Sonatype

Real World Experience

NEXUS LIFECYCLE USER

Strategies to Ban Avoidable Open Source Risk: A discussion with Andrew Wild, Former Chief Security Officer, Qualys

At the Gartner Security & Risk Management Summit in June 2014, Wayne Jackson, CEO of Sonatype, assembled a team of security practitioners to discuss "Strategies to Ban Avoidable Open Source Risk." Andrew Wild, Former Chief Security Officer at Qualys, shared his insights and experiences building an effective secure software development lifecycle that also addressed avoiding open source risk. Here are excerpts from the panel discussion.

Jackson: The 2014 Sonatype Open Source Development and Application Security Survey showed the application is the most commonly targeted attack victor and yet we spend relatively little time and relatively little money securing that attack vector. What are your thoughts about why that might be and how you think that might be changing?

Wild: I think a lot of it has to do with the history of information security in organizations. Information security has worked very hard over the years to integrate with IT and with the business. But integrating with the development engineering side hasn't been a focus in many organizations and, as a result, we do not see the vulnerabilities that are there. We see the exploits.

It is getting attention now, but the tools are often lacking or they are just not deployed within the organization. The information security organization doesn't have that relationship with the engineering team to get them into the Software Development Life Cycle (SDLC) or they are just more comfortable sticking with what they know best: budgeting for firewalls and endpoint security.

Jackson: It's still remarkable how many people in very senior positions have no idea of the extent of their use of open source and whether they are vulnerable or not.

Wild: We are making progress in terms of awareness and policies. It's not uncommon now to read in publicly traded companies' filings public statements

> A bill of materials, whether it's of open source components or in-house components, is part of the overall strategy on large software projects for having trusted, secure components that you vetted and verified are good and acceptable. Reusing those is a key component or piece of the strategy.

that a risk factor is the use of open source software and there could be licensing impacts from that. How that risk is stated and disclosed, actually mitigated or managed inside, well that's a totally different matter I think but it's beginning to appear more and more. There is awareness that it is a risk that needs to be managed. I think in many organizations, it's one of many and it may not get the prioritization it really deserves.

Jackson: Folks are coming to the reality that their infrastructure is built on open source. Human-based policies that follow older methodologies—to establish white lists and black lists, request approvals and so on—are really falling short because too much open source being used is too complex.

Wild: One of my priorities was to gain visibility. At Qualys, we have a defined software development lifecycle. We use an Agile development philosophy, but there was a disconnect between the visibility the security team had and with the developers. You talked about the developers not tracking or knowing. Well, why would they? They built it; it passed through QA testing. Operations is typically the group that on an ongoing basis is managing vulnerabilities. The operations team is handed an assembled piece of code to run. They may not necessarily have detailed visibilities into all the components that make up that system. This creates potential disconnects. So to gain that visibility into the build process—and to identify potential open source components being used before the build process is complete—is something I thought was very significant and would really augment our capability to develop safe code.

Jackson: You are building the code that keeps other people safe too.

Wild: Yes! So it's important that we do it right. Just because you buy a security product, don't assume the people that built it know anything about security … at the code level. It's really hard to get right and not fall into a false sense of security just because something has the word "security" in it.

Jackson: Toyota has a process where they try to optimize their supply chain thinking around producing more products more quickly with more predictability, with more efficiency over time. To your point about awareness being the first step, learnings from the Toyota supply chain philosophy are all about creating awareness so that employees who are empowered to make decisions can do so in a more real-time way. There are guardrails, so that predictability is preserved and optimized over time.

> There is awareness that [use of open source] is a risk that needs to be managed. I think in many organizations, it's one of many and it may not get the prioritization it really deserves.

Wild: In the private sector, third-party risk management is maturing very rapidly, perhaps not as rapidly as it should. But I've seen just in the three years that I've been at Qualys tremendous change in the volume and complexity of the questions that come from customers asking about the Qualys security program, asking about the software development lifecycle, how it works, what we have, what controls we have, and our use of open source software. So this is beginning to become part of the purchasing of the procurement process and, at least in the private sector, I think that is going to have an effect. How that translates to actual implemented security improvements remains to be seen, but I do think there is a concerted effort and push that is coming from the advancement and maturity of third-party risk management.

On the commercial side, we always used to joke that after Enron we finally found the driver for quality. It's prison! We need to send people to jail and then they get religion about quality. Well I think the same thing is sort of happening in security. I look at the Target example, which I know everybody in this conference is talking about because it's the first time I ever remember the CEO got fired because of a data breach. To me, that's going to have a huge ripple effect on the commercial side just in terms of visibility and awareness of the implication of those kinds of things. If you don't do your diligence or governance, it's a bear of a problem.

One of the challenges with regulation, policy, and governance is that you not only have to legislate it and get people to follow it, you've got to enforce it. We've seen contracts that say, "Thou shall build with Agile." Healthcare.gov was sent out as "thou shall build with Agile." But the people who are managing it on the government side don't know what that means and they don't know how to enforce it. So we also need to train acquisition, project management, and program management on the government side to know what questions to ask and to know how to make sure that what they are asking for is actually being done, because a lot of times it is not being done.

The contractors are bamboozling the government into accepting things that really aren't what they were asking for because they don't know how to check that. So that's another piece we can't forget about it if we are going to legislate.

Jackson: What about moving open source security into the development process as a culture? We've started encouraging folks to think more about building a culture where developers can simply make better decisions.

> We need to give the developers the tools to help them with their decision-making when selecting open source components.

Wild: Many organizations have provided safe coding guidelines for developers and that's great if they follow it and it's kept up to date. It does take care hopefully of the security of the code that's being developed in-house. But what's missing in those same organizations is how does a developer know to pick a safe component? How does he know that that component is not vulnerable? What resource does he have? Is he incented or told to check to make sure? Is he given guidelines as how to select a component? He might just pick the component because he's got a buddy that's two companies over that he said this is the greatest thing, it works, and it's fast, so he's going with it.

We need to give the developers the tools to help them with their decision-making when selecting open source components. The security of the component should be one of those factors they consider, in addition to other things like support, understanding, documentation, responsiveness, and performance, when they are looking for a component.

Jackson: Are you thinking about DevOps and continuous delivery at Qualys?

Wild: We are definitely moving towards DevOps. I mean when you talk about a cloud delivery model, where you have updates going out on near continual basis, it's the way to go. It's going to improve our efficiency and our responsiveness. We are going to inject more security into that process and that's what I think DevOps does bring: an opportunity for the security practitioners to really get in with the DevOps community and work tightly with them … I think you have to be part of that process and that's what we are doing at Qualys.