag element defined as an extended element in the TagVault.org certified SWID tag document.
	Product_edition will be registered with all registered values being linked to the regid for the owner. This ensures that the validation steps used for a specified vendor.
Example: **SWID Tag: product_edition = Professional** **CPE: SW_Edition = Professional**	

- **NB: <u>no normalization across products</u>**
- **Deference to vendor choice**

MITRE

CPE “Target_HW”: SWID Equivalent

CPE Attribute	SWID Element
Target_HW	target_platform (new)
	New SWID tag element defined as an extended element in the TagVault.org certified SWID tag document.
	target_platform will be registered as a normalized value list. This normalized list will be consistently verified for every certified SWID tags regardless of publisher.
Example: **SWID Tag: target_platform = x64** **CPE: target_HW = x64**	

- **NB: <u>normalization across products</u>**
- **Valid-values list TBD**

CPE “Other”: SWID Equivalent

- This proposal recommends that when a SWID tag containing a cpe_id is created through a certification process, the cpe_id include the string “certified_tag” as the value of the “other” field in the CPE name.
- This would signal a link between CPE names listed in dictionaries and discoverable SWID tags

Discussion Topics

- Impacts on current Official CPE Dictionary maintained at NIST
- Acceptability of names upon receipt without revision
- Maintaining consistency with existing names and naming conventions
- Lack of cross-product normalization of “update”, “sw_edition”
- Getting to closure

MITRE

TagVault Certification Working Group Draft: Package_Footprint Requirements

MITRE

Background

- TagVault Certification Working Group (TVCWG) preparing requirements document for various SWID tag certification levels
- TVCWG envisages a "Secure Asset Management" certification level, which would require:
  - CPE-related information
  - Package_Footprint
- Requirements for package_footprint under active development
- Seeking security automation community feedback on preliminary draft

MITRE

Example

```
<swid:package_footprint>
        <swid:primary>
                <swid:file>
                    <swid:name>WINWORD.EXE</swid:name>
                    <swid:size>1422680</swid:size>
                    <swid:version>14.0.6024.1000</swid:version>
                    <swid:other name="SHA512"> 3b24….c963be</swid:other>
                    <swid:other name="Path">Office14\</swid:other>
                    <swid:other name="Attrib">A-----</swid:other>
                </swid:file>
        </swid:primary>
</swid:package_footprint>
```

Stakeholder Analysis: Tag Creators

- *Tag creators* need to analyze software applications and prepare corresponding SWID tags for Secure Asset Management certification
- They need to constrain cost/burden of tag creation
- We will assume:
  - Straightforward to comprehensively enumerate the files that comprise an application to be tagged, and
  - This enumeration process can be automated, but
  - Harder to automate "binning" of files into various categories
- Given a file listing, a tag creator must be able to inspect each file in the list and decide whether or not to include it in the package_footprint
- We recommend that any requirements placed on the package_footprint enable tag creators to fully automate the procedure for determining which files should be included

Stakeholder Analysis: Information Security Managers

- **ISMs must ensure that any installed applications are authorized, comply with policy, have not been tampered with, and are free of known vulnerabilities**

- **Tags can help ISMs:**
  - ***Confirm product presence, version, and patch level***
  - ***Confirm executable file integrity***
  - ***Confirm data file integrity***
  - ***Flag suspicious files***

- **ISMs want tags to give them all the information needed to automate the above security processes**

Option Space

- Key files only
- Executable files only
- All files created due to product installation
- All files created or modified due to product installation or execution

Option 1: Key Files Only

- **The package_footprint shall include the product's *entry point*, as well as any other file which, either by its name or attribute (e.g., size, hash value), reliably determines the product's version and patch level as reflected elsewhere in the tag. Each file shall be listed with its full fixed path relative to any variable installation path, along with its size in bytes, a secure hash value, and all operating system-specific file permissions and modes.**
- **We define the "entry point" of a product as the first *executable* file that must be loaded and run by an operating system before the product is able to provide services either to an end user (for products that are intended for direct use by consumers, such as office automation applications) or to another product (for products that are intended to support the operations of other products, such as web servers and print servers).**
- **We define "reliably determines" as: *effectively distinguishes the product's version and patch level from all prior versions and patch levels of the same product.***

DFN: Executable

- We define an “executable file” as any file that contains computer instructions, whether in the form of *machine code* which can be executed by computing hardware or hardware emulators, *bytecode* which can be executed by a bytecode interpreter, or *scripts* which can be executed by scripting language interpreters. *Library files*, whether statically linked at application build time or dynamically linked at runtime, shall also be considered to be executable files. By definition, executable files are considered *static* in the sense that none of their attributes may change unless there is a change to the product’s version or patch level.

Option 2: Executable Files Only

- **The package_footprint shall include every *executable file* that is copied to a computing device as a result of product installation on that device, unless that executable file is part of a component product that is separately tagged. Each executable file shall be listed with its full fixed path relative to any variable installation path, along with its size in bytes, a secure hash value, and all operating system-specific file permissions and modes.**

Option 3: All files created due to product installation

- **The package_footprint shall include every file that is copied to a computing device as a result of product installation on that device, unless that file is part of a component product that is separately tagged. Each file shall be listed with its full fixed path relative to any variable installation path, along with its size in bytes, a secure hash value, and all operating system-specific file permissions and modes. Each *executable file* shall be explicitly marked as such.**

Option 4: All files created or modified due to product installation or execution

- **The package_footprint shall include every file that is either (a) copied to a computing device as a result of product installation on that device, or (b) created or modified during product execution. Excluded are (a) any files that are part of component products that are separately tagged, and (b) any files whose creation and/or modification are user-dependent and cannot be anticipated in advance. Each file shall be listed with its full fixed path relative to any variable installation path, along with its size in bytes, a secure hash value, and all operating system-specific file permissions and modes. Each executable file shall be explicitly marked as such. Each non-executable file shall be marked "static" if none of its attributes are expected to change due to product execution.**

Status

- **TVCWG leaning towards Option 3—All files created due to installation**
- **Would this be of significant value to the security automation community?**

Q&A