>>: John, Josh, you're up.

>>: So, while we're technologying, I did tweet under the SBoM hash tag a couple graphics, um, that we will walk through. This is the use cases and state of practice working group. In a lot of projects like this, I think the scope and goals are, oh, yes?

>>: Could I suggest introducing yourselves?

>>: Yes, we will.

>>: Why don't you do that?

>>: Okay. I'm Joshua Corman, um, chief security officer at PTC.

>>: And I'm John Banghart, senior direct for Cyber Services Eventable.

>>: Um, I've been on the SBoM topic for I think six years and counting, but, um, what we wanted to do was instead of saying, maybe, someday, we could do this, we wanted to capture six plus years of state of practice, and what you'll find, and we tried to, we've had to iterate several times, is it's fairly confusing for folks, what we mean by SBoM, um, which is one of the reasons I tweeted out, um, at the SBoM hash tag three

columns. The green one is really what we tried to frame at the first meeting here on July 19th, so in lieu of me pulling up that graphic for a moment, essentially, um, think of an SBoM at its bare bones, um, the least controversial version is it's an ingredients list without judgment. It's what's in the software, um, the part number, the version names, um, we don't have that debate on XPDX per se, we will, but in its most atomic form, it's a list of ingredients. Where people sometimes get in trouble is column two, where they try to enrich with potentially exploitable vulnerabilities in the national vulnerability database, and that is one of the use cases we're going to show graphically. Um, in a fire fight, when Hollywood Presbyterian Hospital got shut down for a week, they were warned in advance of law in JBoS, they just didn't know what a JBoS was or which thousands of pieces of equipment might contain a JBoS of the affected version. So, answering am I affected and where am I affected can make great use of the column of the list of potentially exploitable vulnerabilities, but it can also create a lot of noise, where people are saying get rid of all these bad vulnerabilities. They're only potentially exploitable, and then column three, we talked about,

um, often, which, um--

>>: Do you want that up?

>>: Yes, please. Column three, we talked about, maybe, is a bad metaphor, call it a, um, nutrition label, which can include the ingredients, but may go beyond that to say, hey, these are, this is the attack surface, so, yeah, there's a lot of parts in it, but only these 12 are exposed. These are the, um, mitigations in place, this is a list of, um, triaged, um, vulnerabilities. So, out of maybe a hundred potentially exploitable vulnerabilities, maybe ten of them are actually exploitable, and someone can make an encystation that can persist across state and across time. We decided on the July 19th meeting, let's focus on the first column, the bare bones ingredients from which the other two could be derived or enriched by someone else at a later date. I don't think we're going to necessarily avoid those in perpetuity, but at least for today's deliverable, we focused on, um, that ingredients list. How's the technology going?

>>: Terribly.

>>: Terribly. Okay.

(Laughing.)

>>: So, um, you can refer to your Twitter, if

you're on the Twitters. I'm @joshcorman. Oh, pretty. Okay. All right, so, we're going to try to tell a little story. Um, one thing to point out is, um, that has not been raised yet, but I'm going to make the point, while commerce depart and NTIA is a best practice working group, to work with industry and all the different ones for capturing and promoting best practices and better practices and whatnot, there is an interesting point of history we're in where the Food and Drug Administration, which is a regular with regulatory authority has declared, it's partly my fault, but has declared, um, that they, their pre-market, um, update will require a CBoM, or a cyber bill of materials, which is hardware and software. I suspect during this open draft comment period, people will push back really, really hard on the hardware part, maybe successfully, we'll see, but the idea, um, was, essentially, surfaced during a correctional task force on healthcare cyber security that came out of the CISA, the computer and information security act of 2015. We did a year and a half long task force, and in that, we realized one of the ways to insulate Hollywood Presbyterian from that JBoS ransomware, that even though they were warned, they were still hit, because

they couldn't answer am I affected and where am I affected, therefore couldn't hatch or remediate, um, one of the things that came out of that recommendation list about 116, if I recall was to require a software bill of materials for all, um, medical technologies, and the committee of jurisdiction, um, in oversight in the House said, hey, we like that, in fact, it was right around this time, it was right before Thanksgiving last year, they said, hey, we like that, HHS, go do that.

So, HHS is doing that, and their public workshop is late January, so they will be using regulatory authority for medical technologies irrespective of what we do here, but their hope and sincere desire is that this collaborative multi-stakeholder process comes up with something really good in evergreen that can apply beyond medical technologies. Um, so, FDA will be using its regulatory on medical devices, but maybe not electronic medical record systems, it's a different jurisdiction. There's also lots of other technology stacks in hospitals. So, we use this graphic to try to make this a little clearer. We're good?

>>: Yep.

>>: Okay, so, let's start our little story here.

Um, this is a bedside infusion pump, it's one of the most common technologies in a hospital. This is a metaphor for what we're doing, not the entire story. Um, this particular make and model of medical device is one, it is but one infusion pump, there are many, but this one is my infusion pump, and, um, hospitals will choose this infusion pump amongst others based on price, functionality, quality, maintenance, all sorts of other things, but increasingly, folks like the Mayo Clinic and others are selecting these based on the relative hygiene of the cyber security and the ability to produce an SBoM, or software bill of materials, and how well they maintain the vulnerabilities within it. Um, even Underwriters Laboratories has this cyber assurance program, which includes, among other things, a requirement to capture a software bill of materials. Um, but that infusion pump is not made exclusively by Acme, we're going to use Acme here, no real company names, hopefully, um, it comes from a whole bunch of open source stuff and third-party license stuff. In fact, um, certain statistics will vary, but some, they argue and hover around 90 percent of the software in these, um, final goods assembled comes from third-party open source sources, often these mega projects, like

what took out Equifax, which took out a financial services company in July 2013, so these big, fat pipes are essentially compound projects, like struts or spring or juice or J query, and those little, um, integrals are essentially the dependent projects that they build upon too, because it's turtles on turtles, and then lastly, um, I think you can see faintly a bunch of tiny, little aerosol atomic products. This could be a single logging framework, it could be a single string parse, like X stream, and those can be pretty nasty when put together.

Now, if we put labels on these, what we've tried to abstract in our use case capture is a daisy chain that starts with parts, gets to compound parts, those can be one through N in between themselves, gets to a final goods assembled, in this case, for the FDA, they're going to regulate that medical device, and then it gets to a consumer or operator, we've oscillated between the terms consumer or operator, but someone buys that thing, deploys that thing, and maintains it safely through retirement. Um, so those are the four columns in which we've started to elicit and capture personas, intents, use cases, supporting business process and artifacts, and is my pace okay? Okay. Um,

well, let's annotate this. Let's tell a little story with this. One more click, please. Okay, so, it might be hard to see, but this is red, and this infusion pump's red, because there's a CVE, or common vulnerability exposure, in that struts II fat, white pipe, right? That fat compound project has a vulnerability in it, and therefore the three hospitals who depend upon this device could be affected. Right now, this is a very haphazard reactive thing, there's an attack in the wild, people say am I affected, I don't know, I can't tell, and we heard passionately from a few medical professionals, including Jennings, who's running a different working group, that they had to make dozens of phone calls to ask their vendors are we affected or not. Phone calls, and they couldn't get answers, so phone calls don't work well at the speed of attack. We can imagine a world, though, where if there's someone like the Mayo Clinic or Cleveland Clinic or Cedars-Sinai, you start demanding these in your purchasing, having a bill of materials allows you to at least short-list, um, hey, which of these things has this version of this particular vulnerability library. So, that's an attack scenario currently done imperfectly, but to increasing value in an isolated use

case. Next.

Now, again, we're not going to use any vendor names, but my a-ha moment was July 13th, 2013, because patchy stress II hit all our banks, so all the banks that wrote their own software systematically eradicated that vulnerable version within a couple days. It was really impressive. The problem was a few days later, their third-party enterprise software infected them again, because it, too, was using this. So, we're not going to use any company names, but HAL cloud cube here is also in the clinical environment, and HAL cloud cube also uses a patchy struts II. So, even if you had patched this through your Acme infusion pump, we now have the problem of, um, HAL got hurt too, right? And these are still at the compound project level, so if you click, I think you should see red on the hospitals again, but the more insidious one, and one of the reasons we really have to do a great job collectively here, so when we're talking about mission and what's the big picture and who the deliverables and stakeholders, um, even if we can get really focused on, maybe, the big projects that attack the most, like a patchy struts II or open SSL or, um, open SSH, things like that, it's the really insidious, little things,

so picture way off in the weeds there a single atomic library, let's say that HAL cloud cube decides to use spring instead of struts, so maybe they're not affected by that anymore, but click one more time, a single vulnerability in that little downstream pigtail dependence can make its way both into struts and into spring, and something like X stream is a particularly insidious example, because it's a vulnerability that never got a CVSS score, so no one prioritized it, people didn't know they had it as a transitive dependency, and if you've read some of the good work by Dennis Cruise, it's trivial to X plight.

So, it's one of those things where the benefit of a more comprehensive and holistic software bill of materials could give, picture in the future where this is less single hand-offs between these stakeholders, what if we could light up like a Christmas tree that that one flaw affects these major libraries and these seven brands of infusion pumps and desktop software, which affect these hospitals or don't? Instead of us asking am I affected and where am I affected after a headline of an attack in the wild, we may simply get an alert saying, hey, this vulnerability needs remediation, please plan accordingly, well before an

attack manifests. So, there's an opportunity to get from current state to desired state, and we're not kidding when we say that people in this room could save lives, because it's not just the medical technologies, like these bedside infusion pumps, which I think the FDA is doing a fantastic job cranking up the heat on, it's also the desktop software, the HVAC stuff. If anybody saw our clinical hacking simulations in Phoenix, Arizona that we did, um, we actually came in through default passwords and the building automation system. Elevators in HVAC and EMRs. So, one of the reasons I don't want to just, my heart's desire, I don't want to just make FDA the experiment here, is if we actually want to protect hospitals, we can't just look at the medical device, we have to look at the software that exposes those. In fact, most of the infections were not through medical devices, but through, um, work stations.

So, that flow of parts, compound, parts, final goods assembled, and operators, there's, you know, dot, dot, dots before and after those, because your customers can have customers as well, you know, if you're a service provider, if you're a factory, so we tended to go fairly deeply here on a medical, but not

only medical, and what we've done is we've sliced up an actual Google Doc spreadsheet, which I've sent, tweeted a link to just before I walked up here, where you could peruse, and we've split it up into rows for domains. So, there will be, you know, domain agnostic software, like enterprise commercial, off the shelf software, there will be, that could be things that you could name right now, what you all use, that could expose any of us or all of us. It could be individual open source projects in a vacuum that are not domain or industry-specific, but then we also have some columns that are more specific, like medical, like financial services. There's been significant use cases in innovation and financial services. In fact, one of the things I wanted to comment on and the first thing is I don't think this working group is all about security and risk management. The overwhelming driving force in financial services was productivity enhancement and developer productivity and reclaiming, um, hundreds or thousands of hours of developer productivity for not doing unplanned, unscheduled rework, so they did these kind of things to be on time, on budget, and we heard some of those voices in the room last time, and hopefully, they're on the phone. So,

one of the use cases we can share with you is actually a financial services procurement one, but there are many reasons and benefits to look at the quality of the parts there.

So, as I pivot quickly, now that the graphics are working, you might notice an S1, S2, S3 in every one of those major columns. Um, for consistency with Demming, a lot of this inspiration came from Edwards Demming, supply chain management in the 40s, he had three principles that drove his activities to make a textile company the most profitable automotive manufacturer in the world for 30 years. Number one, I'll repeat these, but number one, use fewer and better supplier or parts. Number two, use the highest quality parts from those high-quality suppliers. In fact, we have some auto-makers in the room. Number three, track which parts went where throughout the product's life cycle, such that when things go wrong, you do a prompt and agile recall. So, again, fewer and better parts suppliers, highest quality parts from those suppliers, and track which parts go where, which is essentially where we get the notion of a BoM, a bill of materials. SBoM is an unfortunate acronym that is an extension of that. So, what we did is we took those and put them

in our spreadsheet with greater text as, um, there's a set of activities or personas that do supplier selection, so if we're going to buy infusion pumps in the operator environment, let's go to the last column, S1, if I'm procurement for a hospital, I'm going to come up with my purchasing criteria and for which vendors I think are worth my business for the next five years.

So, supplier is really at the vendor level, do they have a disclosure program, as outlined by NTIA a couple years ago, do they have, are they patchable, do they have a UL certification, things like that, but one of them is increasingly do you have a software bill of materials, and how good is the hygiene compared to your peers. Um, so within that operator thing, we have begun to, and we would love actionable help from this room to do, is let's go target and harvest, um, use cases and artifacts in each of those squares on this grid. We have a couple of them already. Um, so, that one's really the medical type one. Um, I want to wrap-up quickly here. To the S2 idea, this might be more, um, okay, we've already standardized on this manufacturer, um, but before we do a go live, you know, push or roll-out that's going to last three to four years as a gold image, let's make sure we're using the least

vulnerability version of that product from that vendor, let's make sure, um, we know all of the residual exposures or CVEs so we can put in compensating controls for those, etc., and the last one, we're calling supply vigilance. So, again, S1 is supplier selection, S2 is supply selection, often MIT or ops team, and number three is supplier vigilance. As a produce of software, for me, supplier vigilance is I have a vulnerability disclosure that I'm monitoring, I watch Google alerts, I watch NVD, we pay attention, any of our downstream dependent software packages to trigger do we need to do a patch, so depending on your leg of this relay race, your nouns and verbs change, but your general objectives don't, and what we saw was isolated value in a vertical or in an S column.

What we'd love to see is as our compadres in the standards group, um, create something more homogenized, more consistent, you can see how a consistent line of sight up and down this chain could enable vigilance and action, even when middle companies go out of business. One of the questions, I think from a trade association earlier, was what counts as, um, a major update or a minor update. Some of the very encouraging conversations we've had in the working

groups is this is as delivered, as built, at build time, so if this is constantly put out as a by-product of building software, then some of those debates, which are fair questions, may be obviated by some sort of harmonization and standardization.

>>: So, I want to, yeah, let me just add one or two things, then we can wrap this up. I think, you know, while my primary job is fighting Windows auto-play in the images, um, when I'm not doing that, um, the other thing that we are doing as part of our group is trying to build on top of this structure that we've put together and actually write up some narratives, some story-telling, right? Not everybody understands what SBoM is, not everybody can easily relate it to their world, but as we think into the future and think about how do we make a business case, how do we make policy cases and things around that, we're going to need something that's fleshed out, that turns it into a real world scenario, um, that people can read and understand and see themselves in. So, in addition to building out the spreadsheet, which Josh mentioned that he tweeted that out, I think we'll get that up on the NTIA website as well, um, we want people to come forward with their stories. You can anonymize them, if you

want, whatever the case may be, but we want to build out the pieces, but we also want to write up those stories, because I think that's going to be a powerful tool, again, particularly for those individuals or organizations who may be new to this, who may not completely understand what we're talking about. So, highly encourage you, um, to get involved in that part of it as well, and I think, can we get the spreadsheet up on the website at some point? I think we can do that. I think we'll get that up there. Um, so, how far did we go? Do we have time for a question now, or do you want us to wait?

>>: Definitely have some time for questions. You'll want to make sure that folks know how to contribute. Can you take a step back and talk about what, take a step forward.

(Laughing.)

>>: Um, for the February meeting, what will you have for us? And what do you need from people here to get that?

>>: So, I think Josh can start answering that by kind of running through a specific example that we've already started to build out, and then I'll build on top of that with some other things.

>>: Yeah, we realize that picture and framing may be pedantic or rudimentary for some of you, but we also found we were meeting every week with some really passionate folks, and we were ahead of ourselves in a couple cases. We hadn't actually contextualized the work. So, here's a concrete cell in this overall graphic, picture we're in financial services, S1, procurement, all right? So, we interviewed, I won't name the bank, unless he chimes in and tells me I can, but we said what's the persona we're speaking with, and the persona in this case was two teammates, one was called a sourcing team within the bank, and they almost are content-agnostic, they really don't care what they're sourcing, they're not security experts, but they work with the risk management team to come up with, um, buyer technical requirements and terms and conditions. Basically, sourcing, I'm going to read this verbatim, but sourcing does not stipulate technical requirements, but the buyer can set those, and we shimmed ourselves in between through something called a master software and licensing agreement template, which is a boilerplate, which is not, which can be red-lined, but has to be there and answered every time.

So, the SBoM shimmed in the criterion and said these are sometimes stricken, they're sometimes, um, modified, and they are always a negotiation opportunity. So, this isn't about so much security, it is, they want to generally reduce the risk to the bank, but their attitude was, um, asking for an SBoM creates a negotiation opportunity during procurement for any and all software we buy at the bank. Um, they did have some internal debates about, um, some IOT is so small that, maybe, it should fly under the radar, but maybe it has a big impact, etc., etc., but we got into about a 40-minute conversation and captured some artifacts, you referenced a financials services ISAC third-party software working group paper, you referenced a financial services sector coordinating council procurement guide for auditors and assessors, so the FFSCC, I think is how you pronounce that, FFSCC, and referenced a new standard, called oasis that he's considering for static analysis in a similar capacity, but I said how does this really work? Don't tell me the process flow, how does it really work? And essentially, they said, and some of them were in the room last time, they always ask for an SBoM now, and if a vendor can provide one, it's a good sign. If they

can't provide one, um, they know it's going to cost more to evaluate, operate, and own this through its life cycle. Um, and I said, so, you don't even care what's in the SBoM right now? He said, in the first phase, no, can they produce one. It's a litmus test, or an indicator of the maturity of the organization they're dealing with. Um, then I pushed a little harder, and I said what do you do with that fork in the road, and he said, um, we know that every time there's a new thing, like a patchy struts II, we're going to have to do a lot of evaluation, which is disruptive, expensive, and difficult, so we know that we're going to bake that into the total cost of operations, so we're going to ask for a massive discount on people who cannot provide an SBoM to equip our triage, and they get it, because they're a big bank, right?

So, they bake that into either free maintenance, or they bake it into reduced cost, because they know they're going to have to augment the cost of ownership, which I thought was logical. I said where are you taking this, and he said there's a new process we're adding on top of this, which is called prohibitive technology list, or PTL. So, they have a few projects that have bit them enough times that are so old and so

dangerous or so radioactive that they're now using this same procurement shim to make sure that anyone who's trying to sell a product that has this really, really old junk in it, um, can articulate compensating controls or add additional contractual language, or they just don't get purchased. So, they create additional friction and impedance to incentivize people to get on more modern and better maintained, um, software and projects. So, this is just an example, it's in the spreadsheet, it's not well-formatted, but what our team has been doing week after week is some of us have peeled off, there are more examples than the one I just read, but what we're doing is saying who works in automotive and can give us an operator S2 thing use case. I'm a developer of IOT middleware, so I'm going to make a use case of myself for some of these things, for what we do to select which open source projects are too risky to use, um, and what we're hoping for from our working groups is we come back each time, by February, this entire spreadsheet filled out, with one or more concrete use cases, with artifacts, when they're willing to do so, with company names, when they're willing to do so, but we know in some cases, they won't, and part of this is to debunk the fact, the

fictions that this can't be done at scale, because it is being done in a lot of places.

Part of this as well is to get the heat map of the divergent nouns and verbs used across these sectors and across these, um, titles, um, because if we heat map that, we believe we're going to have some very consistent use cases emerge, and then lastly, these are just a capture of isolated value, they're not a capture of system-wide value, and we believe the opportunity with our, um, companion team of the standards group could make these more consistent, such that they are machine-readable, machine-speed, and therefore, machine-level value, and the humans can focus on the things that matter.

>>: So, the thing I'll add to that, I think Josh got it exactly right, I mean, the concrete examples, the real examples, whether they're complete or partial, I think are the most important in terms of what we're trying to do, but I also think that doing some creative thinking is okay here too, right? Some theoretical examples, challenges that you may have in your organization, where you think an SBoM type approach might actually be helpful. You may not be doing it today, but you've got some thoughts on how you might

implement it or what that might look like. We're okay getting those ideas as well. Ultimately, as Josh said, the goal here is to try and demonstrate that not only is this working, it can continue to work, it can work in ways that maybe folks haven't thought about or are thinking about, but haven't done. We want to paint that entire picture, and so between now and February, that's our hope, right? Is to build that out, to have the concrete, but also some of the forward-looking as well, to try and bring all that together. So, I think we're done at this point. Are you good? So, we'll stop here, and then maybe a question or two.

>>: So, again, just to play back what you just said, to make sure I understood the answer to Allan's question, so the intent by February is to have something that looks like what's on the chart there, which is roughly 70-odd squares in it, and for each square, you're going to have something. What's the something that goes with each square. That's where I got a little lost. You said a lot of words, but what is the actual deliverable?

>>: So, we have, I'm not happy with how consistently we're capturing use cases, but we want to have an associated doc that captures the persona, the

intent, the business process, the artifacts, and once captured, we're going to do some post-processing on euthanatizing those. Very different words. I mean, the way that, um, we have an automotive manufacturer in the room, they are very, very good at tier one, tier two supply chain management and physical goods, so those nouns and verbs are more mature in that sector than they are in medical, but, you know, FDA has its own jargon too, so they're going to sound different, one of the things we're trying to do is debunk that by just kind of bringing together the common practices or motions, and then if you go to the third tab in the Google Doc I linked to, there's an unstructured capture in an Excel spreadsheet, which is not ideal. What I think we had hoped to do by this morning, but we did not to, because we were working on the graphic, um, was to have a pretty normal product management one-on-one use case capture template that you could just execute. I personally care a little less what method you use, because these conversations go organic really quickly, especially if you're talking to a busy hospital procurement person, um, but I care that we capture them.

>>: I just have one last comment to make, and then we'll move on. I just wanted to say that, I would

be remiss if I didn't point out that one of our group members, Audi Hatch over here, was responsible for putting that graphic together, she did an amazing job under short notice, so neither Josh or I can take any credit for that graphic whatsoever, so I just wanted to make sure we gave credit where credit was due on that.

>>: I'm actually going to keep you up there for a little bit, because what I want to do is, since we, a number of you have sort of emphasized the importance of use cases, um, one, do you have use cases in mind that you can, you don't have to volunteer, we'll guilt you into volunteering afterwards, but just what are some of the things that we can all share now, so that we can start thinking about this while we're looking at this? And it's also important if you have a use case that you think doesn't easily match to this chart, because then they'll have to figure out how to put it in. So, this is a chance for folks to say, you know, this is what I think this data could be used for.

>>: And other was for real estate, the actual spreadsheet has government things, so there are more in the actual spreadsheet.

>>: So, Bruce?

>>: Yeah, I had a comment that it would be good

if we had a higher level statement of what all of these things are for, other than, it just says software bill of materials, and it seems to me that this effort has two parts to it, which would help, um, answer the question that was asked by a gentleman earlier, and that is the function of this, we've seen a lot of people asking for software bill of materials, a lot of organizations all over the world, but we don't, we're not sure what that, what's in their head when they're asking for software bill of materials, and then if we understood that, we would then want to know why did they want it, okay, what is it, and why do you want it, or what are the requirements, and so it seems to me that the effort here is for two things. One is define what a software bill of materials is, and the other part is decide, define why we want one, and what, or another way to put it is, you know, what are the requirements for one, but because that hasn't been stated, it's difficult for people who are oriented towards, okay, write your requirements, and then we'll try to figure out a solution, or, you know, other people looking at it a different way, so I think it would be good to have an overall statement like that for this effort, so that new people joining would understand that.

>>: I think that is a fantastic idea. I'm going to throw the ball back over to Art and Michelle and see, is that sort of the work that you've been trying to get to?

>>: And we're over here conversing, because that's exactly the case. Yes. So, um, and Art is actually drafting some things around this right now, because one of the things that we took away from our, um, by the way, this is Michelle Jump, um, one of the things that we took away from our discussion was adding a problem statement to our guidance document, but also, um, defining a little bit more about this, and I think that some of the things that we're looking at here that'll align a bit more with the problem statement are also these foundational objectives, which are also going to speak back to that problem statement as well, I think. Would you agree with that, Art?

>>: Um, yeah, generally so. I think it was Steve Lipner on the phone, asked the problem statement, the why question, which I believe is about where you're going, Bruce, as well, and that's, I'll claim that our working group is the most responsible, perhaps, for trying to at least propose something there, so we're, we'll try to tackle it.

>>: I mean, it's undeniable, that an SBoM is a, it's a means to an unstated end. I think what I was hoping, at least for our corner, if we capture the keyword, in a use case for us, it's intent, when you start to capture intent, you see the business problems people are attempting to solve. I think the macro level ad hoc for me is we want to dramatically improve the quality and reliability of the software we build and consume and operate, and we believe this can help do that in ways other solutions couldn't, but it is a means to an end, and there may be an alternative or superior means to an end. Some people want to see software reliability, so don't tell me what's in it, just be responsible for what happens, if you give me bad stuff. Other people in this room would have a hard attack, if we introduced software reliability. So, SBoM, absolutely, is an implementation towards unstated things. I'm hoping we at least surface a lot of those use cases and requirements, um, and that the other working group here elevates it to the higher level indicators of success.

>>: Sure, but the question of, you know, using it for making buying decisions is different than using it for determining a vulnerability list, and I'm not

trying to say which is good or if we should do both, but I don't think a lot of people understand that those are open as, you know, requirements.

>>: Yeah, and that's why we wanted not one use case per vertical, but, in fact, today, I said this already, we keep saying security use cases, there are, Demming did not do this to make safer cars. The unsafe speed boat came out in the 70s, 30 years later. He did it to make profitable cars. So, there are many use cases here that are business enablement or agility or profitability, that we should not overlook, especially if we're trying to sell this for greater adoption.

>>: Well, I was just going to say too, I think just from a process perspective, I think one of the, your point is well-taken, I think in part, what has happened for all of our groups is we all left the start line at the same time, and I think as we've been kind of going through this parallel processing, we sort of arrived at a similar conclusion, which is we're trying to define that problem statement through all of the work that we're all doing and trying to arrive at it together, as opposed to having that problem statement right from the get-go. I think there was a sense that we actually have to get through this process a little

bit, let the working groups do their different pieces, and now, we're starting to get to the point where that problem statement is starting to come together, or realistically, problem statements, right, to your point. So, I think we're close with that, I think we've been working very closely with the framing group as well, and I think we're at the point where we can start to lay this out, and then, hopefully, through our group and the others, we can start to demonstrate how the use cases that we're putting together actually go towards addressing that problem statement.

>>: And the tactical opportunity, I'm looking right at Afton from FDA, they have an incredibly clear objective, an incredibly clear set of use cases. It's a subset of all these --

>>: Josh, I think that group is going to talk a little bit later, so I wanted, just again, to see if there's anyone either in the room or on the phone who has said I have a use case that you guys should capture and make sure that while we have this opportunity.

>>: Duncan Sparel. Don't worry, everybody takes my first name and last name and mushes them together. I have a use case I'd like you to consider, which I guess means I'm volunteering, but I'm a security

geek and a software developer, now that I'm retired, I work a lot in the open source area in writing open source security software, and the first thing the Air Force ever attacks is the radar system, so as a hacker, first thing you have to go after is the security systems. I do think that Cobbler's children go barefoot a lot in the security industry, and there could be a security use case on software that you use for security should be following all these rules.

>>: Fantastic. We'll follow-up with you on how to drill down to precision.

>>: That's right.

>>: Omar, and, first, kudos. I think even if you focus on medical devices, it applies to a lot of things. The only comment is, of course, we can comment, or we can bring 20,000 use cases right now, so the scope, you probably have to narrow it. I would suggest to, and I like the example you had with the cloud, because the way that we develop software nowadays, including from open source, to things that, um, again, I'm coming from Cisco, we inherit a lot of things from upstream providers, also from downstream providers, as well as lateral. The patchy struts that you had in there, probably in the use case of the cloud,

there are probably 20 or 30 different points that can be there, so are you addressing, because, like, you can run into scope creep, are you addressing that within your discussions right now?

>>: That's, I'll let this group talk about what they've talked about, but one thing that I want to ask, and maybe we can talk about this this afternoon, you hit a great point. It is a cliche to say that the line between on and the cloud is blurry, so I think something this group needs to think about is what does this notion of software transparency mean when we're dealing with something that's not just on the premise, and I think we should have a discussion about what is the, what makes that use case similar, and what makes it different, and, maybe, we need to drill more into fleshing out the set of use cases, but I think we should certainly start that conversation.

>>: I can't help but think like a hacker, so some of the use cases we're harvesting, even though we kind of left this room on July 19th saying, maybe, cloud services and SASS were out of scope, I don't think they're going to stay out of scope, so I think that our interviews will pull them back into scope, and I think it'll be more similar than different. I think, and it

was very important that we focused on concrete, harvestable, things people are doing as the crawl stage, maybe between now and February, we do the walk stage, and then the run stage, but, um, I have a hunch it's less about what is the SBoM, it's more about how does one consume it, like, you know, an interrogatable API or something, but they become very necessary, especially in ad hoc mash-ups and micro services, and we can't ignore it, right? We don't want to solve for the past.

>>: All right. Any last comments on, um, the idea of a use case? I have, um, Daniel Beard is on the phone.

>>: Hello. Um, so, kind of going off what Josh was saying for profitability, we started doing, um, SBoMs, um, for security reasons and open source compliance reasons, and what we found out was there was an awful lot of libraries, um, in the devices we were making, um, that we didn't need, so by taking them out, we were able to sometimes increase performance, use, you know, cheaper chips, um, so there's actually a business reason why these exist.

>>: Yeah, we used to call that code bloat, like when you look at your SBoM, there's a lot of stuff, and

there's a lot of old stuff, and it makes you go, ew, but then you start cleaning it up, and I think Rob from the task force said he did seven package updates and got rid of over 1400 vulnerabilities by just updating seven packages. A lot of people say do I really need this thing? So, you know, that quote of sunlight is the best disinfectant, sometimes, the act of actually starting this journey ends up getting you to weed out the, um, the bloat, the elective attack surface, an elective risk.

>>: All right. Thank you, John. Thank you, Josh.