Applying DevOps principles to achieve software supply chain security

THIS CONTENT HAS BEEN PROVIDED BY CLOUDBEES

Prakash Sethuraman, chief information security officer, CloudBees

CloudBees.

A recent survey sponsored by CloudBees showed that software supply chain security is top of mind for many senior executives right now. The problem is a general lack of clarity on what to do about it. A recent executive order from President Joe Biden's administration charges several agencies, including the National Institute of Standards and Technology, with releasing guidance around this very issue. NIST's preliminary guidelines were due in early November and not yet released at the time of this article.

But that executive order will have a cascading effect on federal agencies and the contractors who supply them with software. All of these entities are going to need to understand the provenance of the components in their entire supply chain, which will make visibility into their software delivery processes increasingly important. In order to create something like a software bill of materials, they're going to need to wrap their hands around the entire process.

"The current philosophy, and it's been like this for a couple of years now, is to shift security left. Which on the one hand makes sense, because the sooner you can catch problems, the better off everything's going to be," said Prakash Sethuraman, chief information security officer for CloudBees. "But that's placed a huge burden on developers to know and understand what secure and compliant is. But they're not security folks. Security and compliance teams, they've got to try to train developers how to use their tools. And then they also have to figure out how to write that code into all these different tools and keep it current."

The problem is software supply chain security is too big a problem to leave to any one group. It has to become everyone's responsibility. So rather than just shifting security left, in order to truly secure the supply chain and meet the requirements laid out in the EO, agencies and contractors need to go beyond that paradigm and begin thinking in terms of shifting security and compliance everywhere. So how do they do that?

Sethuraman said CloudBees approaches this by applying DevOps principles to the software supply chain through a three-pronged concept: secure in development, secure in delivery and secure in production.

"No one else in the DevOps space is thinking and talking about this triad," Sethuraman said. "One of the problems with DevSecOps is that first off, I believe the 'Sec' is silent; if you're doing it correctly, it's not a separate thing. It's an integral aspect of everything that you do because you've built these controls in. This triad is a lens to view how you apply those things to what's going on. It's getting more into the pragmatic aspects of how we do DevSecOps."

The secure in development part of this triad refers to the code. There are a number of things that have to be done to ensure the code is clean, including verifying that the right tools and libraries are being used.

Secure in delivery refers to the people, the processes, and the controls. It means the right people need to have eyes on it at the right time, and the right controls and processes need to be observed in the approval process.

"There's a concept in software delivery called drift. And this is where a lot of organizations get caught out: Something changes as it goes through the pipeline. And that change isn't detected," Sethuraman said. "So you have to make sure that only immutable, approved objects and components are used for delivering that software, and that if change does occur, it gets detected and stopped, or approved by somebody. And the data and the evidence for that approval should also be attached to that decision."

The final part of the triad – secure in production – is essentially a nod to the fact that these days it's a guarantee that at some point in the future some vulnerability or issue will arise that needs to be addressed immediately. Mean-time-to-detect and mean-time-to-repair are often held up as metrics by which an organization can assess its performance in responding to vulnerabilities. But there's a gap between those two, between when a vulnerability is detected and when it's fixed, when the organization is exposed.

That's why it's important to have systems in place to immediately mitigate that risk. For example, a feature flag can instantly disable a function that's discovered to be compromised, closing that vulnerability so it can't be exploited before it's fixed. Alternately, with the right controls in place, it's possible to do an automated rollback if necessary. It's a question of being able to respond instantly, not necessarily repair instantly, because it will take time to discover what went wrong. But being able to mitigate that risk immediately gives organizations the luxury of time in which to do that.

"The old story about security being the 'Department of No,' or the 'Release Prevention Department' is not relevant in the age of DevOps," Sethuraman said. "Because if you're doing it right, security's built in. You've got the receipts, you've got the evidence there. So you can go faster, securely."

CloudBees®

Compliance, visibility, speed, and so much more - with CloudBees

Build security and compliance into every step of the software supply chain, featuring:

- Continuous Compliance
- Continuous Integration / Continuous Delivery
- Release Orchestration
- Feature Flag Management

Secure and compliant from code commit through production at 5X the speed.

LEARN MORE

cloudbees.com/use-case/streamline-governance-compliance