Key Steps to Integrate Secure by Design into Acquisition and Development

featuring Robert Schiela 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. My name is Suzanne Miller, and I am a principal researcher here in the SEI Software Solutions Division.

Today I am very happy to work with two of my colleagues, Dr. Carol Woody and Robert Schiela from the CERT Division. Today we are here to talk about something called *secure by design*, which is a discussion that we started with the director of CERT, Greg Touhill, a little while ago, and we are going to continue that discussion today. For those of you that are not familiar with secure by design, we are looking at key steps that organizations can use to integrate secure by design into their organizations. We are going to be talking about what are some of the key principles of secure by design.

I want to welcome both of you. I also want to mention that this topic has

been a key tenet of the 2023 White House Cybersecurity Strategy, and related to that, the director of CISA, C-I-S-A, the Cybersecurity and Infrastructure Security Agency, the director of that agency, Jen Easterly, recently visited our Carnegie Mellon University campus to meet with leaders to discuss secure by design and encourage tech companies in particular to incorporate these principles into their products. This is a very big topic, and we are going to have a fun conversation about how do you do this. If you want to make some of these changes as either a government agency or a corporation, how do you make this happen?

I want to welcome both of you to this podcast. Bob, you have not been on our podcast series before, so I am going to start with you to ask you to give our audience a little bit about your background, why you came to the SEI, and what is the coolest thing about your job at the SEI. Then we will get Carol to do the same.

Robert Schiela: Sure, I will try not to take the whole time, but I will say it is a pleasure to be here for my first podcast, Suzie. I have been at the SEI now almost 20 years. I have been working in information security and cybersecurity my whole career. Trying to remember all the questions. I will just skip to what is great about my job, and it is easy to answer if I can have two.

First, I would say what both keeps me up at night and what gets me up and out of bed in the morning is the importance of our job and security. Technology has the potential of transforming our lives and being a great part of our lives, but it also has the potential of really hurting us when abused. Trying to make sure the technology works as we want it to is what I find really exciting, as well as working with brilliant people. I know it is cliché, but we have a few brilliant people that are just really energizing to work with here at the SEI.

Suzanne: Excellent. And one of them is sitting right next to you, Dr. Carol Woody.

Robert: I am sitting with two of them today.

Suzanne: Oh, listen to you. OK, all right.

Robert: It is a great, great day.

Suzanne: You can keep that coming. It is fine. Carol, tell us a little bit for

those that have not met you before.

Carol: I joined SEI actually almost 21 years ago.

Suzanne: And you were in one of my Intro to Software CMM classes way, way back early in your…I remember that, that was my first time I met you.

Carol: Yes, to get educated in a lot of what the SEI was doing at the time. I came here to finish my PhD, and to do a career shift because my background was in software and systems design and implementation. And actually, strategic planning was where I ended up. Then coming into SEI, they wanted people to learn cybersecurity who had the background in development and engineering to begin to help figure out how to build things more securely. We have been really working in this space, although it was not labelled the same thing, for quite a while. We have a lot of pieces, but it is like the goalpost keeps moving out because the attackers get more sophisticated, technology gets more sophisticated, the uses get more sophisticated. I think that plays into why I find it interesting. It is an ongoing challenge, and I have always been one that loved problem solving. Believe me, we have more than enough to go around here.

Suzanne: We do.

Carol: To keep me energized.

Suzanne: Well, and I appreciate…we recently had you speak about the Acquisition Security Framework. You have really been very active in all of these areas. I definitely want to hear what you have to say today about secure by design. Let's get into that, and let's talk about. You have introduced the idea, both of you, that the cyber landscape is really changing a lot, and there are things, everything from software-intensive, third-party component, supply chain, which I know Carol, you are very involved in, DevSecOps, continuous authority to operate is very big in the DoD space. The shifts required to deal with these kinds of ways of looking at systems and these ways of operating systems, they really give us new considerations for establishing the meaning of secure by design. That goes back to, Carol, your engineering approach. We have to bring this in in engineering. It cannot be, as we often talk about, it cannot be bolted on at the end.

Let's get an understanding of what does secure by design mean to you. We did go some directions with Greg, but I want to make sure that we have a baseline of what is secure by design from your viewpoint and what the SEI is

doing with it.

Robert: Sure, sure. Secure by design as a summary form is largely performing more security and assurance activities earlier in the product and system lifecycle. That means instead of waiting until we are testing the system, or it is already fielded, and we are patching, or we are trying to apply controls in a system that has already been deployed, we are trying to do activities that try and ensure the security of the system earlier during the requirements phase, during the design phase, during the development activities, and doing more of those activities early rather than waiting until late.

That is generally secure by design at a high level. What I think has been changing, as well, is…Well first, what is not changing. Unfortunately, I think what is old is new for a lot of this. What I mean by that is unfortunately, we are still suffering from a lot of the same type of security issues as we have for decades.

A lot of the aspects that are causing problems, we have either known about or have been worried about for a long time. But what is new is the amount of interconnectedness of the systems today, the amount of automation that we are starting to apply to the systems today, and the amount of dependence on the system. That all leads to, the risk is more than it has been, and the cost of trying to secure those systems during test and after deployment is just not sustainable.

Suzanne: Do you want to add anything to that, Carol?

Carol: Yes, I would like to backtrack a little bit and have us think about the fact that most systems used to be composed heavily of hardware, and software was an incidental component. I have heard estimates it was less than 7 or [even] 5 percent at times. At that point, if you did not worry about software vulnerabilities and defects, you were really ignoring a very small risk. We find that many, many programs even today still think of software as 100 percent reliable. What has changed now is that software is now handling 90, 99 percent of the functionality.

Suzanne And the decision making.

Carol: And the decision making. And it is integrated across the infrastructure. It is integrated into the way the communications are handled. So you have this system of systems of software, which means that now any vulnerability

or defect suddenly becomes a lot more potentially visible. We really do not have the mindset to recognize this risk early on and figure out how to mitigate it. It is largely being ignored until we get to the implementation phase, and the pen testers come in and basically prove that you have got Swiss cheese. Because you can slap all the controls on it you want to, but if the attackers can get around the controls very easily, they will. Certainly, that makes the system less usable and less secure. We find so many developers and engineers still think that cybersecurity is an implementation problem. Yes, there are implementation issues, but we have to take ownership in the engineering and development side for the pieces that lead to a lot of these problems.

Suzanne: Another recent podcast that we did was on a newer language called Rust. What I will assert is what you are talking about in terms of secure by design is making those kinds of decisions. *What language am I actually going to implement in? Am I going to use a language that has some known, in this case, memory-failure kinds of Swiss-cheese holes to it, or am I going to use a language that may not give me everything I want, but gives me more security?* Those are design decisions. Those are not implementation decisions because that decision has implications for performance and other aspects of things. We are really talking about, as you say, moving up to the left in the lifecycle of where security needs to have a prominent role, not a second-class-citizen role.

Carol: The supply chain is a key element of that too, because decisions about what cloud platform you are going to use, which development tools you are going to use, how you are going to integrate these pieces, how you are going to be applying them, when you are going to be applying vulnerability-analysis tools, all of those are made very early in the lifecycle. And if they are not well structured and well thought about based on the risk concerns that you need to have. We find in many organizations, they have never really thought about the threats they are dealing with. They minimize them based on how it used to be, as opposed to what it really is. All of those converge to create really major issues.

Suzanne: Something that we, I always talk about—Bob is not used to this, Carol is—We are dealing with legacy systems. Part of that interconnected set of systems of systems are legacy systems that were built and secured and tested and certified and deployed without really knowledge, in many cases, of these kinds of concerns and in these kinds of solutions. In addition to secure by design on the new systems, there is this other aspect of secure by design, or at least evaluating what is the state of legacy software that we are

dealing with, and how are we going to protect ourselves from the things that we did not know back in the '80s? Because some of this software is that old when you get into some of these cyber-physical systems.

Carol: It is that old, and it is a black box. You do not have the resources that are active now, that really understand how to change it. Many people are afraid to touch it because if it ain't broke, don't fix it, but what does broken mean?

Suzanne: We do not know. Yes. We do not know what vulnerabilities…There are people, we know there are people in the threat space that will look at that black box and see it as a challenge. *I am going to open that up and I am going to…I see it as an opportunity, I am going to open that up and I am going to find out how to get at that, whatever that is.* Our hesitance in that case can actually be a fault in terms of our vulnerability.

Carol: You really have to bring program management in. It is not just all an engineering problem, because how program management identifies risk is going to determine what gets addressed. If risk is only looking at programmatics and cost and schedule and not considering the vulnerabilities and potential attack surface that you are creating, then you are not looking at the right problems that we need to have addressed.

Suzanne: This especially goes for systems. We have a lot of systems now that we work with that do not just have a 5-year life span or even a 10-year life span. Some of these are designed to have a 50-year lifespan. If it is designed for 50, we know we are going to push it out to 75 if we can get away with it. When you have these very long time frames, the decisions that are made in year 1 or even year 5 have cascading effects way down the pike in terms of ability to make changes that we need to because there are new threats, new technologies, etcetera, etcetera down the line.

Carol: You are pointing out another aspect that needs to be thought about, because the requirements change over time. And the technologies through the supply chain and what the technologies can do change over time. But we do not have a really good way of making sure that the protections that we thought were there to begin with are not deteriorating over time as the pieces are adjusted and changed.

It really boils back to integrating very effective risk management across the lifecycle and also thinking of it as a journey and not an activity. I do not do a risk assessment and then walk away and assume, *Everything is fine because I*

have mitigated all those pieces. What I have to create is a way to manage and monitor over time.

Suzanne: We are talking, now, instead of talking about software sustainment, the language is shifting to continuous modernization. We had the Defense Innovation Board, *Software Is Never Done*. This is another aspect of this, another dimension of it. It is not only continuous modernization, but continuous cybersecurity evolution. We have all these things that we have to think about.

Robert: I think it is a change of mindset of what we even mean by lifecycle, because often when we think lifecycle, we think of those five or seven phases happening. It is basically, traditionally from a waterfall model versus considering you have those phases, but you also have the time element and iterations. Those are happening especially for very long-lifetime or -lifespan systems happening over and over and over for a very long time.

Carol: Adding on to that, you also have a definition of ownership of the problem space that does not really fit with where we are seeing the challenges, because we talk about system owners, but does the system owner only own a small piece of the contents and components, or do they own where it sits in this system of systems and how it ties together? Too many of our engineering focuses are only inward within an arbitrary boundary and not really looking at what is the context that all of this is going to have to live in and survive in?

Suzanne: Some of that context is organizational. This goes back to the supply chain. *My boundary is not the boundary of what my organization is responsible for, even though I may be very tightly interconnected to other system elements that are owned by other organizations and might benefit from some of the kinds of secure-by-design things going across interfaces.*

I think we have established, in terms of the landscape, this is a huge issue, and it is something that I think along with the Acquisition Security Framework, to get more programmatic, especially, attention on this, this is really the engineering side of, *We have got to up our game in the engineering requirements piece for looking at all these issues*.

You have, conveniently enough, at least four steps. I am going to say *at least* because I think these are the initial four steps. I think this is going to grow. The number is not going to stay at four; sorry, people. But the four steps that if people have listened to this and go, *Oh, my goodness. Oh, I have to do*

something about this, what are the four steps that they should be considering now to bring secure by design into their mental model and into their organization?

Carol: They are really basic management steps, but they have to be performed at every level. You have to have some level of planning: how are you going to scope what you are doing? Where is your target? You cannot just assume you will get to some level of security magically. You have to be really thinking about, *What can I support?* And then how open can I make things as I move ahead.

You have to have the right tooling, because we are talking about a humongous interface activity and environment. If you are not automating some of this, it is going to get lost because we have seen with just vulnerability management, the volume is massive.

Training is one of the key aspects. We are coming from a hardware-centric environment, so we have to have all of the players at least have a basic understanding of this problem space and how to maneuver in it or know when they have to get someone that adds more expertise. Too frequently, we have budget constraints that are put in place that say, *I am hiring a rookie, and they are just going to have to wander through and figure out how to make it happen, and the tools are going to help them*. Tools are to be used, but they cannot run the process for you, and they are not designed to cover this full environment. They are still segmented as well.

We have to have a way of scoping how we deal with these. Then from there, we have to have a way of monitoring and measuring. It is going to be constant improvement. If you are not preparing for this because you cannot solve the whole problem right away, then you are basically adding to the problem instead of supporting and helping address it.

Suzanne: I want to go to the training one, because that is actually something that Greg and the director from CISA mentioned. I think it is training and education. This is sort of the same case as waterfall to Agile. Until we really had the bulk of software engineers being educated in Agile and how to do it, in their undergrad curriculum, we really did not see the push towards that in both government and commercial areas. I think that idea that we have to have this secure by design and the mental models associated with security embedded in our engineers, that is part of being a professional engineer and not just being a hack.

Carol: It should be, but the curriculum right now does not include any mention of that. We are actually seeing a few states stepping up to mandating that high school graduates have to have had at least one course. I know I have heard Nebraska…No, excuse me, it is North Dakota and South Carolina I know have instituted mandates that cybersecurity has to be part of the high school curriculum for every graduate. They will at least have heard the word. We can't say that about the current workforce right now.

Robert: I think there is a big challenge to it, the word you used earlier was *mindset* and *mental model*. These are not, as you mentioned, not just simple training, make something…

Suzanne: Go to a three-day class.

Robert: Make it available and they learn it, and that is all. It is not a trivial skill to get. It is a mindset change, which is going to be very, very significant of a challenge.

Carol: It does not translate to checklists and templates.

Robert: Right.

Carol: That is part of what we are struggling with. We see many organizations take the controls that NIST [National Institute of Standards and Technology] has assembled as very valuable guidance and apply it as a checklist. *I have got these, I have got these*. But who is the one that is making sure that the system is effectively implemented and that they cannot bypass that control?

Robert: If I could add, it is all part of the mindset challenge or mindset change is changing from the thought of what the system should do to what the system should not allow. Thinking not just about use cases and thinking through the threads of what the system should do, but thinking about the threats of how an attacker might abuse or misuse part of the system to their advantage.

Suzanne: I have heard people talk about misuse cases and abuse cases as a way of making that a reality, because you have to be able to communicate to people what could happen that you do not want to happen. In some cases, if you are talking about an airplane, it should not fall out of the sky. Some are very obvious. But there is a lot of stuff that is not obvious in terms of what the system should not do, who it should not allow to access and things like

that that have to be specified and have to be designed and have to be implemented and verified and certified, and all the things that we have in our lifecycle before you can really feel certain that you have at least dealt with that threat vector. We have lots of them; as you said Carol, you have lots of problems to solve.

Carol: Oh, yes.

Suzanne: You are going to be busy for a while yet.

Carol: Part of the mindset, too, or one of the examples I always use is that there are a lot of organizations and designers and engineers that assume they are working with embedded systems, and that these are isolated. But in reality, because of the connectivity through all of the different systems, the way software is integrated, updated, all of these pieces, you almost have to *prove* that you are isolated as opposed to *can assume* that you are isolated. We do not see that kind of mindset being applied. So risks are being ignored. And they are very high risks, in many cases, especially for critical infrastructure.

Suzanne: Okay, you are going to keep me up again. Every once in a while, after one of these podcasts, I have a night where I am like, *Oh my goodness. Here we go again!*

Carol: It is an opportunity.

Suzanne: It is an opportunity. You have actually brought up some of the challenges of implementing secure by design. There is the education challenge. There is the basic mindset challenge. There is the programmatic challenge of how do we get this in front of people that are in program management roles and taking it seriously? What are some of the other challenges that organizations should be on the lookout for if they are going to take this seriously, and what are some of the strategies for those?

Robert: I think a lot of the big challenges to me all relate to motivation and incentives. There is a constant pressure for software and system developers, more so to add new features quickly to a system. For a lot of systems and programs, there is a lot of pressure to focus on short-term benefits and costs as opposed to long-term benefits. It is more costly, it seems, up front to spend time and effort on security issues—what might happen or preventing what might happen in the future. Another is who is it hurting versus the cost? So users often get hurt more by these issues, not the vendors.

Those are three big challenges more related to motivation and incentives, which are really hard to deal with because of the economic and incentive structures that we have in place. Some strategies that I think are part of the model of measurement: *What gets measured gets done*. That is a key tenet for a lot of it. Additionally, from the customer perspective or user perspective when you can, make sure you are adding requirements that, thinking about this from your perspective, what do you need the system to not do or protect your users from? There is also the potential need for regulation just because some of these problems, the market by itself and the pressure for features and not considering enough security might require regulation and more governance aspects as well. Those are a few, and for the training and issues, we already talked about that.

Suzanne: Carol?

Carol: Getting to the mindset, there is a way of thinking about the problem space that needs to be there. Too frequently, we decompose everything down to the very smallest level because that is the easiest to build, and then put them all back together with the assumption that integration works well. But what we have to here think about is, we have a requirement to figure out, *Do these composition pieces actually create the whole that I am looking for?*

There are some structured ways of thinking that we can adopt from the safety area that [have] also been used in reliability around assurance cases to focus on that claim that we want to be secure by design and then look at what evidence do we have and start to assemble that. That gives us a structured start to put the information together as we are moving ahead. It also gives us a way to start to isolate, *What have we automated, and how well is that automation working?*

You can potentially farm these pieces out, but somebody has to start owning the big picture. Right now, we do not have that ownership. We do not have ownership for supply-chain risk management. It is scattered all over the place. We do not have ownership for the system security. It only belongs to somebody at the tail end that has to issue an authority to operate. Sometimes they are brought in early enough, but most of the time, they do not hear about it or see it until it lands on their review table. And then they are ending up being the bad cop and telling them about all the things they have not done. And it is back to the drawing board because you did not think about this stuff in advance, whereas you could have because it is not like it is secret.

Suzanne: We are trying to make sure it is not secret.

Carol: Yes. Trying to give more visibility to this type of…

Robert: I wanted to re-emphasize one of the aspects that Carol mentioned, which was automation. Earlier, I mentioned that automation was one of the aspects of software that was making it more risky for us, but it is a tool that we can use for adding assurance more systematically through the system. So adding more automation to the tools as part of our build process, part of our measurement process, so that these aspects and these…

Carol: You get consistency there, which is important.

Robert: You get more consistency to the system.

Suzanne: They are paid attention. You said a critical thing: Controls that cannot be bypassed through automation can be then…You get confidence because they are consistently being applied.

Carol: You have gates that check things, or you can have them if they are put in that way. But that requires a discipline ahead of time to really automate the right things at the right time, and that requires bringing the right expertise in. Again, we are seeing the mindset *of I have done this many, many times over. I am just doing it again even though it is new technology, new capabilities, new interfaces, new structures*. We have to recognize when things have changed enough that we need new mindsets. We are at that point now.

Suzanne: Yes we are. What we have just recently been talking about falls into my mindset as adoption challenges. There are a lot of things that are in the way of us transitioning these ideas into practice because of the adoption challenges. What are some other things in terms of transition for organizations that are thinking about this? There [are] the challenges, but how do you begin this journey? What are the resources, the enablers, that are available for people that take this seriously and want to do something about it?

Carol: I think the key is that we are putting in place capabilities that can be integrated with the way systems are already being built. DevSecOps is a popular way of addressing development. And we are looking at tools, improvement of tools, how you run them, where you run them, what they can provide you with, how decisions need to be affected, all kinds of

guidance related to that.

We have mentioned supply chain. That is becoming a major attack vector. So really thinking about where does the supply-chain risk impact you around your system? We have developed the Acquisition Security Framework to look at which pieces of this does engineering own? Which pieces does the program level own? How do the suppliers need to be managed to show how these pieces interconnect? A least this provides a way of comparing your program to the needs, to start to look for gaps.

Then we are looking very heavily into threat modeling and how do we tie those pieces early into systems engineering when they are also using model-based design and show how the pieces should fit together so that we can influence requirements, we can influence the way the pieces are integrated, and then start to influence some of the outcomes that we have to deal with and really start to highlight these risk issues.

Suzanne: Are we seeing a community building around this? One of the things that we often see when we have big problems like this is we start to see conferences, we start to see communities of practices. Are we starting to see this for this secure by design and the things that it touches? Or are we too soon for that?

Carol: There are pockets.

Robert: I would say there have been pockets for a long time. It ebbs and flows with regard to how much the broader community is concerned. As you mentioned, we just had Director Easterly come and talk about this as well. It changes how many people and at what level the attention is with regard to the communities.

Carol: I would say the pockets have grown out of the high-risk areas.

Suzanne: Infrastructure.

Carol: That is where critical infrastructure, the military, the DoD has had a lot of shared information and eyes on the problems. But moving it to the areas where the general developers understand it has been a challenge. We have to get there, because that is where the supply chain is.

Carol: We have had a lot of wraps and close hold on a lot of this knowledge

for a long time, and getting it sensitized to a point where it can be just broadly the way we do business is something that has to come out.

Robert: I would correct what I said a moment ago, which is, "where it is more critical." What I should have said is, *Where it is recognized to be more critical*.

Suzanne: Fair enough.

Robert: What we need to do is try and help people recognize that their system might be more critical, or there might be more critical aspects and effects of their system and risks and costs than they are currently thinking.

Carol: We have had the mindset of just protecting what we call critical assets. Unfortunately, because everything is connected to everything, then even these very insignificant little pieces can become pathways to those critical assets. Since they are no longer isolated because you have these multi-layers of software that can communicate and do, it is really much more important to look at the whole environment and really understand what is happening, as opposed to focusing on all the little pieces.

Suzanne: I do not want to go deep into this, but you just brought to my mind Internet of Things. I do not have a refrigerator that gives me contents and everything else, but the fact that it could means that it could communicate with me, and that is a pathway.

Carol: But just think of how much we are relying on the handheld devices that we cart around with us all the time. Even those are connected to our systems and doing major work for us. We are becoming more and more reliant on them because they are with us all the time. It is the path of ease from the user perspective. Businesses are relying on that because they no longer have to build that front interface anymore. They can leverage what is already there, but what they are leveraging is a very risky platform, and they are not taking that into account when they are looking at the interfaces.

Robert: One of the approaches that we do at the SEI is try to bring a lot of these communities together and the subcommunities, but there is a lot of relationship across them, and have different workshops, have different conferences for these topics. We have had multiple SecDevOps or DevSecOps conferences and workshops and days and things related to that process. We are planning a Secure Software by Design workshop or conference event to be coming up early in the summer. Right now, we have an open call for presentations or presenters. The intent is to bring experts

from the outside, not just the Software Engineering Institute, and bring them available to discuss what they think the situation and the community should be focused on but bring it into an event where people that are not necessarily experts can come and listen to what it is they should be doing and how they should be doing it from these experts in the fields.

Suzanne: OK. That is the kind of resource we will have in our transcript because that is what we do.

Carol: But we also have some existing training. So folks that are looking to just get their toe in right now and understand the concepts and issues, there are certificates. I know my team has fielded one in cybersecurity and software assurance. We have books that we have written, blogs, and podcasts that touch on lots of different aspects of this. So there is a wealth of exposure that is available, but it requires the people that are in the trenches doing the work to recognize this and raise their heads and expand their perspective.

Robert: Part of that and one of the opportunities or offerings for training is related to, as you mentioned, the blog post about Rust. And for implementation and development activities, we have secure coding standards and guidance for the many systems that cannot just start using Rust, that whether it is because it is embedded or legacy code or whatnot, are using C or C++ or even Java. We have guidance specifically for how to use those languages securely and not inject security vulnerabilities into your code.

Suzanne: Right. Good. Yes, and we have a long history of working in that area.

Robert: We do.

Suzanne: And I think the fact that we have a long history and that we are still talking about some of these things means that we still have people to reach when it comes to these kinds of ideas.

Carol: That, and the goal keeps moving further out. I know I have been working with one of your team members on automated code repair. That is an area that most organizations are uncomfortable with right now, but when the volume of the vulnerabilities is mammoth, we have to look at what can we automate. Some of the key ones that are constant problems that are very high risk, we are looking at how do we automate that?

Suzanne: So here we are. One more problem, but we have the beginning of solutions. Almost all of these things in the security arena that I have come to know about as an outsider is about mindset. And that is not something that happens overnight, and we know that. What are you going to do next to help us to get a better mindset on security? What are you both working on that I am going to bring you back to talk about in six months or a year?

Carol: We are trying to leverage the pieces we have already put in place. For example, we are taking the Acquisition Security Framework and creating very specific, tailored focuses on aspects that are prime in people's minds to get them working with it. Software bill of materials, for example, we are working on creating a tailored version that would focus [on], *If you are going to do software bill of materials, look at these pieces and make sure you do not have gaps and that you have included these.*

We are also looking at zero trust, which is another aspect of the mindset early on for design, and how can we integrate that with what you are doing with supply chain around the Acquisition Security Framework. Those are key pieces that we are in the middle of. We are also looking at what processes can we bring to bear to improve the way we are doing assurance cases. That structure has been more of an art than an ongoing capability and process. We are exploring use of the tools and ways which we can maybe add some pragmatic perspective in terms of looking at the data we have and creating ways to think of what assurance does that give us to help us then identify what are the gaps that we need to be dealing with further on. Hopefully it will help spur this mindset further.

Suzanne: What about you, Bob? What are you going to be working on?

Robert: As Carol mentioned, one of the focuses we have been doing is trying to identify for the development activities and coding, reducing the workload of finding and fixing weaknesses and vulnerabilities. So trying to improve the tools and add automation possibly with better filtering and automation for prioritization or even automated repairs.

I am also working more with some customers on trying to identify more specific approaches for design security. So in the design phases, as you mentioned, selecting your language even though it might impact you in development, it is a design decision. But there are many other aspects about considering encryption, considering different access-control approaches that often, the design phase, there are problems or defects in those designs. So

we are looking at how to provide guidance for what are the most important, impactful aspects of security in a design phase. Hopefully we will be able to abstract some of that work from customer work and generalize and provide some general guidance to the community.

Suzanne: I know some people who need that. That is the other thing is you cross communities. That work you are talking about crosses directly into the systems-engineering community. Some of the other work you are talking about is really more of the automated code repair, much more in the implementation, the software-engineering-test community. That is something that I think is notable about the work that we are doing all over CERT, that we are touching not just security professionals, but all the communities that contribute to building systems. That is really what we are trying to get at is getting the mindset shifted there, not just in the security professionals.

Carol: We need to augment how security professionals think as well, because too frequently, they are focused on the final system. They are used to working in the operational environment, but parts of that problem space are owned by engineering and design. But engineers and designers think differently. They organize their problems differently. So when you come to them and say, *You have to implement these controls*. That does not match with, *Well, what are the requirements that I would put in play that would signal that I need that control?* So we have to educate the security people in terms of how to better communicate the security needs back up into the engineering and design. I think that is one of the challenging areas too, as with the education, that most security people do not think about. They think of themselves as the policemen, the last step before bad things happen in the operational world. That is true, but they have to look at how do they communicate what is needed to get the right behavior.

Suzanne: But they cannot be the boy putting his finger in the dike. You know that old image. That cannot be the only way that we get this done. It is not going to work.

Carol: It also cannot be a game of *gotcha*. Too frequently, pen testing is turning into that. Most of these systems, if they are poorly designed, it is an easy way of communicating the problem, but it is not communicating in a way that you are changing the behavior and the thinking of what needs to be done. That is where we need to be making our impact.

Robert: I would say that it is back to the concept of helping more people

recognize that they can be part of the solution for the security issues and helping the security teams recognize how to better collaborate and communicate with them and be part of the team rather than be a separate team that is a gate that is calling out the engineers. But instead, again, working early during the early phases of design to be a part of the team and developing those requirements early on for the developers to try and implement those security aspects in the code, rather than be called out for not including them after the fact.

Suzanne: This conversation, and the different places that it has gone really highlights the fact that this secure by design is a very broad topic. So I am going to assert that we will have you back to talk some more about different aspects of this in the future.

Robert: That would be great.

Suzanne: I want to thank you for coming today and for giving people these initial steps to think about and blow their minds open again. I love it when we get people thinking in different ways. I want to thank you for that and I look forward to future conversations.

For our audience, we are going to have links in the transcript to all kinds of things we talked about: assurance cases, conferences, secure by design. So look forward to those, and also, look forward to seeing this on our YouTube channel. If you liked the video, we also have this podcast available on Stitcher, SoundCloud, Apple Podcasts, Google, I have to mention all of the different people, but you should be able to find this podcast. But our favorite, of course, is the YouTube channel. I want to thank all of our viewers today for listening in or viewing us, and again, thank you to Carol and Bob for having this conversation.

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 do not hesitate to email us at info@sei.cmu.edu.