SACM Information Model Revisited

Inacio IETF-104 Prague

Previous Information Model

- It was very ambitious
  - 417 elements defined
- Getting further on refining that many data elements was difficult
- Wasn’t easy to determine what was important
- Wasn’t easy to make sense of trade-offs for data models

What if we did the minimum viable?*

- What are the handful of elements we would need to define?
  - Obvious ones: IP addresses, hostnames
  - Mostly obvious: time/date - sample time, “event” time
  - Still kind of obvious: SWID/CoSWID identifiers / firmware versions
  - Practical elements for endpoint ID: serial number, MAC address, HW certs, ?
- But what is the minimum set of things we need to make the information exchange work across our ecosystem?
  - What do folks use as their database keys already?
  - Do we need more than that to start?

* - yeah, I’m all Agile and DevOps and cool and stuff now…

Being a lazy engineer*

- You can only be a good lazy engineer if you can figure out how to make others do your work ☺
  - So the data model has to allow vendor specific extensions into the information model
  - Which means the hard work is still defining what meta's have to be in the information model (name, basic_data_type, byte_length, data_use_type (label, counter, gauge, etc.), description, std/vendor_type, structures/composite, ???)
  - This influences the data model(s)

*Or I'm not smart enough to be a good engineer, so I hide behind the lazy?

Build it?

- Thoughts?
- Is this a way forward?
- Can people imagine a way to use this to connect their systems?
- Could a repository be smart and record what types it had inside?
- Define a new data format? Use an existing data format?
- Build translators?
- IANA registry – when people want to add back the 100 other elements