



SECURE SOFTWARE DEVELOPMENT POLICY

Control Page

ATTRIBUTE	DETAIL
Document Title	SECURE SOFTWARE DEVELOPMENT POLICY
Document ID	OG-SEC-006
Version	1.1
Effective Date	November 2, 2025
Next Review Date	November 2, 2026 (Annual)
Policy Owner	Chief Security & Compliance Officer (CSCO)
Approved By	Chief Executive Officer (CEO)
Classification	Confidential – Internal Use Only



Applicability	All systems, software, networks, cloud environments, and third-party integrations that process, store, or transmit cardholder data (CHD) or support Orchestrate Global's payment orchestration platform
Compliance Frameworks	PCI DSS v4.0.1, GDPR, Financial Regulations
Distribution List	Engineering, DevOps, IT, Security, Compliance, Product, Legal, C-Suite

Approvals

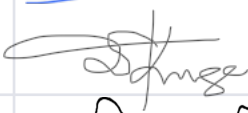
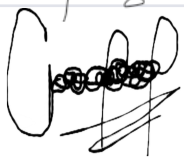
NAME	POSITION	SIGNATURE	DATE
Oleksii Bondarenko	Chief Security & Compliance Officer		1 November 2025
Charles Odhiambo	Technical Lead		1 November 2025
Kevin Kegera	Chief Commercial Officer		1 November 2025
Preston Obadan	Chief Executive Officer		1 November 2025

Table of Contents

1. Introduction
2. Purpose and Scope
3. Software Development Approach Type
4. Regulatory and Compliance Requirements
5. Secure Software Development Lifecycle (SDLC)
6. Secure Coding Standards
7. Code Review and Quality Assurance
8. Security Testing Requirements
9. Development Environment Security



10. Third-Party Component Management
11. API Security Development
12. Vulnerability Management Integration
13. Training and Awareness
14. Monitoring, Metrics, and Continuous Improvement
15. Incident Response Integration
16. Audit, Compliance, and Policy Maintenance
17. Roles and Responsibilities
18. Policy Review and Change Management
19. Appendices

1. Introduction

In alignment with global best practices for secure software engineering and regulatory expectations—including PCI DSS v4.0.1, GDPR, and financial operational resilience standards—Orchestrate Global adopts a structured, risk-informed, and security-by-design approach to software development. This policy formalizes the organization's commitment to embedding security into every phase of the software development lifecycle (SDLC), ensuring the confidentiality, integrity, and availability of cardholder data and payment systems.

2. Purpose and Scope

This policy defines mandatory security controls, processes, and responsibilities for all software development activities related to the Orchestrate Global Payment Gateway. It applies to:

- Payment API development and maintenance
- Merchant and admin dashboard systems
- Database schema and data model changes
- Integration with processors, banks, and third-party partners
- Infrastructure as Code (IaC)
- Third-party library and component integration

All personnel involved in design, development, testing, deployment, or maintenance of these systems must comply with this policy.



3. Software Development Approach Type

Orchestrate Global employs a Secure Agile Development Model, integrating security practices into an iterative, incremental delivery framework. This approach combines:

- **Agile Methodology:** Sprints, user stories, continuous integration, and frequent releases.
- **DevSecOps Integration:** Security is embedded as a shared responsibility across Development, Security, and Operations (DevSecOps).
- **Shift-Left Security:** Security activities (e.g., threat modeling, SAST, secure design reviews) are performed early and continuously throughout the lifecycle.
- **Compliance-Driven Engineering:** Regulatory requirements (e.g., PCI DSS Requirement 6) are translated into technical controls and automated validation gates.

This hybrid model ensures rapid innovation without compromising security, auditability, or regulatory compliance.

4. Regulatory and Compliance Requirements

All development activities must adhere to the following frameworks:

FRAMEWORK	KEY REQUIREMENTS
PCI DSS v4.0.1	Requirement 6: Secure systems and applications; 6.2.4: Secure coding; 6.3.1–6.3.2: Testing & code review
GDPR	Privacy by design, data minimization, protection of personal data
Financial Regulations	Operational resilience, system integrity, incident reporting

Non-compliance may result in disciplinary action, project suspension, or regulatory penalties.

5. Secure Software Development Lifecycle (SDLC)



The SDLC is structured into six phases, each with defined security gates:

PHASE	SECURITY ACTIVITIES	GATE REQUIREMENTS	APPROVAL AUTHORITY
Planning	Threat modeling, security requirements	Approved Security Requirements Document	Security Architect
Design	Architecture review, data flow analysis	Zero high-risk architectural flaws	CISO
Development	Secure coding, SAST	Clean SAST scan, peer-reviewed code	Tech Lead
Testing	DAST, penetration testing	Zero critical, <5 high vulnerabilities	Security Team
Deployment	Secure configuration, runtime protection	Completed Deployment Security Checklist	Operations Lead
Maintenance	Patching, monitoring	Continuous vulnerability monitoring	Development Manager

Each gate must be formally cleared before progression.

6. Secure Coding Standards

All code must comply with the following:

- Input Validation: Whitelist-based validation for all payment inputs
- PAN Protection: PAN never logged; CHD encrypted at rest using AES-256
- Cryptography: Only FIPS 140-2–compliant algorithms (AES-256, RSA-4096)
- Error Handling: Generic messages; no CHD in logs
- Session Management: 15-minute timeout, secure tokens



Prohibited Practices: Hardcoded credentials, CHD in logs, SQLi, weak crypto, debug code in production.

7. Code Review and Quality Assurance

- Payment Processing Code: Mandatory review by 2 developers + 1 security reviewer
- Review Checklist: Includes PAN logging, encryption, TLS 1.3, XSS/CSRF prevention
- Automated Enforcement: Integrated into CI/CD pipeline via SAST and policy-as-code tools

8. Security Testing Requirements

SAST	Every commit	SonarQube, Checkmarx	Zero critical, <5 high issues
DAST	Every release	OWASP ZAP, Burp Suite	Zero critical, <3 high issues
Dependency Scanning	Daily	Snyk, OWASP DC	Zero critical vulnerabilities
Penetration Testing	Quarterly	External firm	Full-stack validation

9. Development Environment Security

- Segmentation: Dev, Staging, Production environments strictly isolated
- Data Handling: No real CHD in Dev/Staging; synthetic/anonymized only
- Workstation Security: Full-disk encryption, EDR, MFA-enabled VPN, no production data locally

10. Third-Party Component Management



- All open-source/commercial components undergo security and license review
- Daily vulnerability scanning; only MIT/Apache/BSD licenses permitted
- Cloud services must provide SOC 2 and PCI DSS evidence

11. API Security Development

- OAuth 2.0, RBAC, rate limiting (1,000 req/min), TLS 1.3
- Webhooks require HMAC signatures and replay protection
- All inputs schema-validated; outputs sanitized

12. Vulnerability Management Integration

SEVERITY	RESPONSE TIME	RESOLUTION TIME	ESCALATION
Critical	1 hour	4 hours	Immediate CISO notification
High	4 hours	24 hours	CTO within 2 hours

13. Training and Awareness

- Developers: Quarterly secure coding & PCI DSS training (certification required)
- Architects: Annual threat modeling (STRIDE methodology)
- QA/DevOps: Semi-annual security testing & infrastructure hardening labs

14. Monitoring, Metrics, and Continuous Improvement

Key metrics tracked monthly:

- Zero critical vulnerabilities in production
- 100% code review coverage for payment logic
- <4-hour incident response for critical flaws
- 100% training completion

15. Incident Response Integration



- Data Exposure: 30-minute response; involves Legal, CISO, Dev, and Security
- Post-Incident: RCA within 48 hours; policy/training updates within 2 weeks

16. Audit, Compliance, and Policy Maintenance

- Internal Reviews: Quarterly
- External Audits: Annual
- Documentation: Code reviews, test reports, training records retained for 3 years

17. Roles and Responsibilities

ROLE	RESPONSIBILITIES
CTO	Policy oversight, resource allocation
CISO	Security approvals, risk acceptance, incident escalation
Development Manager	Team compliance, training enforcement
Security Architect	Threat modeling, design validation
Lead Developer	Code quality, mentoring, gate compliance
DevOps Engineer	Secure deployments, environment hardening
Compliance Officer	Audit readiness, regulatory alignment

18. Policy Review and Change Management

- Annual Review: Mandatory
- Ad-hoc Updates: Triggered by incidents, new threats, or regulatory changes
- Major Changes: Require CTO + CISO approval; 30-day implementation window

19. Appendices



- Appendix A: Secure Coding Checklist
- Appendix B: SDLC Gate Approval Templates
- Appendix C: Approved Technology Stack
- Appendix D: Vulnerability Severity Classification Matrix

This document is the intellectual property of Orchestrate Global. Unauthorized distribution is prohibited.