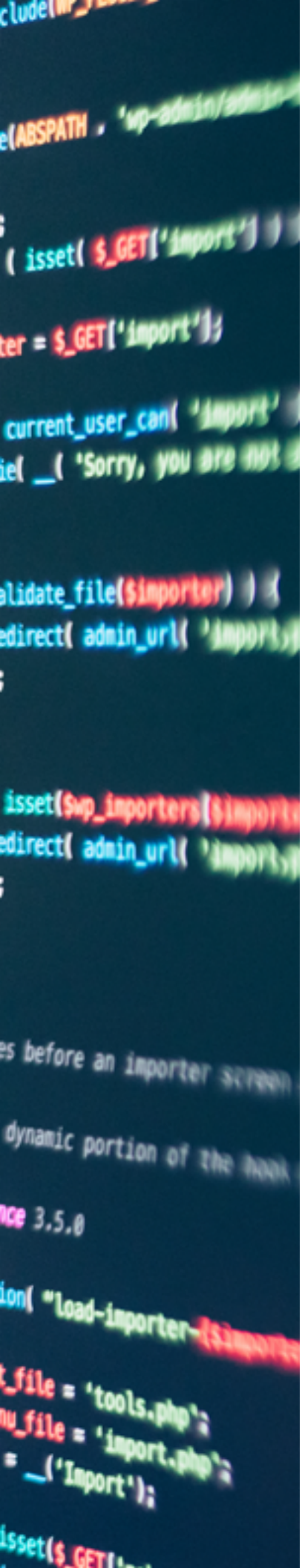
SECURITY COMPASS WHITEPAPER

The Developer's Dilemma

Fast and Risky or Slow and Safe?

Copyright © 2020 Security Compass. Updated April 2020.

Security Compass



The Developer's Dilemma

For years, product management and software development have battled over adding more features or delivering software faster. Faster software time-to-market helps companies gain competitive advantage, so businesses want to release new features quickly. This led many companies to adopt rapid development methodologies to accelerate delivery of new features.

However, to speed up product releases, companies usually compromise on security. Failing to secure software and protect customer privacy is one of the greatest business risks today. This happens because integrating security is perceived as a lengthy process requiring scarce security experts to work with developers daily. Even when companies decide to inject security into development, scaling manual security processes continues to be a challenge.

This seemingly competing goals of speed vs. security result in a developer's dilemma: whether to go fast and risky or slow and safe.

Pressure to go slow and safe

Building secure software is critical for organizations — it's also the right thing to do. Software security professionals want to ensure that the code they write is secure, and the users of software, whether they are banks, military, or consumers, depend on it to protect their sensitive information. When a breach occurs, a company's credibility and brand reputation is on the line, impacting the entire business, not just that of the DevSec team. More recently, software security has become a board-level issue; the Equifax breach in 2017 reduced the company's market capitalization by over 30 percent and resulted in the forced retirement of its CEO, CIO, and CSO. The company's reputational damage continues even today.

Pressure to go fast

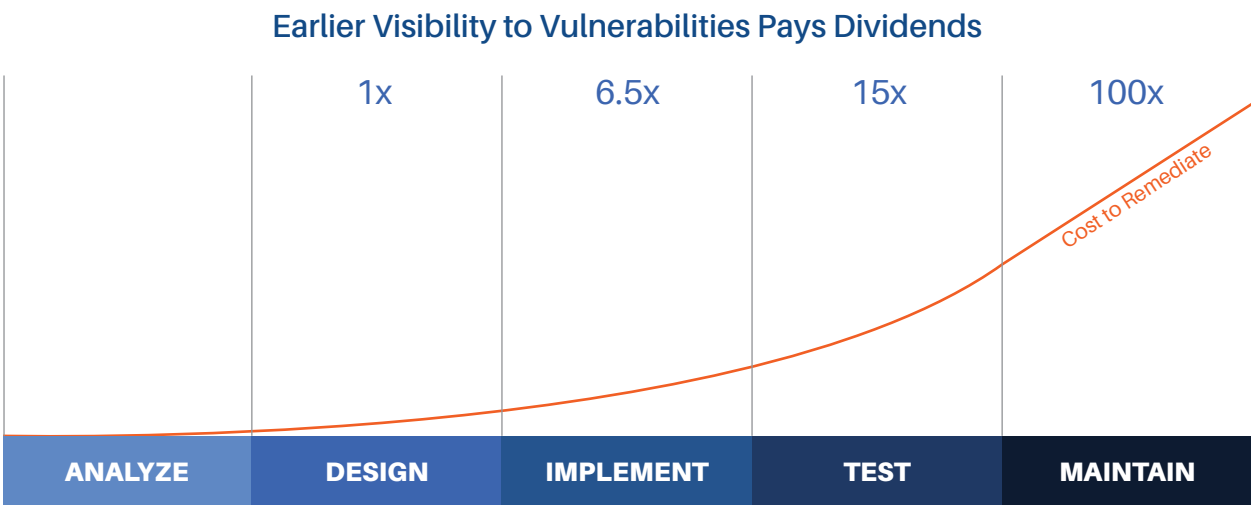
As software development strategies have changed over the past 10 years, so too have strategies for securing software. When software development primarily followed a waterfall methodology, security testing relied on dynamic analysis and penetration testing of the application very late in the development process.

With the growth of software-enabled features, development strategies have evolved. Organizations that can deliver new features faster than competitors can gain market share and reduce customer turnover. Methodologies like Agile, DevOps, and CI/CD accelerate time to market and require a different approach to security. The adoption of open source and the ineffectiveness of traditional testing tools for identifying vulnerable components means a changing attack surface and a wider variety of testing tools.

From the security professional's perspective, the old approach of a separate security testing team using static or dynamic analysis on near-complete applications no longer works. Pressure to deliver new features faster means security needs to be part of the development process – building security into the software instead of testing for security at the end of the development lifecycle.

The financial argument for building security early

While testing for vulnerabilities late in the development lifecycle undoubtedly identifies vulnerabilities, there are two negative consequences. First, since a near-complete version of the software is required for this testing, vulnerabilities discovered are more difficult and expensive to remediate. As shown in the graphic, a study by IBM showed that vulnerabilities identified after a product was released cost 100 times as much to remediate as those identified – and avoided – during the design phase of the Secure Development Lifecycle (SDLC) and over 15 times as much as those identified during the coding phase.



Source: IBM Systems Sciences Institute

To be clear, this does not mean that if a developer could implement a code change in under a day during the coding phase, it would take her 2 ½ weeks of work after the software was released. Vulnerabilities identified post-release involve more resources. There are security personnel who analyze the vulnerability to determine if it is exploitable and triage meetings involving lots of people to discuss and prioritize issues. Once the vulnerability is scheduled for remediation, the code must be refactored, test cases generated, and code retested to confirm the vulnerability was fixed. All of this time and effort adds up even if the vulnerabilities are from coding errors. If a design flaw results in a security issue that is not identified until late in the development lifecycle, the costs can be enormous.

The second challenge with vulnerabilities discovered late in the SDLC is organizational pressure to release the software. As the SDLC reaches its end, the applications are close to their scheduled release dates. Taking time to fix issues can result in lost revenue from missed customer commitments and delayed releases. When the choice is losing revenue or shipping vulnerable software (with a plan to remediate in a future release) the latter
will often prevail.

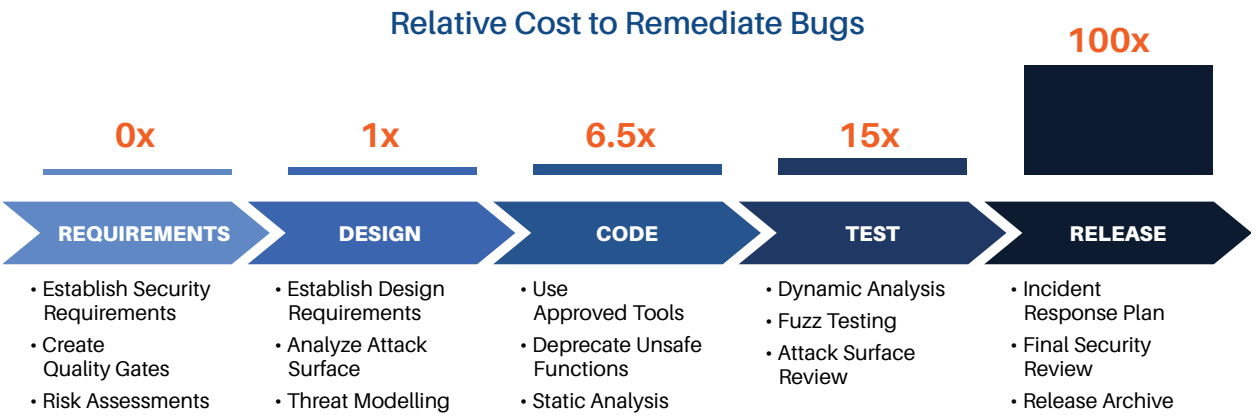
Solving the Developer's Dilemma

Preventing vulnerabilities is better than fixing them, and there are many security activities organizations can perform to identify security issues early in the development lifecycle.

Building secure software is simplified by understanding the threats software faces and implementing secure coding policies to mitigate those threats. Traditional threat modeling can also identify threats, but it can be a very time-consuming process that can't be scaled.

But, it doesn't have to be this way, most threats are inherent to the software's technical stack which can be mitigated through proven strategies.

Automated threat modeling with SD Elements scales the process without adding additional security resources. SD Elements identifies foundational threats in an application's technology stack, including the programming language and frameworks, the deployment environment for the application, and internal policies or regulatory standards to which the application is subject to. SD Elements then provides controls for the threats or standards and translates those into specific activities for developers and security teams, including test plans for validating the implementation of those controls. This provides security, DevOps, and non-security functions with consistent guidance on how to build secure software without slowing down development. SD Elements covers all aspects of the SDLC including configuration of the application infrastructure, risk assessments, and compliance and privacy controls.



Source: Filling your Appsec Toolbox. Which Tools, When to Use Them, and Why - Michael Pittenger

Go Fast. Stay Safe.

Testing for security is different from building secure software. The former is reactive and produces erratic schedules while the latter is process-driven and predictable. While security testing is an important step in any SDLC, proper planning anticipates security issues and allows organizations to avoid common vulnerabilities and weaknesses. This means that security testing is primarily validating that prescribed controls were implemented correctly instead of acting as a primary vulnerability discovery activity.

The result is a balance between speed and security. SD Elements allows companies to build products nearly as fast as if they were being built without any security or compliance at all and as safely as if it were built under the guidance of human experts. SD Elements helps organizations lower risk across all applications and enforce security policies without adding resources.

SecurityCompass

Security Compass, a leading provider of cybersecurity solutions and advisory services, enables organizations to adopt balanced development automation for rapid and secure application development. With their flagship product, SD Elements, the company helps automate significant portions of proactive manual processes for security and compliance that improves time to market for new technology. In addition, they offer advisory services on how organizations can embrace emerging technologies like cloud to strengthen their security posture. Security Compass is the trusted solution provider to leading financial organizations, technology enablers, and renowned global brands. The company is headquartered in Toronto, with offices in the U.S. and India. Follow Security Compass on Twitter @securitycompass or visit them at securitycompass.com to learn more.

1.888.777.2211
info@securitycompass.com
www.securitycompass.com

@SECURITYCOMPASS
SECURITY COMPASS

OFFICES

GLOBAL HEADQUARTERS
1 Yonge Street
Suite 1801
Toronto, Ontario
Canada M5E 1W7

TORONTO
390 Queens Quay W
2nd Floor
Toronto, Ontario
Canada M5V 3A6

NEW JERSEY
621 Shrewsbury Avenue
Suite 215
Shrewsbury, New Jersey
USA 07702

CALIFORNIA
1001 Bayhill Drive
2nd Floor
San Bruno, California
USA 94066

INDIA
#4.07
4th Floor, Statesman House
Barakhamba Road, New Delhi
India 110001

Author; Michael Pittenger

Copyright © 2020 Security Compass. Updated July 2020.