Developers need to start thinking about security now

November 8, 2013 8:30 AM   Andy Chou / Coverity   Venture Beat

http://venturebeat.com/2013/11/08/developer-first-security-2/

Andy Chou will be discussing developer-first security at DevBeat.

The fundamental relationship between security and development is broken.

It's broken because security teams drive security, and development teams let them. There needs to be a re-balancing of this relationship, driven by an awakening in the developer community.

Development teams abdicate security because they don't understand it. They abdicate because they are too busy building features. They abdicate because they are too busy fighting fires. Developers are just too damn busy.

Editor's note: *Developers! If you're good and want to be great, our upcoming DevBeat conference, Nov. 12-13 in San Francisco, is a hands-on event packed with master classes, presentations, Q&As, and hackathons, all aimed at boosting your code skills, security knowledge, hardware hacking, and career development. We'll also have special sessions dedicated to security. Register now.*

And yet, there was a time when developers were too busy for quality. But a new culture has begun to sweep over the developer community. Admired developers and bloggers are flying the banner of test-driven development. It is motivated by the desire to move faster, deploy more frequently and get feedback incrementally. And with this cultural change, we've seen a flood of practical tools and techniques to make test development more efficient and automated. The idea of throwing untested code over to the Quality Assurance (QA) team once every 12 months is starting to look antiquated and unprofessional.

What happened? Developers woke up and realized that they need to own quality. And the job of QA became more about assurance and less about trying to test quality in at the end. QA is becoming about validating that quality was built-in from the beginning. This should give us hope that security could follow the same path.

Let's acknowledge that perfectly secure software is impossible. It's even more impossible at scale, using frameworks and operating systems and services that are themselves riddled with weaknesses. In the limit, security is an infinite cost center. It's never completely done.

Therefore, practical security must be about efficiency. It must be about finding ways to maximize cost-effectiveness. And the only way for anything to be cost-effective in development is for it to be considered as early as possible. An architectural problem is cheapest to detect when the architecture is being designed. A coding error is cheapest to detect when the code is first written.

We can start by taking security seriously as developers. This means facing the fact that security teams don't have the scale or resources to find all the weaknesses for us. And it means realizing we don't have the needed security expertise to make some key decisions about how we design, code, test, deploy, and maintain our applications. We have an obligation to pull in security expertise when it can have the most impact. We have an obligation to incorporate security into our planning, our processes, and our culture. That's what developer-first security is ultimately about.

Andy Chou spent four years hacking through an undergraduate degree in EECS at UC Berkeley and got sick of debugging his own code. He spent the next four years at Stanford researching ways to automatically detect bugs in code by leveraging compilers. After completing his PhD in 2003, Andy cofounded Coverity to commercialize this work. Now, he enjoys running the Coverity Security Research Laboratory.
