Carnegie Mellon University
Software Engineering Institute

SEI Podcasts

Conversations in Software Engineering

An Infrastructure-Focused Framework for Adopting DevSecOps

featuring Vanessa Jackson and Lyndsi Hughes with 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. I am a principal researcher in the SEI Software Solutions Division. Today I am joined by my colleagues, Lyndsi Hughes and Vanessa Jackson. Vanessa is a senior engineer working in the SEI's DevSecOps group in the Software Solutions Division, and Lyndsi is a senior systems engineer in the SEI CERT Division. Today, we are here to talk about a DevSecOps framework that they developed that helps organizations plan and implement a roadmap to functional capabilities in the continuous integration and continuous deployment (CI/CD) pipeline. Welcome, Vanessa and Lyndsi.

Vanessa Jackson: Hello, Suz.

Lyndsi Hughes: Hi, Suz. It is good to be here.

Suzanne: Yes, good to have everybody here. You are both new to the podcast series, so let's start by having each of you tell our audience a little bit about yourselves, how it is that you came to be at the SEI, and what do you think is cool about your work. Vanessa, why don't we start with you?

Vanessa: I have been at the SEI, this will be my 12th year. I came to the SEI from the DoD contracting realm where I was previously serving different roles as a developer, as a system administrator, and as a database engineer. Initially, I did not know what the SEI was or what an FFRDC was, but I was tired of the contracting world and the way business was conducted there. Once a recruiter contacted me about SEI and I had a better understanding for what the organization does, I was excited to join. I am excited about the work that I continue to do right now.

I am helping to set up and manage a DevSecOps team for a Navy development organization, helping the government to transition in a really functional manner into DevSecOps on nonstandard resources. Typically, a lot of government development organizations don't have access to the Azure or AWS [Amazon Web Services] platforms. Setting up best practices in these kind of nonstandard development environments and trying to ensure that the government does things in the way that industry is able to, even though industry oftentimes has a lot more resources, is exciting to me.

Suzanne: Excellent. Yes, we do have a lot of interesting platforms that we have to work in. Lyndsi, what about you?

Lyndsi: I have also been with the SEI for about 12 years. I joined back in 2010 when I was a graduate student as an intern. When I finished school, I joined full-time. My work here has been mostly focused on the infrastructure and hardware side, so I spend a lot of my time, and I have spent a lot of my time here, working on supporting and operating computing environments for developers. A lot of my experience comes from the perspective of a person who is not a developer, but who is in charge of keeping those computers that developers and analysts and researchers need to use to do their work, keeping those computers happy and running and making sure the software that everyone needs is available.

Suzanne: Again, in the government, those are not always the things you might expect to find in comparison to what is in the commercial space.

Lyndsi: Exactly right. There is definitely some specialized-type stuff that we have done here that is not super common.

Suzanne: OK. All right, so this leads us into the idea of DevSecOps. For those of our audience that are not familiar with DevSecOps, or DSO as we also call it, we have lots and lots of resources on the SEI website and the YouTube channel to help you understand what is DevSecOps. We are actually not going to go into that here. We'll put those resources in the transcript for you, but we want to talk about this framework for adoption of DevSecOps that Lyndsi and Vanessa have developed.

I want to start by quoting Richard Seroter. He is the director of developer relations and outbound product management at Google Cloud. He wrote in a post that the Software Engineering Institute put together an interesting DevSecOps adoption framework, which is a bit different than the ones you may have seen already. The first thing I would like you to tell us about is, give us a little overview of the framework, but tell us especially how it is different from what is out there already in terms of DevSecOps adoption frameworks. Lyndsi, why don't we start with you on that one?

Lyndsi: Sure. I think this is different from the traditional, previously existing DevSecOps frameworks because it is focused for a technical audience who is responsible for maintaining the development infrastructure. I think a lot of the existing frameworks that are out there right now are very focused on the developer perspective and how people who are at the keyboard typing code day after day after day to produce a software product, how they can do that work better. I think there are a lot of opportunities for people on the infrastructure side to benefit from the ideas and the practices that are important in DevSecOps. I think our framework really summarizes what those things are from the infrastructure's perspective.

Suzanne: That is what I noticed about it when I took a look at it. This is really coming at things from a different viewpoint. We usually do have the end-user adoption as the adoption target, and here, you are saying, *No, this target is the people that are going to have to develop and maintain all those things that those other people work on*. That is an exciting aspect to this because those folks need to have some love too, and so you are a wonderful representative of that. Vanessa, did you want to say anything else about the differences that you see in what you have developed from other DevSecOps adoption frameworks that you have seen?

Vanessa: Well, I guess just to piggyback off of Lyndsi's point that, from the infrastructure perspective, you can oftentimes find resources as far as—I should reiterate that this paper was built off of our experience going from the ground up. So racking and stacking servers up to building the infrastructure to support the cloud, and then maintaining the actual cloud environment. The resources for infrastructure teams are oftentimes disjointed, so there is not a flow of starting at zero, how can you get to your endpoint, or even understanding truly what your endpoint should look like. Not just kind of this fuzzy dream, but you want to be sure that you've got the necessary…you've got the walls, you've got the foundation to keep this structure running indefinitely.

Suzanne: In our environment, in the DoD, a lot of times, as you were saying earlier, if you are on commercial and you want to hook into AWS, you want to hook into Azure, it is not a very long lead time. Somebody has to say yes and provide the money, but then it's a very easy, typically, connection point from an infrastructure viewpoint. That is not the way it is in the DoD. We have got lots of long-lead items. *Does this thing have an ATO [authorization to operate]?* So I was really pleased to see you actually addressing some of those things that people in the commercial marketplace are going, *Why would you need to do this*? You know?

With that, let's go into what is in the framework. There are six phases in your framework. Vanessa, do you want to take us through the first couple, and then we'll let Lyndsi talk us through some others?

Vanessa: The first framework, we labeled as Stage 0, so it is ensuring that you have the resources deployed for your team that will support building out the next phases. You have a version-control system in place, you have monitoring in place, or at the bare minimum, logging across the existing infrastructure systems. So that you will know, or at least you will be able to easily find out when an issue occurs, and possibly what you may need to do to fix it. So version-control systems, wikis, things that you deploy, and will likely have to maintain as well, but that will take you to the next stage.
After you build that foundation, then we step up to the normalization phase. So we have deployed databases X number of times. Well, now we should know, *You know what? We're going to start documenting how this deployment occurred.* This is especially important from a DevOps perspective but also in the government realm, you are oftentimes working with teams that have varying levels of skillset. I may have deployed a database 100 times, but we need to ensure that across our team, we have a baseline level of skill. So I need to be able to document it, and everybody needs to normalize on how we do that. *We deploy databases in this way. We are going to set up our identity-management system in this way.* It is documented, especially because how you deploy these different types of systems may require different configuration

steps, depending on if you are going against bare metal versus a virtualized environment. Documenting it, those are critical pieces across the normalization, and the team agreeing to, *We are going to move towards using this same system, this same standard procedures, these same standard operating procedures across the board, or at least get to the point where we start building up that knowledge set.*

Suzanne: That normalization would also include, *What are the variations that we need to account for?*

Vanessa: Right.

Suzanne: Because I also know that in some of our environments, how I deploy to a particular classification level has some additional constraints and steps than how I deploy to something that is less classified. So, normalization helps us to see what variation we need to have, and then standardize along those paths. OK, so, Vanessa, you just led us through stages 0 and 1. The next stage is standardization. Lyndsi, why don't you take us through standardization and what comes after?

Lyndsi: Absolutely, yes. Stage 2 is called *standardization*, and it is really a natural following to the normalization phase. As Vanessa mentioned, in normalization, we are spending time figuring out the best ways to do things. In standardization, we are really working hard to determine what those standard operating procedures that our team is going to use are going to be. For example, when an event occurs, when we need to deploy a new database system or we need to deploy a new web server, we are going to have a standard operating procedure that describes exactly what that deployment needs to look like, which means that anybody on our team will be able to do it, because it is well documented and it is well understood.

We are also going to lean really heavily into configuration management, and we are going to use configuration management everywhere that we can. One of the end goals of that might be, *We have deployed a web server. Now we want to make sure it is configured to our standard web-server configuration that we know is functional. So we will also use configuration management to make sure that all of our web servers are configured similarly, if not exactly the same. x*
We are also going to start taking a look at the different systems and applications that we need to support. So, for example, if we have one development team that is using a MySQL database on the backend of their application and another development team that is using PostgreSQL on the

backend of their application, we might get everyone together and say*, Hey, we really don't want to be supporting multiple database systems. Let's become experts on one of those systems, and let's develop all of our applications with that system on the backend.* That really has the nice impact of reducing the overall overhead of just maintaining all these different types of software.

Suzanne: Not to mention licensing costs because that is one of the other things. Government organizations typically are constrained in terms of how much budget they have and proliferating different applications is actually one of the things that gets the budgets all out of whack.

Lyndsi: Oh, absolutely. I will jump to talking about Stage 3 now, which is continuous integration and continuous testing. What this stage looks like is we are going to encourage or require our teams to be frequently using all of those tools and things that we have set up in the stages beforehand. We are going to ensure that our developers are pushing their code very frequently into that source-code repository. We are going to make sure that our system administrators are using the config management system, which should also be managed in a source-code repository. And we are going to start running tests automatically whenever we push that code into those repositories to make sure, number one, that we haven't done something silly like missed a semicolon or missed a tab in a YAML file and created a syntax error that is going to take down one of our servers. We are also going to start testing to make sure that the changes we made actually have the effect that we intended.

For example, if I want to make a change to the configuration of a web server to listen on some different ports for other applications to connect, I can create some automated tests to make sure that when I've pushed that change, the web server is actually accessible on those ports that I intended it to be. This has the really cool result of preventing problems before they become problems. Configuration issues can be discovered and found before they ever see the light of day in any production systems, which is a really good place to be from the perspective of a system administrator.

Suzanne: OK. All right, and so the last two stages, Vanessa, why don't you take those two which would be 4 and 5?

Vanessa: So, Stage 4, infrastructure as code (IaC) and continuous delivery [CD] so…
Suzanne: One of my favorite topics.

Vanessa: We started at the normalization stage and moved up to standardization when we talked about scripting, and we talked about creating SOPs [standard operating procedures] for implementation, configuration, and maintenance. Now, with the advent of infrastructure as code, which is relatively new to the industry, we can start having the same power that developers have as far as managing their code.
Lyndsi talked about configuration management. Now, we can use infrastructure as code to solidify configuration management or the configuration of different systems, different applications, so that when I push out this particular YAML file, the infrastructure as code system controls that. What is in version control is what will be pushed to the specified systems, and if anybody makes a manual change to those files or that system, the infrastructure as code software will reverse those changes. We know for sure the state of a system. We also know if someone has attempted to modify the state of the system, in addition to that being automatically rolled back.
With continuous delivery and building on the power of infrastructure as code, that means as we need to make changes to our web server, to various applications, it is something that we know we can do consistently, so we make the change in version control. That change gets pushed out to our preproduction system, validated, and then it gets pushed out to production. And this happens regularly and consistently, and we can say with a level of confidence what is the state of our system, even as changes may need to be rolled out or rolled back. The power of infrastructure as code really changes the game from a system engineering/system administrative perspective.
Being able to build on continuous testing, and being able to say, *Oh, that port change that we made on a database, that is not what we want because it is now affecting these other applications, so we need to roll back and reconfigure or rethink how we do this.*

Suzanne: You haven't said the word *security*, but that is actually one of my favorite aspects of infrastructure as code. You sort of alluded to it, that I have confidence that even if somebody tries to change something manually which is inappropriate, my system is going to say, *Oh, no, no, no. This is our standard. This is what's going.* if somebody complains about it, they can complain to the system administrator, and we can go through getting that changed in our configuration package, but we are not going to see all of these breaches that often are done unintentionally because somebody is inconvenienced by some aspect of how we are running the pipeline, and they open up a port that is going to allow vulnerabilities and other nasty things that we don't want to happen. I want to double emphasize, this is a really important step to take.

Vanessa: Well, and the tricky part for us as we were building this out is that a lot of these stages kind of blend in. You mentioned security. Then on Stage 5, we touch on automated security, and it is exactly what you say. Now that we have got infrastructure as code, then we can not only ensure that that system is at the state that we have defined in version control, we can also say, *Hey, it looks like user X attempted to modify this system in a way that hasn't been approved by all of our stakeholders, specifically our security team. We need to revert that change, and we need to kind of escalate that, OK, why did admin X do this? and document, log, alert the necessary people.*

Automated security is that critical component that eases the burden of systems admins as far as either attaining an ATO or an authority to operate, or maintaining it. In conjunction with automated security is compliance as code. So for systems that achieve an ATO that you have an enormous amount of rules that need to be put in place that you need to be able to say definitively, *My system does X, just like these rules specified. These rules are codified, not just in our version-control system, but now in our infrastructure as code.* So I can work and bring on the security team and make their job easier too. Their list of hundreds of checkboxes that need to be confirmed, it is not necessarily them going through and checking. It is them reviewing our infrastructure as code, our compliance that is written in its code, and saying, *OK, we agree to these security settings. I can see here that it is implemented, and I know that they cannot change without the necessary people, perhaps the auditor, being informed of these changes.* So everybody is in a happier state that is required to maintain the system.

Suzanne: Yes, and you talked about continuous ATOs. Explain to those that may not understand what is the difference between an ATO and a continuous ATO, so they can understand the impact of this.

Vanessa: The difference between an ATO and a continuous ATO: with an ATO, you usually were specified this authority to operate within a given timeframe, so maybe a year, maybe three years. As your expiration time approached, then it was time for a re-review. So you had to produce and/or provide access to all of the documentation that proved that your *T*'s were crossed, your *I*'s were dotted, as far as the security rules that were put in place for your system—security controls, your risk-management framework, all of those things were still in place according to the rules that were specified in your ATO.

With the advent of automated security and compliance as code, we can now transition to a continuous authority to operate. So periodically with your audit logs and your configuration files, you can package those up and submit them oftentimes through a particular security application, but submit them to your authorizing official, to that office, so that they can review and say, *They are still in good standing. They are still in good standing. They are still in good standing.* So it is not this big deal of *OK, everybody, no one touch the system. We are going for our security review to reauthorize our ATO, so everybody freeze.* It is, *As long as we are following the rules as were specified in our ATO, we will be in good standing, no surprises on our end, and no surprises to our auditors, to our security officials.*

Suzanne: Yes, so this is one of the things that makes this whole adoption framework useful because I think it helps people to see that you can't start there. You can't start at automated security. You have got to go through normalization. You may go through some of these stages faster than some others, but everybody wants automated security, but you have to do the things that will get you to the place where that is actually going to be feasible, and you can actually achieve that continuous ATO status.

I want to switch gears a little bit. The transcript will have actual reference to the paper, so you can read through the stages in more detail, but I want to talk a little bit about what is your experience in using this framework so far out in the real world? Have you piloted this with any agencies or organizations, and what have you learned from that? And Lyndsi, do you want to start with that, and then get Vanessa's take too?

Lyndsi: I would love to answer that question. Before we do that, can we talk about Stage 6?

Suzanne: Of course, Lyndsi. Absolutely. My apologies. Go ahead.

Lyndsi: Great. Stage 6 is really, I think, the pinnacle of this project because by the time you get to Stage 6, which is automated compliance reporting and vulnerability remediation, you have already automated probably everything you can think to automate, and now you are just automating your reporting. When an auditor is coming to visit, instead of your system administrators putting together reports and documents and all of the artifacts that that auditor is going to need to review, those are now being automatically generated for you, which is just so convenient. In addition to that, being able to automatically remediate vulnerabilities as quickly as possible is just going to improve your security posture, because you don't have to worry about people remembering, *Oh, shoot. We saw that vulnerability disclosure. Let's get somebody to patch that.* You can make the computers do it for you, and by the time the computers are doing all of your hard work for you, I think you are in a really good spot.

Suzanne: OK.

Vanessa: I would like to get a chance to expand on Lyndsi's explanation. All the things that she said, absolutely, the beauty of the build-up gets you to this place where you are ready to be best friends with your auditors and with your authorizing officials. But within the government environment, in order to even attempt to attain an ATO, the government requires that you deploy certain security applications. Those security applications will constantly scan all of your systems that are connected to that government network. They are going to not necessarily flash red, but they are automatically generating reports, reports upon reports. Those scans have to be done within a periodic timeframe. Because you have set up your infrastructure as code, all of your compliance as code, all of that is version controlled, those scans usually come off without a hitch, because you build up your system in this right way. Now, automatic vulnerability remediation as a part of this process, you are now automatically rolling out patches to your system. Then, because we have got infrastructure as code and we also have continuous testing, if that new patch breaks something, we can roll back, we can figure out what is wrong, we can start again. So many vulnerabilities are about not being able to keep your systems up to date. Because if I am doing this manually, and I have got 30 systems, then that means I have to take the next week to roll out these updates, patch, rollback, fix. Now that so much of that is automated, then that means that I can say with confidence that, *OK, those patches are going to roll out. We are going to review and be sure that everything is good. Now we scan again. We are good.* That zero-day that just came out, we will have that fixed by this timeframe. We not only have confidence as system maintainers, we can provide that level of confidence to all the stakeholders throughout our organization or within our DevOps team. It finally makes doing system administration within the government not an enjoyable process, but not as painful as it has been in the past.

Suzanne: That is actually a really good point that all of these things are leading up to making system administration, system maintenance, something that can become routine as opposed to something where we are jumping through hoops all the time to get the latest thing taken care of. That makes everybody's life less…Lower blood pressure, all those kinds of things. I can take a vacation for a week, and not worry about everything. All of those are good things, and those are some of the things that automation like this can get us.

Let's get back to where have you seen it work? Where have you piloted? And what did you learn in those early pilots with this framework? Whichever one of you wants to take that, go ahead.

Vanessa: Before we got to the point of seeing it work, deploying or setting up this framework was so critical to us because we had this trial by fire. We had, *How do you set up this development environment in this brand-new building? How do you set up this development environment when the necessary parts haven't arrived? You got this system, but they didn't deliver the right switches.* It is trial by fire in that we tried, and especially when you are doing deployments on bare metal, you try and that fails, the amount of effort it takes to redo that when you can't revert to a snapshot, when you can't delete that cloud VM and just redeploy. Not only were we deploying in a brand new building on bare metal, then we had to build up the virtual environment that was going to then support the cloud environment. We had to do this for an unclass system, a TS [top secret] system, and a SAP [special access program] system. It is kind of the most complicated setup that you could imagine, and it was painful to try to do without the framework.

For us, it is not necessarily, *Oh, we have been able to successfully deploy this in X.* We have been through the pain of having so many failures because we did not have this framework in place, but now, now that I am working in an environment where someone else it is responsibility is to set up and maintain the DevNet [development network similar to a development environment], but we are still responsible for setting up that infrastructure on top of it. In a lot of government places, maybe you've got a DevNet, but you don't have the DevOps component. I have got something like an Atlassian suite or GitLab in order to be able to set up CI/CD, but I still need to build out that infrastructure to do it well.

Even though we at SEI have been talking about DevOps for over a decade now, it is still fairly new within government organizations. That means that what a DevOps team may look like in support of a government contract may mean people with varying skill sets. So I won't necessarily have 10 people that know how to do this thing really well. I may have someone who is brand new to this arena, and so I still need to go through that process of creating my wiki. I still need to say, *Do X exactly this way, exactly in this way, exactly in this way.* It is important to be able to build up the skill set of your team, but also to still build out that standardization of the infrastructure to support development long term. That combined with the fact that because we don't have that AWS environment, we are building up the development environment where the code is being developed, we need to do the same thing in a test environment. We need to do the same thing again in a QA environment across various development teams. Again, that is something that would be impossible to do without having this framework in place. Sometimes you don't necessarily have to start at Stage 0 if you are not racking and stacking. But you need to have familiarity with some of the pain of doing Stage 0 wrong.

Suzanne: Yes. I hear you. Lyndsi, do you want to add anything, any experiences of piloting this that you want to add into this?

Lyndsi: I don't really have much to add. Vanessa really hit the nail on the head. This is really the culmination of a lot of trial and error and learning from some difficult situations and coming on the other side and saying, *You know, we probably could have done that better, and this is how we think a better way to do it would have been.*

Suzanne: OK, that is fair. This is an interesting framework. It is helping people to adopt something new, DevSecOps, but there is also the aspect of, you need to transition this framework itself out into use within the government, especially the DevSecOps community. What kinds of things are available for people that say, *You know what? They are right. I've been through the pain and I didn't go through what they did, and I didn't develop a framework like this. How do I get access to this? How do I get resources to help me do this? Are there resources available beyond the blog post, that will help me to actually do this?* So where are you in that transition of your own framework?

Vanessa: I am blanking because, a couple of different reasons. As you were saying that, I was thinking my problem is I can't remember the exact name of the office. I want to say, the top-level information officer.

Suzanne: The CIO for the DoD, is that what you are thinking?

Vanessa: Yes. They recently deployed, not deployed, released a Dev[Sec]Ops standardization roadmap for the DoD. Especially for when you are working within government organizations that you want to start at the top. Our blog post is kind of starting at the bottom, but you also need to reach up to the top. What is the roadmap that top-level government officials are defining for how they expect this to work across government organizations, across the DoD specifically. Because having both that low-level piece, but also that top level, will help guide you to, *These are the things that I am going to need to do to ensure that I can reach ATO*. Because one of the critical problems, at least that we ran into, was that once we reached the stage of attempting to get to ATO, there was a dearth of information there.

Now that the top level of our government is adopting DevOps and enforcing the adoption of DevOps across government organizations, you need both our blog post where you are starting from the bottom, and then the books from some of the commercial agencies. Google puts out a lot of books as far as how they implemented DevOps, but also that information from government agencies we define this DevOps roadmap, and this is how we see organizations moving towards this in the future.

Lyndsi: Vanessa, it is really interesting that you brought that up, because I was having a conversation with another colleague earlier this week who was talking about how there is a really big difference in perspective from C-level people to people who are actually doing the [hands on keyboard] work. It is so important to be able to bridge that gap in order to have a successful DevOps or DevSecOps deployment. Both ends of that spectrum need to do what they can do to understand the other's perspective in order to make this type of thing successful.

Suzanne: OK. So, what's next? What are you working on either in relationship to this? Are you continuing to work on evolving the framework, adding toolkits, other kinds of support mechanisms or are you switching into some other research? What is on the table for both of you? Lyndsi, I will start with you on that one.

Lyndsi: Well, what is next for me is back to the grind, I would say, of just continuing to maintain and operate those computing environments that I have been working on for a number of years and trying to put some of these higher level things into my own practice, getting some extra stuff automated so that can free up more of my time from doing system administration to doing some other different things and learning some new things.

Suzanne: All right, and Vanessa, how about you? What are you doing next?

Vanessa: My answer will be similar. It is back to the grind, but I think if we are thinking about the next evolution of the framework, it is, *Now that you have got this cloud environment deployed. However, you still don't have all of the resources of a commercial environment*. The next would be, OK, how can you efficiently do CI/CD with these limited resources? What is your best way forward? What open-source utilities can be provided on the one end. Then going to the top level, helping the government to refine that DevOps roadmap, so that they can say, *You know what? We have contracted with these different entities to set up DevNets in these secured environments, but we need to ensure that these DevNets have at least this, this, and this in support of our infrastructure teams and in support of our developers.*

So it is continuing to both do the work, hands on keyboard, but also try and have that larger view of what can be improved, what needs to be standardized or documented so that this isn't as painful for the next group coming behind us, especially because the government, especially within the secure realm, has such a retention problem. A big part of that is that doing this work is so difficult. We want to relieve some of that burden through standardization, through frameworks, through refining and providing more information to say, *You know what? This is how you make this better.*

Suzanne: Yes, make the things that should be routine actually be routine. Go ahead, Lyndsi.

Lyndsi: If I can add one more thing to that, Vanessa, I think it is important also to consider that once you have reached Stage 6 of the framework and you've accomplished everything, I don't think that you get to say like, *We are done. We did it.* Your environments are going to be constantly evolving as vendors release new software, as hardware vendors release new versions of upgraded pieces of hardware. So there are always going to be new things to integrate and automate. I think you are just going to be constantly working through the phases as your environment evolves.

Suzanne: That is a really good point.

Vanessa: To piggyback off of you again, it reminds me too that for all of this, we need to keep...An organization like SEI, we need to keep the conversation going with authorizing officials, whether it is within CYBERCOM, or other government entities to say, *You know what? Right now, the way that you are doing ATO, it is getting the job done, but it is painful in these ways. Here are things* *that we have learned and that our associates have learned, that can make their lives easier, as well as yours, right?* So it is keeping the conversation going, not just for our immediate technical audience, but for the government officials that are making the rules and adjusting them so that we don't have this clash of, *OK technically we are good, but these rules that are coming down don't quite make sense and/or are making this much more difficult than it has to be*.

Suzanne: Right, and we talk about delivering at the speed of relevance and all the things that can be roadblocks to that. This automation approach is one of the ways that we can actually reduce that roadblock and help the AOs [authorizing officials] to actually work at the speed of relevance. I think that is the selling point for them on these kinds of things. I really want to thank both of you for talking with us today. How to adopt is a topic that is always something that I hear when anytime I am in conversations with people about DevSecOps. Sometimes it is from the dev perspective. Sometimes it is from the IT sysadmin perspective. Giving that perspective a chance to see where they fit into all this I think is really, really important. I want to thank you both for that. I do want to remind people that we are going to include links in the transcripts to resources we have mentioned, as well as basic resources on DevSecOps for people that haven't run into that before, of course including your recent blog post. We will have plenty of things for people to look at. I do want to remind our audience that our podcasts are available on SoundCloud, Stitcher, Apple, Google, and of course, our SEI YouTube channel. If you like what you see and hear, you are welcome to give us a thumbs up. We are not the biggest channel on the block, but you get to see lots of important things being talked about in terms of our government world. I do want to thank everyone for joining us today. Thank you again to Lyndsi and Vanessa.