>> So now we have the Framing Working Group, and that's Josh Corman and -- Robin, can you open up Ben Ransford's line, as well, so that he can join us from the ceiling?
>> All right. So we're the Use Cases and State of Practice Group. I will start. So I'm Joshua Corman. I'm the Chief Security Officer at PTC and founder of IAmTheCalvary.org. My co-chairs for this working group are John Bangart (phonetic), and then on the phone for the second ten-minute chunk of this will be Ben Ransford. Hopefully, the lag will allow. I think he's going to -- we have a clever technical solution for that.

We're going to do about ten minutes of -- we keep getting new people -- lots of new people in our Working Group, so we've found that even if it's a little bit of repetition, it's reinforcement, and it's also very useful on-ramps for new voices and new participants. We see some new faces in the room today.

So I'm going to remind us that while most of the State of Practice today is innovation done within a single stakeholder in a supply chain, the promise here and the opportunity is to have to systemwide value across up- and downstream in that supply stream.

So we're going to contextualize this. And we've created some nouns and verbs that are useful for doing so, we're trying to harmonize around them. So State of Practice is really doing two things. We're looking at what's already working within, say, financial services in early adopters that do this very often and very well for the software they write and consume themselves, often for productivity enablement and productivity boost to avoid unplanned work. And then also, some smaller patches of this in safety critical spaces like medical devices.

So what we've done is just some visuals. One of our artistic teammates created this set of motifs that we can walk through in five minutes or less. In fact, we intend to have some sector-specific videos. But for now, let's just run a with singular example of a bedside infusion pump.

There's a bedside infusion pump. This one pump is made by a manufacturer; Acme, for example. It's deployed in 1 through N hospitals or health-delivery organization.

But that device is not made from whole cloth. It's not written from scratch. It has very large open- sourced projects in it, compound projects like Apache struts or the like, maybe an embedded OS, different things. So those fatter pipes are combined into the custom code written by the device manufacturer and then ultimately delivered.

And then those two have parts. This is a chain, right? And across that, we basically have given those some names, right? There's individual atomic parts. There are compound parts, which could be turtles on turtles all the way down. So there could be multiple levels there.

Ultimately, though, there's a legal wrapper of a final goods assembler. They may be regulated in a regulated industry. They may not be. They may be self-consumed. But a final goods assembler is the last stop of creation and construction that is then delivered to an owner or operator.

So in this particular case, you might have a single atomic open-source product to do logging that gets woven into something like Apache struts which gets woven into an Acme pump which gets deployed into hospitals.

One of the challenges we've seen is, if you can see the red there, is a vulnerability in that could affect 1 through N hospitals. And in fact, one of the origins for us putting a software bill of materials requirement recommendation into congressional task force on healthcare was that a single job sterilization flaw and a single JBoss library of an old an unmaintained version affected a single device and took out patient care at Hollywood Presbyterian for a week. So that was one of our origin stories of, Could that be avoidable? Because while warned that this flaw existed, the hospital couldn't answer the simple questions of, Am I using JBoss and where? So they really just wanted to do some quick search or grip to take 10,000 devices maybe down to the 20 that might be affected. And I say "might" because I know Bruce and I are concerned about actual exploitability. So to short-list that. And what we also point out is, even if you did patch and remediate that infusion pump, maybe your desktop software in the hospital, as we saw with WannaCry -- another very damaging ransomware attack -- it wasn't so much hitting the medical devices, although it hit a few. It hit desktop systems in the clinical environment which still had the effect of affecting patient care.

So that same library that might have affected the medical device can also affect desktop software because they're sourcing from similar open-source atomic parts. So line of sight up and down the chain of taint or potential taint when there's a vulnerability anywhere in the chain, the ability to answer, Am I affected? Where am I affected? Or at least short-list which things may need to be taken offline, updated, or mitigated.

And the point of this graphic was really to frame a few more things. I'm going to skip through some of these because this is meant to be the faster version. But it's really important that we don't merely do this for, say, FDA pre-market guidance, which is requiring a CBOM, or cyber bill of materials; but rather, any clinical impact could be from any software used anywhere in there.

So it behooves us to think broader to not make an accidental de facto standard for just one sector because many of us are going to have similar problems and opportunities.

The other thing we did is, I like to borrow from Deming. Some people hate Deming; some people love Deming. But Deming had three principles for supply-chain management in Toyota in the '40s. And we're stealing liberally from there. He said you could improve the quality and profitability of your manufacturing by using fewer and better total number of suppliers; the highest quality parts from those suppliers; and track which parts go where throughout manufacturing so when there's a recall, you do prompt, agile, targeted, and profitable recall.

So we took these terms of S1, S2, S3; and in our larger document, we call them a basic maneuver or activity of supplier selection, a strategic decision of who you're going to have master services contracts with, build your code around, whatnot.

Number 2 is supply selection, which might be after we've chosen Acme as our pump, we want to use the least vulnerable version of the Acme pump before we deploy it into our hospital.

And then lastly, supply vigilance. So watch ongoing through retirement to make sure there's no new CVEs that might affect me or new patches that might need to be taken care of if vulnerability management is your goal.

And as such, let's repeat those. Supplier selection is a strategic decision of your upstream sources of benefit and risk, supply selection is which instance of that supplier, and supply vigilance is your ongoing burden to keep these things hygienic and clean.

So in those, we found the different verticals or different, in this case, horizontals. Our sectors have very different nouns and verbs for those things, but they're all kind of doing them. And if you're an architect in a compound-part project like Apache Struts 2, you may scrutinize which atomic pieces you do or don't use for the long haul.

If you're in Acme, you might be saying, I'm going to choose this middleware for industrial IOT connectivity in Cellemetry or this imbedded Wind River thing or this other BusyBox open-source project. So you're making a choice of supplier selection.

If you're in a bank, for example, we call it "procurement." So one of the more interesting bank interviews we did, they always ask for an SBOM. If they get one, they say, Great. They don't even necessarily look at it yet; but if they don't get one, they want a 20 percent discount off the top because they know it's going to cost them more every time there's a HARP lead or an attack to go investigate those chunks of software.

So we see procurement use cases. And then the DoD doesn't call it "procurement." They call it "acquisition." So the nouns and verbs vary. And as such, we've been interviewing lots of different people in those.

Here are a few of the interviews to date. Ben's going to talk a little bit more about this in a moment. And we're not going to read every single one of these like we did last time. They're very well captured in a document, but there's also a flow within those categories because it tends to be a different persona in each of these companies that does these different functions, or maybe even a couple of personas.

So in those sections, we just kind of looked at what's the indigenous active behavior that's already happening, and how could it be fueled by better SBOMs, more consistent SBOMs, more harmonized SBOMs and the like.

I may circle back to some of these graphics, but here's some of the systemic things to tie off before we get to Ben in my last two minutes. Some of the things we noticed is, if you look at those atomic parts, these little blue SBOMs of 1, they get aggregated into different formats by one or more of those compound projects. Some don't have them at all.

One of the things I'm encouraged about -- hopefully we can get to later today -- is when we talk about which open-source projects support this in the transition from no SBOMs to ubiquitous SBOMs, things like the CII badging -- the Core Infrastructure Initiative badging with tiers could be one of the criterion that having a comprehensive SBOM gets you a better badge and may naturally signal to component selection for developers.

But they are also very disparate. So what we saw is the different colors and shapes. While SBOMs are being created, they're not very standard. They don't have a standard composition. And therefore, when a hospital goes to consume them, they usually have to cut and paste and munge them together in some sort of CMBD, and then they do a grip, and it's not real great, right?

But you could picture a world where we do have a more consistent delivery, and they're just turtles on turtles and machine-readable and machine speed, and there would be less human intervention to get that visibility.

The second problem we think we have is, when we want to answer, Am I affected and where am I affected, if only 30 percent of your supply supplies one, when you go to short-list it, you're going to have a lot of blind spots. And one of the goals here isn't necessarily to invent anything new in the NTIA group. It's just amplify adoption.

So we want to harmonize what these outputs from existing tools look like, from software-composition analysis or build tools, then amplify that harmonized output. And then we might actually see more answered questions during an active attack of, Am I affected and where am I affected, and those blind spots dissolve.

And then lastly, there's other systemic use cases that can't be solved until we have a bit more pervasive adoptions such as, What if a user -- right now, I can have Acme warn me through a customer notification if there is an attack or vulnerability.

But what about when Acme goes out of business, which happens often in supply? The presence, the living artifact of the as-built SBOM is the last will and testament and the last information you have. And you can answer the question irrespective of the longevity of mergers and acquisitions and market failure.

So in that case, we've done some of these interviews. I'm going to hand it over to Ben. Hopefully, I'll click slides contemporaneous with him. And he's going to explain some of the interviews we have done, and we're going to reserve at least ten minutes for discussing what we intend to do next. Ben, are you ready?

>> Yes, can you hear me?

>> We can hear you.

>> Okay. Good. I'm actually inside -- I took a wrong turn at Albuquerque, so I'll let the record show that was a bad Bugs Bunny joke. So we worked -- I'm on the "Overall thrust" slide here, Josh.

>> Yeah.

>> So we figured that in order to figure out what people were actually doing in order to kind of gradually guide this toward a better world, we would have to understand current practices.

So our mandate was to figure out really just what's going on out there. For folks who have adopted SBOM in some form, not even be prescriptive about what that meant, but just really try to understand, you know, what piles of data they're working with, how they're using them, and what values they've been able to pull out of them.

So to adopt the familiar crawl, walk, run formulation. Basically, just kind of figure out at the very beginning, at the most basic level, Are you using SBOM as you would define it, and what are some of the obstacles if you're not, or even if you are?

In the middle bit, to walk, okay, we need to learn how those who are using some form of build materials for their software are doing it. So what workflow does it factor into and what is it like in your organization to use it?

And then the sort of most fun part toward the latter stages, for those who have kind of maturely adopted SBOM or had it going on for a while, what have you noticed that you can't quite achieve that you want to? So we want both people's -- (inaudible) -- and also to understand what people's sort of ideals were. You know, it's always good to synthesize from there.

So I'm advancing to the next slide: Are people using SBOM today? And I think our interview set is a little bit -- there's some selection bias, right? So we got people who have already thought enough about this problem to know that there was a working group and that they might want to join it. But we certainly got some interesting interviews, nevertheless.

So we looked at a few different -- we talked to people in a few different industries or verticals. Healthcare was -- you know, maybe you can say it's overrepresented here, but there's an acute problem.

We've talked to manufacturers who are sort of more horizontal across multiple industries, some big ones. And we have talked to Department of Defense and finance, which I forgot to put on this slide.

We were kind of looking throughout to see if we could kind of draw a picture, like, What's the spectrum here? And I think that we saw maybe the bulk of the struggle was concentrated on the far right ends of the supply chain, to refer back to Josh's left-to-right manufacturer to end user. So end users struggle, and it seems that the struggle sort of accumulates.

However, we have seen some maturity around organizations that really think about the supply chain carefully during procurement. And I think DoD is probably the gold standard for that, although not everybody can get there.

It was really great to talk to people who package software. So probably some software that you or your companies use, Red Hat. We talked to people who have really thought about this in the automotive space, and there's some encouraging work going on at Auto-ISAC and a little bit of cross-pollination that I can't fairly speak to.

And there are some medical-device manufacturers who I know are trying really hard. It's just as you start to get over toward the end users and you sort of have information loss in the present day from role to role, from player to player in the supply chain, you start to run into trouble, and they accumulate.

So I've switched slides to some of the obstacles that we've seen. So really, this was -- I'm sure we've all heard some flavors of these obstacles that come up, and we try to think about the software supply chain and what is in it.

We've certainly seen that there's -- for people who are trying to do a good job about carefully controlling the software that's going into their stuff, there's a pretty heavy vetting workload. I think Art started to speak to, you know, some of the challenges of getting upstream SBOMs and incorporating those.

We've definitely seen that some of the people we talked to had to do some of their own source-code review. They can't just take things from, you know, open-source components and treat those as, you know, sort of gospel. For example, there may be dependencies of which one of your upstream packages just mentions a few and some are sort of implicitly included. So it gets a little tricky when you're trying to vet something carefully to develop your own complete list or manifest.

We have definitely heard a good bit about vendors being intransigent or clueless about SBOM. So some don't know what it means. Some aren't really sure why you would want it or say that nobody's ever asked.

From developers, we've seen some sort of diffusion of responsibility. So I think, for software companies, it's really important to figure out whose responsibility it is. Is it QA? And some of the -- we've definitely seen some trouble around SBOMs, how would we -- (inaudible) -- and so on. These are technical problems.

So next slide. How are people using SBOM concepts today? We've seen that some -- are providing simply a listing of the filesystems. LS or DIR output for Windows users, just a text file. We've seen that people are using a folder full of -- (inaudible) -- and SBOMS. And so -- readable end goals that we've been talking about. Excuse me.

We have seen some people using SCA tools to look at executable or source code with mixed success. And I'm going to skip over my descriptions of who the supply-chain players are except to just note, again, there's some diffusion as you get towards the end users. And unless you're DoD, you often don't have much leverage up the chain. I've heard a lot of frustration around that.

Next slide on, What could SBOM unlock? This led to some really interesting conversations about, you know, what could you do if you had a list? And we got into all the reasons that people had been stymied when they were trying to do things like assessing the impact of a vulnerability.

So some of the things that popped out from these discussions were, We think we could do incident response better, even just being able to match a vulnerability quickly against the stuff that we have. We know that people are struggling with inventory management across industries. And so just having more information -- and really, even if you don't know the reasons in advance that you need all this information, more information about your assets is generally agreed to be better.

We've heard people talking about if you have security tools that are looking out for certain things or applying certain tests to certain devices or even looking out for certain baseline behavior versus observed behavior, you can do some tuning if you know what software is in the thing that you're poking at.

We've heard about end-of-life questions and how you phase out older assets based on, you know, This thing was using version old.old and we actually want new.new.

We have heard of SBOM as a forcing function. And actually, this is maybe one way that some pressure can be exerted upstream from end users or from those parts assemblers. It turns out, there's actually some benefit when end users, or people on the right parts of the supply chain, are asking for things. We've seen some encouraging little hints at people on the left part being able to streamline some of their own processes just through the act of having to come up with an SBOM.

Last slide, I just want to pull out some high-level interviews. As Josh said, we have a lot of documentation of stuff we've discovered and spreadsheets and so on that we're happy to guide people through if they join our group on Fridays.

But I want to say just a couple traits here's. It turns out people are very interested in vulnerability management, perhaps unsurprisingly. We did not see SBOM being used as a major determinant of whether a deal gets done in supplier selection except in maybe the DoD special case.

We have definitely seen this diffusion of ability to handle an SBOM, and end users are really kind of -- I think we're still hunting for good tools. We have noticed -- when we've asked about, you know, What information is on the SBOMs that you have, or whatever you're calling an SBOM, and what would you like to see on there, name and version number and the sort of basics that Art was talking about is really all we heard. Nobody I talked to even mentioned software hashes. So people are just delighted when they can get some basic information about names and versions.

And then one last bit is that it's important -- (inaudible) -- some of the intermediate pieces of code that are in systems. And so perhaps you have Log4j or some other Java library, but it's not specified which version of Java is on the device. And that's an important gap to make sure that we don't leave. And I'll stop there and hand it back to Josh. Thanks.

>> All right. Thank you. We overcame the 30-second lag. All right. Just one -- in light of Art's comments, just a couple pieces of color we didn't prepare. Please also unmute your lines. How do they do that?

>> Star 1.

>> Star 1. And prepare your questions for the last eight minutes or so.

To Ben's point, we haven't seen a lot of supplier selection, but I want to nuance that. We see a lot of financial services asking for an SBOM. So the presence of an SBOM but not the quality or content of an SBOM is coming up more as a procurement barrier or a negotiation tactic. The DoD is the one that wants to know what's in it a lot more for their mission support.

And then if you go back to the crawl, walk, run, as we kind of lean towards what's our final deliverable or what could we deliver by June, in the "run" category, I don't want to speak for Bruce, but I know we share one of the ones that I wanted this group to figure out.

There's some solved problems that people are doing within a bank or within a medical device that just aren't adopted well enough yet. We can help with adoption. But there are, as of yet, unsolved problems where I, as a vendor, get a list from my customer saying, We scan with this software- composition analysis tool. You've got a lot of vulnerabilities. Fix them. Or tell us that you you're not vulnerable to those.

So the ability to have a persistent ATA station across stakeholders and time, as the supplier, to get to add color would not be covered.

The good news is, Art's minimum viable -- what's the "I" stand for? Minimum viable --

>> Yeah. It's not my term.

>> "Inventory." Minimum viable inventory, that particular core solves almost all the problems we have encountered in the use cases, to a certain extent.

If you look at a use case like, Can I determine the licenses, simply knowing the project name -- not even the version but just the project name -- allows you to look at at least a possible set of licenses that were chosen.

But if we had another field for his more extended one, it could be better, right? So we have good with just the core. It could be better because of the licenses I could have chosen, this is the one I chose, right? So were there to be a field for chosen license, it may be more useful.

On the vulnerability analysis, go back to that 10,000 devices. The presence of open SSL may not mean I'm vulnerable, but I might be. So I might go from 10,000 down to 1,000. But if I also had some context, some build information, a vendor ATA station, some of those things that are the advanced fields we haven't done, I might get that thousand down to ten, right?

So to us, most of the use cases are unlocked and unleashed by a harmonization of outputs and an amplification of adoption of just the core MVI. That's it.

As somebody who is getting inundated with SBOM requests or hygiene requests to have a clean bill of health, I and folks like Bruce, I think we want to get to the "run" stage of our interviews, which I would characterize more as, What can't you do? If you're already doing these, what can't you do yet and why?

And that's where I want to shift between now and June is filling out the "what can't you yet do." We couldn't go there initially. We had to capture and codify what's being done. Now we want to kind of shift, as well, to what can't yet be done but for some of these things the Standards Group is working on and Art's outline and his crop circle/flower petal/whatnot. I think we should put each version of your graphics into a blockchain. Okay.

>> (Laughing). Josh, may I add one comment about formability awareness to that, what you just said? So -- and this may be some recency bias, but I heard something really encouraging towards the end of the set of interviews that I was describing there; and that is that I think end users, people who are receiving software -- you know, I asked a fair bit about this. Like, do you expect your vendors to be able to tell you, like, a great level of detail? Are we affected by this CV? Do you expect there to be a constant flow of information about vulnerabilities once an SBOM has been delivered to you?

What I heard a few times was, No. We expect there to be kind of a feedback loop. And so I think as we're thinking about how to structure these processes and these artifacts, it doesn't necessarily -- perfect is the enemy of the good. We don't necessarily have to get every piece of information into the SBOM artifact. We should understand that, you know, at least the more sophisticated end users will expect that there will be some feedback and some back and forth between vendors when a new vulnerability comes out. And so just knowing is kind of half the battle for the end users, and then they can fill in the rest.

>> All right. The first hand I saw in the room was Bruce.

>> You can have a dialogue with your customer if you've got one. It's really hard to do it when it's 100,000, and we have numbers like that. And many other -- especially consumer devices will have similar issues. So we need an automated way to provide this information, not one where we get in a dialogue, because it's just not feasible.

>> The next one I saw was Art and then Duncan.

>> Yeah. So just briefly -- and I apologize if I just missed it -- but roughly how many interviews were there?

>> Well, for the full-fledged, very large spreadsheet we called "sticky notes," I think we have seven captured, but we've done about twice as many of those calls.

>> Yeah. I'm not trying to -- it's a great -- I'm very happy that someone actually went and asked people what they were doing and wanted. What a great idea. Just wanted to check how far you guys had gotten. So thanks.

>> In fact, I think we have two in the room that we're going to do this week.

>> Josh, to that point, do you want to put in a plug of if folks wanted to talk to you, could they still? And perhaps if they wanted to talk to you but didn't want their company's name splashed around, is there something they can do?

>> It sounds like you've done it for me. We would like -- this is kind of -- I forgot to update this one this morning, but this is kind of -- we wanted at least one of each stakeholder type, but not just one. In fact, the Red Hat one was really interesting because it wasn't quite what we thought it would be and we learned a lot from it.

So we do want more. And some people have been afraid to put their company name on. There's absolutely no expectation you'll put your company name on. We just kind of want to understand what industry you're in and what you're doing. We're trying to capture these.

In fact, we haven't said this yet on our march to June, but we also think we're really, really primed to be kind of an FAQ Working Group, getting started by vertical. Because if you take the picture we've drawn and we have one of you from financial services, you can use the nouns and verbs from financial services to walk someone into an introduction to the master document.

So we almost want to have five-minute-or-less videos per role type and/or sector type. And it really isn't that hard. We made a couple already. We just didn't want to have them all be my voice. We need a of proof concept. It's pretty easy to outline this quickly for a brand new person that says, I want an SBOM. How the heck do I SBOM?

And we meet on Fridays at 1:00 Eastern currently. And pretty much every week, somebody's running it. Duncan, I think you were next.

>> Yeah. So this is a question for either Josh or Ben but can be prefaced by -- Allan can say, Defer that to this afternoon when we get into cross-group stuff. But one of the issues that had come up quite a lot in the Framing Group and was an open issue on Art's slide earlier was this whole one-level versus many-levels deep issue.

Did that come up at all in your discussions when you said, Hey, this MVI is all you need? Was it just one level or many when you said it meets all the needs? Or if it didn't come up and you want to defer it, that's fine, too.

>> We did discuss it a lot. We were in the camp of, You've got to go as far as you can, and you have to declare when you have an opaque spot.

What I really love from this morning's conversation is, maybe it's not "I see it" or "I'm opaque." But maybe -- and I don't think JC quite said this, but since I came from a software-composition-analysis-product company a while ago, you could say, "This is what we believe it to be and why we believe it." So it could be, like, a maybe. In a binary system, it could be a maybe, right?

But these kinds of tags could help go further back; but for the operational end user, they got to know everything that might affect them, not just, you know, one hop. But to his point, you know, like, if everyone does their one hop, we'll get there. I just think we'll get there quicker if there's a final -- I was in the camp of, As a final goods assembler, you should know what's in your stuff. You should make a best effort as far as you can and declare when you can't. That was my camp, but it's not a holy war. It's just what gets there sooner versus eventually, you know? Dave Waltermire?

>> So you mentioned that interest by a lot of the interviewees to do more -- to provide more data in an SBOM other than name and version was fairly low. I spent a lot of time talking to vulnerability managers and, you know, many of these sector folks that are concerned about supply-chain security, and they reflect that they often need more information.

Is there a certain -- where do these people sit that you're talking to within the organization? Do they have sort of purview into some of these other use cases where these other fields may be more useful?

>> It varies. When we do an interview in the spreadsheets, which are publicly linked, you can see their name and their title. Sometimes they pull in other colleagues, because

there's really -- if you look at my S1, S2, S3 demarcation from Deming, those will be three different people or even teams.

They're fairly senior people at fairly robust things. I think he misspoke a little. I don't think it's that they don't want it. I think they really, really want the minimum now, and they'll take it. And of course they're going to want more later. But going from 10,000 potential HARP leads to 1,000 is huge.

>> Sure.

>> So of course they want more specifics, and later, they'll ask for more. But in a "let's get going and stop debating what we desperately need" --

>> Yeah.

>> -- that's kind of what we're trying to channel.

>> Thanks for clarifying.

>> And we can pick that up -- -

>> Agreed. This is Ben. I agree, yeah.

>> Eliot?

>> Thanks, Allan. Thanks for your presentation. I hesitate to ask my question given the small sample size because it could lead to identifying information. But looking at your compound-parts provider list, in particular, I could imagine that there's another two-step process here in which you already have a lot of existing tooling to interface -- companies already have a lot of existing tooling to interface with, say, for instance, a Cert or various -- or handle their announcements; and what they really just need to know at some point is, do they need to make -- where do they need to investigate?

And so to the point about non-binary answers, non-binary answers are perfectly fine in some cases for us. We can do our own investigation once we have some reason to at least look.

But then from the operator's standpoint, I think, again, it's a little different problem. In our case, you know, we can tie into existing tooling from the operating standpoint. From the operator's standpoint, they also have existing tooling. There are companies out there that provide them a list of products on an ongoing basis that are vulnerable.

So there is this existing tooling chain. And so one of the questions is, how do you adapt the existing tool chain to all of this SBOM effort?

>> Well, I think we're out of time for now. We might get to some of that discussion, but we do have answers to your questions. The 20-second one is, some of the hospitals are further along on adapting existing tooling. They're just missing the granularity. I think when vulnerability was renounced at product levels, it was easier. Now that the attack surface is going down to open-source shared- dependence models, those tools didn't quite adapt yet. So it's the additional granularity that helps preserve the functionality that you're referring to.

>> So thank you, Josh and Ben.

(Applause)

>> Thank you so much for joining us early on the West Coast.