SECURE SOFTWARE DEVELOPMENT POLICY

CHÍNH SÁCH PHÁT TRIỂN PHẦN MỀM AN TOÀN

Secure by Design • Shift Left • Verify, Don't Trust

Tổ chức / Organization	[Tên tổ chức]
Chủ sở hữu / Document Owner	[Head of Application Security]
Phê duyệt / Approved by	Engineering Governance Council
Ngày hiệu lực / Effective Date	[DD/MM/YYYY]
Phiên bản / Version	1.0
Chu kỳ rà soát / Review Cycle	Hằng năm, hoặc ngay khi có nhóm lỗ hổng/kỹ thuật tấn công mới đáng kể xuất hiện
Tài liệu cấp trên / Parent Documents	Engineering Constitution v1.0 (Tầng 1); Software Development Governance Policy v1.0 (Tầng 2)
Phân loại / Classification	INTERNAL

SAMPLE DOCUMENT

A. Kiểm soát tài liệu

DOCUMENT CONTROL

Phiên bản	Ngày	Người soạn	Mô tả thay đổi
1.0	[DD/MM/YYYY]	[Tên / Vai trò]	Ban hành lần đầu

B. Mục lục nội dung

CONTENTS

- 1. Mục đích
- 2. Phạm vi áp dụng
- 3. Liên kết với Engineering Constitution
- 4. Định nghĩa
- 5. Yêu cầu bắt buộc
- 6. Vai trò và trách nhiệm
- 7. Đo lường tuân thủ
- 8. Cơ chế ngoại lệ
- 9. Không tuân thủ
- 10. Chủ sở hữu tài liệu và chu kỳ rà soát
- 11. Tài liệu liên quan
- 12. Phê duyệt

1. Mục đích

PURPOSE

1. Quy định cụ thể CÁCH tổ chức đưa yêu cầu bảo mật vào từng giai đoạn của vòng đời phát triển phần mềm (SDLC), hiện thực hóa nguyên tắc "Bảo mật và quyền riêng tư từ khâu thiết kế" (Security and Privacy by Design — Điều 2 Engineering Constitution) và mở rộng chi tiết cho Software Development Governance Policy (SDGP), Điều 5.2.
2. Giảm thiểu rủi ro lỗ hổng bảo mật lọt vào production thông qua threat modeling sớm, secure coding baseline, kiểm soát tự động trong CI/CD, và quản lý lỗ hổng có kỷ luật.
3. Thiết lập nền tảng "shift left" — phát hiện và xử lý rủi ro bảo mật càng sớm càng tốt trong SDLC, thay vì chỉ kiểm tra ở cuối trước khi phát hành.

Chính sách này bổ sung cho SDGP, không thay thế — mọi yêu cầu chung của SDGP (SDLC tổng thể, vai trò/trách nhiệm, kiểm soát thay đổi, cơ chế ngoại lệ, không tuân thủ...) vẫn áp dụng đầy đủ.

2. Phạm vi áp dụng

SCOPE

- Áp dụng cho mọi phần mềm do tổ chức phát triển hoặc tùy biến đáng kể: mã nguồn nội bộ, dịch vụ backend, ứng dụng frontend, API, và hạ tầng dưới dạng mã (Infrastructure as Code).
- Áp dụng cho toàn bộ SDLC quy định tại SDGP Điều 5.1: từ thiết kế, viết mã, kiểm thử, đến triển khai và vận hành.
- Áp dụng cho nhân sự nội bộ, nhà thầu, đối tác thuê ngoài phát triển phần mềm cho tổ chức (liên kết SDGP Điều 5.11).
- Không thay thế chính sách bảo mật thông tin cấp doanh nghiệp (nếu tổ chức có) — Policy này chỉ tập trung vào bảo mật trong hoạt động phát triển phần mềm.

3. Liên kết với Engineering Constitution

ALIGNMENT WITH THE CONSTITUTION

Yêu cầu trong Policy	Nguyên tắc Constitution tương ứng
5.1 Threat modeling bắt buộc cho hệ thống rủi ro cao	Điều 2 — Security and Privacy by Design
5.2 Secure coding baseline	Điều 2, Điều 3 — Security by Design / Quality Is Everyone's Responsibility
5.3 Security gate tự động trong CI/CD	Điều 5, Điều 6 — Everything as Code / Observable and Operable by Default
5.4 Quản lý lỗ hổng theo SLA	Điều 8 — Evidence over Opinion
5.5 Quản lý secret/credentials	Điều 2 — Security by Design

5.6 Đào tạo & Security Champions	Điều 10 — Continuous Learning
5.7 Kiểm thử xâm nhập định kỳ	Điều 8 — Evidence over Opinion

4. Định nghĩa

DEFINITIONS

- **Threat Modeling:** quy trình phân tích có hệ thống một tính năng/hệ thống để nhận diện các mối đe dọa bảo mật tiềm ẩn, đường tấn công và biện pháp giảm thiểu tương ứng, thực hiện TRƯỚC khi viết mã.
- **SAST (Static Application Security Testing):** công cụ phân tích mã nguồn tĩnh (chưa chạy) để phát hiện lỗ hổng bảo mật.
- **DAST (Dynamic Application Security Testing):** công cụ kiểm thử bảo mật trên ứng dụng đang chạy (runtime).
- **SCA (Software Composition Analysis):** công cụ quét thư viện/dependency mã nguồn mở để phát hiện lỗ hổng đã biết (CVE) và vấn đề giấy phép.
- **SBOM (Software Bill of Materials):** danh mục đầy đủ các thành phần/thư viện cấu thành nên một phần mềm, phục vụ truy xuất nguồn gốc khi phát sinh lỗ hổng chuỗi cung ứng.
- **Security Gate:** điểm kiểm soát tự động trong pipeline CI/CD có khả năng chặn merge hoặc chặn deploy khi phát hiện lỗ hổng vượt ngưỡng cho phép.
- **Secret/Credential:** thông tin xác thực nhạy cảm (API key, mật khẩu, token, private key...) dùng để truy cập hệ thống hoặc dịch vụ.
- **Secret Vault / Secret Manager:** hệ thống chuyên dụng lưu trữ, cấp phát và luân chuyển (rotate) secret có kiểm soát, thay vì lưu trực tiếp trong mã nguồn hoặc file cấu hình.
- **Security Champion:** kỹ sư trong một đội phát triển được chỉ định làm đầu mối về thực hành bảo mật, không nhất thiết là chuyên gia bảo mật toàn thời gian.
- **Penetration Testing (Pentest):** hoạt động mô phỏng tấn công có kiểm soát, do người/đội có kỹ năng thực hiện, nhằm phát hiện lỗ hổng mà công cụ tự động khó phát hiện.

> **Khung tham chiếu ngành** — Yêu cầu tại Điều 5 được xây dựng có đối chiếu với các khung tham chiếu bảo mật ứng dụng được công nhận rộng rãi: NIST SP 800-218 *Secure Software Development Framework (SSDF) v1.1* — đặc biệt hai nhóm thực hành *Produce Well-Secured Software (PW)* và *Respond to Vulnerabilities (RV)*; OWASP *Application Security Verification Standard (ASVS)* v5.0; OWASP *Software Assurance Maturity Model (SAMM)* v2; OWASP Top 10:2025; và ISO/IEC 27001:2022 Annex A, các kiểm soát 8.25-8.29 (Vòng đời phát triển an toàn; Yêu cầu bảo mật ứng dụng; Nguyên tắc kiến trúc và kỹ thuật hệ thống an toàn; Lập trình an toàn — Secure Coding; Kiểm thử bảo mật trong phát triển và nghiệm thu).

5. Yêu cầu bắt buộc

MANDATORY REQUIREMENTS

5.1 Threat Modeling bắt buộc cho tính năng/hệ thống rủi ro cao

1. Trước khi bắt đầu viết mã (giai đoạn thiết kế trong SDLC — SDGP Điều 5.1), đội phát triển **PHẢI (MUST)** thực hiện threat modeling cho: (a) hệ thống/tính năng xử lý dữ liệu cá nhân hoặc dữ liệu nhạy cảm, (b) hệ thống xử lý xác thực/phân quyền hoặc thanh toán, hoặc (c) thay đổi kiến trúc lớn (ví dụ thêm một service mới lộ diện ra Internet).
2. Người chủ trì threat modeling là Tech Lead của tính năng/hệ thống đó; đối với hệ thống rủi ro cao nhất, **NÊN (SHOULD)** mời AppSec/Security tham gia.
3. Kết quả threat modeling (mối đe dọa nhận diện và biện pháp giảm thiểu) **PHẢI (MUST)** được ghi lại thành tài liệu ngắn gọn, gắn kèm thiết kế kỹ thuật hoặc ADR liên quan (SDGP Điều 5.8), làm bằng chứng tuân thủ.
4. Đối với tính năng rủi ro thấp/thông thường, đội **NÊN (SHOULD)** áp dụng checklist rút gọn thay vì một phiên threat modeling đầy đủ.

5.2 Secure Coding Baseline

1. Kỹ sư **PHẢI (MUST)** tuân thủ baseline lập trình an toàn dựa trên danh mục rủi ro ứng dụng web được cộng đồng bảo mật công nhận rộng rãi hiện hành (OWASP Top 10:2025), bao gồm tối thiểu: kiểm soát truy cập bị lỗi (broken access control), cấu hình sai bảo mật (security misconfiguration), rủi ro chuỗi cung ứng phần mềm, injection (SQL/command/LDAP...), và sử dụng thành phần có lỗ hổng đã biết.
2. Với hệ thống có yêu cầu bảo mật cao, đội **NÊN (SHOULD)** đối chiếu thêm với OWASP Application Security Verification Standard (ASVS) v5.0 — khung kiểm tra xác minh bảo mật ứng dụng theo cấp độ rủi ro (level-based) — để xác định mức độ kiểm soát cần đạt.
3. Chi tiết kỹ thuật cụ thể (coding pattern được phép/cấm theo từng ngôn ngữ, thư viện mã hóa được phê duyệt, cấu hình an toàn theo framework...) do **Secure Coding Standard (Tầng 3)** quy định — Policy này chỉ đặt ra yêu cầu ở mức nguyên tắc.
4. Kỹ sư **KHÔNG ĐƯỢC PHÉP (MUST NOT)** tự triển khai thuật toán mã hóa/băm mật khẩu riêng khi đã có thư viện chuẩn ngành được tổ chức phê duyệt.

5.3 Security Gate tự động trong CI/CD

1. Pipeline CI/CD của mọi hệ thống thuộc phạm vi Policy này **PHẢI (MUST)** tích hợp tối thiểu các lớp kiểm tra bảo mật tự động phù hợp với loại hệ thống: SAST (quét mã nguồn tĩnh), SCA (quét thư viện/dependency), quét container image và Infrastructure-as-Code (IaC) nếu áp dụng, và DAST — DAST **NÊN (SHOULD)** bắt buộc cho ứng dụng có giao diện web/API lộ diện ra ngoài.
2. Tổ chức **PHẢI (MUST)** thiết lập ngưỡng nghiêm trọng (severity threshold) làm điều kiện chặn: phát hiện lỗ hổng ở mức Critical **PHẢI (MUST)** chặn merge vào nhánh chính; phát hiện ở mức High **NÊN (SHOULD)** chặn deploy lên production trừ khi có ngoại lệ được phê duyệt theo Điều 8.
3. Pipeline build cho hệ thống triển khai lên production **NÊN (SHOULD)** tự động sinh SBOM (Software Bill of Materials), phục vụ truy xuất nguồn gốc thành phần khi phát sinh lỗ hổng chuỗi cung ứng mới.
4. Ngưỡng chi tiết theo từng công cụ, từng loại hệ thống, và quy trình xử lý false positive do Secure Coding Standard hoặc một Standard/Procedure chuyên biệt (Tầng 3/4) quy định cụ thể, để tránh cứng nhắc hóa

ở tầng Policy.

5. Đội phát triển **KHÔNG ĐƯỢC PHÉP (MUST NOT)** tự ý tắt/bỏ qua (skip) security gate để đẩy nhanh tiến độ; việc bỏ qua tạm thời chỉ được thực hiện qua cơ chế ngoại lệ tại Điều 8.

5.4 Quản lý lỗ hổng (Vulnerability Management)

1. Mọi lỗ hổng phát hiện được (qua security gate, pentest, bug bounty, hoặc báo cáo từ bên ngoài) **PHẢI (MUST)** được phân loại mức độ nghiêm trọng theo thang Critical/High/Medium/Low, và xử lý theo SLA (thời hạn khắc phục tối đa) tương ứng.
2. Khung thời gian SLA cụ thể cho từng mức độ nghiêm trọng do tổ chức tự quy định trong Vulnerability Management Standard (Tầng 3), dựa trên năng lực đội ngũ thực tế và mức độ tiếp xúc rủi ro của từng hệ thống — Policy này không ấn định con số cứng để tránh cam kết không khả thi ở tầng Policy.
3. Lỗ hổng ở mức Critical **PHẢI (MUST)** được thông báo ngay cho Tech Lead và AppSec/Security liên quan, không chờ đến chu kỳ báo cáo định kỳ.
4. Tình trạng xử lý lỗ hổng (đặc biệt các lỗ hổng quá hạn SLA) **PHẢI (MUST)** được theo dõi tập trung và báo cáo định kỳ cho Engineering Governance Council.

5.5 Quản lý Secret/Credentials

1. Kỹ sư **KHÔNG ĐƯỢC PHÉP (MUST NOT)** hard-code secret/credential (API key, mật khẩu, token, private key...) trực tiếp trong mã nguồn, file cấu hình commit vào kho mã, hoặc log ứng dụng.
2. Secret **PHẢI (MUST)** được lưu trữ và cấp phát qua secret vault/secret manager được tổ chức phê duyệt, có khả năng luân chuyển (rotate) và thu hồi khi cần.
3. Pipeline CI/CD **PHẢI (MUST)** chạy công cụ quét secret tự động (secret scanning) trên mọi commit/pull request để phát hiện secret bị lộ trước khi vào nhánh chính.
4. Khi phát hiện secret bị lộ (kể cả trong lịch sử commit đã cũ), đội liên quan **PHẢI (MUST)** thu hồi/luân chuyển secret đó ngay, không chỉ đơn thuần xóa khỏi mã nguồn hiện tại.

5.6 Đào tạo và Security Champions

1. Áp dụng chương trình đào tạo nền tảng tại SDGP Điều 5.12; ngoài ra, mọi kỹ sư viết mã **PHẢI (MUST)** hoàn thành đào tạo secure coding baseline (Điều 5.2) theo chu kỳ do tổ chức quy định.
2. Mỗi đội phát triển **NÊN (SHOULD)** chỉ định ít nhất một Security Champion — đầu mối kỹ thuật về thực hành bảo mật trong đội, tham gia threat modeling và làm cầu nối với đội AppSec/Security trung tâm.
3. Security Champion không bắt buộc phải là chuyên gia bảo mật toàn thời gian, nhưng **NÊN (SHOULD)** được đào tạo chuyên sâu hơn mặt bằng chung và được phân bổ thời gian hợp lý cho vai trò này.

5.7 Kiểm thử xâm nhập (Penetration Testing) định kỳ

1. Hệ thống được phân loại rủi ro cao hoặc production quan trọng (theo Điều 5.1.1) **PHẢI (MUST)** được kiểm thử xâm nhập định kỳ (tần suất cụ thể do tổ chức quy định trong Standard liên quan), thực hiện bởi đội/bên có năng lực độc lập với đội phát triển hệ thống đó.
2. Trước khi ra mắt một tính năng lớn làm thay đổi đáng kể bề mặt tấn công (ví dụ mở API mới ra ngoài, thay đổi cơ chế xác thực), đội **NÊN (SHOULD)** thực hiện kiểm thử xâm nhập hoặc đánh giá bảo mật tương xứng trước khi go-live.
3. Lỗ hổng phát hiện qua pentest được xử lý theo đúng quy trình quản lý lỗ hổng tại Điều 5.4.

> **Không lặp lại nội dung của SDGP:** Các yêu cầu về SDLC tổng thể, vai trò/trách nhiệm chung, kiểm soát thay đổi, cơ chế ngoại lệ và không tuân thủ đã được quy định đầy đủ tại SDGP (Điều 5.1, Điều 8, Điều 9). Policy này chỉ mở rộng chi tiết phần bảo mật phần mềm đã nêu tại SDGP Điều 5.2; khi có xung đột diễn giải phần chung, SDGP được ưu tiên áp dụng.

6. Vai trò và trách nhiệm

ROLES & RESPONSIBILITIES

- **Head of Application Security:** chủ sở hữu Policy; phê duyệt Secure Coding Standard, Vulnerability Management Standard; quyết định ngưỡng security gate ở cấp tổ chức.
- **Engineering Governance Council:** phê duyệt Policy và sửa đổi; nhận báo cáo tình trạng lỗ hổng quá hạn SLA.
- **Tech Lead:** chủ trì threat modeling cho hệ thống mình phụ trách; bảo đảm security gate được tích hợp trong pipeline của đội.
- **Security Champion:** đầu mối thực hành bảo mật trong đội; hỗ trợ threat modeling và secure code review.
- **AppSec/Security Team:** duy trì công cụ SAST/DAST/SCA và secret scanning; điều phối pentest; tư vấn threat modeling cho hệ thống rủi ro cao nhất.
- **Kỹ sư phát triển:** tuân thủ secure coding baseline; xử lý lỗ hổng thuộc trách nhiệm mình theo SLA; không hard-code secret.

Các vai trò chung khác (Product Owner, QA, DevOps...) áp dụng theo đúng quy định tại SDGP Điều 6, bổ sung trách nhiệm bảo mật đặc thù nêu trên khi liên quan.

7. Đo lường tuân thủ

COMPLIANCE MEASUREMENT

- Tỷ lệ tính năng/hệ thống rủi ro cao có threat modeling trước khi code.
- Tỷ lệ pipeline CI/CD có đủ SAST + SCA + DAST hoạt động.
- Tỷ lệ lỗ hổng được xử lý đúng SLA theo từng mức độ nghiêm trọng; số lỗ hổng Critical quá hạn.
- Số lượng secret bị lộ phát hiện qua quét tự động; thời gian trung bình để thu hồi/luân chuyển sau phát hiện.
- Tỷ lệ đội có Security Champion được chỉ định.
- Số hệ thống rủi ro cao được pentest đúng chu kỳ; tỷ lệ phát hiện nghiêm trọng từ pentest đã được khắc phục.

8. Cơ chế ngoại lệ

EXCEPTIONS

Áp dụng theo đúng cơ chế và giới hạn tại Engineering Constitution, Điều 8 và SDGP, Điều 8. Riêng việc tạm thời bỏ qua security gate (Điều 5.3.5) hoặc trì hoãn SLA xử lý lỗ hổng Critical (Điều 5.4) **PHẢI (MUST)** được Head of Application Security hoặc người được ủy quyền phê duyệt bằng văn bản, có thời hạn cụ thể.

9. Không tuân thủ

NON-COMPLIANCE

Áp dụng theo đúng nguyên tắc và quy trình xử lý vi phạm tại SDGP, Điều 9: báo cáo và khắc phục cho vi phạm thông thường; chỉ xem xét kỷ luật lao động — qua HR/Pháp chế — đối với hành vi cố ý (ví dụ: cố tình

hard-code secret sau khi đã được đào tạo, hoặc cố tình tắt security gate mà không qua phê duyệt ngoại lệ).

10. Chủ sở hữu tài liệu và chu kỳ rà soát

DOCUMENT OWNERSHIP & REVIEW CYCLE

Document Owner: Head of Application Security (Trưởng nhóm Bảo mật Ứng dụng). Chu kỳ rà soát: hằng năm, hoặc ngay khi có nhóm lỗ hổng/kỹ thuật tấn công mới đáng kể xuất hiện — lĩnh vực bảo mật ứng dụng thay đổi nhanh nên chu kỳ rà soát cần linh hoạt hơn mức tối thiểu.

11. Tài liệu liên quan

RELATED DOCUMENTS

- Engineering Constitution v1.0, Software Development Governance Policy v1.0 (Tầng 1-2)
- Secure Coding Standard, Vulnerability Management Standard, Data Protection Standard (Tầng 3)
- Threat Modeling Procedure, Penetration Testing Procedure, Secret Management Procedure, Security Incident Response Procedure (Tầng 4/5)

Bảo mật không phải là cổng kiểm tra ở cuối, mà là câu hỏi được đặt ra ngay từ dòng thiết kế đầu tiên;
không secret nào nằm trong mã nguồn, không lỗ hổng Critical nào bị lãng quên quá hạn;
mọi công cụ tự động đều hỗ trợ con người quyết định, không thay thế con người chịu trách nhiệm;
và không hệ thống quan trọng nào ra mắt mà chưa từng bị thử tấn công bởi chính đội mình.

12. Phê duyệt

APPROVAL

Vai trò	Họ tên	Chữ ký	Ngày
Chair, Engineering Governance Council			
Chief Technology Officer			
Head of Application Security (Document Owner)			
Head of Engineering			