This document was compiled in real time from the multistakeholder discussion at the July 19 multistakeholder meeting on Software Component Transparency. The notes were recorded live in front of participants. No further attempt to edit or organize the raw notes has been made. For more information, please go to: https://www.ntia.doc.gov/SoftwareTransparency

Open questions for discussion

- **Opacity / Known Unknowns in the graph**
  - Field to indicate level of confidence (if not certain/unknown)?
  - Open source (exe) – if it's exe then an SBOM should come with it. When you compile the open source yourself, then you are the compiler, which means it's not SOUP.
    - Confidence isn't necessarily transitive. (confidence vs lack of knowledge)
    - JavaScript is a different model. You could be accessing 1000 files for one function. Tooling should be built in.
    - "Compiled from" = opaqueness?
    - Authoritative vs nonauthoritative tags – if person creating tag is person who had role in creating/distro'ing software, that's authoritative. If there's a mismatch (multiple parties) then that's nonauth (in SWID). Re manifest of software – 1. Payload = authoritative manifest, provided by creator; 2. Evidence = more forensic, third-party discovery tool, nonauthoritative.
  - All parties must be "well-behaved" for this to work. Different gradations of confidence.
  - Duncan's axiom
  - No substitute for authoritative info.
  - Fewer suppliers increases security etc
  - Art framework: known knowns, known unknowns, data with caveats
  - How to capture the best effort amid uncertainty?
    - SCAs may not give us what we need.
  - Known knowns and known unknowns -> standards group
    - This group's work will allow for translucency (way to indicate neither known or unknown)
  - Translucency high-level discussion -> framing group
  - Obstacles: both internal and external (data can be hard to get inside an org due to multiple product teams etc., possible naming issues)
  - Any relevant info from POC group will be shared
  - Low confidence can become high confidence based on new information
- **Namespace – more consistent naming**
  - We need to create alias lists for vendors, products, brand names, etc.
    - FIRST is creating a supplier list.
    - Individual supplier creates names for themselves.
  - CPE = what not to do. Standardized identifiers for product names. Created when we see a vuln for the first time. Leverage existing mechs for uniqueness, like UU ID. Low likelihood of collision. Something like DNS. In SWID, guidance: use UU IDs or prefix w DNS name. Or, put authoritative parties in charge of providing the info. Supplier could provide.

- Properties of a name: security properties, how will it be used, does it need to be recognizable, authoritatively assigned or randomly assigned?
- Entity resolution – depends on the criticality and assurance posture of the system. In some you need to know exactly where a package came from. Need for extensibility in all formats.
- Software Heritage: wayback for software. Version at point in time. Longitudinally auditable.
- PURL: Use the vector through the package mgr to resolve location of a piece of software – geographic understanding (address – lat/long) is more unique than a name.
- Linux Foundation badge with criterion. Could add criterion for SBOM, certain percentage of certainty. Could motivate developers.
- Example: SBOM system that uses CPE format bc initial purpose was to assoc with vulns. Established, relevant. If no CPE we make a placeholder and update when there is a CPE.
- Just need simple product identifier.
- Suppliers may be interested in supplying CPEs.
- Entities could create their own names, but they may use different formats
- Could still have multiple names for the same thing.
- Properties of name:
    - Human readable
    - Indexable
    - Cross-referenceability
    - Use-case specific
    - Does not need to be unique way
    - Vulnerability use case has naming issue (bc name happens after vuln discovered). May be independent of this process.
- Could identify relationships without needing an alias field
- This needs to be flexible bc there are existing systems in place.
- Federation, aliases for short run, convergence as we move forward
- Using package managers is swell idea.
    - However, they can be compromised.
- Orgs should be able to map across use cases. Tools may ID software different ways. Need to synthesize data across different info sources.
- When manifest and dependency files are not present, still have to account for known unknowns.
- If possible there should only be one way to do each thing.
- Not all fields apply to all use cases.
- Need generic ID system.
- Elements of CPE can be useful. Can be useful framework if you have a big engine doing entity resolution.
- Work to be done: (Framing group will define problem more explicitly)
    - Guidance on what naming looks like
    - Description of existing guidance in naming
    - What are characteristics of identifier? (Who is using the name and who is securing it)

- **Transparency: How can we share SBOM data**
  - Across the supply chain
  - Last mile to an enterprise owner/user
    - Last mile may not be knowable (and can change)
  - Web resources aren't always available, can go out of business, can't always look them up bc of closed environment
  - Model depends on previous supplier having done the same thing, should continue the same format
  - SBOM acquisition use cases – if I hold a piece of software, I need to have a way to find the SBOM. There are ways to deal with resources to address air gap problem. Also there are lots of reasons orgs need to know something about the collection of software it has – they are vuln mgrs. And want to know what's avail. NVD needs to attribute vulns to given software. Suppliers could post repository of their own SBOMs. Must be done in standardized way.
  - Define "as built"? –known open question, where/when to generate SBOM. Can be discussed in framing group.
  - Potential natural custodians of super-set (out of business problem)
  - ISOA can in-source data? Good solution if already in place.
  - Terminology for suppliers of software but also of SBOM (could be different)
  - Centralized repository, possible ML to work on similar names (aliasing)
  - Think about federated capabilities. How to turn into manageable space?
  - Standard exists- Exchange protocol for metadata format- transport protocol. Assumption that data is federated.
  - For users of data- consistency is desired in transmission.
  - Rough thinking of framing group: two formats, multiple ways to get, but only 3-4 collectors. Sector specific, it may be possible to get consistency. High level will need to be general.
  - Important not to conflate topics- need consistent tooling, not necessarily paths.
  - For supplier: Define format, suppliers will produce.
  - From HDO side, 300 different ways of figuring out where SBOMs are
  - Unless it is explicit, problem will be realized on multiplicity of paths
  - Will be possible to produce, especially if tied to CBEs. At least large companies.
  - Talking about SBOM- not only element of product information, need to expand beyond just SBOM. Possibly beyond scope of this group. Shouldn't have just a solution for SBOMs. Have portals for suppliers information and format, as more information is shared it is becoming more of a problem.
  - Collapsing communication patterns
  - Problem also that vendors will want different means of formatting. Perhaps through standards body.
  - SIROLI could work, could be extended to support medical device use cases. Desire to look at where there are standardized ways to share SBOMs.
  - Standards groups has looked at how standards meet needs.
  - In thinking about federated approach, 1) prevents needing a book for each suppliers SBOM, 2) small device manufacturer doesn't need to build up SBOM by hand

    - Different markets want different presentation layer, but can be done. Doesn't help with airgaps, etc. Will transmit in many ways. Current attestation of vulnerability is ephemeral, SBOM is not. Needs different transmission means.
    - This issue can go into standards and formats guide.
    - Support centralized area at least for medical devices.
    - This issue delegated to standards and formats group
- **Terminology**
    - Possibly for framing group
    - Actually, part of their ongoing work
- **Tooling**
    - Producing BOM data
    - Difference between manufacture and supplier (in medical device context), two steps- software manufacture built, or platform software runs on
    - Internally need to create identity mapping, in production of SBOM need to pick and choose.
    - Consuming BOM data
    - Would be good to be able to easily identify "bottom" component. Easy tools.
    - Can't thing name itself? Everyone else uses that name rather than central management.
    - Supplier should work to make unique.
    - By root node do you mean leaves? Open SSL are root nodes- leaf is HDO with no production.
    - Atomic component has no further components- should name itself.
    - Gets into use cases, if you take components and compile yourself- name at space of uncompiled code, vulnerability at space of compiling
    - Deliverable should be "what does it take to be a well behaved supplier"- for framing group
    - Can look into badging/scoring

<u>Outcomes and Deliverables</u>

- POC will be delivering info that will inform the rest of the groups. POC does NOT want to define minimum viable product.
- Be very clear that the proof of concept is to look for surprises but not to be the minimum viable product. FDA may use what we produce. Want to see what HDOs' internal TTXs encounter (will occur by end of May). Will collect and report out on MDM experience. Findings will be shared with other working groups.
- Tentative timeline:
    - June: draft deliverables from Framing, Use Cases, and Standards groups, and will have input from POC group. These would be drafts ready to be shared more widely. (Drafts should include indication of what is in Phase 1, and if other issues are in Phase 2, indicate that and a timeline for those items.)
    - Phase 1 – MVI concept.
    - Phase 2 – Outside scope of MVI, reflecting additional work that we'd like to do.

    - Recommendations on how we may share
    - Sort out redundancies prior to completion (e.g. multiple use case lists in WGs), need document structure and swim lanes. (Terms to avoid, choose language carefully.)
    - Audience for drafts:
        - Procurement team, operators, producer, lawyers, govt affairs, doers and deciders (executives, regulators). Text should be friendly for tech journalists. Packaging will be key. Production value assistance is welcome!
        - Standards: How-to guide. (Can include with with framing group's guide)
        - Use Case: Capture obstacles – package for ease of understanding
    - Awareness and adoption
        - How? Need coordination and strategy
            - Define need for SBOMs (WG?)
            - World tour
- Groups that may be interested
    - Defense in US
    - UK
    - ISACs, auto and finance
    - Electric utilities
    - Public transportation (e.g., DC area trains)
    - Telecoms
    - Security industry
    - OT/ICS
    - Smart cities (offers more centralized points for outreach)
- Power comes from industry getting in front of the problem
- Cloud – pure SAAS to API infrastructure
    - NOT Phase 1
    - SBOM benefits are different for suppliers and customers. Supplier has same benefits if it's cloud or not.

Post links to additional resources – groups have done a lot of good work. How can we help new people get up to speed, capture lessons for the future?

- Next meetings:
    - Week of June 24 – webex? – share draft documents 2 weeks before meeting, ideally comments submitted one week prior to meeting, then remaining comments discussed at meeting.
    - Week of September 2 or September 9 – in person

MVI, what, why, how – core of Phase 1