5 Principles for Securing DevOps

DevOps, a new model for software development, is transforming the way the world creates software.

To successfully integrate security testing into DevOps, it's critical to understand how DevOps and continuous integration/continuous deployment (CI/CD) are different from Agile development and how this difference changes the requirements for your application security.

The following five principles will get you on the right path to securing code at DevOps speed:

1 Automate Security In

Automate security from day one. By automating security, teams can ensure that each component gets all the security testing it needs without taking up any extra resources, making security a part of the development process itself.

Look at your existing tools for continuous integration, continuous deployment, defect tracking, and more to determine the best points to introduce automation.

Read this datasheet for more on automating application security.

2 Integrate to "Fail Quickly"

It's much faster, cheaper, and easier to ask a developer to fix something they just coded compared to something they wrote six months ago.

Use automation to conduct testing while the developer is writing the code, giving them instant feedback so they can make a fix before it ever becomes a problem.

Check out this guide for more on application security integrations.

3

No False Alarms

If your automated testing has too many false positives, you'll unnecessarily stop the development team's delivery process, force them to deal with the false-positive findings, and delay their schedule.

Earlier generations of AppSec tools often returned false alarms due to the way they were designed. These tools were intended to be used by an AppSec professional who would screen the findings before passing the actual issues on to the developer. Instead, modern tools are designed to provide both maximum coverage for finding critical flaws while tuning out the noise of low-level issues.

Read this blog post for more about reducing false positives.

4

Build Security Champions

Security professionals are scarce, and most developers aren't trained in the practices of secure coding.

Tackle this problem by designating one person on the development scrum team as the go-to resource for application security. When you have a security champion embedded with every scrum team, you're guaranteed to have security represented in every design decision and at every stand-up.

Watch this video to learn more about security champions.

5

Keep Operational Visibility

In a DevOps approach to software development, the development team's job doesn't stop once the product is in production. Development teams should also continue to monitor security in live-running applications.

This can be done with a variety of tools and practices, many of which the team already uses, though their tools may need tuning to isolate attacks from other events.

Learn more about protecting apps from development to production in this eBook.

Get more details on these five principles in our eBook

5 PRINCIPLES FOR Securing DevOps

DOWNLOAD

VERACODE

Veracode is a leader in helping organizations secure the software that powers their world. Veracode's SaaS platform and integrated solutions help security teams and software developers find and fix security-related defects at all points in the software development lifecycle, before they can be exploited by hackers. Our complete set of offerings helps customers reduce the risk of data breaches, increase the speed of secure software delivery, meet compliance requirements, and cost effectively secure their software assets — whether that's software they make, buy, or sell.

Veracode serves more than 1,400 customers across a wide range of industries, including nearly one-third of the Fortune 100, three of the top four U.S. commercial banks, and more than 20 of Forbes' 100 Most Valuable Brands. Learn more at **www.veracode.com**, on the **Veracode blog**, on **Twitter** and in the **Veracode Community**.