CI/CD / DEVOPS / SECURITY

Engineering Solutions to Security Issues

Tools engineers can use to identify security issues early in the software development lifecycle to help achieve a more secure product.

May 24th, 2021 7:36am by Ron Powell

Lead image via Pixabay.

Circle CI sponsored this post.

Software engineering teams have always looked for ways to increase code creation efficiency, reduce code vulnerabilities and improve security processes. Many are now shifting security left, establishing security controls and testing — specifically integration testing — at an earlier phase in the software development lifecycle (SDLC). Developers are incorporating security monitoring and remediation practices as early as possible in development, rather than waiting for testing and quality assurance (QA) to find security issues.

Traditionally, security was separate from software development, with most software testing occurring right before deployment into production environments. This makes issues challenging to fix: You need to identify where specific problems exist in the code, remove the problematic code from your application and replace it with updated code, all without breaking the rest of the application that depends on the original, albeit incorrectly written, code.

In comparison, shift-left testing encourages software delivery teams to test code right after writing individual units of code. This not only helps with early bug and security problem detection, but also helps maintain the health of the software development lifecycle in all stages so that you can release secure, quality applications to market.

Early testing is especially important in software delivery pipelines based on continuous integration and continuous delivery (CI/CD), where

We will look at some tools engineers can use to identify security issues early in the software development lifecycle, then tools to help engineer solutions for a more secure final product.

Identifying Security Issues

Engineers can use tooling to discover security issues early, often in real time as they write code. This includes static analysis tools, which scan as the engineer writes code, then flag any security issues in the engineer's integrated development environment (IDE) or editor. That way, the engineer is aware of the issue immediately and can fix it right away.

Developers typically use static analytical methods to design and test components. In this instance, the code is not running or executed, but the tool itself executes using the source code as its input data.

Static code review software also helps developers understand structure and coding standards and implement their own coding standards. As developers push code directly into production, static code analysis also validates code quality, decreases later errors and reduces bugs. This upfront embedded approach to security also makes it easier and faster for the QA team to perform functional and performance testing later on.

Complement static code analysis with static secure code review. Security audits test the environment and coding practices security to ensure that developed apps are resilient to operational and environmental threats.

Dynamic analysis tooling runs an engineer's code and probes it for

can analyze and identify potential issues arising during the application's actual execution and that impacts its reliability. Dynamic code analysis tools help debug running threads and processes. They also highlight performance problems, memory use issues and memory leaks.

A modern CI/CD pipeline checks feature branches the engineer is working on and runs detailed scans to find any security issues. This uses more powerful static and dynamic analysis tools than the engineer can run locally. Vulnerability scans also check for security issues in all the container images and libraries the engineer uses to build the application.

CI/CD pipelines check code for any security flaws continuously: during staging, during production, when the code is live or when the code deploys. This saves time and effort scanning and fixing bugs after release.

Engineering Security Solutions

Developers feel pressured to ship fast and often, and as a result, are loath to introduce tasks that slow their process. However, static code analysis identifies security problems early, including common security vulnerabilities identified by the Open Web Application Security Project (OWASP). This eliminates the costly and time-consuming task of fixing errors that appear downstream.

Fixing problems at the eleventh hour risks either delaying release dates or shipping code with errors and bugs passed to the end user. This may require emergency patching or even recalls in certain situations, contributing to a poor customer experience and loss of sales.

Software engineers can solve many security issues early in the development process if they have timely, automated feedback about what is wrong. The exact solution an engineer applies depends on the nature of the problem and how it was discovered.

Static analysis tooling tells the engineer exactly what code they need to change. However, dynamic analysis tools tell you what went wrong but not exactly why, making it challenging to prevent future occurrences. Still, sending early feedback to the developer shifts security to the left and forces software engineers to develop a beneficial security mindset.

You can integrate purpose-built security tools with existing DevOps practices, reducing the impact of adding another process. Security scans and checks trigger within an IDE when a pull request executes or as an additional step in secure CI/CD pipelines.

gets a report of anything that went wrong so they can fix their code immediately. This rich and contextual information provides actionable insights, helping improve an organization's overall security practices.

How Shift-Left Security Works

Say an organization practices engineer-led early security issue analysis and remediation. One day, a routine security scan finds a vulnerability in an image.

When they see this alert, the engineer rolls forward to an updated version of the image and discovers where this breaks the code. They then build a solution that solves the breaking changes, but maintains the integrity and intention of the original code.

Just before production, this image vulnerability would be harder to detect and more challenging to fix. However, with this early scan in place, the developer was able to quickly fix the issue long before the software reached QA.

Conclusion

Shifting security left creates novel solutions to security issues, by exposing the problem to the engineers developing the app in real time. This enables the security team to work on other high-level priorities and provides an opportunity for developers to increase their knowledge and experience of good security in action, which is critical as many organizations struggle to fill security roles due to a lack of appropriately skilled staff.

requirements from early in the build phases, they enhance their agility, maintain their integrity and ensure the speedy deployment of secure applications.

CI/CD changes developer team culture. When it comes to developer team success, finding the right DevOps metrics to measure is crucial. Learn how to measure DevOps success with four key benchmarks for your engineering teams in the 2020 State of Software Delivery: Data-Backed Benchmarks for Engineering Teams.

TNS

Ron Powell is senior manager of Marketing Insights and Strategy at CircleCI producing content that enables developers to build, test and deploy their projects faster. He has a background in space physics, having worked as a Cassini team member analyzing...

Read more from Ron Powell →

RELATED STORIES

4 Tips to Avoid Downtime Risks and Start the New Year Strong

25 Most Popular Programming Languages Used By DevOps Pros

Cloud Cost-Optimization Strategies to Rein in Cloud Spending

Is a Multicloud Strategy Right for Your Organization?

INSIGHTS FROM OUR SPONSOR

circleci

CircleCI is the leading continuous integration and delivery platform for software innovation at scale. With intelligent automation and delivery tools, CircleCI is used by the world's best engineering teams to radically reduce the time from idea to execution.

Learn More →

Solving the top 7 challenges of ML model development with CircleCI
29 June 2023

Mocking API requests with Mirage
26 June 2023

Reduce cycle time with effective pull requests
20 June 2023

A guide to dynamic application security testing (DAST)
9 June 2023

Zero trust security for CI/CD pipelines
1 June 2023

A guide to static application security testing (SAST)
24 May 2023

the week's most important stories & analyses.

EMAIL ADDRESS

SUBSCRIBE

ARCHITECTURE

Cloud Native Ecosystem
Containers
Edge Computing
Microservices
Networking
Serverless
Storage

ENGINEERING

Frontend Development
Software Development

FOLLOW TNS

Machine Learning
Security

OPERATIONS

Platform Engineering
Operations
CI/CD
Tech Life
DevOps
Kubernetes
Observability
Service Mesh

CHANNELS

Podcasts
Ebooks
Events
Newsletter
TNS RSS Feed

THE NEW STACK

About / Contact
Sponsors
Sponsorship
Contributions

FOLLOW TNS

FOLLOW TNS

help you choose your path and grow in your career.

Frontend Developer Roadmap
Backend Developer Roadmap
Devops Roadmap