

SECURITY COMPASS WHITEPAPER

# Enhancing Cybersecurity in Financial Services





## The first mover advantage

The “first mover” concept holds that the organization first to provide a new product or service to market will gain market share, better brand recognition, and customer loyalty.

Obvious examples of this in the online world include Amazon, that changed forever the way books are sold, then changed the way everything else was sold; Netflix (streaming services); and eBay (auction sites).

Similar disruptions are occurring in financial services, where consumers now embrace online banks like Ally and Chime and brick and mortar “retail” banking is less of a factor. In this world, software-derived features increasingly set apart organizations. The ability to deliver new features, faster than competitors, is a critical competitive differentiator and can provide substantial operating efficiencies.

## Speed to market in the financial services industry

Software development strategies have evolved to meet changing demands. Competitive pressure requires financial services organizations to respond quickly to customers’ needs. Rather than introducing quarterly software updates, it is now common for development and operations to push dozens of updates to production every day. Large banks and insurance companies that employ thousands of software engineers, QA, and operational professionals, and manage hundreds or thousands of custom applications have shifted from waterfall methodologies to agile, DevOps, and Continuous Integration/Continuous Delivery to introduce required features far more quickly.

## Go to market strategies

When faced with requirements for a new feature, product and engineering teams have several different options, each with advantages and disadvantages.

- » **Proof of Concept:** The fastest time to market option is a proof of concept (PoC). A PoC gives users rudimentary functionality and allows them to test the utility of a feature and provide feedback on design and execution.


PoC's are, by design, prototypes and not enterprise-ready releases and are often limited to internal users. Depending on the organization and exposed attack surface, they may or may not be reviewed by security teams or thoroughly tested for quality and performance issues.

- » **Minimum Viable Product:** A Minimum Viable Product (MVP) offering expands requirements to include the least possible features and functionality needed for a commercial offering. The MVP allows organizations to deliver new products and features quickly, but typically with the intent of adding more functionality in subsequent releases.

An MVP meant for a production environment may be internet-facing, greatly increasing exposure to adversaries. These applications will typically undergo standard security testing. However, because the applications are developed in a compressed development cycle – often to exploit the first mover advantage – teams may face pressure to release applications with vulnerabilities identified by traditional security scanners late in the development process.

- » **Enterprise Quality:** From a security and quality standpoint, it is preferable (when possible) to build and test software thoroughly prior to release. Enterprise quality applications are full featured (though enhancements are always on the product roadmap) and receive full quality and security testing regimens.

While enterprise-quality applications are fully tested, this testing – like that for PoC and MVP software – occurs late in the development lifecycle. This means that security bugs and design flaws identified by static and dynamic analysis are more expensive to remediate and teams will face pressure to release software.



## Security and compliance can introduce speed bumps

As time to market has become more critical, so too has security. Financial services firms are a top target for hackers. The combination of personal information, access to funds, and complex systems that require constant patching means financial services organizations are **300 times more likely** to face a cyberattack than companies in any other industry.

Financial services firms are also heavily regulated. Standards and frameworks must be considered, including the Payment Card Industry Data Security Standard (PCI DSS), Gramm-Leach-Bliley Act (GLBA), and the FFIEC guidelines in the US, the OSFI standards in Canada, the European Banking Authority (EBA) in the EU. Some of the requirements, like PCI DSS, are very prescriptive and specify which controls are required and the classes of vulnerabilities for which organizations are obligated to test. Others, like the California Privacy Protection Act, Maine's Act To Protect the Privacy of Online Customer Information, and the proposed federal Consumer Online Privacy Rights Act are less specific and simply require organizations to protect any personally identifiable information they collect.

In addition to external requirements, most financial services firms also have internal security policies with which development teams must comply. These require various levels of testing and security controls based on the criticality of the application to the organization's goals and the threats the application faces by virtue of its deployment environment.

Determining which requirements and policies apply to each project, and which controls need to be implemented to mitigate risk is a necessary activity. In today's rapid development environment, security and engineering teams struggle with the perceived trade-off between security and time to market.

## Is security testing enough for applications?

Security testing and ensuring compliance with overlapping and changing internal and external requirements can challenge the most mature organizations. Verifying that all requirements are met, and all controls are implemented properly using spreadsheets and traditional testing methodologies like static and dynamic analysis is inefficient and incomplete. **These tools are reactive** and designed to identify coding errors later in the development lifecycle, not prevent them from entering the codebase. Further, these tools can require hours to run and produce “noisy” results that require review, introducing friction and slowing development.

There are well established methods for building secure software. They require organizations to understand which internal and external policies apply to each application, identify security requirements in addition to product functional requirements, identify likely threats to the application, then identify and implement appropriate controls to mitigate risk. These are perceived to be

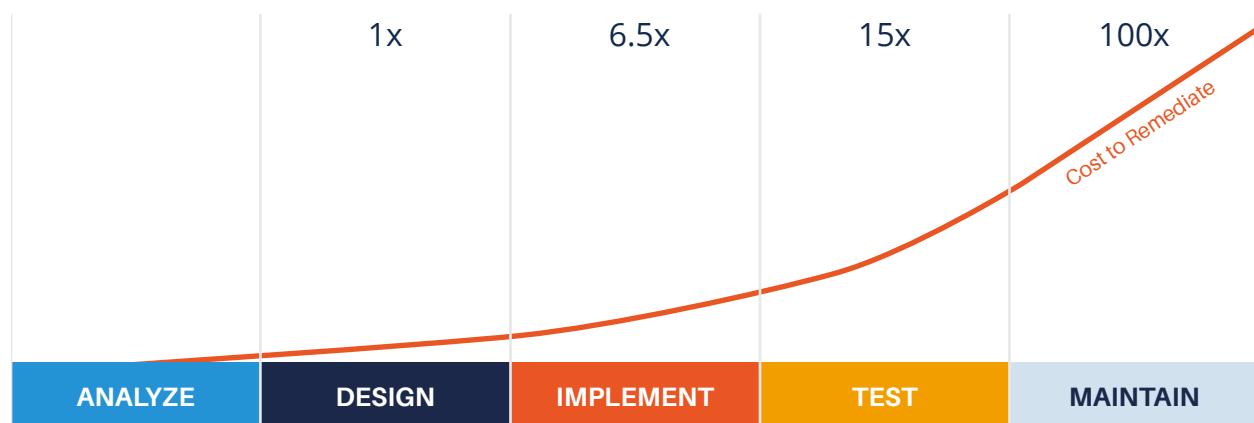
activities that delay product development when compared to building applications then testing for vulnerabilities. The opposite is true.

## Reactive security slows development

Application Security Testing technologies like static and dynamic analysis identify common coding errors that can result in security vulnerabilities. These include issues like failing to validate untrusted input from users and other systems, weak cryptography, and improper access control.

While finding these issues prior to releasing software is good, the cost to remediate vulnerabilities found late in the development process is far higher than if they could be found or prevented earlier in the process. 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.

### Earlier Visibility to Vulnerabilities Pays Dividends



Source: IBM Systems Sciences Institute

## Ancillary costs of security testing

The time spent by development refactoring code to fix vulnerabilities is only a portion of the additional costs, of course. Teams will meet to prioritize issues, QA must generate new test cases, and code will be rescanned to ensure that the issue was fixed (and nothing else broken). Before any of this happens, however, scan results need to be triaged to eliminate false positives and insignificant issues, both of which are [common with static analysis](#).

Research by Grammatech found it takes an analyst [10 minutes to triage a single finding](#). This means a scan producing 1,000 findings can take over 20 labor days to triage.

Excess false positives are, in part, due to the number of rules that organizations use in the static analysis tools. Thousands of rules are available covering hundreds of types of vulnerabilities. Security teams often run all the available rules to find any possible security issues. In many cases it can be more efficient to focus on those classes of vulnerabilities required to be checked by policy and those that have been prevalent in applications scanned previously. This is only possible, of course, if other controls are in place to prevent vulnerabilities from entering the code base.

## Proactive security accelerates development

Organizations can build secure software more quickly by anticipating threats and weaknesses then building controls into the development process. Traditional threat modeling is one way of accomplishing this, but the time required from architects and scarce security resources

makes this practical of only the most critical projects.

A faster and nearly as effective alternative is to classify the application for criticality, determine which internal and external standards apply, and map the project's technology stack and deployment environment to known threats associated with those factors. This allows teams to [identify up to 90 percent of the threats an application faces](#) in a fraction of the time. For example, if a team is modeling a web application for which a user needs to authenticate to the system, several threats and controls can be identified irrespective of the purpose of the system, including:

- » An attacker may attempt to learn user credentials by logging into the system. On failed logins, don't provide more information than necessary about what was incorrect.
- » An attacker may attempt brute force attacks to guess passwords, therefore ensure that the system only allows a fixed number of failed logins for a fixed period.
- » A man-in-the-middle attack could capture login credentials, therefore ensure that login pages use HTTPS.
- » An attacker may trick a user into revealing their credentials. A requirement for two-factor authentication would mitigate this risk.
- » An attacker could "shoulder surf" a legitimate user to steal a password. Masking passwords by default would increase the difficulty of this tactic.

These threats and controls require knowledge of the project's functionality, information managed, technical stack, applicable regulatory standards, and criticality to the business; all readily available information that requires no understanding of data flow, trust boundaries, or attack trees. Once the threats are identified, controls to mitigate risk can be mapped to each item and assigned as part of the project requirements.

## Go Fast. Stay Safe.

First movers gain a competitive advantage by delivering innovative features quickly. In today's environment, it is not acceptable to prioritize speed over security. By anticipating threats and assigning controls, organizations can achieve both. With proactive planning, security testing becomes a validation exercise to ensure that assigned controls were completed correctly.



# SecurityCompass

**Go Fast. Stay Safe.**

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](mailto:info@securitycompass.com)

[www.securitycompass.com](http://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  
07702, USA

### CALIFORNIA

600 California Street,  
San Francisco, California  
94108, USA

### INDIA

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