Secure Software Development Lifecycle (SSDLC) & Vulnerability Management Overview

Version: 2.0
Last Updated: 1st September 2026
Classification: Public

Secure Software Development Lifecycle (SSDLC) & Vulnerability Management Overview

This document describes how Concord builds, deploys, and maintains software that is secure by design and continuously improved. Read together with the Security Overview, Infrastructure & Architecture Overview, and the Information Security Schedule.

Security by Design

Security is incorporated at every stage of development rather than treated as a final testing activity: secure by design, secure by default, least privilege, defense in depth, privacy by design, and continuous improvement.

Secure Development Lifecycle

Planning — business, security, privacy, AI governance, and regulatory requirements are considered up front, with risk assessment as part of scoping.

Architecture & design — architecture review, threat modeling, authentication/authorization design, encryption planning, API security planning, data flow analysis, and tenant isolation review happen before implementation begins.

Development — secure coding standards, peer code review, dependency management, secrets management, input validation, output encoding, and structured error handling.

Security testing — static application security testing (SAST), software composition analysis (SCA), dependency vulnerability scanning, secret scanning, Infrastructure-as-Code scanning, and manual security review where warranted, before production deployment.

Release & deployment — automated CI/CD pipelines, deployment approval, configuration validation, rollback capability, environment separation, and post-deployment verification.

Operations & maintenance — vulnerability monitoring, patch management, security monitoring, incident response, and continuous improvement after deployment.

Threat Modeling

Structured threat modeling considers authentication, authorization, data flows, API security, AI workflows, administrative interfaces, tenant isolation, and external integrations, with identified risks prioritized by impact and likelihood.

Code Review

All source code changes are peer-reviewed before merging to production branches, covering functional correctness, security, coding standards, maintainability, and performance impact.

Dependency Management

Concord maintains a dependency inventory, monitors published security advisories, updates dependencies on a regular cadence, and reviews high-risk packages before adoption.

Secrets Management

Credentials are centrally managed with restricted access, rotation, secure storage, and environment separation — no hardcoded credentials in source.

Vulnerability Management

Vulnerabilities are identified, risk-assessed, prioritized by business impact and exploitability, remediated, and verified, with continuous monitoring throughout.

Security Monitoring

Authentication events, administrative actions, infrastructure changes, security alerts, application logs, and API activity are monitored to support both incident detection and operational improvement.

Patch Management

Vendor security advisories are monitored, risk-assessed, tested, and deployed under controlled, verified procedures, with critical updates receiving prioritized attention.

Penetration Testing and Security Assessment

Concord periodically evaluates platform security through security testing, which may include penetration testing conducted internally or by qualified third-party security specialists. Scope and frequency are risk- and architecture-driven. Detailed reports are not publicly distributed but may be made available to customers under confidentiality arrangements.

AI Security Considerations

Prompt validation, output handling, access controls for AI services, secure AI provider integration, audit logging, and human oversight are part of both product development and operational monitoring for AI-enabled functionality.

Security Defect Management

Security defects are tracked, risk-classified, assigned, remediated, verified, and closed through documented processes, with lessons learned feeding future development.

Shared Responsibility

Concord is responsible for: secure software design, secure development practices, vulnerability management, security testing, patch management, and incident response.

Customers are responsible for: securing their own endpoints, managing user access, configuring integrations securely, and reviewing AI-generated outputs before relying on them for business decisions.

Additional Information

Available on request, subject to confidentiality obligations: Security Overview, Infrastructure & Architecture Overview, AI Governance & Responsible AI Overview, Information Security Schedule, and Service Level Agreement.
