NTIA Multi-stakeholder meeting on SBOMs
June 27, 2019, 1-4pm

NOTE: These notes were recorded live during the "virtual" multistakeholder meeting, but are not meant to be a complete record, and may have the occasional error. Documents and presentations are available here: https://www.ntia.doc.gov/SoftwareTransparency

Please contact NTIA if you have any questions.

- **Welcome**
    - While we wait for everyone to join: should we spell it SBOM or SBoM?
    - 11 months since first meeting
    - NTIA said we need to make some progress, but no one org or government should decide. Hence this multi-stakeholder process.
    - Today: we'll hear the progress of each of the four groups. Hear their progress, get feedback.
    - Review of the agenda: the "what," the "why," and the "how" of SBOM, then tackling next steps.
    - Healthcare proof of concept completed their exercise and have some preliminary results.

- **1st presentation: Framing Group**
    - Wrestled with the structure, but now happy with results.
    - "Minimum viable" etc. terms – all that is out. Now use "baseline." See Section 2 of the group's document.
        - 5 identity elements in there.
    - What has not changed is, once we have that core identity, other info is necessary for other applications. E.g., licensing info is not currently required but obviously needed for IP use cases.
    - The existing structure can be represented in existing formats.
    - Relationships section: Focus on a "nested" approach to capture the recursive structure. SBOM must have at least one thing in it: a component. That's what the SBOM is about. Any additional components in the SBOM are subcomponents.
    - This may bear some further discussions with the other groups.
    - When do I create an SBOM, when do I change it, etc.? We have a section on that.
    - We see feedback on this definition of SBOM, including the "relationship" part of it.
    - (Made some last-minute changes right before the meeting.)
    - Still struggling to determine whether each field is required or not.

- Framing Group needs to make sure it’s in alignment with the other working groups, is producing stuff that all other groups agree with.
- Please comment in the Google Doc! (http://tinyurl.com/sbom-framing-draft-june) Looking for feedback. There is also a mailing list and Tuesday meetings.
- Apologies from the group chairs to the rest of the group – lots of rework since last meeting on Friday.
- Turning to the document itself…
- Nested SBOMs are like Russian nesting dolls.
- The 5 Baseline Elements: component name, supplier name, version string, author, hash.
- Author is NOT the supplier name. We foresee someone creating an SBOM for a 3rd party component that does not come with one.
- Blue row vs white rows: identify the overall product that the SBOM refers to = blue. Subcomponents are in white.
- Baseline elements: please provide all of these, or as many as you can. (e.g. there may be a few cases where you can’t provide a hash).
- Goal: map components to other sources of data. Name and hash will be useful for different use cases.
- You can have an SBOM with one line – one with no subcomponents.
- Important to establish that something is a component without dependencies versus a component with an unknown number of dependencies. But this is not addressed in the current document and approach.
- QUESTION: One SBOM objective = identify vulnerabilities. How do you address which are going to be mitigated when supplier supplies patches? Sometimes we don’t want to go through lengthy QA process.
  - Vulnerability management is one of the primary use cases for SBOMs.
  - Two answers. One is more complicated but better; the quicker answer probably doesn’t scale well.
  - Discussed in the group: vulnerability mapping is potentially not answered in the current doc. Requires non-trivial efforts – must list vulnerabilities somewhere, CVE is good place to start; there is work that is needed to map CVEs to products; we think the data to be mapped needs to be an external effort. Current problem: current lack of data. Suppliers are not creating these entries right now.
  - Known issue: CVE + subcomponent may or may not expose that vulnerability to parent components. We’ve left that unaddressed for now.
  - Benefits Group will talk more about what extra steps are needed.
  - Re: compiler flag set or not set, next group will talk about this.
- QUESTION: This group argued a lot over tree/leaf/branch etc. In the current doc, there are two SBOMs; the first one references the second one. But see the graph on page 20. What if it has 1 component in it, Apache -- could all 3 lines be in one SBOM, or would each line in the SBOM call out a separate SBOM?

        - I believe this is still allowed. One-hop vs multi-hop, going up the supply chain.
        - The recursive nature that they came up with does not prevent one hop setup.
        - If unable to obtain an SBOM from e.g. Apache, you are allowed to create one.

- **2nd presentation: Use Cases and State of Practice**
    - One goal of our doc, which isn't complete but is pretty far along: separation between producers, choosers, and operators of software.
    - This is the lifecycle of software. Illustrates SBOM benefits.
    - Producer benefits: less unplanned maintenance work, reduce code bloat, etc.
        - Example: several examples of SSL within one executable.
        - Monitoring/reviewing for vulnerability is a primary use case.
        - (Continuing to list benefits in the slide)
    - Choosing benefits for those selecting software (commercial or OSS) for their org.
        - The core of everything: identify vulnerable components!
        - (Continuing to list benefits in the slide)
        - Audit and verify supplier claims = super important. Need tools and techniques to do that analysis.
    - Operating software benefits. = IT, manufacturing line, network guy at datacenter, etc. The first responders when incidents occur.
        - E.g. TLS has new 0-day. Identify that thanks to the SBOM.
        - Drive independent mitigations – maybe more important than 1st benefit.
        - (Continuing to list benefits in the slide)
    - Describe process for how they looked at what people were doing out in the world.
        - Most actors are some combo of producing and choosing.
        - Most people are primarily buyer of software.
        - The group conducted formal and informal interviews.
        - OS distributors are more mature distributors. Healthcare end-users, kept largely in the dark by their manufacturers, are less mature.
        - Lack of participation from suppliers, e.g. medical suppliers – no awareness from them that SBOMs would be useful. Lots of people are waiting for their vendors' thinking to catch up.
        - Unclear division of responsibilities around SBOM: who is responsible for producing them? QA?
        - More mature outfits:
            - "End of life" benefits were strong.
            - SBOM forced them to do things they should have been doing anyway.
    - System-wide benefits for the full chain. Analogy: patient health vs public health.
        - Sometimes vendors go out of business. SBOM created at packaging time may be your best notification for machine-readable impact analysis.

    - One of their strongest voices: DoD. But some things missing from the minimum viable can enable provenance, pedigree, integrity, etc. that are highly valued.
- Suggestion: call them “potentially exploitable vulnerabilities,” since a vulnerability in a subcomponent may or may not expose vulnerability to its parent.
- QUESTION: Website has documents. Will slides be on the website too?
    - Yes, they will be posted by early tomorrow.
- QUESTION: DoD is on the line. SBOM brought up in a lot of forums; willing to facilitate more stakeholder discussion.
- QUESTION: Have you interviewed software developers and testers, or just security folks?
    - Many docs linked on our project work site: architects, operators, security, vigilance – captures title, organization, upstream and downstream.
    - Will work with Allan to make sure this info is better discoverable on the website.
- QUESTION (Allan): Is there further work that you need in order to do your work?
- QUSETION: 3 components to the value obtainable from SBOMs. 1: produce inventory informing you of software composition. Other components: 2: monitoring software that is the subject of the SBOM. There might be a match between what you learn in the SBOM vs known/potential vulnerabilities. 3: Fixing it. If you know the inventory and you are aware of the vulnerability, you still need to correct it. From DoD’s standpoint, it makes the most sense for DoD to call on SBOMs as part of a bundled activity or responsibility, so software sources are required to build an SBOM and monitor the software for vulnerabilities as they may emerge and take responsibility to correct it. Concerned about risk: if you gave DoD 100 SBOMs on 100 software packages, that’s great but it doesn’t give DoD any ability to monitor, and even if it did, how do we fix those vulnerabilities? Could that responsibility to fix it be contracted out?
    - Darwinism may apply here.
    - What do you do if you can’t fix it? – turn everything off, worst case.
    - How do you hold supplier accountable? Don’t know much about DoD, but in auto industry, you buy whatever it is from a supplier and hold them accountable for anything that happens inside that component (e.g. exhaust, body panel). They have to then tell you what to do when something goes wrong.

  Q: Is it worthwhile to pilot initiatives where DoD calls on current/prospective suppliers to assume this responsibility for all 3 components? Monitoring, fixing?

    - Allan: idea worth exploring. We’re starting with the transparency layer in these groups. We can help facilitate, but for now in our proof of concept, focus on what we need to produce usable data.

- QUESTION: We don't know who our ultimate customer is, so we can't just push patches and integrate with other software because our dependencies are not 100% known. Not represented in the 3-column model.
    - May be accounted for in the more verbose model.

- **3rd presentation: Standards and Formats**
    - Topic: How do we SBOM?
    - Each ecosystem has its own lifecycles etc. Blurred line between open source and proprietary source. Want to serve both.
    - Important goal: ensure that we survey which formats are available. Must be machine readable so you can act on it. If you have beautiful data in PDF form, that doesn't really help you.
    - Success is: machine-readable format that links software publisher and components, is signed by the publisher, automatable, and verifiable.
    - One format doesn't necessarily apply to all use cases.
    - Want SBOMs to be natively generated by the software packaging process.
    - SPDX (Linux Foundation) and SWID (commercial): examined and compared them.
    - We do not want to proclaim a "winner" format. They come from diff use cases, and each has a sweet spot.
    - SPDX is normally created early in the lifecycle of software.
    - SWID is meant to be created upon installation: where and how software is installed. Originally intended for entitlement management (e.g. # of installs allowed).
    - There are good reasons to use both.
    - Software is global. Similar efforts afoot elsewhere too. Want to know how our efforts resonate or don't resonate around the world.
    - See the infographic in the slides re: software lifecycle and SBOM assembly line
    - Went over the baseline component information table in the slides
    - We have some toy examples – not quite baked yet. They give a high-level understanding of how SBOMs will be generated.
    - Work to come: compare with other groups' work, incorporate feedback, and finish the toy examples and move onto a how-to guide.
    - Next steps: describing the tooling landscape, creating a quick start guide,
    - Allan: could use more help, great place for others to get involved.
    - QUESTION: Security, licensing, assurance (applications): Should they be working at both 50,000ft, but also real-life examples? We've been working with those real-life examples but they're not in our documents. Should we be documenting these use cases?
        - Allan: great point; in automation, sectors, stages etc. require different tooling.
    - QUESTION: from enterprise software side: we produce SBOMs in Cyclone.

        - Is Cyclone based on SPDX? No, standalone but uses purl.
    - QUESTION: Dynamically loaded libraries, configurations, complex use cases. Does the roadmap consider how software is evolving? Is there a plan for addressing these new functionalities?
        - Current plan probably does capture these more modern, complex software practices. Different methods of producing, choosing, and maintaining are all rapidly evolving, but the elements themselves (producing/choosing/maintaining) stay the same.
    - QUESTION: In DevOps and security world, there's a big interest in using CycloneDX today. Wasn't on the radar –
        - It was on the radar, and was discussed in the group, along w efforts like software heritage.
        - Black Duck works with SPDX natively.
        - Even in DoD there's new standards and transfer formats being proposed for critical infrastructure.
    - …Q: It is fast-moving, true. I invite people to join in and contribute to Cyclone, it's open-source.
        - Great.

- **4th preso: Healthcare Proof-of-Concept.**
    - Objective: collaborate between healthcare delivery orgs (HDOs) and medical device manufacturers (MDMs) to explore SBOM production and consumption with a provisional SBOM format; demonstrate successful use of SBOMs. Bring experience back to the other work groups.
    - Stuff we did and did not deal with
        - No hardware. Used both SWID and SPDX.
        - (Continues going over slide)
        - Dropped out of scope: Context was too complex. API for data access also dropped out of scope.
    - Timeline.
        - Wanted to collect feedback in standardized way, so we could present it in findings that were across the board.
        - Dec.: Define uses cases and formats.
        - April: began execution. MDMs produce SBOMs.
        - May: HDOs consume SBOMs and execute use cases. Experience recorded via forms.
        - June: Write report. Will be delivered to the group.
    - Apollo: wanted to express how we did. Number scale used: 1, 8, 11, 13. Apollo 1 = catastrophe (astronauts died before liftoff); Apollo 8 (astronauts circled around moon but didn't land); Apollo 11 (moon landing); Apollo 13 (explosion, but astronauts brought back safely).
        - 2-dimensional assessment: success-failure, preliminary-operational.

- Use cases executed: 2 main use cases: procurement and asset management.
    - Clarity = executed.
    - Lifecycle management = executed.
    - Vulnerability mitigation = executed.
- Asset management: focused on risk assessment and mitigation.
- Risk management: identified unsupported software; HDOs thus had mitigations in place. Monitored for new vulnerabilities. Assessed new products.
- Vulnerability management: monitored inventory against new vulnerabilities, assessed new products being added before added to network, assessed risk.
- Raw experience, feedback (have not developed these into findings report yet; just semi-random sampling of feedback):
    - MDM: Slight preference for SWID: less error prone.
    - HDO: SPDX more human readable, but SWID preferred for automatic ingestion.
    - Hard problem of naming that we're trying to solve.
    - SBOMs allowed identification of vulnerabilities; correlation was difficult.
    - Lack of trust in the completeness of info provided by manufacturers to HDOs.
    - Asset mgmt.: sometimes SBOMs provided info used later to protect the asset, and sometimes usable for end-of-life planning.
    - Risk mgmt.: mitigate new vulnerabilities was successful. The naming convention problem interfered. Requires manual work for correlation.
- Experience comment:
    - Digestion not possible with current CMDB; more work required for customization.
    - We relied on Splunk for the system of record. Very easy ingestion, parsing and mapping of SWID, and fairly simple correlation to info in the NVD database. CMDB, specifically ServiceNow, is something we'd like to explore.
- QUESTION: Did anybody talk to their suppliers to get BOM data?
    - That was within scope, a best-effort point for device manufacturers. Whether they would use the info on hand or find other sources (e.g. next level down supplier) for the info.
- …Q: The supply chain has many layers. If we ask each other serially, if we're late in the process, it could take a long time. Perhaps ask now, because it may take a while to get the political/legal will to get this done. Ask suppliers to be prepared and be ready.
    - Sure. Let's do everything we can to accelerate things.
- There wasn't any automated tooling. What would that look like moving forward? 3 big manufacturers (hardware, others) had a similar "gosh this is really hard" reaction.
    - Not an easy problem to solve.

    - Don’t have any great insights into tooling from the proof of concept. Everyone took a different path. Some SBOMs were constructed manually, others automated or partially automated by individual implementations of special tools based on current repositories of SBOMs.
    - Problem is producing the SBOM in the form that’s needed. Whether it’s in the build process or some other level, like vulnerability management.
- QUESTION: Doc like device master record, brought to market, can this be leverage for easier creation of SBOMs for MDMs?
    - One reason we went with manual approach is because we had the design history file, specs, other sources providing info -- but not in an electronic format easily usable by tools. PDF etc.
    - Repos already exist and sometimes need to be developed. There’s no tool that exists that help with extraction.
- QUESTION: Want to think about a 2.0 version. What about developing tooling to support creation/publication of SBOMs, like a web service or something used to update a device? It would be disappointing to do all this and then not push it forward. Want more than a POC next.
    - Agreed.
    - Want to close some of the gaps in 1.0.
    - NYP is creating new database for their team. That company has agreed to work on a POC 2.0. They have an API that could be used to consume the info.
    - ServiceNow, other similar services. Excited about supporting this, helping to ingest info.
    - There are major equipment manufacturers outside of this POC that have already built some of the infrastructure needed. They were able to produce an SBOM-like thing, even if it wasn’t in the right format.
- QUESTION: One-hop vs multi-hop. In actual experience, did MDMs produce a full tree down to the subcomponents, or just the top component and then the HDOs had to determine whether Apache contains this or that? Or maybe both? (rephrased) Is it the responsibility of an MDM to provide the full tree, or is the SBOM creator only responsible for one layer? (rephrased) Does the final goods assembler provide one layer or the full tree?
    - Didn’t have visibility into everybody’s SBOMs.
    - Still a point of discussion. Would probably make sense for MDMs to provide all info available at that times, including as many hops as possible. This would lead to better consistency.
    - HDOs have to give MDMs time to think about and operationalize this. All hops won’t be included on Day 1.
    - More info is good, but perhaps that will be solved by the market. If one product we can see everything and the other we can’t, that will be a force in the marketplace.

- Potential next steps
    - Allan: Any other overall reactions? What would you like to see as next steps?
        - RESPONSE: Need to gather and report on 1.0 feedback first. That's a pile of work. Next pile of work: 2.0.
    - Insurance use-case? Keeping honest people honest. Use-case group is planning to work on this.
    - Standards group: Talked about tooling. Let's make this machine-readable and -writeable. Github approach maybe? Perhaps discuss how that would look like.
    - Broader awareness and adoption approach: natural fit for the framing group.
    - QUESTION: What can we at Github do to best assist? Planning to sync with the Formats Group after this call.
        - RESPONSE: How-to is the obvious way to engage: examples of both SBOMs, tools used to produce them, available in a form that devs and managers understand.
    - Plan to refine drafts
        - We want to share these docs beyond the NTIA community. Let's work these docs from their draft form to a more polished form.
        - How do you see these four draft documents fitting together? Should they be free-standing? Tightly bound? Loosely bound?
        - RESPONSE: Agnostic. Defer to Allan's judgment.
        - RESPONSE: I'm going to be promoting the standards doc in particular. Having that doc as a bit of background would be helpful. These docs seem to have different audiences. So perhaps standing alone would be best.
        - RESPONSE: Maybe loosely joined, if they're going to be available online somewhere. Chapter or document links, perhaps, so they exist within a context, including links to tools, reference implementations, etc. Choose-your-own-adventure.
        - RESPONSE: Whatever we do, prefer to use consistent terms and concepts/constructs across the documents.
            - Really important point. A fresh set of eyes would really help! Newcomers very welcome to work on this.
        - RESPONSE: Agree some efforts stand on their own, but they shouldn't feel like they were done in isolation. Also, sometimes not all docs will agree on something – explain why. Biggest concern: making sure that all docs are in one location so that if you find one doc you should find the others together. (Also proposed some specific refactoring of info from one document to another.)
        - Suggestion: think about what this presentation should look like. Perhaps should be part of the Framing Group's purview, since they've been thinking about global scope for a while.
        - RESPONSE: Info architecture. 1. Newegg comes to mind as a contributor. 2. Nouns and verbs vary wildly by sector; perhaps have a table, e.g. government procurement = bank acquisition, etc.

      - Note: for things going on the NTIA website: the content management system is simplistic.
    - RESPONSE: 1. Finish the docs we have, don’t get hung up on perfection. On the software side of SBOM, get more agile. Is it good enough. 2. Since Github has volunteered to help, we here actually use Github for our docs. Designed for collaboration; let’s use it in these working groups to collaborate on our docs.
      - Most working groups are currently using Google Docs currently.
    - …R: Github may do collab/approval better than Google Docs.
      - RESPONSE: Standards Group is interested in using Github. BUT: Very easy to underestimate the level of unfamiliarity and discomfort that people have with systems like Github. It’s surmountable – and an amazing achievement to have non-coders working on Github – but unlikely to happen.
      - Hard enough time to get orgs to use Google Docs since limited by their company’s security practices. There are lots of considerations in play.
    - How do we tell the story of early adoption with our good-but-not-perfect docs? CMDB, vulnerability scanners, large data tools – what else do we need beyond these four docs available this summer?
      - QUESTION: Nothing is more convincing than seeing people doing something. Healthcare POC is fantastic.
      - QUESTION: Excited to explore the assurance stuff. Ready to dive in and help with specifics.
    - What are the 2.0 aspects that we want to pivot to? Identified based on last few meetings 3 priorities: awareness/adoption, tooling, extending beyond baseline SBOM model. Anything else we should tackle? (Healthcare group already volunteered 2.0 POC.)
      - (No comments.)
    - Let’s make sure we are clear about our audience, avoid conflicts.
    - Anyone can join a working group at any time!
    - Any other topics people would like to discuss?
      - (No comments.)

- **Closing**
  - We’ll be posting the slides and these notes online.
  - Allan will be in Japan; Japanese government is really interested in this. This project is going to be impactful very soon.
  - NTIA is here to help your org get up to speed and involved in the process.
  - Special thanks to Megan at NTIA.