GitHub

Three AppSec pitfalls every security leader can avoid

With web applications causing 70 percent of all security breaches*, secure software is critical for business success today. Security can be easier said than done—thanks to complexity, siloed teams, and slow processes—but it doesn't have to be. Here are some common application security pitfalls every software team can watch out for.

*: 2019 Data Breach Investigations Report, Verizon

1 Security as an afterthought

Developers heavily outnumber security specialists in most organizations. Having developers create code and involving security at later stages in the development cycle is a losing battle because of the high speed and volume of releases. This approach doesn't scale to cover all applications and keeps vulnerabilities from being discovered until it's too late—resulting in vulnerable code being pushed to production.

SOLUTION:

Shift security left and scale security efforts to cover all applications, starting from the early stages of development.

2 Silos and Dev-Sec friction

Traditionally, communication between developers and security teams tends to be issue-driven or incident-driven. But bulk communication by only email, PDF reports, or GitHub issues leads to friction and is frustrating for everyone.

For developers, issues raised by security might not matter for day-to-day development or may include feedback for a project that should've already been finalized. Fixing these issues only adds stress around rescheduling sprint tasks and effort—making security a roadblock for innovation.

For security teams, it's frustrating not to be involved during the architecture, design, and early phases of development. It can also be challenging to explain organizational and application security risks to developers who don't have years of security expertise. In the end, poor communication leads to less collaboration and empathy overall.

SOLUTION:

Make security part of development by integrating tools into your developer workflow. Promote discussions and asynchronous collaboration between both teams.

3 Security as a checkbox exercise

Just like the importance of security varies between organizations, actual security practices also vary within organizations themselves. The difference between formal security policies and how they're put into practice can be confusing and make prioritizing security issues even more complicated.

Also keep in mind that application security is just a small part of an organization's overall cyber security efforts, and tends to be isolated from development and CI/CD. This leads to bad habits like:

- **Valuing quantity over quality.** Focusing on a high number of low-quality security scan results or vulnerabilities doesn't solve larger problems and only adds more work for developers.
- **Making security a development problem.** Plugging raw security scan results into issue trackers and just assuming that developers will fix them all desensitizes developers to security-related issues.
- **Not measuring value.** Just like any other initiative, the value and ROI of application security efforts should be continuously measured and evaluated. Otherwise, a lack of data can hold your organization back—and not show evidence that security improvements are being made.

SOLUTION:

For an immediate fix, focus on pushing a limited number of real security issues, then prioritize and present them to developers instead of sharing a flood of false positives. On a larger scale, look for security tools that can "codify" new security issues and prevent them from ever being merged into a production branch. Remember: your security tools should actually improve your code—so keep measuring the value and impact of your application security program over time.

Questions about application security?

We’re here to help.

sales@github.com
github.com/features/security