SWID Message and Attributes for PA-TNC

draft-coffin-sacm-nea-swid-patnc-01

https://datatracker.ietf.org/doc/draft-coffin-sacm-nea-swid-patnc/

SACM Virtual Interim Meeting

June 15, 2015

Agenda

- Recent milestones
- Data Model
- Next Steps

Recent Milestones

- Submitted to the list a draft of how many of the technical changes might be implemented
    - This led to good discussion; some changes will be needed

- Released version -01
    - Basically just the removal of references to IF-IMC and IF-IMV, per earlier consensus
    - Other changes have a dependency on the data model discussion

Recent Milestones (2)

- Adopted as a WG draft
  - Basically means “This seems like a reasonable framework for collecting inventory information.”
  - Significant evolution remains likely
- Gunnar Engelbach added as an author
  - He and Charles Schmidt will be the main editors going forward

Data Model

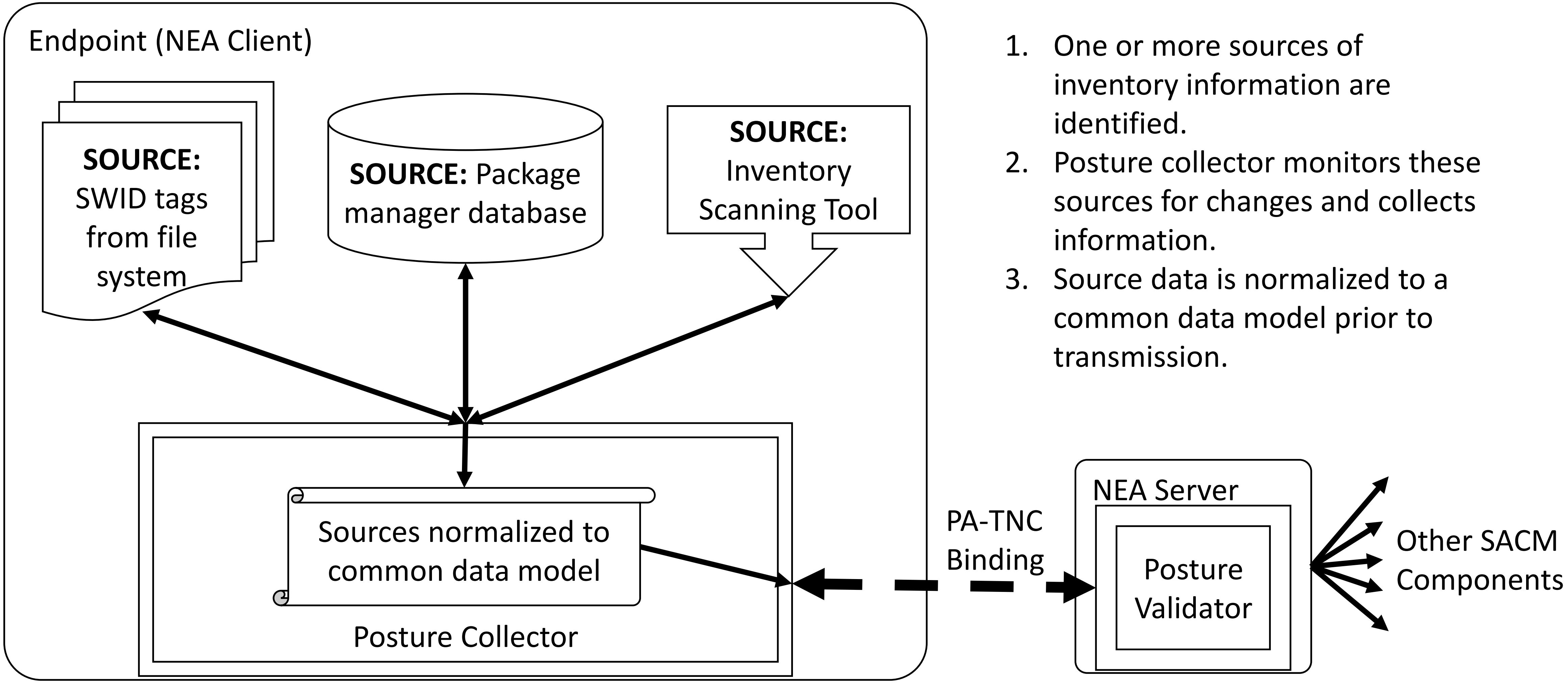
Everything is under discussion!

Data model

- Multiple discussions on the mailing list point to and hinge on the question of the data model of messages
    - Current specification uses ISO SWID 2015 or ISO SWID 2009 as the data model

- Most open issues are can be subsumed by the question of message data model
    - “Add field to indicate application location” -> field of the data model
    - “Add field to indicate tag source” -> field of the data model
    - “Indicate the type/format of tags in messages” -> type/format of the message is the data model of the message
    - “Add field for SWID tag versions” -> field of the data model

Nominal data flow


Endpoint (NEA Client)
SOURCE: SWID tags from file system
SOURCE: Package manager database
SOURCE: Inventory Scanning Tool
Sources normalized to common data model
Posture Collector
PA-TNC Binding
NEA Server
Posture Validator
Other SACM Components

1. One or more sources of inventory information are identified.
2. Posture collector monitors these sources for changes and collects information.
3. Source data is normalized to a common data model prior to transmission.

* Sources are provided as examples. This slide makes no assertions as to any source being required.

Inherent assumptions of this flow

- We are not constraining Sources for software inventory information

- Each piece of data is associated with a single Source
  - Posture Collector receives data and alerts about changes from Sources. Details of this can vary:
    - PC periodically queries for changes
    - Source automatically alerts PC to changes

What should the software inventory data model look like

- There are existing data models for characterizing installed software. Seems reasonable to use one instead of inventing new.

- 3 Criteria suggested:
  1. Is extensible – not dependent on outside bodies/long-timescale procedures to add new fields
  2. Readily accessible – model can be acquired by all without significant burden (e.g., fee, license, export restriction, etc.)
  3. Complete/Sufficiently expressive – want to be able to support reasonable/common use without extension

Use cases of data model

- Identification of software that is vulnerable/needs patching
- Identification of software for license management
- Support software whitelisting/blacklisting
- Support integrity checking of files the comprise software

Necessary data model information

- Software name
- Software version (to whatever precision is necessary)
- Software publisher (arguably necessary to disambiguate names)
- Software location

- Others?

Next steps

- Continue discussions on-list
- Develop a draft data model ahead of IETF 96