UNITED STATES

6 security risks in software development and how to address them

Experts share how software development teams can 'shift security left' and improve governance of open source usage, software deployment, and data management.

By Isaac Sacolick
Contributing Editor, InfoWorld
MAR 8, 2021 3:00 AM PST

CIOs and their IT departments face significant business pressure to modernize applications, improve customer experiences, migrate applications to the cloud, and automate workflows. Agile development and devops comprise the cultures, practices, tools, and automations that enable software development teams to achieve these goals and deliver business value with greater quality and in faster release cycles.

The most advanced development teams have fully automated continuous integration and continuous delivery (CI/CD) pipelines with integrated test automation and deploy with infrastructure as code. They connect change management and incident management workflows with agile development tools and use AIops platforms to find the root causes of production issues faster.

Tech Spotlight: Security

- 4 ways to keep the cybersecurity conversation going after the crisis

Yet security issues in software development persist. In ESG's Modern Application Development

(CSO)

- Mitigating the hidden risks of digital transformation (CIO)
- WFH security lessons from the pandemic (Computerworld)
- WAN challenges steer Sixt to cloud-native SASE deployment (Network World)
- 6 security risks in software development — and how to address them (InfoWorld)

Security research, only 36% of respondents rate their application security program a 9 or 10, while 66% said that application security tools protect less than 75% of their codebase, and 48% acknowledged that they push vulnerable code into production regularly.

These security shortcomings are not for lack of technology, consulting, or security service providers. The Cybersecurity Almanac 2020 identifies more than 3,500 potential security partners. Ultimately, the key to delivering business value while minimizing security risks in sofware development is clearly defining security principles and communicating them to software development teams.

Here are six risks that CIOs and IT leaders should focus on and ways to address them.

Risk #1: Not treating security as a first-class devops citizen

It's easy to say the organization puts security first, and many organizations do follow best security practices in agile and devops. But with infosec often understaffed compared to the number of development teams, it's easy to see how other business and technical debt priorities dominate agile team backlogs and why security practices are not adopted uniformly across the organization.

The ESG research supports this conclusion. While 78% of respondents say their security analysts directly engage developers, only 31% review individual features and code. That's a sizable gap, and it's unlikely most organizations can hire enough security experts to have them permanently assigned to agile development teams. But here's what many organizations can do:

- Require ongoing security training and education for the whole software development team.
- Ask infosec to document security acceptance criteria standards in tools like Atlassian Confluence or Microsoft Teams and require agile teams to reference them in user stories.
- Formalize collaboration on agile planning and release management so that infosec can flag higher-risk features and user stories early in the development process.

- Record and publish sprint reviews so that infosec can watch more of them and flag risky implementations.
- Require that all newly developed APIs, microservices, integrations, and applications instrument the required security tests in their CI/CD pipelines.

Defining principles, ensuring cross-team collaboration, improving culture, and promoting team happiness may be the most important ways CIOs can contribute to improving software security. In the 2020 DevSecOps Community Survey, happy developers proved to be 3.6 times more likely to pay attention to security.

Risk #2: Developing proprietary technical implementations

Software development teams love coding and developing solutions, and organizations need their wizardry, innovation, and technical chops to address pressing business challenges. But sometimes the requirements send development teams down the path of solving difficult technical challenges and implementations that they potentially could adopt from third-party sources.

Low-code and no-code can sometimes mean more secure solutions. There are at least two reasons for this. First, agile product owners don't always know the security implications of their top features. Second, many struggle to formulate requirements without dictating elements of the solution, which sometimes leads teams to implement code-intensive solutions that introduce security risks.

Agile development teams should begin by asking the product owner questions about feature priority and negotiate its scope and requirements. One way to do this without being confrontational is to enforce rigor in writing user stories

and estimating them so that complexities get exposed before coding begins.

Once the team agrees on priorities and feature scope, development teams should consider where they can leverage third-party technologies in the implementation. The review should include low-code and no-code platforms, open source libraries, commercial frameworks, public cloud services, and software-as-a-service tools.

Of course, there's no free lunch. Using third-party solutions carries its own risks.

Risk #3: Poor governance and management of open source and commercial components

Have you heard the one about how devops teams are the best equipped to pick their own tools? It's an oft-stated belief from advanced devops teams, and I know of several well-known devops books that promote this principle.

However, many CIOs, IT leaders, and CISOs warn against empowering devops teams with carte blanche decision-making authority over tool and component selection. At the same time, most leaders also acknowledge that too many restrictions and complex approval processes slow innovation and frustrate talented developers. CIOs, IT leaders, and CISOs must define clear and easy-to-follow rules and sensible governance around technology selections, upgrades, and patching.

Recent survey findings illustrate the risks. In a survey of 1,500 IT professionals about devsecops and open source management, only 72% of respondents report having a policy on open source use, and only 64% reported having an

open source governance board. That's only the tip of the problem, as 16% of respondents believe they can fix a critical open source vulnerability once identified.

These results are concerning given the number of reported breaches tied to open source components. In the 2020 DevSecOps Community Survey, 21% of respondents acknowledged breaches related to open source components. It's not just an open source issue, as any commercial system can also have API security vulnerabilities or other software component vulnerabilities.

Clearly defined policies, governance, and management practices around open source usage, tool selection, and technology lifecycle management are needed to mitigate risks. But organizations differ on best practices; some lean toward more openness and others toward less risk tolerance and stricter procedures. To strike a balanced policy between security and innovation, CIOs should establish a multidisciplinary team to define governance procedures, practice standards, tools, and metrics.

Having tools that integrate developer capabilities with security best practices can alleviate some of the challenges of selecting open source components. Jay Jamison, chief product and technology officer at Quick Base, shared this insight regarding Quick Base's approach to innovating with open source:

"We are an early adopter of GitHub Advanced Security, which makes it easier to root out vulnerabilities in open source projects managed on its platform. This is an important step to moving security earlier in the software development lifecycle, or as it's known among developers, shifting left."

Risk #4: Unfettered access to source code repositories and CI/CD pipelines

Securing in-house software used to amount to locking down version control repositories, scanning code for vulnerabilities, defining minimum privileges to facilitate deployments, encrypting connections, and running penetration tests. Locking down the network and infrastructure was a completely separate security realm involving separate tools and disciplines managed by IT operations.

Today, there are more risks and more tools, but also better integrations. I spoke to Josh Mason, VP of engineering at Cherwell, about Cherwell's approach to securing code. "At Cherwell, we layer automated static analysis security testing (SAST), dynamic application security testing, and human-driven penetration testing, which in unison tend to improve productivity. Implementing SAST as part of the CI/CD pipeline moves the discovery process further left in the software development lifecycle, resulting in quicker and less expensive resolutions," he said.

Mason also recommends locking down the version control repository. "Taking guidance from the zero-trust model and the principle of least privilege is a good practice that limits access to source-control repositories and its functions. Source control repository [solutions] such as Azure DevOps, GitHub, Bitbucket, and others provide fine-grained user permissions to limit developers — or whole development teams — to a smaller portion of the codebase related to their work."

Rajesh Raheja, head of engineering at Boomi, a Dell Technologies business, recommends several security disciplines where development teams should take responsibility. "If the software isn't developed properly, the security risk is magnified at a scale far greater than if an individual system was breached. You can mitigate risks by securing the CI/CD pipeline, locking down systems with

the principle of least privilege, implementing secure workarounds for automation with multifactor authentication, driving security awareness within the team members, and developing secure coding practices."

Risk #5: Securing and managing sensitive data

Although many devops teams are versed in security practices for developing, testing, and deploying applications, they must also layer in security practices around data management and dataops.

Chris Bergh, CEO of DataKitchen, explains the issue and an approach to automating more data operations security. "Data privacy and security challenges prevent companies from monetizing their data for competitive advantage. Manual processes can't address the issue — there is simply too much data flowing too rapidly to cope with it. Datasecops is a methodology that automates data privacy and security, integrating privacy, security, and governance into automated workflows that execute alongside data analytics development, deployment, and operations."

The main dataops challenge for CIOs and IT leaders is adopting proactive data governance, labeling sensitive data, and educating developers and data scientists on acceptable data practices. Centralizing identity management, defining role-based entitlements, and masking sensitive data in development environments are important data security and data privacy practices.

Managing sensitive data goes beyond data security. For example, many companies, especially those in regulated industries, must capture data lineage showing who, when, where, and how data changes. These companies often utilize data integration and data management platforms that have built-in data lineage capabilities.

Risk #6: DIY security expertise and solutions

My approach to managing risk and security has always been to seek advice from different experts. Security threats are growing in intensity and complexity, and it's unlikely that most organizations have all the required expertise. Furthermore, when security issues do arise, having a list of people to consult with on reducing risks, addressing issues, collecting forensics, and shoring up vulnerabilities is critical to minimizing the impacts.

Although tools and practices help CIOs address today's issues, we need the experts to help with the next set of security challenges.

Next read this:

- *The best open source software of 2022*
- *Devs don't want to do ops*
- *7 reasons Java is still great*
- *Why Wasm is the future of cloud computing*
- *Why software engineering estimates are garbage*
- *Continuous integration and continuous delivery explained*

Isaac Sacolick is president of StarCIO and the author of the Amazon bestseller Driving Digital: The Leader's Guide to Business Transformation through Technology and Digital Trailblazer: Essential Lessons to Jumpstart Transformation and Accelerate Your Technology Leadership. He covers agile planning, devops, data science, product management, and other digital transformation best practices. Sacolick is a recognized top social CIO and digital transformation influencer. He has published more than 900 articles at InfoWorld.com, CIO.com, his blog Social, Agile, and Transformation, and other sites.

Follow

InfoWorld Technology of the Year Awards 2023. Now open for entries!