VERACODE Infosheet

Understanding Modern Software Development:

What Security Professionals Need to Know

Ryan Lloyd, VP of Products at Veracode, recently sat down with Scott Ward, Principal Solutions Architect of Emerging Partners at AWS– Amazon Web Services, to discuss the recent changes in software development, their implications on security, and what security professionals need to know about this new landscape.

How have development tools and processes changed in recent years?

There has been a noticeable shift from a monolithic architecture to services- or microservices-oriented architecture. Microservices architecture entails breaking applications into smaller, interconnected services instead of one large, monolithic application. The reason for the shift is likely due to the fact that microservices architecture is faster to develop and easier to maintain. Unlike monolithic applications that are hard to make changes to, microservice applications can be adjusted on the fly.

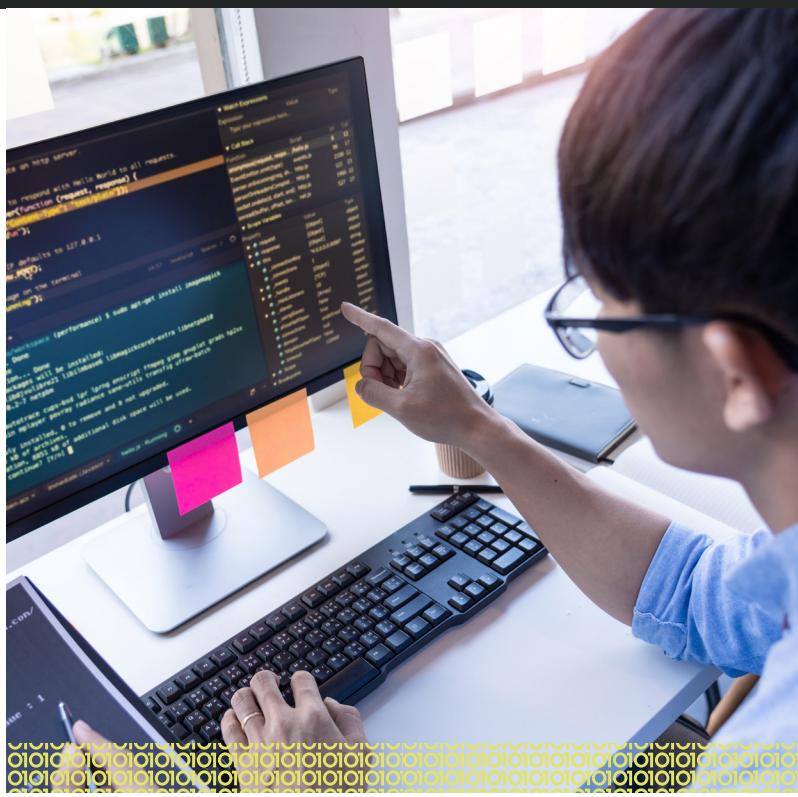
Organizations are also moving away from manual processes in favor of automation – this includes both automated deployment and application security (AppSec) tests. Automation has enabled organizations to move from quarterly software releases to weekly or monthly releases.

Lastly, there is increased flexibility with testing new environments and an increase in CI/CD processes like blue-green deployment. With blue-green deployment, organizations are able to stand up multiple copies of a software stack and run them both with little downtime. It's a useful technique for transferring users to an updated version of an application.

What are some of the challenges that developers run into while trying to balance the complexities of new tools and processes with security measures?

The new tools and processes have resulted in faster deployments. With faster deployments comes increased security scans. To help security professionals with the influx of scans, developers took over the scanning, leaving security professionals to concentrate on security oversight and regulatory compliance.

But, since developers are also responsible for ensuring timely deployments, they're at a crossroads. As Scott Ward expressed, developers are left wondering, "do I move faster, or do I stay secure?" There is a fear that by taking the time to scan applications for vulnerabilities, deployment time will suffer. Developers want their applications to be secure but – since most developers are compensated on speed of deployment – security often takes a back seat.



What are some approaches security professionals can take to get developers fully onboard with security?

Security teams and development teams need to build a foundation of trust. Ward stated, "The security team needs to make sure that they aren't presenting themselves as a group with their arms crossed saying 'no.' That's the typical way that security has been looked at, and that's a perception that will prevent the development teams from letting them in and participating." Security should be more of a "guiding light" to help developers – not the gatekeepers. They should aim to involve developers in security processes early on.

One proven method for gaining developer trust is the implementation of a security champions program. A security champions program consists of nominated or self-elected developers with an interest in security that attend security trainings to become the voice of security on their development team. Security champions advocate for best practices to make sure security is included in the software development lifecycle (SDLC) early on.

Another way to ensure that developers will implement AppSec scans is by making it easy to conduct the scans. Developers understand the importance of AppSec scans, but the scans often slow down deployment. By integrating the scans into the developers existing tools and automating them, developers can easily conduct AppSec scans without compromising time to deployment.

Finally, help developers write secure code so that they aren't bogged down with remediation later in the SDLC. The best way to help developers learn to write secure code is with hands-on tools and training. Consider a tool that allows developers to practice fixing broken code in their chosen language – like ***Veracode Security Labs***. Or as Ryan Lloyd mentioned, consider adding a solution to the ***IDE phase*** that provides developers with real-time feedback as they code, "almost analogous to a spell-check capability."

For more information on the changes in software development, or for additional tips on unifying the developer and security professional roles, check out our webinar with AWS, ***Understanding Modern Software Development***.