Getting Your Software Supply Chain Intune with SBOM Harmonization

Featuring Jessie Jamieson as Interviewed by Matt Butkovic

Matt Butkovic: Hello. Welcome to the SEI Podcast Series. My name is Matthew Butkovic. I am the technical director for Cyber Risk and Resilience in the SEI CERT Division. Today, we are here to talk about SBOMs, software bills of materials, a critical tool in understanding supply chain risk and ensuring that we produce quality software. A key challenge is ensuring that no matter what system or tools we use, comparable SBOMs are produced. I am joined today by Dr. Jessie Jamieson, a cybersecurity engineer in the SEI CERT Division. Welcome, Jessie.

Jessie Jamieson: Thanks, Matt. I am really happy to be here.

Matt: Jessie, for folks who maybe aren't as familiar with you and your body of work, let's start with you telling us a little bit about yourself.

Jessie: Yes. Sure. Thank you for the welcome and for the warm introduction. As you said, I have a PhD in mathematics actually. I love using math to solve all kinds of difficult problems. Something that you may not actually know about me is I am also published in two medical journals using mathematics to help solve medical problems [here] and [here]. I have had internships at Oak Ridge National Laboratory and at NASA Goddard, two awesome places to work. At Oak Ridge National Lab actually, that's where I was first

introduced to cybersecurity. Got a taste for it there working with their SOC [Security Operation Center], which was an awesome experience. And after that first taste, I said, *Yes, this is great. I love using math to solve this kind of difficult problem*. And cybersecurity is a field that really touches everyone, so everyone can see themselves affected by cybersecurity technology these days. And if there's any chance, I can use math to help solve these difficult problems and make the world a safer place, I'm all about it. I've touched a lot of different things. I have a lot of different interests. I follow Formula One very religiously, and I love cars. I've come up to the Pittsburgh Grand Prix a couple of times. I am just very interested in a lot of different things and topics. But I was very interested in helping do some research here on the problem of SBOMs.

Matt: Well, excellent. So, Jessie, thank you for providing your background and a little bit of a biography. One of the things you said I think is so important, which is using interdisciplinary approaches to solve problems. We are so glad you joined us here at the SEI, where you are applying your enormous skills in mathematics and other techniques. For instance, I know that you have a specialization in functional analysis and PDE [partial differential equations] theory. When you were earning your doctoral degree at the University of Nebraska, these are things that you learned, applied, and then it sounds like found outlets for, not only in cybersecurity but also in healthcare, which is great to hear.

Jessie: Actually, the PhD was a really interesting journey because studying PDEs and functional analysis was actually not my original goal. In undergrad, I studied graph theory and combinatorics, and I love the application of mathematics to network science. When I got to grad school, I changed fields and went in a slightly different direction, which was quite a balance because when I was doing my internships, I applied different kinds of math than what I was actually studying in my PhD. Although my PhD is actually in the theory of beams and plates and bridges and things, I studied all kinds of mathematics. Even within mathematics there is an interdisciplinary kind of approach there, where you can apply graph theory, you can apply network science or even some numerical analysis and mathematical modeling. There are all kinds of things we can apply to some of these problems. That is one of the reasons why I love math so much is there is all kinds of different fields within it field almost. And you kind of draw from all of it in order to really chip away at some of these problems.

Matt: Excellent. I know we care deeply about making the world a safer and better place. One of the ways that we are looking to do that is by advancing

the state of the practice and state of the art of software bills of materials. Maybe let's start at the very beginning for members of the audience who may be less familiar with the concept, can you maybe explain software bills of materials and what they are intended to do?

Jessie: Sure. Yes, that is a great way to start. I am sure most people in the audience are familiar with the concept of a nutritional label or of a shipping manifest. If you purchase something from the internet, you may get a label that tells you exactly what is in your package. And wouldn't it be great if we had that for software, too? That is really one of the goals. When we ship a package of software or a piece of software, and you are putting that in your environment, you should be confident in knowing what is exactly in that package and what you are actually incorporating into your environment. That gives you a better idea of risk, right? Ideally, we should be able to go down a list and see a full list of everything that is coming with that, with that software package, and our analysis and the work that we have done has shown that there is a little bit of trouble there. And we are going to get into that. But really, if you think about SBOMs as almost an ingredients list. That is one way that I usually think about these tools and these artifacts. That is basically a very brief, explain it like I am five, version of that.

Matt: No, no, that was that was great. An SBOM then, as you mentioned, is a list of ingredients. Maybe, a little more aspirational it is actually nutritional information. It is not just the contents of something, but also the implications of those components, right? I can see where that is immensely useful in having traceability in your software, but also then when you are presented with specific, threat scenario, it helps you then make risk decisions. I know we will talk more about that concept broadly, but I wanted to ask more specifically about a recent effort. In 2024, you were an integral part of an SBOM Harmonization Plugfest. Could you explain and maybe decode that name for our audience?

Jessie: Sure. Yes, the plug fest was the brainchild of Dr. Allan Friedman, who had this idea of, *Let's use a bunch of different SBOM tools to produce SBOMs for a set number of software targets*. If you standardize the software targets, and when you produce the SBOMs in the course of producing that software, we should be able to then produce a comparable set of SBOMs and go down the line between all of the SBOMs and say, *Is there agreement across all of the SBOMs for this one particular set of software*. We are holding variables constant in this experiment. So we are producing the SBOMs at the same lifecycle stage of the software and when we are producing the software, and we are holding the pieces of software constant, so those aren't changing. Actually,

for our plugfest, what we did was choose a number of software targets across a myriad of programing languages and software types, and we froze them at certain commits. Then we asked participants of the plugfest to use their tools to produce SBOMs for those pieces of software at that snapshot in time. Theoretically, the SBOMs are being produced at the same time for the same targets, at the same lifecycle stage, and all things should be the same. Spoiler alert, they weren't. It was really awesome, a really great experiment design and really allowed us to start digging into sources of divergence in SBOMs, as they are produced by a number of tools.

Matt: OK, this is real interesting, so the divergence is a problem for a couple of reasons, right? It calls into question the fidelity or the usefulness of the SBOM. If we can't trust the output of the systems, then you really can't trust as an input to the risk management that you might need to perform with it.

Jessie: Absolutely. Yes, that's a huge sort of source of trouble.

Matt: If we want to improve accuracy and veracity of SBOM, how would you suggest that we go about making these improvements, or maybe, to put it differently, what are some of the key challenges we found driving this divergence in output?

Jessie: There were actually quite a number of sources of divergence in the SBOMs that we studied. First of all, there are a particular set of minimum elements that are prescribed to be present in all SBOMs. These include things like the title of the software that the SBOM is produced for, the version of the software, cryptographic hashes, other elements that really allow you to get a very good idea, a precise idea of the software that the SBOM was produced for. We saw actually that even though these minimum elements are very well prescribed, there was a lot of variance even amongst those elements. For instance, the version number for the software packages that the SBOMs were produced for, you could see version *0.2* or spelling out of the word *version,* or just *Version 2* or *V2*, or just within the strings that populated the artifacts that we were given there was quite a lot of variance. Now, also, another source of variance was actually in the dependency structure of SBOMs. Now, as I said before, SBOMs are almost like a nutritional label, or they should be, right? You should be able to see direct links between the nutritional elements of your food and where they are in your food. So you have iron and all of these other nutrients. You should, for an SBOM be able to say, *Here's my main piece of software, the software package that the SBOM is for*, and be able to trace down through your software the individual submodules and packages that the software depends on. It turns

out that what some organizations, not only the producers of SBOMs but also the consumers of SBOM considered to be dependencies, results in a lot of deviation across SBOMs that are produced for software. For instance, if a software package calls another submodule as it is built or in a runtime environment, sometimes you would see that submodule listed as a dependency and sometimes you wouldn't. On the flip side, for different use cases of SBOM, you would see end users looking at the SBOM and say, *I expected this package to be listed as a dependency for my use case, whereas for other use cases that the dependency being present or not being present may not have made such a huge difference*. You have on one side the use case is driving the production of these SBOMs and what they are used for. And on the other side, the producers of SBOMs having to almost guess what they think the actual dependency structures or substructure within the artifact should actually be to maximize the usability for the end user. I touched on use cases. There are a huge number of use cases for SBOMs: not only asset management, inventory management. We touched on, I think you mentioned, vulnerability management and risk. Depending on what folks are using and end users are using the SBOMs for, they may need different information. Even that can drive differences across SBOMs because you may have one producer who is really targeting one sub audience and another producer targeting another sub audience and use case. Lots of sources of divergence in these SBOMs that we saw produced, and it came from a number of different sources.

Matt: It sounds like fit for purpose is a significant principle here, which is there are different use cases. It is not as simple as you just turn up the accuracy knob. Right? It sounds like the technical solutions to improve SBOM are probably multifaceted. There is some additional technical complexity that needs to be layered in to have more accurate and useful SBOM.

Jessie: Yes, I believe that is true. I know that recently, at least as of the time we are recording this podcast, CISA had put out a call for information for improving the minimum elements in SBOMs. I think, in a number of the recommendations and findings of our work, we are seeing reflected in the new guidance that they were given. I think that regardless of whether they directly used our input or not, we are at least seeing that movement in a positive direction with respect to making SBOMs better—more accurate, more precise, more usable—for folks interested in managing and understanding risk in their environments.

Matt: Yes. One of the most significant reasons we are both here is that, working for a federally funded research and development center, we can

make contributions to the profession. I know that recently there was a paper published that captures the lessons learned and the next steps working with your colleagues here at the SEI. I was wondering if you could highlight maybe one or two things that we have not touched on already, that you would want to draw the audience's attention to regarding where we are headed in SBOM, the key challenges yet to be resolved, really what other work you think that needs to be done?

Jessie: Yes, that is a great question. I think there is quite a lot of work yet to be done. We had some ideas for experiments we would like to run. For instance, one of the variables that we were not able to hold constant is knowing what is in the software package itself. So, if we had a software package that we had a ground truth, honest-to-goodness, golden-standard SBOM for, that we could then run the tools against and compare it to reality. We didn't quite have that for the software packages that we were using in in our plugfest because we didn't even know which tool would be accurate in the first place. That was the whole point of the plugfest, right? But if we design a software package ourselves that we know exactly what is in it and then run the plugfest again, that is one more variable we could control for and see and compare to an actual ground truth or gold standard. Another element is investigating the dependency structure of SBOMs. I talked about that earlier. Some SBOMs that were produced for the pieces of software that the tools apply to, they produce very, very flat SBOMs. Even though the contents were the same, the dependency structures weren't the same. I would really like to spend some time investigating that and understanding what additional information we might be able to glean from SBOMs given a more accurate dependency structure. *Does risk percolate uphill? How quickly does it percolate? Can you see individual packages that are present across many different elements of software in your ecosystem? Are they almost like the single points of failure in your environment? Can you use the dependency structure to understand how risk flows in your environment?* I think there is quite a lot more research that we could do. That is just a couple of ideas. I think in our paper…

Matt: which is available on the SEI website.

Jessie: Yes. The paper we published, our blog posts, highlight additional areas of future research and additional findings. We had plenty of findings. Also, I did want to highlight that we published our code that we use to analyze the SBOMs as open source software packages.

Matt: That's great.

Jessie: Yes, huge support from the open source community. I really want our results to be validated and reproducible to the degree that they can be. You can find all of those resources on the SEI [website] like you said. Even though we only used a few very simple Python scripts, they really enabled us to do the analysis, and I am hoping that they can be helpful for other folks as well.

Matt: I am hearing there is a great deal of additional experimentation possible. It sounds to me that developers, those that need to think about risk and supply chain, those that build tools that analyze software, that there is many roles in this ecosystem and that all could sort of benefit from increasing the fidelity and the trustworthiness of SBOMs. Clearly this is work of high value. A related question, Jessie. Artificial intelligence is something we talk about all the time. Are there any lessons we are learning looking at traditional computing SBOM that could be applied to artificial intelligence?

Jessie: I think that is a great question. I just want to highlight that we can barely get agreement in SBOMs across traditional software, much less very complex, AI-enabled systems. I think this is a challenge that we are going to have to tackle. I know that there is some work in that direction, I think here at the SEI and at other venues. AIBOMs are a thing. One thing that I am a huge proponent of is data strategy and getting data right. Without data, there are no AI-and-machine-learning-enabled tools and capabilities. We have talked about the importance of commit-stamping and time-stamping pieces of software and being able to map an SBOM, and when it was produced to the exact moment that the software was produced. Version controlling data sets and time stamping data sets. *Do we need SBOMs for our data sets? Do we need SBOMs or bills of materials for the parameters that inform the decisions and outputs that an AI or machine learning enabled model make?* I think there actually are a number of lessons to learn here. And I think, if anything, we have learned how difficult it is, like I said, in traditional software. There is just an added layer of complexity here once you introduce AI into the mix. Especially when you consider the AI models that are constantly training, constantly updating are changing continuously. So, continuous monitoring of changes of systems in your environment is a challenge ubiquitous across all software.

Matt: It sounds like if we can focus on some of the foundational elements, it pays benefits in lots of areas. A question Jessie, so for folks in the audience that are curious about SBOM, maybe not as familiar as you are with the subject. Maybe they are like you, maybe they are a mathematician that has a nascent interest in cyber. Where would you suggest they start? We have mentioned our publications and those are all be linked to this podcast, but

thoughts about other resources or where someone can understand the basics of SBOM.

Jessie: CISA and DHS have a number of resources on their website that folks can go to and read about their work in, in the SBOM world and in the supply chain world. But also, two of the primary standards and formats of SBOM are CycloneDX and SPDX. Those standards and the folks who really produce and maintain those standards also have a number of resources available for folks who can read all about how these SBOMs are formatted, reasons for including or excluding certain fields and formats and how they are turned into machine-readable artifacts. If you are interested in the actual modeling, those are two good places to start. In addition to all of the resources we've already mentioned.

Matt: Excellent. Well, I have a challenge for both of us. We both mentioned that we are fans of Formula One racing. I think about how many of the cars have software manufacturers' names on them. There must be a way for us to leverage our interest in SBOM to get into the Formula One world and ask questions.

Jessie: Oh, one hundred percent. I would love nothing more than to have my name plastered on the side of a Formula One car.

Matt: Jessie, I would love to have you back for an additional podcast. I know you have a wide range of interests, and you have so much to offer the audience based on your experience. Thoughts about other topics you would like to return for, to have a discussion like this?

Jessie: That is a great question. I think it depends on where math takes me next. There are a lot of applications of math to cybersecurity, which we have already kind of discussed. For instance, one area that may not be obvious is the application of epidemiological modeling to vulnerability remediation. You can build models that track the remediation of diseases as people go from susceptible to a disease to infected with a disease to recovered and not sick anymore. Those same concepts actually apply to vulnerabilities and vulnerability management. I know that there has been some research out there done on this topic but directly tying that system of differential equations to risk in an environment might be a fruitful area of research that we could address and come back and talk about in the future. It is just another instance of the ubiquity of mathematics and how awesome it is that you can find applications of these little subfields in math to a cybersecurity topic. It really is the marriage of two fantastic areas of research.

Matt: The world runs on math.

Jessie: It absolutely does.

Matt: I am always fascinated to meet someone that has the depth of understanding that you do of mathematics.

Jessie: Oh, I appreciate that. It wasn't easy. I know everyone had a terrible high school algebra experience. I promise, that is not normal, and math really is an awesome tool around the world. So, yes, if you're struggling with math keep trying.

Matt: Excellent. Well, that is a great place to end the conversation.

Jessie: Of course. Thank you, thank you.

Matt: Well, Jessie, I want to thank you for joining us today. Sincerely. This was an excellent conversation. As mentioned, we'll include links in the transcript to all the resources you mentioned, including recent blogs, a recent blog post and the technical report written by your team. And finally, reminder to our audience that our podcasts are available on *SoundCloud*, *Spotify*, *Apple podcasts*, as well as the *SEI's own YouTube channel*. If you like what you have seen here today, please give a thumbs up. Thank you for joining us.