Developing and Using a Software Bill of Materials Framework

featuring Michael Bandor and Carol Woody as Interviewed by Suzanne Miller

Welcome to the SEI Podcast Series, a production of the Carnegie Mellon University Software Engineering Institute. The SEI is a federally funded research and development center sponsored by the U.S. Department of Defense. A transcript of today's podcast is posted on the SEI website at sei.cmu.edu/podcasts.

Suzanne Miller: Welcome to the SEI Podcast Series, a production of the Carnegie Mellon University Software Engineering Institute. My name is Suzanne Miller, and I am a principal researcher in the SEI Software Solutions Division. Joining me today to talk about their work on designing and building a software bill of materials framework are my friends Michael Bandor, a senior software engineer, and Dr. Carol Woody, who's a principal researcher over in CERT. Welcome to you both.

Carol Woody: Hello.

Michael Bandor: Good afternoon.

Suzanne: Let's begin by having you tell our audience a little bit about yourselves. I know you have both been on before, although, Mike, it has been quite a while. Let's start with you. Tell us what brought you to the SEI, what you do here, and what is one of the coolest things about your job.

Michael: I joined in May 2005 just after wrapping up an almost 23-year military career with

the U.S. Air Force as an enlisted programmer. I was aware of the Software Engineering Institute from their process improvement days and was a practitioner. They were a known factor. The coolest thing to me is the way we work across our various customer bases, the DoD, and federal government. We can see trends and see things that might not readily be apparent to other organizations. And we can bring some of that insight into our customer work and say, *Hey, we are starting to notice this problem is creeping up across several programs. Maybe there needs to be a way to address it*. We are early adopters of many things. We were one of the early folks assisting the DoD with Agile implementations. SBOMs [software bill of materials] is going to be another case where we are getting an early foot in the door.

Suzanne: OK, and, Carol, what is something you haven't told us before about you in terms of your career here?

Carol: Now, that is a tough one because I have done lots of podcasts. Well, I came to the SEI to actually finish my Ph.D. I was a consultant for New York City, having a five-and-a-half-hour commute every day. One does not get things done like in-depth research doing that. I stayed after I finished my Ph.D. because it was such a neat team of people to work with. It is like being one of the superstars with all this in-depth knowledge that you can wander down the halls and connect with. Right now, I am leading a team of experts. We are focusing on the challenges of cybersecurity early in the lifecycle, because too frequently organizations are kicking the can down the road because they don't know what to do about cybersecurity until it is almost too late. By that time, they have designed in all kinds of problems, they have built in all kinds of problems, and they are going to have to start over, which nobody is willing to do. That is why we have a lot of operational problems that we still don't know what to do with. Mike and I are working with other cybersecurity experts, delving into how we integrate cybersecurity in the supply chain because that is almost the holy grail at this point.

Suzanne: We have a podcast on that. That is one of the things we will reference in the transcript because that is a very important piece. It is good to have you back. Mike, it is good to talk to you on the podcast. It's been a long time, as we were talking about earlier. Today, as Carol mentioned, and Mike, we are here to talk about the idea of a software bill of materials and the framework that you have put together to help people to deal with that. But let's first talk about the threat landscape. You talked in your blog post about a recent report from Security Scorecard that examined more than 230,000 organizations, and they found that the systems of 98 percent of them—so, well over 150,000 of them—have had third-party software components breached within the preceding two years. Can you talk a little bit more about this. And talk about sort of how the SBOM idea might actually have helped to avoid all of those breaches, or at least some of those breaches. Carol, let's start with you.

Carol: Yes. Since the Heartland incident, which was in 2009, that was one that really hit a broad group of people. A lot of credit and debit card transactions were exposed. A lot of people got very nervous, but not nervous enough to really address the problems. But, they have been growing exponentially since then. Essentially, we are dealing with confidentiality, integrity, and availability of all of this critical data that gets compromised in every one of these activities, being a real challenge for identity theft, for organizations' structures in terms of financial management, et cetera. Everything is being exposed these days. A big chunk of this is now coming through the third-party products and services that we have all been adopting very energetically, because it saves us time and money. We can do things quickly. We can implement them very, very quickly. But, what you are doing is you are basically adopting risk from other organizations, and you don't really have good insight into what they are doing, what pieces of other products they have actually integrated into what they are selling you, or giving you, in the case of open source. Without an SBOM, you really don't have any level of transparency into what is going on aside from the marketing material and maybe some implementation guides. This is a lot of risk. To start reducing this risk is really going to be massive. We are hoping that SBOMs will start to be at least a step forward to provide us transparency, but it is no silver bullet. Mike, you have been participating in some of the experiments with that. Maybe you want to add your two cents' worth.

Michael: Sure. Some of the earlier work we did for another customer revolved around software obsolescence. It was when we were doing that, the SBOM information came out. I started looking at this thinking, *This looks like an SBOM*. It is pretty much the same information. What we found is a lot of organizations tend to track things on spreadsheets with the lowest common denominator, and that was one of them. They were trying to track components and relationships on a spreadsheet. You can't see all the interrelationships that way. We started using a graph engine, Neo4j, to injest the data and start to see where, *Oh, this has dependency on this package, but so does this, and so does this*. And you start seeing things from a broader perspective than just those spreadsheets would show you. That was some of our early experimentation.

Suzanne: When we are talking about an SBOM, if I have an app on my phone, let's say I get an app for a debit card, a bank. Right now, I am guessing I couldn't send an email to the bank and say, *Hey, could I see the SBOM of all the different applications that your phone app depends on?* I am kind of assuming we are not there yet in terms of that level of transparency.

Michael: No, we aren't there yet. The concept of an SBOM has been around for a long time. The hardware world has had bill of materials or HBOMs for a long time. But SBOMs, there has been attempts over many years, and there has been some reluctance to provide, *Okay, what is the ingredient list?* That is actually one of the things, *Oh, am I revealing something? I don't want to do this.* No, you are saying, these are the ingredients, but I am not giving you

the recipe for how to assemble those ingredients. That is a big difference.

Suzanne: But, some people don't even want to let you have the ingredients because they fear that you are good enough chef that you can actually…

Michael: To replicate and to reverse engineer it, yes.

Suzanne: Right. So that is where some of that fear comes from. There is this balance between transparency and privacy in terms of the privacy of the people that are building the application and intellectual property as well. Go ahead, Carol.

Carol: The other piece is that, if you know you are using products that are of sketchy origin, by providing this SBOM, you are essentially providing information to those that might want to do something to your product that you may not want to have added. It is somewhat of a listing of potential vulnerabilities if they can identify them in the components that you are using. All of those become very much concern points for vendors in terms of putting these out in the open. I think we are not looking at them as something that somebody is going to be able to just call up and arbitrarily get their hands on. There will be a little bit more control on it than that.

Suzanne: When I look at the commercial, there is that side. We then also have the DoD, and we have the Executive Order 14028 from 2021. That basically says the U.S. government has to enhance supply chain software security and integrity. That is one of the places that you are looking at SBOMs really being a part of that understanding. Is that correct?

Carol: It is, but the SBOM by itself is not going to really do anything more than provide you with a picture of what you have got and maybe some insight into where you might be vulnerable. But, certainly, it is going to take a lot more integration with other information to actually make use of it. We have been exploring a lot of the potential for how we can actually leverage an SBOM. Face it, you have a lot of third-party software. You are going to have a lot of data around SBOM. If you are going to the expense and energy of collecting it, putting it together, we need to make more use of it than just having a nice package on the shelf that checks the box and says, *Yes, I did compliance*. We have been exploring how do you get value out of this? How can you really make it work for your organization? Some of that comes with what Mike was talking about with the visualization. I think a lot of it also comes with just increasing your familiarity with what you really have as far as risk. You should be able to start to assemble a realistic perspective on what you have in-house and potentially how you need to improve your protections. Mike, did you want to talk a little bit more maybe about the graphics that we were doing in terms of use cases?

Michael: Yes. On behalf of one of our customers, we started looking at some use cases for SBOMs and what an SBOM has in it that can answer that use case on its own versus *I need*

additional information beyond the SBOM. For example, an SBOM will tell you, *I have component X, and it depends on component Y*. It won't necessarily tell you where in the software architecture it is. That information is typically not part of an SBOM. It won't tell you what hardware element or segment it is in. You are going to need additional information from there. An SBOM, plus other information, gives you a lot better sight picture on what's going on with your system, what the risks are. It could turn out quite easily that—arbitrary example Log4j—that you have got more than one instance in your system that multiple components depend on that you may not necessarily be aware of until you start looking at those dependencies. This also includes your toolchain that developed those products.

Suzanne: We have been talking around SBOM—without actually defining what's in and out of the SBOM. Before we go further, why don't we go ahead and, for those that aren't familiar with this concept already, what would they expect to see in an SBOM? And, as you were saying, Mike, what would they expect not to see?

Michael: In the guidance that came out, there is a minimum essential elements document. What that laid out was, *We expect to see, OK, the author. Who provided this information? Where is the component from? The name of the component. The version number. If there is some sort of unique identifier. What is the date of the SBOM? How current is it? And, does it contain or depend upon other components?* That is where you get your primary, your secondary, your tertiary, all those layers down till, *Oh, I don't depend on anything else, so I know that that line stops as far as my dependencies are concerned.* That is the minimum set. Most of the SBOM tools—like right now the guidance that came out from the minimum essential elements talk about either using the SDPX format, CycloneDX, or the software identification (**SWID**). Each of those standards has additional elements that a tool may support. There is additional information, but at a minimum, there are those elements that they want you to have that complete compliance with the executive order. Now, where things get interesting, there was a recent report that came out earlier this year. The company looked at 3,000 SBOMs. I am guessing, based on the source, it was mostly open source. Only one percent of them were compliant with the minimum essential elements. The rest of them had problems, incomplete information, and version information—other things like that—that they weren't fully compliant. There are still some quality issues that need to be addressed that hopefully will slowly start to converge to where there is a better set of expectations.

Suzanne: Have you guys done a survey of—I just thought about this—how many systems are out there that you would want to have SBOMs for? It is got to be in the hundreds of thousands at this point.

Michael: I would guess BMX. You have got the primary product and then any components it uses, and then those components have components they use. This becomes a problem that expands exponentially rather quickly.

Carol: That is what we are talking about with a lot of data. Because you are going to have to go many, many levels deep to really know what you have got in your products. The framework we built, though, really is focused on how to use this stuff. Because an organization is going to have to have some really new practices and effective processes in place to really make use of this. They have got to have a whole series of processes integrated into their bills that feed the software bill of materials to appropriately insert their information. They are going to have to have some way through their acquisition of acquiring the SBOMs for each one of the products, libraries, services, you name it. It just goes on and on and on because we do a tremendous amount of reuse across the technology base. And then there has to be a way of vetting them in terms of quality, which gets to the issue Mike was talking about. *Garbage in, garbage out* is one of the standard statements, and this will just feed that engine if we are not careful. Then, if we are going to effectively use it, we have—we were exploring several use cases. If a new vulnerability comes out and you need to determine, *Do I have this?* you have got to go beyond just accessing the SBOM. That will tell you, *Yes, there is the piece in there somewhere in your ingredients list. It is spread all over your organization*. *Good luck in finding it*. So it has got to integrate with other areas in your environment, your configuration management, all of the controls that you have in your operational environment. You are going to want a way to flag this stuff automatically. There are new vulnerabilities discovered minute by minute. That volume is huge. You are not going to want to look these up one at a time. This is going to have to be something that also has automation related to it, so we are really talking about an ecosystem here. Our framework is really just scratching the surface. It is saying, *We know we need-ish changes in all of these different areas*. This is where you need to at least think about, get started on, some of them you will start manually, but you also need to be thinking about automating them.

Michael: Yes. One of the approaches we did when we did the framework wasn't, *How do you generate an SBOM?* We went from the standpoint of, *You have an SBOM, now what do I do with it? What should I be doing with it? How should I handle it? Should it be a configuration item? Does it affect my acquisition? Does it affect engineering? Does it affect tests in the battle?* Things like that. That is kind of a unique approach we have seen in the SBOM literature itself. No one seems to be addressing the *Now what?* questions.

Suzanne: You are making me think about sort of one of the connections that you briefly mentioned, but I want to come back to it. There is the configuration piece, but then there is also the architectural view of the system. I am going back to what you said earlier, Carol, about we need to get security in earlier. This is one of the places where, if the architecture is identifying the elements, and then that goes into an SBOM, you have got a connection

back to that. If we decide that we are not going to use this component, here are the effects that has because there is all these other places that use it or not. It seems like there is a strong connection between having an SBOM and having a well-defined software architecture that you can understand how the pieces in the SBOM essentially are part of the recipe, if you will, for the whole thing.

Michael: Right, and you are going to have more than one SBOM. You are going to have multiple SBOMs, because your product coming through your developmental pipeline is going to have one or more SBOMs (I don't know how many things are coming through there). SBOMs of the components that make up your pipeline, your tools, because some of those vulnerabilities may not only be in your product, they may be in your toolchain. Then you have got tools that may support the end product out in the field that didn't necessarily come through your pipeline that could also still have those vulnerabilities. We are talking multiple SBOMs. That is why one of the things that tools need to do is be able to ingest SBOMs besides generate them. *OK, I've got an SBOM for Adobe Reader or some other package. How many programs have we seen where they are using multiple versions of the same component?* You are going to have to have SBOMs for each of those. This is, *OK, this is not the current version. This was like Windows 8. I need an SBOM for that.*

Carol: Yes. And you are going to have to maintain that information about those multiple versions as you roll upgrades through your organization. All of these are issues that are coming to the forefront. Having an SBOM is nice, but actually making use of it is the only thing that makes sense. Because this will help us at least better understand what we have got in the supply chain. And also, as we start to characterize these, we should be able to start to quantify the risk somehow. That is more research.

Suzanne: More research. You always have a research agenda, Carol. You are never bored.

Carol: It is a target-rich environment. What can I say?

Suzanne: There you go. Another target I am thinking about is legacy systems. I know that many of our government customers don't often always have the option of upgrading. They have to use systems that have been in play for a long time that were way before anybody thought about SBOMs. Is it better just to leave it alone? Or are legacy systems something you really want to pay attention to and create SBOMs for if you can?

Michael: Legacy systems do present a challenge. If you have got access to the source code, you could use an SBOM tool to try to generate an SBOM to the best of your ability, at least at that first level of components. From there it is going to take some analysis. If you don't have access to the code, you may have to derive one through software composition analysis, scans of some type, trying to go figure out, okay, *What's in this recipe? What do I need to be worried about?* Because it all comes down to managing the risk. When they talk

about, on the executive order and then the subsequent minimum elements talk about the, *What unknowns do I have? How do I manage those?*

Carol: Well, too, you really need to prioritize the legacy products. Some of them have more exposure than others. Again, it looks at, how are you organized in terms of technology protection? If these are stand-alone tools that aren't widely connected, then your risk is fairly minimal. Maybe you put that on the lower priority. But if this something that is a key component to your product, and it is widely used, you really need to analyze it and understand what you have got. Because we know supply chain risk is growing.

Suzanne: I love talking to you all, and then I hate it because then I can't sleep for two nights afterward.

Michael: Exactly. You are not the first person to have made that statement.

Carol: Welcome to our world.

Suzanne: Welcome. I don't know how you sleep. I swear I don't know how you sleep. All right, so let's switch gears a little bit. Tech manufacturers, many of them are taking this seriously. But what are the biggest mistakes that they can make? What cautions would you give to tech manufacturers that do want to take this seriously and do want to create software bills of materials for these products?

Michael: So not necessarily mistakes, more challenges. As I said earlier, the quality. Is the data complete? That is a big one. Do I have all the fields filled out so the end user, whether it is going into another product or I am the product, knows what is in there? Consistency in like the version schemes. You name the versioning scheme, it is probably in an SBOM. But taking that version information and—because right now the emphasis seems to be on vulnerabilities—and tying it to a vulnerability database. If I can't make that match, there is a lot more manual analysis that has to be done. Again, drilling down those dependencies beyond the primary component. Can I figure out what's out there? That is one of the challenges. There are discussions back and forth I have seen in some of the articles about, *How do we make the SBOM data available?* One method potentially could be, okay, most of these developers or vendors have a support page. You can make it publicly available that way. Others may be a little bit more cautious about, *Well, we really don't want to tell you all the components and sensitivities, for whatever reason*. In that case you may have to get a direct submission from that vendor to whoever is developing your system or to use the acquiring organization. There are those challenges. Again, there is always discussion about intellectual property. There are ways to do it without…You show them the ingredients, you are not showing the recipe.

Carol: I also think it is a case where they need to crawl before they run. Start

experimenting with a few of your products. Work with some of your close customers so that you can start to create a smooth way before you roll something out, and then have everybody screaming at you because it is missing this, that, and the other. Or they can't get to it the way they need to. There are so many variables right now, and we don't have what I would consider to be well-structured processes ready for this yet. The tooling is just getting started. It is still very much on the hands-on stage, what I like to think of where you have to babysit it a lot to get out of it what you need. And we are still discovering, what do we ultimately need? You don't want to get halfway through a major development and realize, *Oh, my god, if we had done three or four other things at the beginning, it would be so much smoother*. Let's work our way into this. But the value I think is definitely to be had.

Suzanne: You are actually kind of moving into the transition aspects of this. What kind of piloting have you done for this framework? Can you share any of the results or any of the in-progress things that you are seeing that have informed how you are thinking about the SBOM problem?

Carol: I think from piloting it would be more expert eyes on the problem right now. We are really trying to assemble the best minds to figure out for the framework. Now, Mike, you have been exploring a few other avenues a little more closely.

Michael: Right. Like I said, we, with our approach that, *What do we do with this now that we have it?* that has been what we are trying to look for it for piloting on. We are always interested in hearing from organizations that are interested in applying it. Getting feedback. *Is it working? Is it not working? Do we need to tweak something, something that was not addressed?* There is some growing interest from some of our DoD and federal customers. A lot of it is through word of mouth. People have seen on our website. I know I was contacted from at least one commercial company that was interested in *How do we apply this? This might make sense to us as a company that provides software to the federal government*.

Suzanne: Right. Yes, that is going to be interesting because I am also thinking about even things like small businesses that provide support software and other support to the federal government. This could be very impactful if you are a small business, and you are trying to do this as opposed to one of the larger companies that have a lot of resources. I can see a lot of implications for this as this becomes a more popular way of communicating about what are our ingredients in our recipe?

Carol: Well, I think small businesses are going to have an easier time because they have fewer ingrained processes. They have got a little more flexibility than major organizations. Which, especially if you have a lot of divisions and you want to have things done consistently, structuring that the right way to roll out at large is challenging.

Michael: Yes. There are commercial tools. There are open source tools out there. Pick what makes sense for you, what works. There are other ways to get involved with this. The Cybersecurity and Infrastructure Security Agency, CISA, website has a lot of good SBOM information. They also have several working groups that the public or organizations can attend. I know I sit on one of them just to see what is going on with the adoption, the transition. The National Telecommunications and Information Administration website also has a lot of good software bill of materials information on it. They were the ones that originally hosted the original minimum essential elements that came out of the Department of Commerce. That website also has interesting SBOM myths. There is one of them that Carol actually touched upon earlier about, *Well, if I give you an SBOM, it will show you what is in my recipe. I can attack this.* While that is a risk, that is the first myth that attackers aren't using SBOMs to get into your system. They have other ways they are finding out. There are other things to worry about.

Suzanne: Any other guidance that you would want to give government agencies typically who are acquiring…They are starting out in this. What should they be asking for? What should they be looking for from their vendors?

Carol: They are going to have to ask for SBOMs in the contract if they want them. That is not something they can assume that a vendor is just going to hand them. We have already seen several agencies running into that problem of, *Oh, this is a new mandate. Let me just have the SBOM.* The vendors are saying, *It is extra work for us. You are going to have to pay for that.* I think that is one of the starting areas.

Michael: Yes. It is not a, *Give me the SBOM. Check the box, I got an SBOM*. Again, you need to do the analysis. What are the impacts of it? It is a configuration information, and it needs to be treated as such, just as any other configuration item in your system. Take the time to do the due diligence for the analysis. That is the big payoff.

Suzanne: But the other thing that I heard you say is, don't assume that just because it is standard, if you will, configuration information that it is going to be provided. You must ask for it explicitly at this point in the way vendors are treating that kind of information.

Michael: Right. A lot of times the way these executive orders and other policy changes that come down from the federal government, those aren't reflected in the contract. It takes a contract change from the acquiring organization to say, *Hey, we need to start doing this. Let's figure out how much it is going to cost. When can we do it? When can we start expecting the outputs from it?* It is not, oh, a matter of, *Get a tool and run it and send it to us when you got it*. Also, figuring out the frequencies of the deliveries. Right now, a lot of the wording says, *major release* or *when the software's delivered*. Well, define *major release*. Define *delivered*. If you are using an Agile methodology, every build should have an SBOM coming out of it. There is some language (lawyer wording) that they need to be careful of to make sure

everyone is interpreting what those terms actually mean when it comes time to implement the contract.

Suzanne: OK, we don't call it a series. But, basically, we have been essentially doing a series of podcasts on supply chain security over time. Carol, that is a passion of yours, I know. Beyond the SBOMs, what is next for you in that realm of supply chain security?

Carol: Well, we are really trying to figure out, how do you characterize what the supply chain risk is? How do you think about it? If I have a supply chain, how do I characterize that risk? Right now, we are just trying to figure out qualitatively, but ultimately there should be a way that we can measure that risk. We don't want to count vulnerabilities. That doesn't really tell you anything. Because in order to be useful, the measurements have to give you a way to figure out how to respond in terms of what is too risky. *How do I know when it is reasonable, when it is not going to create additional problems?* All of those are questions we really don't know how to answer yet. We don't know how to evaluate it. We can look at it within a single product for a supply chain and look at the product itself, but all of the pieces that go into it are now just becoming part of the attack surface. We have this expanded window that we need to add to our risk picture. That is what we are working on right now, trying to characterize it and then figure out, what do we do with it?

Suzanne: OK, and, Mike, what is next for you? Are you going to continue with the SBOM stuff for a while? Or do you have something else.

Michael: I am actually. I still have a keen interest in the potential of graphing out the SBOMs when combined with architectural data and the vulnerability data, in particular, to show how you can take those different sets of data, put them together, and you have this, *Oh! light bulb moment! I've got a problem, and here's where it is at.* One aspect I am currently working on for another customer is looking at how SBOMs might effectively be used to support test and evaluation activities. Various things there. Yes, it is a very specific area that, yes, that we are looking at some potential use cases there. Like I said, we also had an opportunity recently to provide some comments back for the proposed federal acquisition wording.

Carol: We have probably had enough information to be dangerous right now, I am afraid.

Suzanne: Yes, that is the thing. All right. Well, thanks again for keeping me awake tonight.

Carol: Providing tonight's insomnia here.

Suzanne: There you go. But, seriously, I do want to thank you for talking with us today. I think that this is one of the many topics in cybersecurity that are starting to get attention and that needs a lot more attention. We do appreciate you taking the time to help us

understand it. As always, are going to include links in the transcript to resources, websites that were mentioned during this podcast. Finally, a reminder to our audience that our podcasts are available pretty much every place you download podcasts as well as our own SEI YouTube channel. If you like what you see and hear today, you are always welcome to give us a thumbs up. We always appreciate that. I want to thank Carol and Mike again for joining me today. I want to thank all of our viewers for paying attention to this important information.

Thanks for joining us, this episode is available where you download podcasts. Including SoundCloud, TuneIn radio, and Apple podcasts. It is also available on the SEI website at sei.cmu.edu/podcasts and the SEI's YouTube channel. This copyrighted work is made available through the Software Engineering Institute, a federally funded research and development center sponsored by the U.S. Department of Defense. For more information about the SEI and this work, please visit www.sei.cmu.edu. As always, if you have any questions, please don't hesitate to e-mail us at info@sei.cmu.edu. Thank you.