Giddy-up, GitOps

4 core software security problems—and what to do about them

Johnathan Hunt
VP of Security, Indeed

In 2020, in the US alone, more than 37 billion records were exposed in nearly 4,000 reported data breaches. What's worse, recent research from Stanford University Professor Jeff Hancock and security firm Tessian found that 88% of breaches are caused by human error.

So what's the problem with security? We have all the right processes, we have dedicated cyber teams of people, and we adopt and integrate the latest and greatest security tools. Yet companies, consumers, and customers still fall victim to code's vulnerabilities.

Your team needs to think about the root of security issues and stop addressing only the branches and symptoms of software security challenges. Here are the four biggest problems facing software security today.

Vague requirements

The first problem is a question of policy and guidance. Currently, there's no specific requirement for the application of security tools and no process standards in existing security frameworks. Most frameworks set guidelines that are entirely conceptual. For example, the official language of PCI DSS security compliance standards simply states: "develop software securely." And PCI 6.3 states: "Develop internal and external software applications (including web-based administrative access to applications) securely, as follows:

- In accordance with PCI DSS (for example, secure authentication and logging).
- Based on industry standards and/or best practices.
- Incorporate information security throughout the software development life cycle."

The issue with loose policies such as these is that they make software security subjective. Security team members and developers applying tools in their approach to security are effectively engaging in a hobby, albeit a serious one, if there are no standard guidelines requiring the use of security tools.

developers develop secure code, companies and security policy makers should introduce specific requirements focused on incremental steps to improving security. The road to creating secure software is long, but progress is only made if companies choose to start somewhere and get going.

Fault is irrelevant

The second issue stems from the way most enterprises think about security. Currently, to most organizations, security is all about the size of the security team and the certifications they hold.

This approach is flawed. Security isn't about team size, expensive tools, or certifications collected; it's much simpler than that. At its core, security is about the vulnerabilities you have and the risk to the organization that those vulnerabilities create. Unfortunately, most enterprises focus on fault instead of long-term solutions, but fault is irrelevant. Holding one set of people wholly accountable for an incident—whether it be developers, security officers, or an organization's leaders—won't fix vulnerabilities, but it will foster resentment and alienation between teams.

Ensuring that security teams are doing more than assigning blame internally requires an attitude adjustment. Security teams should focus on vulnerabilities by evaluating their scope, managing the risks they present, and configuring tools as needed to recognize the problem. In turn, developers should focus on fixing the problem that led to the vulnerability, and leadership needs to hold both groups equally accountable for their roles in security.

Misaligned management

The third problem plaguing software security arises from team alignment. Most organizations' efforts are severely misaligned. Security goals are missed, existing vulnerabilities are allowed to age, and eventually whole workforces find themselves frustrated by what appears to be a never-ending, uphill battle with software security. This feeling of frustration is also another side effect of the "fault model."

This problem is created when separate organizations are given separate goals to work toward. For example, the product organization works toward a release date, while the security organization tries to ensure that the new release is secure. When each player in the development process is assigned an individual goal, they can wind up working in competition with each other.

Instead of collaborating to minimize the risks vulnerabilities create, teams work to ensure that their own individual assignment is completed. The problem here is individuals measure their success based on whether or not their individual assignment was completed. In this scenario,

This problem can be solved by a slight adjustment from leadership. Executives, directors, and managers need to align their individual verticals and teams by prioritizing the same security goals. Once these new goals are in place, leadership can measure success and progress beyond individual teams, because the entire organization is operating on the same guidelines.

Doing better

Lastly, the most important problem facing software security today is each individual member of the development process. It sounds harsh, I know. But it's not a matter of skills or work ethic; it's all in the way we approach software security. The industry has long relied on tools to check vulnerabilities, automate corrective fixes, and expedite tedious audit bottlenecks. However, you shouldn't be relying solely on tools to address vulnerabilities.

Think of security tools like car insurance. All drivers are insured, but they still drive cautiously, because car insurance is a restorative measure, not a safety guarantee. The same applies to the development process. Everybody, from operations to security teams to developers, should drive more cautiously.

This is a tall order, but working toward better security practice doesn't just benefit the company; it also benefits individual workers. For example, developers who can take a proactive approach to security in the development process can make a strong case for a promotion or raise.

Security teams need to act as an anchor by supplying stability. However, in the end, responsibility for improving security doesn't fall on security teams alone, the solution falls on all of an organization's people. Improving software security requires collaboration and cautious coding from all parties involved.

Hope is on the horizon

While all of these problems may sound dire, the situation is not completely doom and gloom. My company's most recent state of DevSecOps survey found that more organizations than ever are prioritizing security across the board and that shifting left is becoming a reality for most DevOps teams.

There's a long road ahead to improving software security, and no $19.99 solution available can replace an organization's willingness to reframe the way they think about building more secure software.

Keep learning

practitioners.

- **Get up to speed fast on the state of app sec testing** with TechBeacon's Guide. Plus: Get Gartner's 2021 Magic Quadrant for AST.
- **Get a handle on the app sec tools landscape** with TechBeacon's Guide to Application Security Tools 2021.
- **Download the free** The Forrester Wave for Static Application Security Testing. Plus: Learn how a SAST-DAST combo can boost your security in this Webinar.
- **Understand** the five reasons why API security needs access management.
- **Learn how** to build an app sec strategy for the next decade, and spend a day in the life of an application security developer.
- **Build a modern app sec foundation** with TechBeacon's Guide.

Read more articles about: Security, Application Security