The Role of the CISO in Developing More Secure Software
Transcript

Part 1: Gain the Authority; Build the Business Case; Start Measuring

Julia Allen: Welcome to CERT's Podcast Series: Security for Business Leaders. The CERT program is part of the Software Engineering Institute, a federally-funded research and development center at Carnegie Mellon University in Pittsburgh, Pennsylvania. You can find out more about us at cert.org. Shownotes for today's conversation are available at the podcast website.

My name is Julia Allen. I'm a senior researcher at CERT, working on operational resilience and software assurance.

For our listeners' information, we have posted several podcasts that discuss what development teams and project managers need to do to build more secure software.

And today I'm pleased to welcome Pravir Chandra, Director of Strategic Services for Fortify Software. And we'll be talking about the leadership actions, in particular, that are necessary to get software security off the ground, and the critical role of metrics; that oft and often pursued subject that we're all grappling with. So our discussion, in particular, is based on a CISO -- Chief Information Security Officer -- checklist that Pravir and his colleges at Fortify have made available, as well as Pravir's leadership in developing the Software Assurance Maturity Model.

So welcome Pravir; glad to have you with us today.

Pravir Chandra: Thank you.

Julia Allen: So the first item on the checklist, that you've so kindly made available, addresses CISOs, in particular, having the authority to enforce standards -- something that really helps get the ball rolling on software security. So how do leaders go about getting that authority and making it stick?

Pravir Chandra: So I think the best way to go about getting authority is to really just demonstrate the business case for software security. And there's typical ways you go about doing that. I think one of the most obvious ways that people think of first is to try to basically just address the impacts; what are the potential business impacts from software security-related risks? So you have the known risks, and even some of the unknown risks; which for a lot of organizations that are getting started, that's probably the bigger slice of the pie. And if there are unknown risks, you can do things like just conduct an assessment on a critical application and use that to light a fire under the organization to try to at least get people thinking about the impacts of software security problems.

I think another way to go about building that business case is to also, instead of just focusing on the negative, the negative impacts of software security, is to also focus on the benefits. So things in terms of reducing risk obviously. But also I think there's been a lot of customers that I've worked with that have done a lot of interesting work in demonstrating cost savings because of software security programs, which I think is another angle to go at it from.

Julia Allen: Okay, so when you're talking about this business case -- that's always been an interesting idea for me too -- is when you stack up software security and standards and the authority to enforce standards and the things that developers and managers have to do to develop

more secure software and make investment in that, how -- when you're making the business case -- how you compare and contrast investing in that set of endeavors with maybe other technological investments? So how do you put software security on the same playing field as other technology investments?

Pravir Chandra: Well there's a couple of ways because it's often challenging. I think a lot of times you do struggle with that.

I think the first thing to do is to actually have a good plan going into it. So instead of saying, "We need to start investing in software security," and just start acquiring products or tools in almost a haphazard, reactionary way, I think it's important to have kind of a balanced plan. And, in fact, that's why we built out the Software Assurance Maturity Model was to help people think about things in a more methodical way, in terms of what are the different activities that are really required. And to really start considering the risk management side of the scale, which is how much do we really need to be investing in software security? And where are we currently weakest, and what -- the easy wins on improvement? And I think if you can do that, you can actually pretty successfully boil it down to concrete budgets, to concrete plans for improvement that are iterative in nature. And then you can compare them against other iterative, project-related tasks that are on the development plate. So things like making improvements and doing feature upgrades and things of that nature.

Julia Allen: Okay, okay. So what are -- we started off talking about standards and authority a little bit. But are there some ways that CISOs can ensure that software security standards are used across all development projects -- recognizing that all development projects aren't equal and maybe you won't need to do the same level of practice adoption as you might do for something more critical. So how can the CISO get that momentum moving in the right direction?

Pravir Chandra: Yes, this is another really good question that I think a lot of folks out there that are building software security programs have to figure out how to crack this nut. I think that the easiest way to go about it is to, well first maintain an application inventory. It's actually kind of funny because a lot of the organizations that I work with, on Day 1, when they're starting down the road of building more secure software, they don't even know how many applications they have or what's even out there. So getting a handle on that first is maybe Step 0, I should say. And then really it just comes down to establishing some consistent risk rating criteria for applications. So, I mean, this basically -- I mean, it could be just as easy as having a couple of factors like: is it internet facing? Is it internal only? Is it something that's a boxed, downloadable piece of software or something like that? And just having those risk rating criteria that help you effectively establish a few different buckets of things that you care about. So a high-risk bucket, a medium-risk bucket, a low-risk bucket. And then actually dividing and conquering.

So basically establishing some rules for the high-risk bucket, and maybe having little less stringent rules for the medium-risk bucket, and maybe you just ignore the low-risk bucket to start out with. And in that way you at least are slicing things horizontally, across the organization, and trying to apply some semblance of a consistent plan there.

Another way you can go about slicing it -- because a lot of organizations are quite big and have been grown by acquisition and so forth -- is to try to slicing it based on the line of business. So if your company is very vertical-oriented -- so you have different businesses that are operating almost independently -- then you can really just try to establish a different set of criteria or a different set of rules for each line of business. Now, of course, you have to make sure that they're all consistent with one another and that they're being administered fairly and that you're

benchmarking appropriately. But I think that's sort of the general strategies that work -- either slicing it by line of business, or slicing it by risk rating, or some combination of those two.

Julia Allen: Okay, good advice, good advice. So let's talk about metrics a little bit; that elusive subject that we all hope for one day, but are still struggling to come up with the right definition. So when it comes to software security, how can metrics be used to report compliance of standards; maybe used as a forcing function for issue resolution; maybe to heighten awareness? How have you seen metrics being used to help move this forward?

Pravir Chandra: Metrics can get used in a lot of different ways. But I think that the most effective way to do it is to really establish some baseline for how you're actually collecting data, and make sure that it's consistent.

I've worked in a lot of organizations that were using poor metrics and I think they tend to hurt a lot more than they help. Because you end up with different parts of your organization that are just gaming the metrics themselves as opposed to working towards the actual goals of improving software security. So I think really spending a little bit more time up front focusing on what's a good metric to actually collect, and does it actually reflect what you care about? And that's a good way to get things started.

Now as far as rolling them out, you could do it in a lot of different ways. I mean, I think that some organizations that are a little bit more competitive in nature have tried doing things like making the metrics public, so that all of a sudden one manager can see their peer's metrics. And then all of a sudden it becomes a bit of a pride thing and a bit of a competition to make sure that you have the best metrics in your section of the development organization, or something like that.

And I think the other thing that's kind of required is that you may have some very detailed metrics on the ground, so to speak, but when they roll up you have to have some way of aggregating those metrics to present to upper management. I mean, I think really you want to have people paying attention to software security metrics all the way through the management chain. So having some way of rolling those up or aggregating metrics for a given group is important as well.

Julia Allen: Okay, so let's talk about a few examples. Do you have some of your favorites or some key metrics that you've seen really serve the purposes that you described?

Pravir Chandra: Well I think that they vary a little bit. So I think some of the simplest ways to get started are just things like vulnerability hot- lists, where you say, "Okay, this set of cross-site scripting issues, for instance, or SQL injection issues" or you just pick one or two or some small number of particular vulnerabilities. And you say, "That's a metric that we're going to track." And you basically make sure that that number is being collected Consistently through some reporting process or through some sort of governance automation technology, and you use that as a simple way.

Other metrics that I've seen are a little bit more in accordance to whether or not you're following the process as outlined or as established in that organization. So are you producing design review documents? Are you filling out the correct forms and so forth? I think that one of the interesting things about it is that they tend to correspond to the culture of an organization. So if you're a very command-and-control oriented company, then those metrics like get pushed down from the top, and so forth. So it's a little bit variable. But I think those are probably the simplest ways to get started.

Part 2: Security Responsibilities that Stick; CISOs that Add Value

Julia Allen: Okay. One of the real interesting items that I saw in your checklist talked about making sure that your project guidance leaves no room for anyone to deny that they understand their security requirements and their responsibilities as they're developing a software. So how does that really show up in action, for both developers and managers?

Pravir Chandra: Well that's a real challenging one. I mean, I think this sort of -- the whole idea of leaving room is that ultimately if you want something to get done in an organization, you have to someone that's responsible for it. And usually that rolls up to somebody like a CISO or someone that's running an application security center.

When it comes to actually pushing it down the line, it really comes down to, I think most importantly, having education programs in place. So making sure that you're arming folks with the knowledge that they need so that they even know what their responsibilities are when it comes to actually developing secure software. So I think that usually tends to start with the developers themselves and usually also architects. If you have software architects that are there, usually some of the more senior folks, you start with them and make sure that they understand what the impacts are. And I think a close second is actually making sure that managers have some awareness of what they're expected to be producing.

And then as time goes on, you try to roll that out to other folks that are involved in the development process; so people like project managers, people that are specifying requirements, such as business owners or business analysts. And even in a lot of cases out to folks that are doing QA and testing. Basically make sure that everyone understands it. And I think the easiest way to do that, once you have a training program established, is to try to roll it into people's individual MBOs, or their actual requirements that they have, or their goals, I guess, as employees for getting bonused out and things like that. And then carrying it even further up the chain, so that you actually put it as part of their job description that they need to be able to develop secure software and build secure interfaces and so forth. And you can - - taking it a step further and hiring folks that have those characteristics or have those skills.

Julia Allen: Have you -- I know we're at the beginning of this journey of getting secure development practices into the lifecycle. But have you actually seen instances in the clients and customers you've worked with, where these needs are showing up in job descriptions and being measured in performance reviews? Or are we just getting to that point?

Pravir Chandra: I think there's a few companies that are rather mature -- so they've been thinking about software security for awhile -- that have definitely gotten there. I think that it's going to take a little bit longer for us before it becomes sort of a mainstream thing that we see popping up. Because it's really not the first -- it's not the right place to start, right? I mean, you really have to start with an educational program and then move it further towards, into the job descriptions and so forth, over time. So I think probably in the next five years or so we'll see it popping up more prevalently. Right now I see it most often in the financial sector but don't really see it too often anywhere else.

Julia Allen: Okay, good. So the developers and managers of projects, they've got a lot of things on their plate. And here comes the CISO, the CISO, with yet more requirements. So how can the CISO be viewed as a value-add to the development process -- not a burden, if you will?

Pravir Chandra: Yes, yes. I think the most important factor that I've seen for CISOs that are very successful at rolling out software security programs is that they demonstrate that they understand what development is all about -- in terms of the fact that development groups are under

tremendous budget and timeline pressures to actually get things out the door, under lesser and lesser resources as each day goes by and so forth. So I think they have to demonstrate the fact that when they're proposing security programs and things of that nature, that the immediate reaction from a lot of folks that are on the development side of the house is going to be, "Well where is the time and budget to actually do this?" So they really have to understand that and take that into account when they're proposing strategies and so forth.

They also have to demonstrate understanding, that it's not like -- a development group doesn't operate like a bunch of line workers in a widget manufacturing plant. It's a fundamentally different process, building software. So you can't really go in and impose Draconian, strict rules and things of that nature. Usually it's more about consensus building and fostering good working relationships with the dev group, so that they view you as an asset rather than the guy that comes in with all the new rules that we have to follow. So--

Julia Allen: And I would think that on occasion, depending on the nature of the personal dynamic and the relationship, they could maybe even start to see the CISO as a champion on their behalf, right?

Pravir Chandra: Yes, absolutely. And I think the most effective CISOs, that's exactly what they do, is that they basically get in there and minimize things like process re- engineering and continual change and things like that. They minimize that. So they try to dovetail with the existing processes that dev already has in place and just add their team of security folks into the appropriate checkpoints.

They try to do things like listen to feedback. So when the development team is saying, "Okay, this step is really not working," they try to come up with some creative solutions to that. And make sure that basically there is kind of full feedback cycle there so that when people have input on what's working and what isn't, that they're taking it into account and actually responding.

You really have to have a plan that fits into the culture of the company. So if you're a very command-and-control-oriented company, then maybe the best way to do it is through mandates and actually drilling down policies and guidelines and standards and so forth. If you're more of a very process-oriented, very regimented, documentation, recorded guidelines and things like that -- it's a little bit more bureaucratic in nature -- then maybe the best way to roll out your process is through having forms that people fill out or having documentation that they've produced that lets people know. If you're more of a lean and mean organization -- that you have champions that are the ones that run each product group and they're the ones that have been there forever, the gray beards, so to speak -- then maybe getting them as allies and using that same model for rolling out software security is the right way to do it.

So coming up with a solution that actually culturally fits your organization is important. Because if you're a lean and mean organization and you come in with a documented, rigorous process-oriented approach, it's more likely to be met with I guess with some negative reaction.

Julia Allen: Sure. And what about the critical point where the CISO has the authority to halt a release at a key milestone, if particular requirements aren't met? How have you see that handled with skill and diplomacy?

Pravir Chandra: Well it varies a lot. There's a lot of CISOs in organizations that don't actually do this. And there's a lot of CISOs in organizations that I've worked with that do this extremely rigidly and rigorously.

I think it's tough because it's not always the right answer to stop a piece of software from going out the Door, which is, as a security guy, kind of makes me feel a little creepy to say that. But the idea is that, I mean, there really is a balance to it. You have to pick your battles. So if it's a critical piece of software that has significant business impact in that piece of software rolling out the door on time, then potentially cutting it an exception and so forth is the right thing to do.

So I mean I think the best way to go about being gatekeepers is -- I mentioned it before a little bit -- is to use the existing process. Every development process has some choke points or tollgates, so to speak, that are already there, just based on the nature of software development. So don't try to invent new ones. Just go and dovetail your security requirements into those same processes.

And make sure, above all at a minimum, that you have a very well understood exception process, so that you can actually document and track exceptions. Not to mention -- going back to metrics as we were talking about before -- that's another very important metric to have, I think, is to actually track exceptions to your -- anytime you cut an exception to a project, just to make sure that's being tracked. And that you have some notion of exactly what percentage of exceptions are going out the door.

Julia Allen: Well and I also heard you do a full circle back to risk management, when you talked about critical applications that have high business impact, picking your battles. To me those statements imply that you also have a risk perspective when you're making a go/no-go or alternative decision, right?

Pravir Chandra: Yes, absolutely, absolutely. And I think that's one of the -- I think the most effective CISOs have a very tangible notion of risk. And have really good methods of determining what's riskier than -- is a riskier than b? And a lot of times that boils down to value judgment. Sometimes you can get it down to very concrete decisions where you can collect some facts and make those decisions. But other times it's based on gut feel based on the organization and the kind of business that you're in.

Part 3: When a CISO Should Walk Away; Handling Outsourced Software

Julia Allen: Great. So let's say as a CISO you've done all the right things, you've built all the right bridges, you have a good relationship with the development team. And even with the best of intentions you're not able to make it all work. So is there ever a time where a CISO should walk away from the responsibility of the secure development process?

Pravir Chandra: So I will say that usually when a CISO has a good relationship with the dev team, they figure out some way of making it work. I think that's by far the majority case.

But there's definitely -- I mean, I could definitely see the possibility where it wouldn't work. Now to kind of preface it, I will say I've never seen a dev organization that was intentionally trying to produce something that was a bad product. So usually, like I said before, you can make it work. Because really it's just a matter of framing the problem correctly because it's a competing priority.

I think that in situations where it just can't work, are usually just management problems. So where you as a CISO have the responsibility to actually sort out and own the software security problem but you don't have the authority to change anything. And that you can't, through grassroots or other means, actually get the development organization to cooperate or to try to work with you at all. Now it could be a matter of just picking a new technique or trying something a bit different. But in some cases it could be that it's just not going to work, because you just don't have the right setup from a management perspective.

Julia Allen: Okay yes, and I guess you just really -- obviously that's a tough decision to make. But there are times when it might be the right decision, to get the right level of attention, right?

Pravir Chandra: Yes, yes, absolutely. Well and I think that the other thing that's kind of interesting is that in some companies they just don't value security. So even as a CISO coming in, you may have a lot more experience and a lot more of an idea of what that company should be doing to make sure that they're building software more securely. But everywhere else in the company you're just sort of met with the blank stare of, "Well why is this even important?" And it's a big challenge. Even if you do have a really well-framed business case, it's sometimes tough to get an organization that's been operating at a status quo to change that.

Julia Allen: Okay. Well in today's business climate, as you well know, so many times we're acquiring or picking up various pieces of software, kind of mashing things up, putting different services together. So how does what we've talked about today shift or not, when it comes to either outsourced software or software services in the cloud? What is the CISO's role there when it comes to making sure that all works in a secure way?

Pravir Chandra: Yes. Well I think that -- well it's a lot more well known, I will say, about the right way to go about handling outsourced software. I think cloud is still a little bit ephemeral because it's just a little bit newer.

When it comes to outsourced software, I think a couple of techniques that a CISO can use are different than just setting standards. Because you're not really as intimately involved in the development process and you don't have that level of control that you might have with an in-house development team. So it really just shifts over to being pretty strictly verification activities. So you're doing testing, you're doing code reviews, things of that nature. And really concentrating on those, and trying to, as much as possible, roll those requirements into your agreements with your outsourced vendors.

So anyone that's writing software for you, just have some upfront notion that they're going to deliver something that's secure, basically just explicitly documenting that as an expectation. As well as, if you wanted to get more specific documenting, the kind of testing procedures that you're going to use to verify that. So if you're going to conduct a code review; or you're going to have a third-party conduct a code review; or you're going to do some security testing of some sort, and just laying that out explicitly.

Julia Allen: Do you find that in the organizations that you've worked with, when they're actually acquiring a piece of software, or they're using maybe open-source or some other type of available software package and integrating it into a critical business application, do you find that there's some fairly rigorous due diligence, in the same way that if they were building the software themselves?

Pravir Chandra: When people are using open-source software and third-party software?

Julia Allen: Right.

Pravir Chandra: I think that's usually not the case that people are very rigorous with it. I think that they probably should be, and I think a lot of companies that are a bit more sophisticated or a bit more mature because they've been thinking about software security for a bit longer, have really focused on that and started paying attention to it quite a bit more. But I think that most

organizations, when you're getting started, you just focus on the stuff that you're working on and the code that you're developing; just because it's the only way to scope down the problem.

Julia Allen: Right, right, and like you said, build up that knowledge base in your staff skill set so that people understand what they're trying to do, right?

Pravir Chandra: Uh-hum, uh-hum. Now in a lot of cases -- some organizations build all of their software through outsourced agreements and things like that. Now those organizations in some cases actually have a bit of a challenge on their hands, because they don't really get that experience learned by trying to do it internally first. And so in a lot of cases it's a bit more of a challenge; mostly just because they lack the experience of the right way to go about determining what's a good goal for software security?

Julia Allen: Well Pravir, this has really been a really nice, tangible, practical approach I think to some tough questions about how you -- how a CISO in particular -- can go about being the right kind of stimulus for software security, secure software development in their organization. Do you have some places where our listeners can pick up a few more good ideas?

Pravir Chandra: Yes, absolutely. So on the Fortify website, there's a section there about CISO guides. And I think there's about eight of them up there now, just focusing on different aspects of software security problems. So those are quick two-pagers or three-pagers that are just checklist oriented to give you some good ideas on how to get started.

The other website that I'll mention is the Open Software Assurance Maturity Model website, which basically has all the details and a lot of the most recent notes about the Software Assurance Maturity Model, which is basically the framework, or a framework, for building out a software security program in a balanced, iterative fashions. So that's all available. And both resources that I just mentioned, they're all freely downloadable PDFs.

And of course the other thing that I'll mention is that I'm always happy to help. So people can contact me directly, if they're interested.

Julia Allen: Yes. In fact I notice with the Open Software Assurance Maturity Model, there's a really nice treatment of metrics in that model. And so that particularly caught my attention to help our listeners with more on that subject.

Pravir Chandra: Yes, yes. We tried to put down a lot of different example metrics. We tried to try to capture basically just some ideas on what you can try to collect. So for anyone looking to get started with metrics, that's a good place to start.

Julia Allen: Well Pravir, I so appreciate your time and good counsel to a tough subject that we're all trying to raise our community's level performance on. So thank you very, very much for your comments today.

Pravir Chandra: Thank you very much for having me. It was a lot of fun.