Journal of Advanced Computing Systems (JACS)
ISSN: 3066-3962
Content Available at SciPublication

DevSecOps in Fintech: A Maturity Model for Integrating Security into the AI-Driven SDLC

Utham Kumar Anugula Sethupathy[1], *Vijayanand Ananthanarayanan*[2]
[1]*Independent Researcher, Senior IEEE Member, Alumni, Nanyang Technological University, Atlanta, USA*
[2]*Independent Researcher, Alumni, Fairleigh Dickinson University, Atlanta, USA*
DOI: 10.69987/JACS.2023.30805

Keywords

DevSecOps in Fintech, Financial Technology, DSOMM, Artificial Intelligence

Abstract

The financial technology (fintech) industry operates under a dual imperative: the need for rapid innovation to remain competitive and the non-negotiable requirement for robust security to protect high-value assets and comply with stringent regulations. Traditional, siloed security models fail to meet these demands. This paper argues that the unique risk profile of fintech necessitates a specialized DevSecOps maturity model. We introduce the Fintech DevSecOps Maturity Model (Fin-DSOMM), a novel, four-level framework adapted from the Open Web Application Security Project (OWASP) DSOMM. Unlike generic models, the Fin-DSOMM prioritizes compliance-as-code and proactive threat modeling at early maturity stages. It further posits that Artificial Intelligence (AI) is a critical accelerant, enabling organizations to progress from a reactive, automated security posture to a predictive and adaptive one. This paper outlines the structure of the Fin-DSOMM, details the role of AI in enhancing each maturity level, and provides a strategic roadmap for fintech organizations to build secure, compliant, and agile software development lifecycles.

1. Introduction: The Dual Imperative of Speed and Security in Fintech

The fintech sector exists at the confluence of rapid technological advancement and immense risk. On one hand, firms are under intense market pressure to accelerate the delivery of new features, applications, and updates to meet evolving customer expectations.[43] On the other hand, the financial industry is a primary target for sophisticated cyber adversaries, with one study indicating that the sector experienced over 21,000 cyber-attacks between 2003 and 2023, and another noting that 60% of unique malware attacks target finance, healthcare, and retail.[44] For fintech companies, a security breach is not just a technical failure; it can lead to catastrophic financial losses, severe regulatory penalties, and an irreversible erosion of customer trust, which is the bedrock of any financial institution.[45]

This high-stakes environment renders traditional software development models, where security is treated as an afterthought or a final gate before release, completely untenable.[43] The only viable path forward is the deep integration of security into every phase of the software development lifecycle (SDLC). This philosophy, known as DevSecOps, is not merely a best practice for fintech but a fundamental requirement for survival and success. However, generic DevSecOps frameworks often fail to account for the specific regulatory and risk context of finance. This paper addresses this gap by proposing a specialized maturity model designed to guide fintech organizations on their journey toward building fast, resilient, and secure systems.

2. Background: From DevOps to DevSecOps

2.1 Core Principles of DevSecOps

DevSecOps represents a cultural and technical evolution of DevOps, the practice of combining software development (Dev) and IT operations (Ops) to shorten the SDLC. DevSecOps integrates security (Sec) into this continuum, transforming it from an isolated function into a shared responsibility.[46] This approach is founded on several core principles:

Security as a Shared Responsibility: In a DevSecOps culture, security is not solely the domain of a dedicated security team. Instead, developers, operations

engineers, and security professionals collaborate throughout the SDLC, with everyone taking ownership of building secure software.[46]

Automation: Manual security reviews and testing are replaced with automated tools integrated directly into the development pipeline. This allows for continuous monitoring and identification of vulnerabilities without creating bottlenecks.[46]

Shifting Left: This principle involves moving security practices as early as possible (to the "left") in the SDLC. By identifying and remediating vulnerabilities during the design and coding phases, organizations can significantly reduce the cost and effort required for fixes, as issues caught late in the cycle often necessitate extensive rework.[43]

The benefits of this approach are tangible. Organizations that successfully adopt DevSecOps principles have reported that their engineers can release code up to 60% faster, demonstrating that security, when integrated correctly, can be an accelerator, not an inhibitor, of speed.46

2.2 The OWASP DevSecOps Maturity Model (DSOMM)

To provide organizations with a structured path for implementation, the Open Web Application Security Project (OWASP) developed the DevSecOps Maturity Model (DSOMM).[49] A maturity model serves as a compass, allowing an organization to assess its current capabilities, define its target state, and create a roadmap for incremental improvement.[50] The DSOMM is typically structured into several levels of increasing maturity (e.g., four or five levels), which are applied across various dimensions of the SDLC, such as Governance, Design, Implementation, and Operations.[51] This leveled approach is valuable because it helps organizations prioritize investments, communicate progress effectively to both technical and non-technical stakeholders, and measure their evolution over time.[50]

3. The Fin-DSOMM: A Proposed Maturity Model for Financial Services

While the standard DSOMM provides an excellent general-purpose framework, it is not sufficiently tailored to the unique constraints of the fintech industry. A generic model might treat regulatory compliance as a feature of higher maturity levels. However, for a financial institution, adherence to regulations like the Payment Card Industry Data Security Standard (PCI DSS), the General Data Protection Regulation (GDPR), and the Sarbanes-Oxley Act (SOX) is a foundational, non-negotiable requirement. A security failure in fintech is almost invariably a compliance failure, carrying the risk of severe legal and financial repercussions.[45]

Therefore, a specialized maturity model for fintech must elevate compliance from an advanced-stage goal to a Level 1 prerequisite. This paper proposes the **Fintech DevSecOps Maturity Model (Fin-DSOMM)**, a framework that integrates this compliance-first mindset. It provides a more realistic and effective roadmap for financial organizations by embedding regulatory controls at the very beginning of the maturity journey. Table 1 highlights the major differences between the generic OWASP DSOMM and the proposed Fin-DSOMM, emphasizing the compliance-first and AI-enhanced security posture required in financial services environments. Figure 1 illustrates the four progressive maturity levels of the proposed Fin-DSOMM, from foundational compliance controls to AI-driven adaptive defense.

Table 1. Comparison of Generic OWASP DSOMM and Proposed Fin-DSOMM

Dimension	Generic OWASP DSOMM	Proposed Fin-DSOMM
Compliance Priority	Higher maturity stage	Level 1 prerequisite
Regulatory Focus	General security posture	PCI DSS, GDPR, SOX, AML-specific
Threat Modeling	Mid-level maturity activity	Mandatory from Level 3
Infrastructure Controls	Standard IaC practices	Compliance-as-Code enforced
Security Automation	Rule-based CI/CD checks	AI-assisted predictive security
Incident Response	Manual + automated alerts	SOAR + AI-driven autonomous response
Financial Risk Alignment	Generic enterprise security	Fraud prevention + transaction integrity

Dimension	Generic OWASP DSOMM	Proposed Fin-DSOMM
Compliance Priority	Considered in higher maturity stages	Level 1 prerequisite and foundational requirement
Regulatory Focus	General security posture and best practices	Fintech-specifty regulations: PCI DSS, GDPR, SOX, AML, KYC, FFIEC, local mandates
Threat Modeling	Introduced in mid-level maturity	Mandatory from Level 3 and continuously updated
Infrastructure Controls	Standard IaC and security configurations	Compliance-as-Code with policy-as-code enforced
Security Automation	Rule-based CI/CD checks and scanning	AI-asbased automation with predictive risk scoring and contextual analysis
Incident Response	Manual + automated alerts and playbooks	SOAR-integrated with AI-driven anomaly detection and autonomous response
Financial Risk Alignment	Generic enterprise security objectives	Aligned with fraud prevention, transaction integrity, and customer trust protection

Figure 1. The Fintech DevSecOps Maturity Model (Fin-DSOMM)

3.1 Level 1: Foundational Security & Compliance-as-Code

At this initial level, a fintech organization establishes the cultural and technical bedrock for a secure SDLC. The focus is on basic hygiene and, crucially, codifying compliance requirements.

People & Culture: A security-first mindset is fostered through the appointment of security champions within development teams. Regular, mandatory training on secure coding practices and the specific regulatory landscape (e.g., PCI DSS, data privacy) is established.[43]

Process: Foundational security and compliance policies are documented and version-controlled. The practice of Infrastructure-as-Code (IaC) is adopted using tools like Terraform or CloudFormation. IaC is critical because it ensures that development, testing, and production environments are provisioned in a consistent, repeatable, and auditable manner, which is a key tenet of both security and compliance.[45]

Technology:

Compliance-as-Code: This is the cornerstone of Fin-DSOMM Level 1. Instead of manual audits, compliance rules are translated into automated checks. Tools like Open Policy Agent (OPA) are integrated into the CI/CD pipeline to automatically validate that infrastructure configurations (defined in IaC) adhere to financial regulations, such as data residency rules or network segmentation requirements for PCI DSS.[48]

Basic Hygiene Automation: The pipeline must enforce fundamental security controls. This includes mandatory Multi-Factor Authentication (MFA) for all developer and administrative accounts, Role-Based Access Control (RBAC) to enforce least privilege, and automated secrets scanning using tools like truffleHog or Gitleaks to prevent credentials, API keys, and tokens from being committed to source code repositories.[45]

3.2 Level 2: Automated Security in CI/CD Pipelines

With a compliant foundation in place, the organization moves to integrate a comprehensive suite of automated security testing tools directly into the CI/CD pipeline. At this level, security testing is no longer a separate phase but an automated, non-negotiable gate that every code change must pass before progressing toward production.[46]

Technology:

Static Application Security Testing (SAST): Tools like Checkmarx or SonarQube are integrated to automatically scan source code for vulnerabilities (e.g., SQL injection, cross-site scripting) on every commit or pull request.[48]

Software Composition Analysis (SCA): Given that modern applications are heavily composed of open-source libraries, SCA tools (e.g., Snyk, Sonatype Nexus) are used to scan all dependencies for known vulnerabilities (CVEs) and license compliance issues.[45]

Dynamic Application Security Testing (DAST): DAST tools are used to test the running application in a staging environment, probing for vulnerabilities from the outside-in, simulating how an attacker would interact with the application.[58]

Container Security: As applications are increasingly deployed in containers, tools like Trivy or Clair are integrated to scan container images for vulnerabilities within the base image and application layers.[53]

3.3 Level 3: Proactive Threat Management

Maturity at Level 3 signifies a shift from a reactive posture (finding and fixing known vulnerability types) to a proactive one (anticipating and designing against potential attacks).

Process: The focus moves further "left" to the design phase. Threat modeling becomes a standardized and mandatory practice for all new features, microservices, and APIs.[48] Teams use frameworks like STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) to brainstorm potential attack vectors and design mitigations before code is written.

Technology:

Advanced Infrastructure Security: A Web Application Firewall (WAF) is implemented to protect against common web-based attacks on ingress traffic.[53] The organization adopts the principle of immutable infrastructure, where servers and containers are replaced rather than patched, which reduces configuration drift and hardens the environment.[53]

Enhanced Monitoring: Security logging is centralized, and Security Information and Event Management (SIEM) systems are deployed to correlate security events from across the infrastructure, enabling more effective monitoring and incident response.[49]

3.4 Level 4: AI-Enhanced Security Intelligence and Adaptive Defense

The highest level of maturity in the Fin-DSOMM is achieved through the integration of AI and machine learning. This marks a fundamental qualitative shift from *automated* security, which executes predefined checks and rules faster, to *intelligent* security, which can discover novel threats and predict future risks. While the preceding levels focus on codifying and automating responses to known threats, Level 4 introduces the capability to learn and adapt to an evolving threat landscape. Rule-based automation is insufficient against zero-day exploits and novel attack patterns, whereas AI/ML models excel at identifying subtle anomalies in vast datasets that deviate from normal behavior, even if that behavior does not match a predefined attack signature.[60]

Process: The security posture becomes fully proactive and adaptive, with a focus on real-time threat detection and automated response.

Technology:

AI-Driven Threat Detection: Machine learning algorithms are deployed to analyze logs, network traffic, and application performance metrics in real-time. These models establish a baseline of normal behavior and can flag anomalies that may indicate a sophisticated attack in progress.[60]

Predictive Vulnerability Analysis: AI is used to analyze code repositories, commit histories, and public vulnerability databases to predict which parts of the codebase are most likely to contain future security flaws, allowing for targeted code reviews and testing.[60] Some models can even anticipate potential zero-day vulnerabilities by identifying patterns similar to those in past exploits.[60]

Automated Incident Response: Security Orchestration, Automation, and Response (SOAR) platforms, enhanced with AI, can take autonomous containment actions. For example, upon detecting a suspicious pattern of API calls, the system could automatically isolate the affected microservice or block the source IP address.[48]

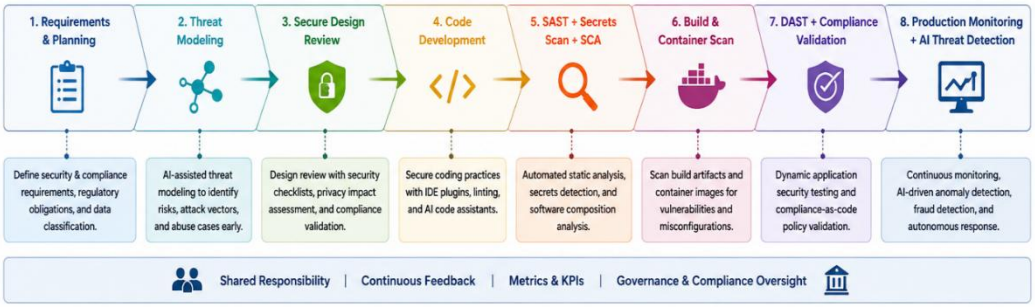
4. The Role of AI in Accelerating DevSecOps Maturity

AI does not just represent the highest level of maturity; it also acts as an accelerant that helps organizations progress through the earlier levels more efficiently.

AI for Automated Code Review: Beyond traditional SAST, AI-powered tools can provide more nuanced code analysis. Trained on billions of lines of code from open-source projects, these models can identify complex security vulnerabilities and logic flaws with greater context-awareness than rule-based scanners.60

AI for Streamlining Compliance: The process of mapping hundreds of technical security controls to complex regulatory frameworks like PCI DSS or SOX is a monumental manual task. AI can automate this process, analyzing control implementations and automatically generating evidence and documentation for audits, ensuring that compliance is continuously verified rather than checked periodically.[60]

Figure 3 demonstrates how security controls are shifted left across the fintech SDLC, ensuring that compliance validation and threat detection begin during planning and continue through production monitoring.



Securing the AI Pipeline (AISecOps/MLSecOps): As AI is integrated into the SDLC, the AI/ML pipeline itself—comprising data, models, and code—becomes a critical piece of infrastructure that must be secured. This emerging discipline, known as AISecOps or MLSecOps, applies DevSecOps principles to the AI lifecycle.[44] It involves securing data pipelines against poisoning, ensuring model integrity, and managing the unique vulnerabilities of AI systems, a topic explored in greater depth in Article 5 of this series. Table 2 summarizes how AI capabilities evolve across the four maturity levels, supporting the transition from compliance automation to predictive and autonomous security operations.

Table 2. Role of AI Across Fin-DSOMM Maturity Levels

Maturity Level	Primary Goal	AI Contribution
Level 1	Foundational compliance	Automated policy validation
Level 2	CI/CD security automation	AI-assisted SAST and code review
Level 3	Proactive threat prevention	Threat modeling recommendations
Level 4	Adaptive defense	Real-time anomaly detection and SOAR

5. Implementation Challenges and Strategic Recommendations

The path to DevSecOps maturity is not without its challenges. Organizations frequently encounter cultural resistance, as ingrained silos between development, security, and operations teams can be difficult to break down.[43] The proliferation of automated scanning tools can lead to security data overload, overwhelming teams with a high volume of alerts and making it difficult to prioritize the most critical vulnerabilities.[46] Furthermore, integrating modern security tools with legacy financial systems can present significant technical hurdles.[56]

Strategic success requires a multi-pronged approach. Strong, top-down leadership is essential to mandate security as a core business priority and drive the necessary cultural change.[43] Continuous education and cross-training are needed to equip developers with security knowledge and security professionals with an understanding of DevOps methodologies.[43] From a technical perspective, organizations should begin with a solid foundation, prioritizing the implementation of IaC and a robust CI/CD pipeline, as these are the enablers for all subsequent automation.[48] The success of a large fintech organization that transitioned from two releases per year to bi-weekly releases after a comprehensive DevSecOps transformation serves as a powerful testament to the achievable benefits of a committed implementation.[58]

Table 3. Common DevSecOps Adoption Challenges and Strategic Mitigations

Challenge	Impact	Recommended Mitigation
Cultural resistance	Slow adoption	Executive sponsorship

Challenge	Impact	Recommended Mitigation
Alert fatigue	Ignored vulnerabilities	Risk-based prioritization
Legacy system integration	Security gaps	Incremental modernization
Compliance complexity	Audit failure risk	Compliance-as-Code
Tool sprawl	Pipeline inefficiency	Platform consolidation

6. Conclusion

For fintech organizations, the question is not whether to adopt DevSecOps, but how to do so in a way that addresses their unique security and compliance obligations. Generic maturity models provide a useful starting point but are insufficient for the high-stakes financial environment. The proposed Fintech DevSecOps Maturity Model (Fin-DSOMM) offers a more tailored and effective roadmap by embedding compliance-as-code as a foundational principle and charting a clear path toward an AI-enhanced, predictive security posture. By systematically progressing through these maturity levels, fintech firms can build the resilient, secure, and agile development capabilities necessary to innovate safely and maintain trust in an increasingly competitive and hostile digital landscape.

References

[1] N. Soni and A. Kumar, "DevSecOps adoption in financial technology systems: Security-first software delivery in regulated environments," *Journal of Information Security and Applications*, vol. 72, p. 103394, 2023.

[2] M. Shah, R. Patel, and K. Srinivasan, "Cybersecurity risk management in fintech ecosystems: Challenges and strategic responses," *Computers & Security*, vol. 128, p. 103168, 2023.

[3] P. Gupta and S. Raman, "Integrating security into CI/CD pipelines for high-assurance software delivery," *IEEE Access*, vol. 11, pp. 44892–44908, 2023.

[4] A. Verma and T. Joseph, "Security automation and shift-left testing in cloud-native DevSecOps pipelines," *Software: Practice and Experience*, vol. 53, no. 9, pp. 1781–1798, 2023.

[5] R. K. Singh and L. Zhao, "Compliance-as-code for financial services infrastructure using policy-driven DevSecOps," *Future Generation Computer Systems*, vol. 145, pp. 311–324, 2023.

[6] Open Web Application Security Project (OWASP), "OWASP DevSecOps Maturity Model (DSOMM)," 2023. Available:

[7] S. Brown and J. Harris, "Infrastructure as Code and immutable infrastructure for secure banking platforms," *Journal of Cloud Computing*, vol. 12, no. 1, pp. 1–16, 2023.

[8] D. Fernandes and P. Roy, "Static and dynamic security testing integration for secure software release pipelines," *Empirical Software Engineering*, vol. 28, no. 4, pp. 1–28, 2023.

[9] V. Kumar and H. Lee, "Software composition analysis for open-source dependency risk management," *ACM Transactions on Software Engineering and Methodology*, vol. 32, no. 3, pp. 1–24, 2023.

[10] Y. Chen, M. Ali, and S. Roberts, "Container vulnerability scanning and runtime security in financial cloud platforms," *IEEE Transactions on Cloud Computing*, vol. 11, no. 4, pp. 3561–3573, 2023.

[11] L. Martin and K. Hughes, "Threat modeling using STRIDE in enterprise API security design," *Information and Software Technology*, vol. 160, p. 107224, 2023.

[12] J. Wilson and T. Baker, "Security information and event management in financial cyber defense operations," *Computers & Security*, vol. 126, p. 103064, 2023.

[13] P. Das and R. Srinath, "AI-driven anomaly detection for fraud prevention and threat intelligence in digital payments," *Expert Systems with Applications*, vol. 233, p. 120912, 2024.

[14] M. Hassan and E. Kim, "Predictive vulnerability analytics using machine learning for secure software engineering," *IEEE Access*, vol. 12, pp. 22411–22427, 2024.

[15] S. Tran and B. Lopez, "Autonomous incident response using SOAR and artificial intelligence in enterprise security," *Journal of Cybersecurity*, vol. 10, no. 1, p. tyad018, 2024.

[16] A. Narayanan and G. Foster, "AI-assisted secure code review and intelligent vulnerability prioritization," *Software Quality Journal*, vol. 32, no. 1, pp. 87–109, 2024.

[17] R. Mehta and J. Olsen, "Continuous compliance automation for PCI DSS and GDPR using policy-as-code," *Information Systems Frontiers*, vol. 26, pp. 541–558, 2024.

[18] K. Ibrahim and S. White, "AISecOps: Securing the machine learning lifecycle in regulated industries," *Future Internet*, vol. 16, no. 2, p. 61, 2024.

[19] T. Nguyen and H. Park, "Legacy system modernization challenges in DevSecOps transformation programs," *Journal of Systems and Software*, vol. 209, p. 111918, 2024.

[20] B. Edwards and M. Silva, "Organizational transformation and DevSecOps maturity progression in large financial enterprises," *IEEE Software*, vol. 41, no. 2, pp. 74–82, 2024.