Title: How to Start a Secure Software Development Program
Transcript

Part 1: The Evolution of Secure Software Development

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.

Show notes for today's conversation are available at the podcast website.

My name is Julia Allen. I'm a senior researcher at CERT, working on security governance and software assurance. Today I'm pleased to welcome Gary McGraw, Chief Technology Officer for Cigital, the editor of Addison-Wesley's Software Security Series, and the author of *Software Security: Building Security In*. We'll be discussing Gary's views on the state of the practice in software security and how to integrate security practices into your software development life cycle (SDLC). So welcome, Gary. Tall order for today.

Gary McGraw: Hi Julia.

Julia Allen: Good to have you.

Gary McGraw: It's good to be here.

Julia Allen: So you've been a leader in software security for many years. And I'd be interested in how you've seen the landscape changing in, say, the last five years or so.

Gary McGraw: Yeah, I got started in software security, really, in the mid '90s because of Java. So Java came out and Ed Felton and I poked a bunch of major holes in it. And then, we started wondering why it was that amazing people, like Guy Steele and Bill Joy, screwed things all up. And if you wanted to do the right thing when it came to building software, where would you turn? And the answer really was there was nothing written down about software security. So John Viega and I wrote *Building Secure Software* in 1999. It was almost ten years ago, and it took about a decade to get things rolling.

But the last five years, what's happened is a move from evangelism about just being aware of the problem at all, to trying to do something about the problem — to integrating best practices into the SDLC, and to setting up metrics and governance programs at the highest levels in business. And what we've seen is real leadership from the financial services industry in terms of software security adoption.

So our practice at Cigital, which is a consulting firm — we've got about 100 people — is focused around software security. And most of our software security work is in the financial services vertical today. So what I see happening now is the early adopter curve is spreading and things are getting wider and we see other industries beginning to get interested in software security as well — like independent software vendors led of course by Microsoft; and embedded systems providers led by Qualcomm and other cell phone technology companies; and even hospitality. So things are spreading a lot wider which is nice to see. So I'm optimistic that we're actually making progress in this field.

Julia Allen: Well, it sounds like some really good uptake. So for a business leader, our target audience and listener group, how do you define software security? And why do you think it's something that business leaders need to start paying more attention to?

Gary McGraw: Software security is the activity of trying to make software behave under intentional malicious attack. And so, we already have a problem, as most software professionals know, just trying to make software work in normal day to day circumstances without adding in malicious attackers. When you add malicious attackers to the equation, things get a little trickier. I think what happened in terms of business, especially in the financial services arena, is that Sarbanes-Oxley caused public companies to realize how much exposure they have on the software risk front. And they started asking questions about "where all this software came from," and the answer was "they built it themselves." "And how many developers they had on staff," and the answer was "sometimes tens of thousands." And this notion of trying to get a handle on software risk, in order to get a handle on business risk, was what really drove financial services.

So if you look today at the leadership of big firms, like Morgan Stanley and Goldman Sachs and others, you see that there is an officer called the Chief Risk Officer, who is often in charge of figuring out how to handle a deal with software risk. And of course, the sexiest and most appealing and interesting and fun software risk to deal with is security risk. So there you go. So that's what I think is really kind of driving the business front. It's this realization of how much exposure we have to software induced business risk.

Julia Allen: So in your experience, how is developing software with security in mind different from normal software development? You talked about how challenging it is just to get software to function as it should be. But when you have to think about security, what are some of the differences that you see?

Gary McGraw: I guess the biggest difference is thinking about an attacker and pondering the attacker's perspective while you're building something. So good solid software engineering of the sort that you guys practice and preach at the Software Engineering Institute involves trying to make software behave itself under normal circumstances. And good engineering, lots of requirements, engineering and design, and design analysis and good coding standards, and all that stuff. So if you look at normal software methodologies, what you see is we have many possible ways to build software that works properly in normal situations.

The issue with software security is this attacker's perspective. And I think that a lot of developers, being very, very optimistic people, don't often think about attackers. So after breaking software for more than a decade, I can tell you, about 80 percent of the time when you break a system and you go and you talk to the guys that built it, they sort of look at you like you killed their puppy dog, right?

Julia Allen: Right.

Gary McGraw: And they say things like, "Well, nobody would ever do that. You're not supposed to do that. That's against the rules when you do something like put in a 5 gigabyte username." And I think that getting people to think about and actualize in testing the attacker's perspective is a lot of what many of the software security best practices or touchpoints are designed to do. And as you know, these are best integrated throughout the entire software development life cycle.

And my own approach, from a philosophical perspective, tries not to put one particular software methodology over another. I don't care if you use agile methods or XP or the spiral method or CMM SEI, whatever. It doesn't really matter which methodology you follow, as long as you're doing the right sorts of best practices in the software development life cycle. So I think that adopting this is something that has to be driven that way.

Part 2: Key Practices for New Software; COTS and Legacy Software

Julia Allen: Well, you started to mention the touchpoints, which is kind of an introduction to integrating security practices into the software development life cycle. Do you have kind of a short list of key practices that you recommend when you're working with your clients?

Gary McGraw: Absolutely. Well, there's seven touchpoints that — identified in the book *Software Security* that you mentioned at the top of the show, and that we talk about in our new book *Software Security Engineering*, which you and I helped to co-author. The two that are the most important are source code analysis, with a static analysis tool, and architectural risk analysis.

And the reason that there are two of those best practices that are important is that software security flaws come in two flavors — bugs found in the code, usually on line 42, and flaws found in the design. And it's divided up 50/50. So if you're going to target both bugs and flaws to remove all security-related defects, then you need best practices that target things both at the code level — that's code review with a static analysis tool — and at the architectural level — and that's architectural risk analysis. So we spend a lot of time in our book, and also I spend a lot of time in *Software Security*, talking about those best practices.

Now there are others. There are five other things that you can do. But frankly, right now, I think it's worth focusing all the attention on those two best practices and making sure that your organization is doing those. So if you're a business leader, and you're wondering whether or not you have software security issues and what you're doing about them, look and see whether you're doing those two best practices.

One other quick piece of advice — I think it's worth mentioning, and that is make sure that software security is somebody's actual job. And one of the issues in software security that we've found over the years is that technical people on the software side think it's the problem domain of technical people on the network security side. So they point the finger at them and they say, "Well, isn't that what those firewalls and anti-virus stuff and all that money that we're spending on security? Isn't that what that's for?" And then on the other side, you have the network security guys who say, "Hey, we built this beautiful pristine network, which we manage properly, and we've handled our risk correctly. And then these darn software guys come along with their broken software and ruin everything." So they point the finger squarely back at the software guys.

Now if you're in upper level management and you have two groups pointing at each other and nobody taking responsibility, you have a classic management problem. And that is that it's nobody's job. So the first thing that you can do, as an executive, is make it somebody's job, and both empower them and give them the budget to solve this problem. And we've seen that happen.

About ten years ago, we started our first software security group at Cigital. If you look at Morgan Stanley, they've got a group dedicated to this. They've got one dedicated at Goldman Sachs. They've got one dedicated at Qualcomm. They've got one dedicated at Cisco. Microsoft has quite a large software security group with 165 people in it. So a lot of progress has been made in some organizations, and I think that the first step is to make it somebody's job.

Julia Allen: Well, this is pretty interesting, because we've both been kind of in this landscape for a bit. But I sometimes wonder how practical is it to target software security particularly for operational systems. I mean I can see why it would be a good and useful thing to undertake when you're building something new. But what do you do about everything that's in production today?

Gary McGraw: Oh, Lord.

Julia Allen: How do you get your heads around and hands around that issue?

Gary McGraw: Well, so there turns out to be much more COTS (Commercial-Off-The-Shelf) and legacy software than even software than we're building today. And in fact you can do many things to get a handle on that sort of risk.

One of the important things that's happening today in the business community that we see in our own practice at Cigital is software consumers beginning to push back on their suppliers. By consumer, I don't mean guy on the street who buys Windows from Microsoft. But rather, say, Dell who's buying some software from a software supplier — and asking hard questions like, "Well, what did you do to make sure that this is secure? Can you provide me some evidence (or what we would call an assurance case in our book *Software Security Engineering*)." And so I can look at the assurance case and figure out whether I think that what you're doing is appropriate from a security perspective. And that is real market shift, which is fantastic.

So it goes to show that it's not just new software that people are worrying about managing risk with. They're also worried about COTS software. They're worried about legacy software. And it turns out that there are many, many things that you can do. Some of them are strictly legal. That is, you can build your service level agreements and your acceptance criteria to include legal constructs around security so that your suppliers are forced to do the right thing. And they need to provide evidence that they're doing the right thing. And I think we've made some progress there, too. So I'm happy to say that software security applies in that domain just as well as it does in the domain of new software.

Part 3: Two Ways to Get Started; Two to Avoid

Julia Allen: Well, that's great advice and good tips for folks who find themselves in that situation. So you mentioned, obviously, putting someone in charge. And you certainly highlighted a couple of the critical touchpoints. When you're working with your clients and they're just getting started, what do you find to be effective ways to ease them into this problem domain?

Gary McGraw: Well there are four ways that we've seen people get started in practice. And I wrote a little article about that for, I think, it was Dark Reading, before I switched to InformIT. But generally speaking, it sort of depends, in terms of effectiveness, what your organization looks like.

So one of our organizations that we've done a huge software security initiative with over the last two and a half years, going on three now — and they're still underway because they have 10,000 developers even though they thought they were an investment bank. These guys had a very strong centralized IT leadership that had the power and the budget capability and the political clout to cause big cultural change inside the organization, in the name of risk management. And so those guys ran a program that was much more top down in nature. And the first thing that they did was build a big plan based on where they wanted to be in two or three years and where they were now, and how they were going to get there — (1) that involved training lots of people; (2) that involved building a portal where there are places for people to learn and steal code and figure out how bugs and exploits really work; (3) buying the right tools and integrating those into both the code review and the testing process; (4) and also thinking about other aspects of integrating best practices into the SDLC. So you put all those things together, you get a big Program and that's a top down approach.

Another approach, where we had a customer that was equally large — but in this case they had strong centralized IT leadership but the rest of the organization was very much stove piped. And the development organizations in each of these — separate stovepipes, in each of these separate vertical business units — were not reporting directly to IT. The way that we approached that software security program was to perform what we called a portfolio risk analysis over their couple thousand apps (applications) to try to determine which apps were the most important and carried the most risk so that we knew where to invest the most resources.

Julia Allen: Right. Because given that you can't tackle everything, you've got to have some scheme for picking and choosing, right?

Gary McGraw: Exactly. And that worked great for those guys. So those are two ways to start a program. Now we've also had two ways that were sort of sub-optimal starting programs. One common way is that people will buy lots of tools because they get sold this, kind of, tool bill of goods and they think that if they just buy all these web application security tools or all these static analysis tools that the problem will solve itself. And they find themselves, kind of, sitting next to a several million-dollar pile of tools and looking at them and going, "What were we supposed to do again?"

Julia Allen: Right. Right.

Gary McGraw: I wouldn't recommend that as a way to start but you could salvage it. And then another common way that people get started that's often driven by the geeks is by trying to train their people. And there's lots of training and whether or not it sticks is unclear. But training only really works, in my opinion, when it is properly integrated into an entire software development life cycle approach. And so there are people who are just doing lots and lots of training and all of a sudden, they find out that their training program is not as effective as they thought and they look around for some more help, and we help people with that, too. So those are, sort of, four common ways of getting started with software security. The first two being optimal — a top down approach led by strong centralized leadership and a portfolio risk management approach.

Julia Allen: Right. Because the two that you said were sub-optimal, you're kind of bringing tools and training in with no context, with no process, with no foundation to build upon.

Gary McGraw: That's right. So as consultants, we, kind of, get called in to help people salvage those situations that they've pretty much painted themselves into the corner. But it is common to see those two things.

Julia Allen: Well, Gary, as we bring our conversation to a close, do you have some — other than maybe the sources that we've already called out — do you have some other places where our listeners can learn more about software security and software assurance?

Gary McGraw: Yes, absolutely. So I have a podcast called The Silver Bullet Security Podcast, where every month I interview some famous person in security and we often talk about software security and software security best practices. I also have a blog called Justice League. But I think the best place to go these days when you're just trying to get your head around the problem is, in fact, the Addison-Wesley Software Security Series. And there are many books in the series, some of which I've helped to author, some of which I authored myself, and some of which other people wrote. For example, the book *Secure Programming with Static Analysis* by Chess and West is a fantastic place to get a handle on how to do that code — the best practice static analysis that I mentioned before. So those are three sets of resources that people can go to. And then, finally, I also have a monthly column that I write for InformIT.

Julia Allen: And of course, we'd probably be remiss if we didn't mention our own, Build Security In website, right?

Gary McGraw: Absolutely. That's a great place to go and it's lots of free resources. It's a website that we built with Joe Jarzombek's help for the Department of Homeland Security.

Julia Allen: We'll be putting all the URL's in the show notes. So that's great.

Gary McGraw: Super.

Julia Allen: Well Gary again, thanks so much for your time today. I enjoy reading everything you write and kind of following your journey and path as you guide us in this whole conversation and this whole journey in software security.
So thanks so much for your time.

Gary McGraw: Well thanks Julia. And you know, last note. I am optimistic about the progress that we've made, which as you know as a security professional, Julia, is kind of a rare thing, finding an optimistic security person.

Julia Allen: Absolutely. So keep it going.

Gary McGraw: There you go.

Julia Allen: Thanks so much.