Silver Bullet Talks with Tanya Janca

Gary McGraw | Synopsys

Hear the full podcast and find show links, notes, and an online discussion at www.synopsys.com/silverbullet.

Tanya Janca is a senior cloud developer advocate for Microsoft, specializing in software security. Her job involves evangelizing software security and advocating for developers through public speaking. She's a leader in the OWASP DevSlop project and believes in hands-on teaching via workshops and real technical examples.

How did you become interested in security?

I was a software developer for a really long time, and I never thought that I could like anything more than building things. And then I met this guy. I organized these lunch-and-learn sessions at work for my team, and we had an ethical hacker come in and he broke in through the login screen using SQL injection, which I'd never seen before, and it just completely broke my mind.

He came back and talked to us a few more times, and he was in bands and I was in bands so we became friends, and then one day he's like, "Tanya, join the dark side! Be a hacker! You'd be so awesome!" And I was like, "No. Nothing is better than software development." So he spent a year and a half convincing me to be his apprentice.

Well, we really need people in our field.

Yeah. So I guess he's doing a great job of it, because he brought me in and I'm bringing others.

How long were you a developer before that happened?

Around 16 years. I started programming as a teenager, and then I got my first job in IT as soon as I was legal.

So I've always held that the best way to create software security people is to start with software people versus, say, network security people, and go from there. Do you agree with that?

I agree with you 100 percent. I don't know how someone can understand how to secure a thing if they don't know how the thing works. That doesn't mean a network person can't learn software or learn the security of software; it just means they have a bigger uphill battle.

Maybe I'm biased because I'm one of those software people, too, but I think that software is harder to learn to do and to learn about and to practice than, you know, some aspects of network security.

I personally like software better than networks; for me, it comes more naturally. I kind of stink at network security. I can do the basics, and I can scan all the networks and look for things, but when it gets deep into it I'm like, "no, I'm afraid."

Well, 20 years ago when I was first starting out in software security, there was no field, and about half of the people or more were these normal security people who were trying to think about software, but they didn't really even know what a build was, or a compiler. They were kind of Perl people, so they knew something about scripting, but that was a problem. And it has always seemed easier to me to start with software people. We have a lot of software people at Synopsys because we think it's easier to teach software people about security than the other way around.

Oh yeah, it's way easier. Especially if you think of a software developer that's curious about security. That's the magic ticket.

That's a problem, too, because I'm not sure you can teach somebody how to code. I started coding at 16, and I took some classes later, but they were just kind of teaching me stuff I already knew. How do you teach somebody to code? Do you, or is it a natural born problem?

I'm not sure. I never even thought of that to be quite honest, because

1540-7993/18/$33.00 © 2018 IEEE

About Tanya Janca

Tanya Janca is a senior cloud developer advocate for Microsoft, specializing in software security. Her job involves evangelizing software security and advocating for developers through public speaking. She's a leader in the OWASP DevSlop project and believes in hands-on teaching via workshops and real technical examples. As an ethical hacker, OWASP project and chapter leader, software developer, and professional geek of 20 years, Janca is fascinated by the "science" in computer science. Previously she worked as an IT security coordinator for the 42nd general election in Canada. She is also an avid gardener and has been the front woman of multiple bands. She holds a computer science diploma from Algonquin College, and she grew up in Ottawa where she still lives, and she's fluent in French.

some people have told me how it's really hard, but at least 50 percent of my family are computer scientists or computer engineers, and the rest are mathematicians or mechanics. So when I was like, "oh, I've studied computer science," the whole family was like "obviously." So when I started coding, it was like, off to the races.

They wanted to play a guitar, so I made a "how to play a guitar" program, and for me it was exciting and awesome immediately, but other people just bang their heads against the wall. Maybe it takes a certain type of personality trait, or maybe it's the way our brains work. I'm not sure. But some people just pick it up like it's nothing.

And that has implications for software security people, too, if you hold, like we do, that you should start with software people first, and turn them into software security people.

I've actually worked at a bunch of places where regular security people are trying to do application security, and it's very painful. It's painful for me. It's painful for them. And I find that the people who suffer the most are the software developers. Because you have someone that's thinking, "Well, can't we just put an appliance in front of that? Can't I buy a box that does the security?" No, you have to teach the developers. I gave a one-hour overview of the OWASP top 10 to the security team I used to work with, and I felt like I had done something bad to them because all their faces looked like they were melting off. And I was just trying to explain the concepts—not even how to protect against them. And they were like, "How can anyone ever learn all this?" I'm like, "Guys, this is just 1 through 10."

There turn out to be 10,000 of these.

And then you need to know how to protect against them, and they were just like, "this is impossible." I'm like, "Well, if you've never coded before, it certainly seems that way." It's an advanced thing, right? I feel like security is—you can't really start a career immediately in security, or it's really hard to be really good at it. If you are a network engineer then you do network security. If you're a software engineer you do software security. You have to master the thing underneath first.

It gets even worse, too, if you try to figure out where software architects come from. But let's don't even go there. I want to pursue a slightly different angle. You've been on the receiving end of the "ugly baby" phenomenon of security, where somebody declared your code bad during a review, and they're like, "your baby is ugly and your code is bad, and everything is terrible, and don't come back until you get it right."

It's so true. It feels like someone said your baby was ugly. Actually, I have a talk called "Insecurity in Information Technology," and I tell that story for the first five minutes of my talk. Basically, I had to go back and forth three separate times with the person running the VA automated scanner that he did not understand whatsoever, and he shamed me. He told me that if my code was good, that I should have passed the first time, and that if I was a good software developer, I would have known how to fix all of the issues. But now that I've done some pen testing and some vulnerability assessments, I'm like, "Oh, that guy had no idea what he was talking about, and he felt more insecure than me."

He's just letting the tool do the work.

Yeah, and he was doing that on purpose to stop me from asking more questions. I worked somewhere once (that will remain unnamed), and the two people on the team that I'm supposed to help prop up and make functional, the security people, were making faces at each other. I'm like "what are you doing?" And they said they were practicing their "you're-so-stupid" face so when developers ask questions, the developers know not to ask more questions.

That is terrible. That is just absolutely wrong. So what do we do to eradicate that attitude?

Well, I have a talk all about it. I did some research into this. Social scientists study all sorts of things about culture change and all of that, and it turns out when people feel insecure in their jobs, there's predictably bad behavior that they do, and that predictably bad behavior consists of things like that. So if you're on a security team that doesn't even understand software and doesn't feel supported and doesn't have training and doesn't feel there's enough of you and all of that, those people are going to feel really insecure, and unfortunately some people take out their insecurities on other people.

If I feel insecure about a thing, what I do is book a talk or, for instance, an open security summit where I'm going to be unleashing a new project. Because I don't know how to do that thing very well yet, I'm just going to make a big commitment for myself so I have master the thing so that I don't feel uncomfortable about it anymore. But I see a lot of people instead shy away. Everyone handles their lives differently, but we should support security teams, and especially support developers.

We also need leadership. Sometimes, you'll have someone at the top who doesn't take security seriously or a CSO who doesn't understand software at all and doesn't understand application security at all and thinks it's like network security. I've had senior security people say, "tell them to just patch it." We're not getting a magical patch that custom fits the software.

Exactly. Where do you think the patch comes from, sir? So let's compare and contrast working for the Canadian government, which you did for a long time, versus working for a big corporation, which you've done for a very short time.

It's so different, it's amazing. In the government, your role is very defined. This is your job, and you just do those things, and you try to do those things as well as you can. At Microsoft, I do my job, but I can volunteer to be a part of all these other teams, which to me is insanely exciting.

In the government, if I've worked somewhere a long time and I have clout, then I could say "I want on that project," and usually people will be like, "it's Tanya so it's okay by us." You have a reputation for things, and you can kind of push your social capital to get the things that you want. But at Microsoft, I'm just like, "I'm really curious about this technology you're building. Can I look at it from a security angle?" They line up for security, as opposed to the government where I felt sometimes I was chasing them around to do security. At Microsoft, they're looking forward to it. The attitude is just so exciting.

Let's poke into the stuff you did for the government—in particular election stuff. In your view, what's the biggest risk to fair and free elections? It is insecure voting machines, social media propaganda campaigns, or apathy?

All the countries that have elections are really worried about voter suppression, and people using social media to try to trick people into going to a wrong voting session, for instance, but it can get so advanced to having armed men outside of where you're supposed to vote, and they're going to shoot you if you try to vote.

Clearly we don't have that problem in Canada, but in the last election we did notice some tricky things being done on social media that previously we'd not seen. And there's been some articles out about it. I can't go in depth about things that aren't publicized, but I mean, the US has been having quite a bit of that.

And as it turns out, our elections used to be approximately when the US elections were, which meant all the people who were employed doing that trickery were busy. But this one was in the spring, so they had time off.

So they came and screwed around with the Canadian elections. That's terrible. We're sorry.

I believe that is a possible explanation. I don't have proof one way or another, but to me it's unethical at a level that I can't even express with words. Even if someone's going to vote for someone that I wouldn't vote for, I'd still feel that it's their right. And so I have strong positive feelings about elections, democracy, and about fairness. I actually spoke on this in Switzerland at the Swiss Cyber Storm in 2017, and I don't know if you know, but Switzerland is switching to e-voting, but they currently do a lot of their voting by mail.

Their risks and concerns are very different from Canada's. In Canada, we actually vote on paper, and we count with something like 36 members of the media in the room, and a member from every single party—so we have a giant room and everyone counts together. And then they actually recount it again to make sure, and we keep it for four years just in case.

The most important risk to democracy in my opinion is the public not believing the results. And in Canada, we'll do anything to make sure that they can trust it—including letting members of the public in while we count, because that's how important it is to us. Complete humility. No egos. Even if it takes twice as long, we'll work all day. It doesn't matter.

You champion learning as witnessed by your OWASP DevSlop project, among other things that you've done. Can you briefly describe that project and the importance of getting containerization right to software security?

I'm so excited about this project. There are four of us on the team right now, but we're growing pretty quickly. Nicole Becher and I are leading it. Basically, each of us is making different types of pipelines to automate as many security processes as possible. The next step is to add containers, and we're each going to release to different cloud providers so you can see how to safely release to each one of them.

And one of our team members is on the OWASP Core Rule Set team, and they're the ones that created the signature set that's used with the free WAF ModSecurity, the web application firewall. So Azure Cloud's community has created a plugin in conjunction with Microsoft and Core Rule Set team, so you just press a button and it turns on the Core Rule Set for free. That's insane. That's so amazing. So I want to set that up so I'm creating a team website for us, and then I'm going to deploy it through our pipeline live so people can watch and see. My stuff is going to be Microsoft stuff, but everyone's using different platforms. We have Apache; we have all sorts of different things. Whichever tech stack you're using, we're hoping we'll cover it, and you can just copy the steps of our repo and then adjust it for your environment. Which is so exciting because it's really hard to make a pipeline from scratch, especially if you don't know.

We're going to be at the Open Security Summit. Nicole and I are going to be at NorthSec in May. We're also doing microservices, so Nikki and I are going to crush this thing named Pixi that we made, which are insecure microservices. Basically we're trying to teach people by watching us crush it. "Watch us crush it. Okay, now all of you are going to crush it," and then we walk around and make sure everyone understands and go through the lessons with them.

It's kind of like dragging WebGoat into the modern world. A little bit. Pixi is, anyway.

If anything, the newest kind of WebGoat is Juice Shop by this guy named Bjorn and his team. It is a really nice lab app, but we want to cover the weirdo things that aren't covered in Juice Shop. They have some microservices, but we're really interested in DevSecOps. Basically we're obsessed. Yep.

That sounds cool. I think you should get a logo that's somehow a pig, because I love the DevSlop name. That's a hilarious name.

Our logo is actually the two gears is from DevOps, except it says "DevSlop." We should have a pig, though, because we like animals.

You could just say, "slop the pig." That could be your motto. Let's keep pushing down the DevOps thing a little bit. I have some worries. In my view, the critical danger of DevOps is that the rush to automate everything and speed everything up leaves secure design analysis or threat modeling—or whatever you want to call it—sort of lost or left out. What should we do about architecture in the DevOps paradigm?

I really feel that it needs to be DevSecOps, and by that I mean it can't be the Dev and the Ops teams doing stuff and then security on the other side. In my opinion, the security team needs to be in there with them. Someone on the security team should be on their project. Maybe they have to tag team and swap out for different activities, because some of us are better at code review than threat modeling, et cetera. But there needs to be security sprints as well.

So if you're doing design or you're adding a new feature, you should tap in your threat modeler to come in and threat model for that activity. And then there has to be a full sprint—or multiple ones depending on how big your project is—where it's just all security for the entire sprint to identify the little things you've not been keeping up on.

I see the point there, but I've also been involved in many analyses in projects where we looked at the architecture, and we were like "uh-oh, we're going to have to refactor this whole thing now." Even though you're a multinational bank, and it's going to take five years, it has to be done. And I'm hoping that we don't lose sight of that kind of work in DevOps. I think DevOps—DevSecOps or SecDevOps, or OpsyDevsySexy, or whatever you want to call it—is fine, and it's got a lot of positive characteristics, but we cannot forget what we already know. That's super important.

Oh, I agree. I guess I'm seeing a lot of waterfall where the security team's model is "stop while we do some security." Which is crazy. And it doesn't work. You ask for a threat model, and they get back to you in four months. And you're like, "you can't just stop me." So what do the software developers do? They're like water. They just go around you. You don't want me in the database? Well, I need in there so I'm in there now. I did not realize how much hacking I did until I became a hacker.

If you need to get your job done and you have a deadline, you're just not going to stop. So if you have a software developer that's like "okay, I guess I'll just sit on my hands until…"—that person's fired. They're not going to make it. You want the people who are like, "nothing's going to stop me."

That's the good part of DevOps because if you're integrated tightly and it's automated, it's way easier to do, and it doesn't stop you or slow you down too much. I totally agree with that. But somehow we've got to strike a balance, and I don't think we've figured it out yet. All right. Another topic. What is more of a challenge: being a woman in security, or being a woman in the Ottawa punk scene?

Definitely music. Music is much more complicated. Sexism is completely overt and in-your-face in music, not in a way that it is like in tech. Trying to find musicians to play music with that I didn't have to sleep with was next to—no. They're just like "I'm not interested then." Or, "I don't play with girls." Being told during sound check to go change into a miniskirt and a thong.

So it's just totally, totally blatant.
Yeah, but because of that, most of this is never a problem for me in tech, because I'm used to it being so much worse.

I've got one last kind of crazy question. It's about a song that you have called "Heartbleed."
"Heartbleed" is about the heartbleed vulnerability.

I knew it.
That group, the Zero Day Reapers—all their songs are about different aspects of security, because I was in my apprenticeship trying to learn what different things were and understanding things, and for me writing a song is the best way.

The Silver Bullet Podcast with Gary McGraw is cosponsored by Synopsys and this magazine and is syndicated by SearchSecurity. ■

Gary McGraw is vice president of security technology at Synopsys. He's the author of *Software Security: Building Security In* (Addison-Wesley 2006) and eight other books. McGraw received a BA in philosophy from the University of Virginia and a dual PhD in computer science and cognitive science from Indiana University. Contact him via garymcgraw.com.