VERACODE

Whitepaper

Leveraging Automation to Achieve DevSecOps for Secure Web Applications and APIs

Strengthen your software against evolving threats while maintaining the speed and flexibility of modern development practices.

Table of Contents

Modern Software Development Requires Automated Security Testing

Traditional security testing methods are no longer sufficient for modern development workflows. The days of year-long projects with manual security checks are over, unable to keep up with frequent software releases, which often occur multiple times a week.

To adapt, security testing must be automated and seamlessly integrated into the software development lifecycle. Quality management principles, such as the rule of ten, highlight the rising cost of fixing vulnerabilities in later development stages, emphasizing the critical importance of early detection and remediation when issues are more cost-effective to address.

In modern software development, automated security testing becomes vital to ensure software is built securely. Unlike manual testing usually reserved for major releases, automated security tests not only find vulnerabilities earlier in the development lifecycle but also help save time and reduce costs while improving overall security posture.

Security Challenges in Modern Software Development

Teams face several security challenges when using modern software development process in today's in today's business world. The following chapter outlines the most critical challenges.

- Importance of time-to-market
- Wide range of security tools
- Lack of security expertise
- Sole reliance on frameworks

1

Functionality and time-to-market are more important than security

Continuous integration and continuous deployment (CI/CD) pipelines are more responsive to customer needs and help shorten development cycles. However, usually this increased pressure to deliver more software faster leads to negligent security testing.

Research indicates that security is not always seen as a business-critical concern. Security testing is typically conducted only when there are specific customer requests or when the perceived risk in releasing significant software versions deems it necessary. Consequently, version 1.0 undergoes a manual penetration test (pentest), and subsequent vulnerability testing commonly occurs only for version 2.0. Minor versions are typically released without undergoing additional security tests.

Essentially, the security level is based solely on the skills of the software developers. This is due to two reasons:

- Security testing and remediation of results takes time.
- Security testing, especially manual testing, is costly.

To launch software on time and within budget, the developers' focus is on creating features, not on increasing the security of the software.

2

Wide range of security tools covering only individual vulnerabilities

A second challenge for today's software developers is the multitude of different tools available on the market. A specific test for almost every security vulnerability checks that one vulnerability very well but only covers one vulnerability.

For example, SQLMap is a great tool to find SQL injection vulnerabilities, but that is only one of the attack vectors hackers use. If a developer uses multiple of these scanners, they face the next challenge.

Each scanner must be individually configured and customized to meet your needs. The scanners are based on different programming languages and often come with other requirements such as system libraries, so it takes the developer a lot of time to set up the internal security scanners. In addition, when there are several different security tools, developers must decide which one to use. Since they often do not have specific security training, evaluating the tools is complicated.

Moreover, the output formats differ from tool to tool, so a vulnerability management solution also comes into play, which must be configured and set up. If the developer wants to run a security scan, he must then run each scanner individually and consolidate and standardize the results by hand. He must also ensure that each scanner is always up to date to detect the latest vulnerabilities.

3

Security expertise is not easy to come by

Another problem that mainly concerns management is the lack of qualified software developers - especially in security.

In particular, it is challenging to recruit the experts they need in security management. This is another reason why security testing is often not carried out to the required extent.

Even when developers see the urgent need for security products, they often lack the skills to operate various security tools and interpret the results. Learning these skills, in turn, prevents developers from creating new features for their software.

4

Managers and developers must be able to rely on each other

Managers often don't have the time or expertise to review and assess the security of the software they develop, so they must rely on their software developers to create a secure applications. However, developers are under additional pressure to develop one feature at a time.

As a result, they have to rely on the frameworks they use to secure their code and have no security themselves. This only leads to the illusion of a secure application, which is not tested.

Practical Examples of Integrated Security Measures

It is crucial that security is embedded in existing systems and structures so that the security layer does not hinder users. A practical example is a revolving door that only allows one person to pass at a time. This gives security personnel a clear picture of this part of the building's security, while the doors are not perceived as security measures by their users.

Ease of integration is one of the prerequisites for security measures to be successfully implemented in modern development practices. When WhatsApp introduced end-to-end chat encryption overnight without users noticing, it increased the security of its' millions of users' communications.

As a widely accepted form of communication, there were no delays or additional installations to enable the security feature, so users didn't have to change anything in their application.

Implementing such usable security in software development helps you meet deadlines more consistently. It also improves inter-departmental relations as conflicts between the software development department, and the security department can be more constructive and empower software developers.

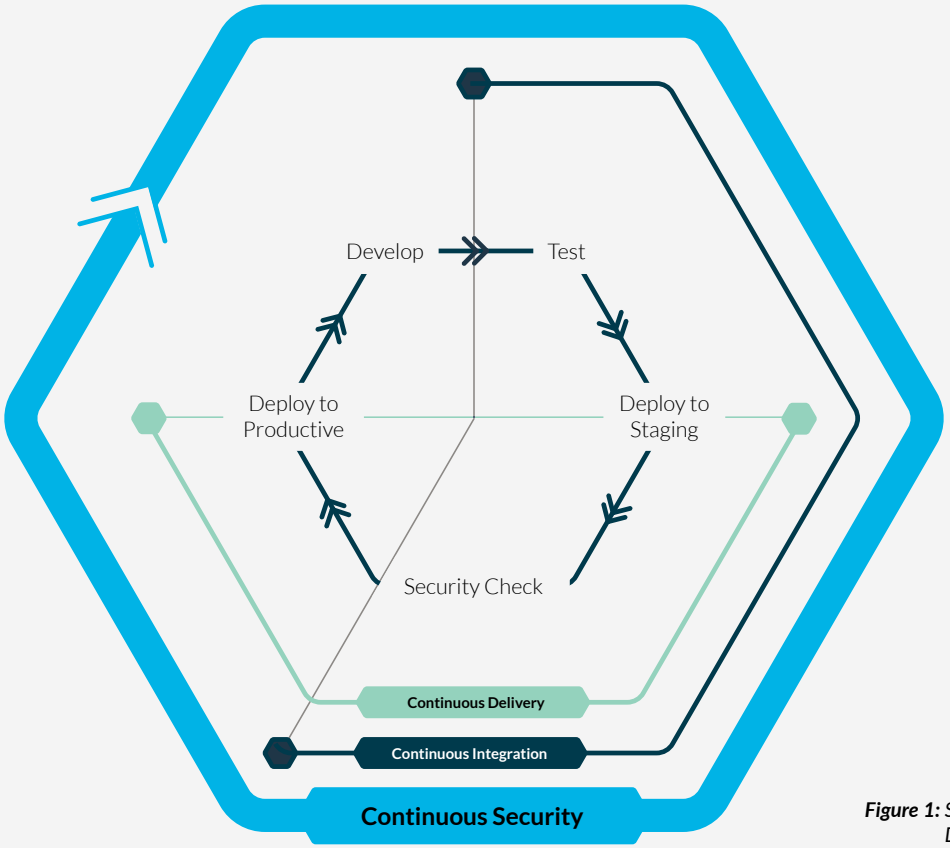


Figure 1: *Security in the Agile Development Environment*

Steps to Integrate Automated Security Testing Seamlessly into Your Workflows

One way to solve the mentioned problems is integrating security tests into the CI/ CD process (see Figure 1). To successfully incorporate such scans into the daily routine, several aspects need to be considered:

- Safety tests complement other forms of testing
- Integration into the CI/CD process is essential for continuous security

Security testing must integrate seamlessly with the current development environment so that software developers do not have to leave their familiar environment.

A normal CI/CD process includes a build server that triggers certain functional tests when specific actions occur (e.g., the software is deployed). This may mean that certain unit tests are run each time a push is made to a repository. When a pull request is created or merged, additional tests such as integration tests may be run. The results are usually collected, presented in the build server‘s user interface, and pushed to different communication media such as a Slack channel.

Implementing functional security tests means that the build server automatically triggers the security tests based on the developers‘ configuration. Automated tests can additionally be triggered at specific time intervals to ensure secure software.

One way to implement this is to scan all nightly builds with the automated security scanner. Just as a unit and integration testing have become standard tools for modern software developers, security testing must be fully integrated into their daily workflow.

The security scanner must bundle security scans for the developer and make sense of identified vulnerabilities. This ensures that developers whose core competency is not security engineering are not hampered by the need to integrate multiple tools or interpret command line output from security scans.

Automated Security Testing Empowers You to Deliver More Secure Software, Faster

Instead of security being an obstacle for developers, they are adequately supported with integrated and automated security scans and get near-instant feedback. For example, after a pull request is created and the code is deployed to a test or staging environment developers still know what code they worked on in the hours or days before and don't have to re-dive into previous work.

Proper testing provides feedback on existing vulnerabilities and provides links to resources or guidance on how to resolve an issue. This level of quality control gives developers back sovereignty over the security of their code. Instead of creating manual security tests, they can focus on their revenue-generating work, like developing new features.

In addition, these tests can free up cognitive resources that are tied up by supervisor pressure. Because a software developer's job is often seen as just a feature producer, (non-tech) managers often see it as a waste of time to put too much work into things like code quality or security.

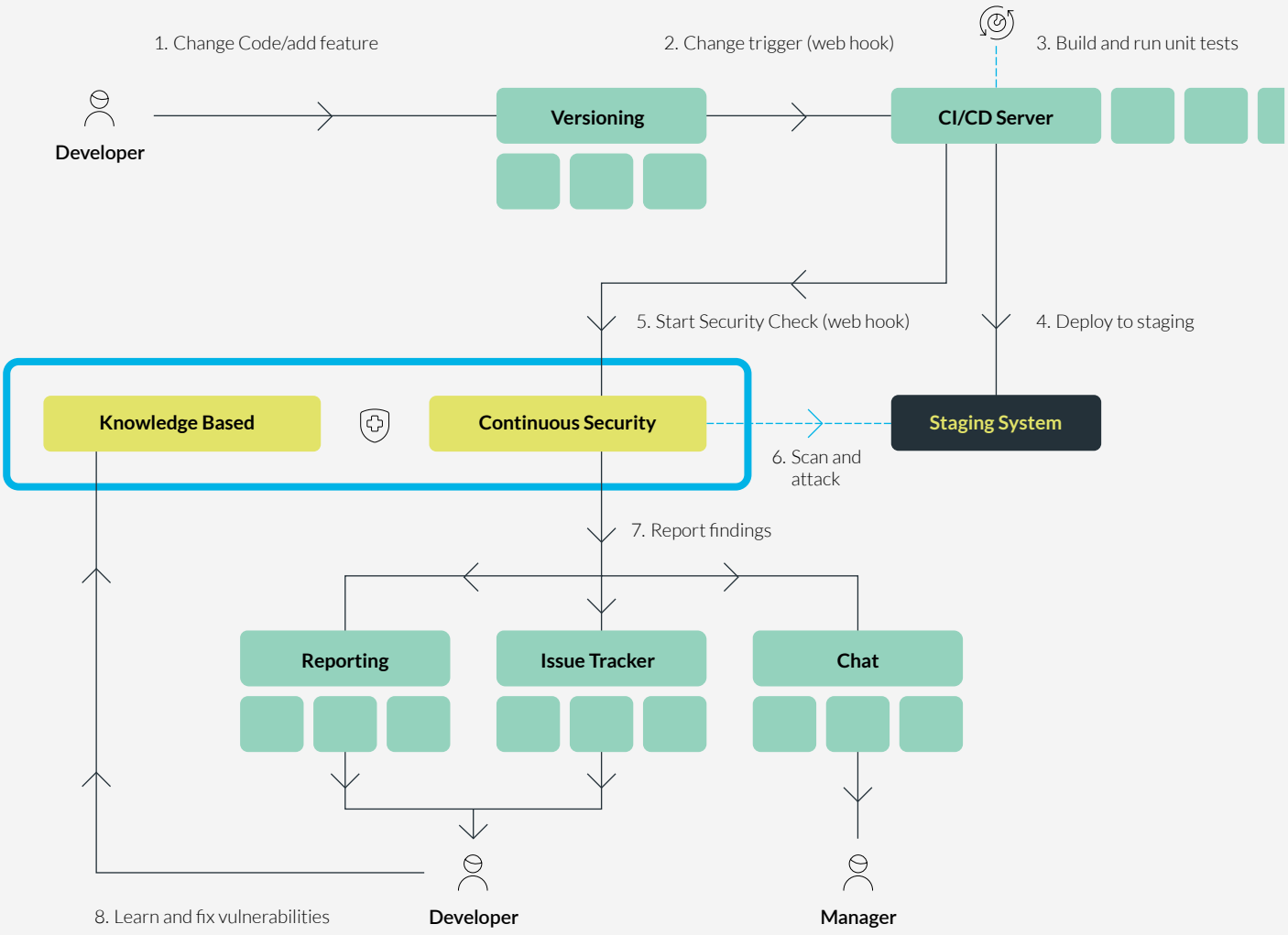
However, in a security emergency, the immediate responsibility for fixing vulnerabilities lie with the developer who implemented the code that contains the vulnerabilities. An automated security scan gives the developer a great tool to continuously check the security status of the software while focusing on their core tasks.

Once automated security is integrated into the daily workflow, developers can independently verify that their code has passed a security scan with each released product version. In addition, a solution that always includes the latest security checks - such as a SaaS security scanner - ensures that any newly discovered vulnerabilities (known as zero-day attacks) are also included in the scan.

Secure Innovation: Seamless Security, Uninterrupted Workflows

Verocode helps you detect web application security vulnerabilities in real-time. With a cloud-based security scanner that is offered as a SaaS model. We ensure that developers can fully concentrate on their revenue-generating work while we detect the vulnerabilities. As software developers ourselves, we know how difficult it can be to produce software with a high user experience and all the features you need - and to do it all securely.

That's why we're writing a toolchain of security scanners that a webhook can trigger. Reporting is done so that developers can focus on solving security problems, and executives can focus on monitoring the software security state. Integrations for existing communication channels, such as Slack, inform managers as needed. If a vulnerability is found, the developer is notified immediately (see Figure 2 for a detailed explanation of the process).



Achieving DevSecOps with Automated Security Testing

Automated application security testing that integrates seamlessly into the development process helps developers find and fix vulnerabilities as early as possible.

This way, developers can deliver more secure software faster. The time saved by targeting security issues as they arise (rather than after weeks or months of productive use) allows tdevelopers to focus on what they do best - writing code.

Automated and integrated security testing are crucial aspects of a successful DevSecOps program. And Veracode can help you empower your teams to integrate automated security testing seamlessly into their workflows, ensuring the delivery of secure and high-quality applications.

Start Free Today

VERACODE

Veracode is a leading AppSec partner for creating secure software, reducing the risk of security breach, and increasing security and development teams' productivity. As a result, companies using Veracode can move their business, and the world, forward. With its combination of process automation, integrations, speed, and responsiveness, Veracode helps companies get accurate and reliable results to focus their efforts on fixing, not just finding, potential vulnerabilities.

Learn more at **www.veracode.com**,
on the **Veracode blog** and on **Twitter.**