Incorporating Supply Chain Risk and DevSecOps Into a Cybersecurity Strategy

featuring 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: Hi, my name is Suzanne Miller. I am a principal researcher in the SEI Software Solutions Division. Today, I am joining you in the middle of an ice storm in Pennsylvania, in Pittsburgh, Pennsylvania. We have had some power fluctuations, so I apologize if that affects how this video looks today. But even so, my colleague and friend, Dr. Carol Woody, is joining me today, who is also a principal researcher. She is over in the SEI's CERT Division. Her work has focused on government agencies, higher education, medical organizations, a wide variety of situations, and she has been a frequent guest on our show to talk about her research in the field of cybersecurity engineering and government agencies.

Today, we are going to be talking about supply-chain issues and the planning that needs to take place to integrate software from the supply chain into our operational environments, and there are a lot of different operational environments that we are going to be talking about. We also are going to talk about building a cybersecurity engineering strategy for DevSecOps that addresses those supply-chain challenges.

Welcome, Carol. It's always good to talk with you.

Carol Woody: Thank you. Good to be here.

Suzanne: Excellent. I want to start off by having you tell us a little bit about yourself, what brought you to the SEI, and the work that you do here on a day-to-day basis, for viewers who may not know you as well as I do.

Carol: Appreciate that. I am focused on research and how to mine what we know in the operational environment from cybersecurity and figure out how we better engineer and structure our systems so that they can more effectively join this very challenging environment. Too frequently, we are finding that the knowledge we expect our systems and software engineers to have is not there. So we have been focused on educating them extensively in terms of what is happening in the operational environment. Tied with that, what we are talking about today is that, really, they are not planning for the real context that they are having to address.

So we have got to make that integration much smoother. As we move into the DevSecOps environment, that integration is going to pick up speed. We have to be better prepared for how we are going to be fielding incrementally built systems in a way that we can keep them secure sufficiently as we increase their capability and functionality. So it's very much of a balancing act, but it's one that is going to require a lot of expertise, and we have got to make sure that we have the right eyes on the problem at the right time. One other area, tied with that, is that more and more of our software is coming from third parties, so we know less and less about the details of it, and the people that are building it are much more integrators as opposed to writers of software.

Suzanne: So that is a lot. We are going to unpack that one step at a time. But before we do that, just tell people a little bit about how you ended up at the SEI. Because I think everybody always likes to know, how did Carol get here? How did Suzie get here? Just give us a little bit of that background.

Carol: Well I started in software in college learning how to program and eventually moved out and into the software development and worked my way up career-wise into systems design and then strategic planning. I was in consulting in New York City for many years. At that point, I decided that I really wanted to finish my Ph.D. and move into research. At that point, the Software Engineering Institute was hiring, but they wanted to hire me on the cybersecurity side, and it was very perplexing. It didn't really fit what I had known, but we talked a good bit, and eventually, they convinced me that I needed to learn cybersecurity and really focus on that. So it has been an exciting ride ever since. Major learning curve initially, and then, beyond that, trying to figure out how we converge the languages of the operational security side into the languages and approaches that have been structured around acquisition and development. It is truly two separate stovepipes that really don't converge very well. We are continuing that effort and also trying to build educational material that we can drive out into the academic environment, because they don't really teach cybersecurity in the schools. So if you are hiring your engineer, you can't even assume that they've heard the word *security*, much less know how to do it. It is a fascinating and major area to be working in.

Suzanne: The reason I like you to tell your story is that switch in career. You are the example for me of, *I'm not stuck anywhere. If I want to learn something new, it doesn't matter if I am 20 or I am 40. I can learn something new and become an expert in a new field*, and you have done that so beautifully. So, thank you for sharing that with us.

Carol: I always love a challenge.

Suzanne: In a previous podcast with us, you talked about how to build a cybersecurity strategy, and I would reference people to that. We will put it in the transcript, but today, we want to take a deeper dive and examine how to build a cybersecurity strategy for DevSecOps that integrates with the supply chain, taking those two things and putting them together. Let's start with the challenges that organizations face—you introduced some of them already—when you are integrating software from the supply chain. We have got, *I don't know what's in it*. We have got, *I don't know where it comes from*. What are some of the other challenges that our organizations are facing?

Carol: One of the key areas that we see is that organizations really don't figure out what is at risk. So they are making many of their risk decisions around cost and schedule and thinking about only the components relative to cybersecurity. They are looking at, *How do I worry about cybersecurity of the code. How do I worry about the cybersecurity in my design model*? and not really looking at when you are fielding a system, what you are actually doing is connecting up suppliers and supplier software through the design that you put together into an operational environment, and none of this is static. All of it has to be continually updated and refreshed, because we have a very complex, operational attack environment that has to be factored in.

The systems that we field need to be built to recognize problems, resist them, and then recover from them quickly. All of that has to be part of the way we build systems. One of the examples that we studied in starting this project was a major four-part system. Each one of the parts had to be started up separately and then integrated, and it took four hours to restart the system. Operations would make choices of not bringing up some of the pieces if they happened to have a problem, because it took everybody down for four hours. That is not an acceptable process if you have an attack on one of those four components that you need to resurrect from quickly.

Suzanne: One of the things that we talked about before is that when we say supply chain, we are really talking about multiple layers.

Carol: Exactly.

Suzanne*: I may know a lot about my immediate supplier, but what about their suppliers and what about their suppliers that are using open-source code, even though my supplier, I don't*

allow to use open-source code. How does cybersecurity strategy address that layered view of the supply chain?

Carol: Well, we have to recognize it as a reality, and we are not going to be able to fix everything all at once. It is a journey, and we have got to continually be thinking about what risks do we care about the most immediately. We are basically building maturity at the same time as we are trying to deal with fielding. This is one of the challenges as we migrate into a DevSecOps environment. We are fielding more quickly, but we are building maturity into the tools, what we know about how to build our systems, how our suppliers are feeding us information and code that we can integrate. But then we are also learning from the operational environment and have to factor that back into how we are functioning and building systems and respond more quickly.

It is a culture shift, as well as just the mechanics of what we are doing and how we are dealing with this. Every one of these pieces represents some level of risk. We cannot be totally risk averse. We have to recognize that there will be risks that we have to accept. Which ones of them are acceptable? Well, that depends on what your system is doing. That depends on the context in which it is functioning, the mission it is supporting, and how robust your recovery mechanisms are. If you can recover in two seconds, maybe you care a lot less about how robust it is if you can immediately bring it back than [you would] if it takes you four hours. If you are dealing with data that is highly sensitive, and you are dealing with processes that are highly critical, like healthcare and banking and these areas that people care a lot about, then you are much more risk-averse than you would be if you are running a Zoom session like we are on. You want a certain amount of protection, but maybe not to the same level. All of these have to be planned for, and that is really what your cybersecurity strategy is. It has got to include all of the elements you are dealing with and recognize that these are parts of the anatomy of the product that you are building. You can't just focus on the code or just focus on the design, which is too frequently what we are seeing.

Suzanne: You have given a beautiful description of a lot of the whys that we need to do this, and especially with DevSecOps and the speed that DevSecOps gives us for being able to deploy updated software more frequently. You have given us a couple of the elements of a cybersecurity engineering strategy. What are the pieces of it that are specific to the supply-chain aspect that people, if they have built a cybersecurity strategy before, but have only really dealt with it within their own code. What is different about addressing a cybersecurity strategy for an integrated supply chain?

Carol: What is different is that you need to factor in the time delays. Who controls which parts have to be addressed? For example, the latest supply-chain risk we are dealing with, with Log4j, the suppliers can't fix their code overnight, but you have an immediate risk that you have to deal

with. So you are going to have to temporarily mitigate those risks in some way, until you can get the information from the supplier and apply the patch or the fix or whatever needs to be done for a more permanent solution. So you are always in a situation of waiting for the entity that owns the code to make the fix.

Now if you look at a supply chain that goes four or five layers down, if the problem is way down in those layers, that is a lot of waiting you may have to do. Also, you run the risk of the suppliers that are underneath not recognizing that they have a problem. You may be at risk for longer than you expect, which really puts looking at and understanding your operational environment to be very critical and knowing when the system is functioning properly and how it behaves if it has a problem.

Too frequently, right now, we don't know what our operational environment really looks like. I mean, think about all the changes that we have dealt with in the last three years. What is normal? I don't think we know right now, and so, detecting something that is out of normal that would be behavior that we not quite what we expect is becoming harder. In these instances, we're learning a lot of it by accident and then trying to scramble and figure out how to respond. Part of what we have to recognize in planning is that these things are going to happen. They are not once and done. This is a new pattern that we have to factor into how we think about and build systems, and it is that thinking that needs to be on the plate when you are building your cybersecurity strategy.

Suzanne: So I am a software manager, and you have scared me to death. Well done.

Carol: Sorry.

Suzanne: If I am trying to address this in my organization, what are some of the first steps that I need to take? I have already heard I need to understand my operational environment much better than I probably do. I have heard that I need to know who my suppliers are, and to the extent possible, understand who their suppliers are. What are some of the other things that are steps that I need to go through to address this, especially if I am trying to integrate my brand-new DevSecOps pipeline that is just chugging away beautifully. How do I go about doing this?

Carol: Well, software people need to be working closely with the systems engineers as a start, because what you really need is a very good picture of the system. The reason you need that is that you need to understand the attack surface that you are dealing with. What kind of risk are the ways that you are operating and executing forcing you to have to deal with? And the attack surface is much more now than just, *What inputs come into my system and how does my code operate, and what outputs am I producing?* Because you are dealing with a development environment that is tightly coupled now with your operational environment, your development

environment now becomes part of your attack surface. So you need to look at the robustness of the development environment, the tools that you are using, how they are integrated, how they are managed and monitored, and the potential impact that could have on the code you are actually fielding.

Also, from a software perspective, you need to make sure that as you are building the system incrementally, that you are creating the right level of robustness, reliability, with each increment that gets fielded, to meet the risk environment that it is being fielded in, so that you can continue to support the mission. You can't just say, *Oh, I made all these lovely features* and throw the features out first, and, *We'll get to security later*. That has been too much of a traditional approach in terms of how systems have been built. It has got to be thought about, designed, and tied together right from the beginning, and you have to link in and design for how you are going to recognize, resist, and recover from these things.

What monitoring is built in the system, so that when operations is looking at things, what can they see? What can they identify and discover, or are you just not tracking the things that they might need? All of those are part of planning and design that need to be considered. A reality is that insider threat is what we need to be worried about as well. In some cases, your organizational structure that is around how you are building and fielding the system becomes part of your attack surface.

That is maybe beyond what software itself would consider, but it does become an entry point that then fielding the software needs to be able to look at and say, *Someone that has access to these capabilities probably should not also have access to these other capabilities*. That might be part of how you think about the design as well as how you think about setting up your authentication and authorization. Where do you need to have your risks mitigated, and how important are they?

Those are all very tough decisions, and too frequently, we have got them being made either just in the supply chain, so that your suppliers are making decisions for you that then feed into your system that you are not aware of, which you had mentioned early, open source as being involved. We are looking at trying to institute things like software bill of materials (SBOM) to make those more visible, so that you can decide, *Do I want to accept this, or is the risk too great, and I don't happen to want that component in my system?* Right now we are not prepared to make those decisions, but we have got to think about them and how we want to deal with them. So it is a journey. We want to improve how we are doing things. We know we have got a baseline with gaps. What are those gaps and how risky are those gaps? We may need some mitigations temporarily in place until our tools and our capabilities and methods are in better shape to handle these risks.

Suzanne: One of the things that I can imply from what you said is that a cybersecurity engineering strategy isn't just a software strategy. That it really is a system and an organizational-level strategy. Did I get that correct?

Carol: Yes, it has got to be thought about as a key component there, tying those pieces together. And it has got to tie with your acquisition efforts, because, really, you are dealing with your suppliers through contracts. You are setting up relationships with them. It is not like you take their product and you never see them again. In this situation, you need them to be maintaining what they are delivering for the long haul, because you are depending on it, and how important that dependency is has to be factored into your risk management.

Suzanne: If I am trying to build a strategy like this, what are some of the challenges, red flags, that I should be looking for within the organization that would help me to know that, one, we are not ready for this or, although you have to be ready for this.

Carol: You have to start dealing with this.

Suzanne: Yes, I have got to start somewhere, but what are some of the challenges that you know of that people run into when they try and do this?

Carol: Too frequently, we are seeing a lot of these different components being handled in stovepipes. We have one group that is building a DevSecOps pipeline in isolation. Requirements are being thrown over the wall from systems, and even software engineering, and fed into the pipeline with no real linkage to what does that mean to the product or the cybersecurity? So you need all of those eyes working together, and most organizations don't have the organizational structure in place to support those communications. They don't have visibility. In some cases, the systems engineers don't even know who the software engineers are. Things are separated, and they definitely don't know who is handling the operations on the IT side.

We are building data-rich environments, but what we are not building are information-rich environments. So we have got to be looking at the data analytics and how it can make what we are producing and all the tools we are using and the pieces that we are operating useful to us in the long run to improve. So everything has to be organized around how do we improve not just the processes and the mechanics for faster delivery, but also, the integration and the information sharing and the analytics, so that we could do a better job of *recognize, resist, and recover*.

Suzanne: Excellent. This is a lot. This is probably one of the richest podcasts, in terms of information, that we have done in a while. This says to me that there is a big transition challenge in getting what you are talking about out into the hands of the people that are trying to get to it. What are some of the resources that we have available to people that are trying to adopt this to help them work their way through all the things we have talked about to get to an improved

posture, in terms of their cybersecurity strategy. I will preview that by saying I know you happen to have published a blog post and a webcast on this. Those are certainly things I know that people would want to look at.

Carol: We co-authored a book that goes into the basics of what needs to be done. There is a cybersecurity certification for software assurance as well that covers a lot of the background and some of the methods in this area, touching on what you need to do for requirements, what you need to think about for the supply chain. It is not a how to. It is more of an understanding, because the real challenge we have is that each system that you are creating and each operational environment you are targeting, as well as the way you are building it, is unique. There are a lot of similarities that can be shared, but what we are also looking at is how can we provide basic levels of understanding that can be taken in templates that can be used.

We have a lot of information about threat modeling that we are looking at, which would say how do you look at this attack surface that you're building to begin to figure out which threats are important to you, and then factor that into the way you are thinking about the system and what you need to do. We have also been working on an independent model that will assemble the issues in DevSecOps that need to be considered relevant to software assurance and cybersecurity. Things like what requirements are critical? What processes and practices need to be in place? So that you can begin to compare yourself to the model to say where are we? Where are the gaps? Where the areas that we need to improve?

You are probably not in a situation where things are immediately going to go up in flames in any one moment unless you happen to have the joy of being one of the victims for the latest attack. But you want to create an environment that is as resilient as the product that you are trying to push out. As we more tightly couple all of those pieces, we have got to have them working together more tightly. The communication and the integration among those have to be reflected in how we talk about problems, who we talk to, who we share information with. All of that coupling is becoming more and more critical, and your suppliers now are becoming part of that integration need. They need to understand what your issues are, and they need to be lined up to support you. All of these relationships have to be initially established and then nurtured over time. We are currently not organized for that. Everybody is really structured in very highly specialized stovepipes. In many cases, we all use different terminologies for the same thing. We talk past each other a lot, and that has created some amazing dialogues that I can't even begin to describe.

Suzanne: I actually really resonated with your statement that said, *We need to create environments that are as resilient as the products that we are pushing out*. I think that is a summary in many ways of what we are trying to achieve; the cybersecurity engineering strategy is one of the ways that we are trying to achieve that as a big goal. I really like that as a big goal

for that. You have done many things related, and you have talked about some of them in terms of transition mechanisms. What are you looking at doing next in this area of work? What are your research interests in moving this forward?

Carol: We are in the process of assembling the practices that are really critical to integrating program management, engineering, and the supply chain, and articulating those in a way that we can begin to move ahead and determine, *Do we have the right things in place*? because we are seeing those relationships as being really critical. One of the participants in that research with me has been working with this effort under DHS [Department of Homeland Security], and they have been using supplier risk-management assessments through the critical infrastructure for a good number of years now. We are looking at leveraging that and augmenting it with the engineering aspects to define what it is that needs to be in place so organizations can begin to look at themselves more critically and say, *What are we missing?* If that is a risk, then we have got to begin to figure out how to address it.

Suzanne: I want to thank you for this conversation. I think there are a lot of people that hear the words but don't really get the message that this supply-chain stuff really does have an impact on many, many aspects of our operating environments, the products that we rely on, the electricity infrastructure, and all the things that go along with that. So thank you very much. For our listeners and viewers, thank you for joining us today on this ice storm day in Pennsylvania. We managed to get through the podcast without any further power fluctuations. Somebody has got the *recover* piece going, even if we don't have *resist* all the way in place.

We will include links in our transcript to all the resources that Carol has mentioned, including the blog and the webcast, and some of the other things that she talked about. For those of you that aren't familiar with where to find us, this podcast is available on the SEI website at *sei.cmu.edu/podcasts* and anywhere else you get your podcasts—iTunes, Stitcher, SoundCloud, Spotify, and even YouTube, because we are a video too. If you find us on YouTube, please feel free to give us a thumbs up. We love thumbs up. As always, if you have any follow-up questions about this discussion or Carol's work, please don't hesitate to email us at *info@sei.cmu.edu*. Thank you very much.

Thanks for joining us. This episode is available where you download podcasts, including SoundCloud, Stitcher, TuneIn Radio, Google Podcasts, 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 email us at info@sei.cmu.edu. Thank you.