SECURITY INNOVATION PERSPECTIVES

5 Steps for Moving to Mobile DevSecOps

It's time mobile development adopted a DevSecOps approach that fits the way developers build apps — automated, rapid, continuous, iterative, and auditable. Here are five steps to get there.

By Dinesh Shetty, Director of Security Engineering, Security Innovation

Why wouldn't secure mobile app development follow the same DevOps process as developing enterprise apps? DevOps organizations are agile and adaptable, thanks to automation and integration. Unfortunately, mobile app security is anything but. It's often manual and monolithic. It is a multilayered process with maze-like interdependencies that doesn't easily fit into an agile delivery model.

Securing mobile apps is almost seen as an obstacle to innovation and on-time release. However, not securing them gives cyber adversaries easy access not just to a phone or device but to any data that is stored on the device, as well as the networks it accesses. Access to enterprise apps, and the data they store, can be heavily restricted to authorized users and devices. Mobile apps, on the other hand, can be installed on any device, regardless of security settings. This makes them more vulnerable to data leaks. Even if the app is secure, a committed cyber adversary will exploit any exposed mobile interface, like the platform, calls to the OS, or background operations. It's time mobile development took a huge leap forward in security with a DevSecOps approach that fits the way developers build apps — automated, rapid, continuous, iterative, and auditable.

Five Steps to Getting to DevSecOps for Mobile

The goal is to build security into mobile development as part of an automated agile release

Previous Next

process while meeting the security requirements of a given app. This means integrating security tasks across all phases of the development lifecycle.

Plan

Have a solid plan. Determine risk factors for the app and define the security needed to defend it adequately. Identify gates or checkpoints where you will perform security checks applying to that stage. Security checks should evolve in focus from automated pattern matches to human-driven penetration tests. Additionally, empower developers in security best practices so they understand risk and stay current with the mobile threat landscape — which changes quickly. Lean on resources like the OWASP MASVS and security checklists that everyone can reference and follow. OWASP MASVS can help organizations identify the security risks associated with their mobile apps and develop a plan to mitigate those risks.

Use Secure Coding Techniques

Secure coding is critical. While infrastructure scans and SAST tooling can help secure the environment and avoid common, known vulnerabilities, they aren't enough. To stay ahead of emerging threats, implement ongoing training specifically designed for mobile application development. Effective training ensures that developers build and test their security skills to reduce vulnerabilities, improve productivity, and eliminate rework. Recent research on cybersecurity training benchmarks from the Ponemon Institute indicates that training programs incorporating real-world conditions and featuring role-based content are highly effective and deliver the greatest ROI.

Test Everything in the Build

At the build stage, testing should verify that things have been done correctly up to this point. Test application code, platform usage, API communication, databases, other back-end software, automation files, and certificates and certificate chains. Manually check for vulnerabilities identified in threat models, and use automated tools to check for a broader range of vulnerabilities.

Third-party libraries and components can unintentionally introduce risk. Continuously test your software supply chain,

libraries, code repositories, open source software, and binaries with software composition analysis (SCA) tools to ensure a clean software bill of materials. Also, be ready to apply patches to third-party code when new vulnerabilities are discovered. Test all internally written code with static source code analysis. Test your mobile app build with SAST/DAST/IAST, which tests it like a cyberattack would. A layered approach with current automated tools goes a long way toward a secure app.

Perform full-scope pen testing for mobile apps designed to contain sensitive data, intellectual property, business-critical data flows, or regulated content. If continuous testing is integrated into your pipeline, pen testers can focus on the tough security issues instead of obvious flaws.

Deploy Confidently

Security doesn't stop at deployment. At this point, referencing artifacts and policies created in earlier phases is paramount. Can the top threats identified in the threat model be realized? Do you have what is needed for logging and audit purposes? Is there any threat specific to your deployment configuration that needs further examination before "going live"?

Monitor and Improve

Continuous feedback loops are what make DevOps effective. Besides traditional bug reports, error logs and telemetry data provide valuable feedback to developers. Good sources include:

1. Penetration testing reports
2. Monitoring metrics
3. User-reported issues and bug bounties
4. Vulnerability reports and third-party patch/update notifications
5. Alerts from security intelligence and event management (SIEM) systems

Deploy Confidently

Shifting to DevSecOps for mobile development is a lot easier with someone who has done it before. Save time and accelerate success by leveraging an experienced partner with real-world insight into emerging technologies, cyber threats, attacks, and controls. Now, go make your mobile apps resilient enough to make attackers look elsewhere.