BỘ THÔNG TIN VÀ TRUYỀN THÔNG
CỤC AN TOÀN THÔNG TIN

Số: 166 /CATTT-ATHTTT
V/v Ban hành hướng dẫn "Khung phát triển phần mềm an toàn (phiên bản 1.0)"

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

Hà Nội, ngày 10 tháng 02 năm 2022

Kính gửi:

- Đơn vị chuyên trách về CNTT của các Bộ, cơ quan ngang Bộ, cơ quan thuộc Chính phủ;
- Sở TT&TT các tỉnh, thành phố trực thuộc Trung ương;
- Hiệp hội An toàn thông tin Việt Nam;
- Các doanh nghiệp phần mềm;
- Các doanh nghiệp cung cấp sản phẩm, dịch vụ ATTT.

Thực hiện chức năng quản lý nhà nước về an toàn thông tin của Cục An toàn thông tin, Cục An toàn thông tin ban hành văn bản "Khung phát triển phần mềm an toàn (phiên bản 1.0)".

Tài liệu Hướng dẫn này nhằm khuyến nghị các cơ quan, đơn vị, doanh nghiệp phát mềm có thể phát hiện lỗ hổng, điểm yếu bảo mật và kịp thời khắc phục trong quá trình phát triển phần mềm.

Bản mềm tài liệu hướng dẫn đăng tải tại địa chỉ: https://ais.gov.vn/thong-tin-tham-khao/ khung-huong-dan-phan-mem-an-toan.htm

Trong quá trình thực hiện, nếu có vấn đề vướng mắc, đề nghị Quý cơ quan, tổ chức phản ánh về Cục An toàn thông tin để được hướng dẫn thực hiện.

Chi tiết liên hệ: Ông Mạc Đức Nam Khánh, Phòng An toàn hệ thống thông tin, Cục An toàn thông tin, số điện thoại: 0702213756, địa chỉ thư điện tử: mdnkhanh@mic.gov.vn.

Trân trọng./.

Nơi nhận:
- Như trên;
- Bộ trưởng (để b/c);
- Thứ trưởng Nguyễn Huy Dũng (để b/c);
- Cục trưởng;
- Lưu: VT, ATHTTT.

CỤC TRƯỞNG

CỘNG HOÀ X.H.C.N VIỆT NAM
CỤC AN TOÀN THÔNG TIN
BỘ THÔNG TIN VÀ TRUYỀN THÔNG

Nguyễn Thành Phúc

Hướng dẫn
KHUNG PHÁT TRIỂN PHẦN MỀM AN TOÀN PHIÊN BẢN 1.0

(Kèm theo Công văn số 166 /BTTTT-CATTT ngày 10 tháng 02 năm 2022 của Cục An toàn thông tin)

1. Giới thiệu

Vòng đời phát triển phần mềm (SDLC - Software Development Life Cycle) là phương pháp luận hay một chuỗi các quy trình để phát triển phần mềm từ thiết kế, xây dựng và duy trì phần mềm. Có một số mô hình SDLC phổ biến gồm mô hình thác nước (waterfall), xoắn ốc (spiral), phát triển phần mềm linh hoạt (agile), đặc biệt mô hình agile kết hợp giữa phát triển phần mềm và triển khai. Rất ít mô hình SDLC đề cập chi tiết đến vấn đề an toàn phần mềm, vì vậy các phương pháp phát triển phần mềm an toàn thường cần được bổ sung và tích hợp vào từng mô hình SDLC. Bất kể mô hình SDLC nào được sử dụng, các phương pháp phát triển phần mềm an toàn cần được tích hợp trong suốt quá trình vì ba lý do: để giảm số lượng điểm yếu (vulnerability) trong phần mềm đã phát hành, để giảm thiểu tác động tiềm ẩn của việc khai thác các điểm yếu chưa được phát hiện hoặc chưa được khắc phục và để giải quyết gốc rễ nguyên nhân của các điểm yếu để ngăn chặn sự xuất hiện lặp lại.

Hầu hết các khía cạnh của an toàn phần mềm có thể được giải quyết nhiều lần trong SDLC, tuy nhiên, vấn đề an toàn được giải quyết càng sớm thì càng cần ít nguồn lực và chi phí để đạt được cùng một mức độ an toàn. Nguyên tắc này, còn được gọi là *dịch chuyển sang trái (shifting left)*, là nguyên tắc quan trọng bất kể mô hình SDLC nào. Dịch chuyển sang trái giúp giảm thiểu bất kỳ tồn tại kỹ thuật chúng đòi hỏi cần khắc phục các lỗi bảo mật sớm trong quá trình phát triển hoặc sau khi phần mềm được đưa vào sản xuất.

Tài liệu này xác định và khuyến nghị các phương pháp phát triển phần mềm an toàn, trọng tâm của tài liệu sẽ nhằm đưa ra các hướng dẫn về mặt quản lí, cách thức để thực hiện, không liên quan đến các yêu cầu về mặt kĩ thuật, công

cụ và cơ chế, hướng đến:

• Có thể được sử dụng bởi các tổ chức trong bất kỳ lĩnh vực hoặc cộng đồng nào, bất kể quy mô hoặc mức độ phức tạp về an toàn thông tin mạng

• Có thể được áp dụng cho phần mềm được phát triển để hỗ trợ công nghệ thông tin (CNTT), hệ thống điều khiển công nghiệp (ICS), hệ thống vật lý mạng (CPS) hoặc Internet of Things (IoT)

• Có thể được tích hợp vào bất kỳ quy trình phát triển phần mềm hiện có nào và chuỗi công cụ tự động; không được ảnh hưởng tiêu cực đến các tổ chức đã áp dụng các hoạt động phát triển phần mềm an toàn.

• Có thể áp dụng rộng rãi, không phân biệt riêng cho các công nghệ, nền tảng, ngôn ngữ lập trình, mô hình SDLC, môi trường phát triển, môi trường hoạt động, công cụ, v.v.

• Có thể giúp một tổ chức lập tài liệu về các thực hành phát triển phần mềm an toàn hiện tại của mình và xác định các mục tiêu thực hành trong tương lai như một phần của quá trình cải tiến liên tục.

• Có thể hỗ trợ một tổ chức hiện đang sử dụng mô hình phát triển phần mềm cổ điển trong việc chuyển đổi các phương pháp phát triển phần mềm an toàn sang sử dụng với mô hình phát triển phần mềm hiện đại (ví dụ: Agile, DevOps).

• Có thể hỗ trợ các tổ chức đang mua sắm và sử dụng phần mềm hiểu được các phương pháp phát triển phần mềm an toàn do các nhà cung cấp của họ sử dụng.

2. Khung phát triển phần mềm an toàn

Tài liệu này xác định phiên bản 1.0 Khung phát triển phần mềm an toàn (SSDF - Security Software Development Framework) được chia thành bốn nhóm:

- *Giai đoạn chuẩn bị của tổ chức (PO - Prepare the Organization):* Các tổ chức phải đảm bảo nhân sự, quy trình và công nghệ để thực hiện phát triển phần mềm an toàn tại cấp độ tổ chức. Giai đoạn này cũng có thể áp dụng cho quá trình phát triển phần mềm an toàn, như các nhóm phát triển riêng lẻ hoặc các dự án. Phần này tập trung vào môi trường vận hành của phần mềm.

- *Bảo vệ phần mềm (PS - Protect the Software):* Các tổ chức cần bảo vệ tất cả các thành phần của phần mềm phòng chống giả mạo và truy cập trái phép. Giai đoạn này tập trung vào môi trường vận hành của phần mềm

- *Sản xuất phần mềm đảm bảo an toàn (PW - Produce Well-Security Software):* Các tổ chức cần sản xuất phần mềm đảm bảo an toàn với tối thiểu các điểm yếu tồn tại trong các phiên bản phát hành. Giai đoạn này tập trung vào quy định các khâu phát triển phần mềm từ thiết kế, kiểm định chất lượng – độ an tòa, lập trình, cấu hình môi trường phát triển phần mềm, kiểm thử,…tức là quy trình phát triển phần mềm.

- *Ứng phó với các lỗ hổng bảo mật (RV – Respond to Vulnerabilities):* Các tổ chức phải xác định điểm yếu bảo mật còn tồn tại trong các phiên bản phần mềm phát hành và ứng phó tương ứng để xử lý những điểm yếu này và ngăn ngừa tương tự xảy ra trong tương lai. Giai đoạn này sẽ ưu tiên khắc phục sự cố và ngăn ngừa khả năng tái hiện lỗ hổng của phần mềm sau này.

Mỗi quy định được xác định bao gồm các thành phần sau đây:

- *Quy định:* Tên của quy định và một kí hiệu riêng, mô tả tóm lược về nội dung của quy định đó.

- *Nhiệm vụ*: Một hoặc nhiều tiêu chí cần để đáp ứng quy định đưa ra.

- *Ví dụ triển khai:* Một hoặc nhiều ví dụ của loại các công cụ, các quy trình và các phương thức có thể được sử dụng để giúp trển khai một nhiệm vụ, không nhằm ngụ ý rằng bất kỳ ví dụ hoặc tổ hợp các ví dụ nào là bắt buộc hoặc chỉ những ví dụ đã nêu là các phương án khả thi.

Danh mục từ viết tắt

SDLC	Software Development Life Cycle	Vòng đời phát triển phần mềm
SSDF	Secure Software Development Framework	Khung phát triển phần mềm an toàn
KPI	Key Performance Indicator	Chỉ số hiệu suất chính
KRI	Key Result Indicators	Chỉ số rủi ro chính
PO	Prepare the Organization	Chuẩn bị của tổ chức
PS	Protect the Software	Bảo vệ phần mềm
PW	- Produce Well-Security Software	Sản xuất phần mềm đảm bảo an toàn
RV	Respond to Vulnerabilities	Ứng phó với các lỗ hổng bảo mật
ICS	Industrial Control System	Hệ thống điều khiển công nghiệp
CPS	Cyber-physical systems	Hệ thống vật lý mạng
IoT	Internet of Things	Internet vạn vật

Khung phát triển phần mềm an toàn (SSDF)

Quy định	Nhiệm vụ	Ví dụ triển khai
1.Chuẩn bị của tổ chức (PO)		
1.1. Xác định yêu cầu bảo mật cho phát triển phần mềm (PO.1): Đảm bảo tính bảo mật luôn gắn liền với SDLC. Bao gồm các yêu cầu từ các nguồn nội bộ (ví dụ: chính sách, mục tiêu kinh doanh và chiến lược quản lý rủi ro của tổ chức) và các nguồn bên ngoài (ví dụ: luật và quy định hiện hành).	**PO.1.1:** Xác định và ghi lại các yêu cầu bảo mật cho cơ sở hạ tầng trong quá trình phát triển phần mềm, đồng thời thực hiện các yêu cầu này liên tục trong suốt SDLC.	• Xác định chính sách bảo mật cho cơ sở hạ tầng phát triển phần mềm, bao gồm cả việc bảo mật điểm cuối xuyên suốt SDLC và duy trì tính bảo mật liên tục. • Xác định chính sách để đảm bảo các quy trình phát triển phần mềm xuyên suốt SDLC và duy trì tính bảo mật liên tục, bao gồm các phần mềm nguồn mở, phần mềm bên thứ ba đang được sử dụng phục vụ phần mềm được phát triển. • Định kỳ thực hiện kiểm tra, đánh giá về bảo mật an toàn thông tin. Việc kiểm tra đánh giá ngoài thực hiện định kỳ, còn cần phải thực hiện khi có các yêu cầu mới từ các nguồn nội bộ và các nguồn bên ngoài, hoặc ngay khi có một sự cố lỗ hổng bảo mật lớn xảy ra. • Đào tạo, nâng cao kiến thức về an toàn thông tin cho nhân viên để đáp ứng được với các yêu cầu thực tế.
	PO.1.2: Xác định và áp dụng các yêu cầu bảo mật đối với các phần mềm để đáp ứng tiến độ đề ra.	• Xác định các chính sách, các yêu cầu từ khâu thiết kế và bảo đảm kiến trúc phần mềm dựa trên quản trị rủi ro. • Xác định các chính sách bảo mật cho phần mềm và kiểm tra tính tuân thủ quy định trong SDLC. • Phân tích rủi ro khi tích hợp nhiều công cụ, công nghệ trong cùng sản phẩm. Sau khi phân tích rủi ro, cần đề xuất các công cụ, công nghệ khác có khả năng giảm thiểu các rủi ro. • Quy định các yêu cầu về sao lưu và thời gian lưu trữ phần mềm dựa trên mô hình SDLC.

Quy định	Nhiệm vụ	Ví dụ triển khai
		• Tuân thủ các chính sách như: Vòng đời phần mềm, thông báo cho bên sử dụng về thời hạn và hỗ trợ đối với phần mềm. • Kiểm tra, đánh giá tính bảo mật của phần mềm.
	PO.1.3: Các yêu cầu cho nhà cung cấp sản phẩm, dịch vụ.	• Xác định yêu cầu bảo mật trong các hợp đồng và tại các cam kết của bên cung cấp phần mềm. • Xác định các tiêu chí liên quan đến bảo mật trong quá trình lựa chọn, phát triển phần mềm. • Yêu cầu nhà cung cấp sản phẩm, dịch vụ chứng minh nguồn gốc và tính bảo mật của sản phẩm. Cần bảo đảm tính bảo mật của sản phẩm phải tuân thủ các yêu cầu bảo mật của tổ chức. • Thiết lập và tuân thủ các quy trình đánh giá rủi ro. • Thiết lập và tuân theo các quy trình để giải quyết rủi ro.
1.2. Thực hiện các vai trò và trách nhiệm (PO.2): Đảm bảo mọi yếu tố liên quan đến SDLC được chuẩn bị để thực hiện các vai trò và trách nhiệm liên quan đến SSDF trong suốt quá trình phát triển phần mềm.	**PO.2.1:** Xác định vai trò và trách nhiệm cho các vị trí của SSDF. Thường xuyên đánh giá, xác định lại vai trò và trách nhiệm của các thành viên trong nhóm.	• Xác định vai trò và trách nhiệm liên quan đến SSDF cho tất cả các thành viên của nhóm phát triển phần mềm. • Tích hợp các vai trò bảo mật vào quy trình phát triển phần mềm. • Xác định vai trò về bảo đảm an toàn thông tin của từng vị trí (từ quản lý cấp cao đến nhân viên và bên sử dụng vận hành sản phẩm) có liên quan đến SDLC. • Định kỳ kiểm tra, đánh giá vai trò, trách nhiệm của từng vị trí nhân viên. • Đào tạo, nâng cao kiến thức về những xu hướng mới cho từng cá nhân liên quan.

Quy định	Nhiệm vụ	Ví dụ triển khai
	PO.2.2: Đào tạo, nâng cao kiến thức về an toàn thông tin cho tất cả nhân viên. Định kỳ kiểm tra, đánh giá trình độ của nhân sự để cập nhật, bổ sung kiến thức cần thiết về an toàn thông tin.	• Xác định yêu cầu kiến thức đào tạo đối với từng đối tượng. • Xác định loại hình đào tạo hoặc chương trình đào tạo cần thiết cho từng vị trí nhân sự. • Tổ chức các khóa học phù hợp với từng vị trí nhân viên. • Kiểm tra, đánh giá năng lực định kỳ của từng nhân viên từ đó có định hướng và thay đổi chương trình đào tạo cho phù hợp.
	PO.2.3: Xây dựng và áp dụng các quy định về SDLC cùng các vai trò và trách nhiệm liên quan đến SSDF.	• Chỉ định một lãnh đạo hoặc nhóm lãnh đạo chịu trách nhiệm về toàn bộ quy trình phát triển phần mềm an toàn. • Nâng cao nhận thức về các rủi ro sẽ gặp khi phát triển phần mềm không tích hợp tính bảo mật trong suốt vòng đời phát triển và giảm thiểu rủi ro do thực tiễn SSDF cung cấp. • Phổ biến kiến thức cho nhân viên có vai trò, trách nhiệm liên quan đến SSDF về các quy định và tầm quan trọng của SSDF đối với tổ chức.
1.3. Triển khai các công cụ hỗ trợ (PO.3): Sử dụng tự động hóa để giảm bớt nhân lực, cải thiện độ chính xác, tính nhất quán, khả năng sử dụng và tính toàn diện của các phương thức bảo mật trong suốt SDLC, cung cấp cách lập hồ sơ và trình bày việc sử dụng phương pháp.	**PO.3.1:** Chỉ định công cụ hoặc nhóm công cụ cần thiết để giảm thiểu rủi ro và bảo đảm tính bảo mật khi tích hợp các công cụ.	• Xây dựng danh mục công cụ và xác định công cụ phù hợp đối với từng danh mục. • Xác định công cụ bảo mật để tích hợp vào chuỗi các công cụ dành cho nhà phát triển. • Đánh giá khả năng/năng lực của từng công cụ. • Sử dụng công nghệ tự động để quản lý và điều phối chuỗi công cụ. • Thường xuyên tối ưu công cụ để phù hợp với hoạt động, đặc thù ngôn ngữ, framework tại tổ chức.

Quy định	Nhiệm vụ	Ví dụ triển khai
.		
	PO.3.2: Thực hiện theo các phương pháp bảo mật được khuyến nghị để triển khai và duy trì các công cụ và chuỗi các công cụ.	• Kiểm tra, đánh giá tính bảo mật của từng công cụ. • Tích hợp các công cụ và tuân theo các quy trình phát triển phần mềm hiện có. • Áp dụng công nghệ mới và các quy trình cần thiết cho các bản dựng mẫu. • Cập nhật, nâng cấp hoặc thay thế các công cụ có tính năng vá lỗ hổng bảo mật và thêm các tính năng mới cho công cụ. • Định kỳ kiểm tra, đánh giá các công cụ để phát hiện lỗ hổng bảo mật. • Định kỳ kiểm tra, đánh giá tính toàn vẹn của sản phẩm để xác định các rủi ro tiềm ẩn.
	PO.3.3: Cài đặt, cấu hình các công cụ để đáp ứng về tính năng bảo mật theo quy định	• Sử dụng công cụ sẵn có để tạo sự kiện kiểm tra về các hành động liên quan đến phát triển an toàn. • Xây dựng và áp dụng các chính sách bảo mật và sao lưu dữ liệu.

Quy định	Nhiệm vụ	Ví dụ triển khai
1.4. Xác định và sử dụng các tiêu chí cho bảo mật phần mềm kiểm tra, đánh giá (PO.4): Bảo đảm phần mềm theo SDLC đáp ứng các tiêu chí khi được kiểm tra, đánh giá bảo mật trong quá trình phát triển.	**PO.4.1:** Xác định tiêu chí kiểm tra, đánh giá tính bảo mật của phần mềm và theo dõi trong suốt SDLC.	• Đảm bảo các tiêu chí về quản lý rủi ro an toàn thông tin. • Xác định các chỉ số hiệu suất chính (KPI) và các chỉ số rủi ro chính (KRI) để bảo mật phần mềm. • Bổ sung các tiêu chí bảo mật phần mềm vào các nội dung kiểm tra, đánh giá. • Theo dõi, phát hiện các bước bỏ qua không tuân thủ quy định vè kiểm tra đánh giá yêu cầu bảo mật. • Báo cáo kết quả kiểm tra, đánh giá an toàn thông tin.
	PO.4.2: Thực hiện các quy trình để thu thập và bảo vệ thông tin cần thiết để bảo đảm các tiêu chí về an toàn thông tin.	• Sử dụng các công cụ để thu thập thông tin cho việc đưa ra các yêu cầu bảo mật. • Bổ sung các công cụ cần thiết để hỗ trợ việc thu thập thông tin hỗ trợ các tiêu chí. • Tự động hóa các quy trình áp dụng các tiêu chí. • Chỉ cho phép nhân viên được ủy quyền để truy cập thông tin thu thập và ngăn chặn bất kỳ thay đổi hoặc xóa thông tin.

Quy định	Nhiệm vụ	Ví dụ triển khai
1.5. Thực hiện và duy trì bảo mật môi trường phát triển phần mềm (PO.5): Bảo đảm các yếu tố tác động từ môi trường để quy trình phát triển phần mềm được bảo vệ khỏi các mối đe dọa từ bên trong và bên ngoài.	**PO.5.1:** Tách biệt và bảo vệ từng môi trường liên quan đến phát triển phần mềm.	• Áp dụng phương pháp xác thực, nhận dạng riêng biệt với xác thực dựa trên rủi ro và quyền truy cập có điều kiện cho từng môi trường khác nhau. • Sử dụng phân đoạn mạng và kiểm soát truy cập để tách môi trường phát triển thành từng phần, từng khâu riêng biệt để giảm thiểu ảnh hưởng khi bị tấn công. • Triển khai xác thực và hạn chế các liên kết để tránh bị ảnh hưởng lẫn nhau giữa các môi trường khi xuất hiện các rủi ro. Chỉ sử dụng Internet khi thật sự cần thiết. • Thường xuyên ghi nhật ký giám sát và kiểm tra các môi trường liên kết giữa các thành phần. • Thực hiện ghi log thường xuyên và giám sát các hoạt động để kịp thời đưa ra các cảnh báo và có biện pháp ứng phó khi xuất hiện các sự cố mất an toàn thông tin mạng. • Cấu hình các biện pháp kiểm soát bảo mật và công cụ thực thi bảo mật để bảo vệ an toàn cho môi trường phát triển phần mềm. • Thường xuyên kiểm tra dữ liệu các phần mềm và kiểm tra, đánh giá lỗ hổng bảo mật trên các phần mềm đó.
	PO.5.2: Thực hiện quản lý, đánh giá rủi ro an toàn thông tin cho các sản phẩm, phần mềm trước khi đưa vào sử dụng.	• Cấu hình phần mềm trước khi đưa vào sử dụng dựa trên các quy định đã được phê duyệt. • Cấu hình phần mềm trước khi đưa vào sử dụng để cung cấp chức năng, dịch vụ cần thiết cho người dùng. • Liên tục theo dõi tình trạng bảo mật của các sản phẩm, phần mềm trước khi đưa vào sử dụng. • Áp dụng các biện pháp bảo mật và công cụ đánh giá, bảo đảm tính bảo mật cho các sản phẩm, phần mềm trước khi đưa vào sử dụng.

Quy định	Nhiệm vụ	Ví dụ triển khai
		• Yêu cầu xác thực đa yếu tố cho quyền truy cập vào các sản phẩm, phần mềm trước khi đưa vào sử dụng.
2.Bảo vệ phần mềm		
2.1. Bảo vệ tất cả các dạng mã nguồn không bị xâm nhập và giả mạo trái phép (PS.1): Ngăn chặn những thay đổi trái phép đối với mã nguồn.	**PS.1.1:** Lưu trữ các dạng mã, bao gồm cả mã nguồn và mã thực thi, chỉ những người được ủy quyền, các công cụ, các công cụ mới có thể truy cập khi cần thiết.	• Lưu trữ tất cả mã nguồn và hạn chế quyền truy cập. Ví dụ, các mã nguồn mở được sử dụng dùng cho mục đích truy cập công cộng, trường hợp này tính toàn vẹn và tính khả dụng phải được bảo vệ. • Sử dụng các tính năng kiểm soát để theo dõi các thay đổi được thực hiện đối với mã. • Xem xét và phê duyệt tất cả các thay đổi được thực hiện đối với mã sau khi mã đã được tự động quét các lỗ hổng bảo mật. • Sử dụng chữ ký số trên mã nguồn/mã thực thi để bảo vệ tính toàn vẹn của chúng. • Sử dụng mật mã (ví dụ: hàm băm mật mã) để giúp bảo vệ tính toàn vẹn của tệp.
2.2. Cung cấp cơ chế xác minh tính toàn vẹn của phần mềm (PS.2): Bảo đảm người mua và người sử dụng phần mềm là hợp pháp, không giả mạo.	**PS.2.1:** Cam kết bảo đảm tính toàn vẹn cho người mua và người sử dụng phần mềm.	• Với các tệp được công bố, cung cấp mã băm trên các trang web an toàn. • Cung cấp các chứng chỉ đã được thẩm định bởi cơ quan có thẩm quyền, thiết lập ký mã khóa để hệ điều hành, công cụ và dịch vụ khác có thể xác nhận tính hợp lệ của chữ ký trước khi sử dụng. • Định kỳ đánh giá quy trình ký mã khóa, bao gồm: Gia hạn chứng chỉ, luân phiên, thu hồi và bảo vệ.

Quy định	Nhiệm vụ	Ví dụ triển khai
2.3. Sao lưu bảo đảm tính bảo mật cho các phần mềm (PS.3): Sao lưu các phần mềm để giúp xác định, phân tích và loại bỏ các lỗ hổng được phát hiện trong phần mềm sau khi được đưa vào sử dụng.	**PS.3.1:** Lưu trữ an toàn các tệp cần thiết, các thông tin bổ sung (ví dụ: thông tin xác minh tính toàn vẹn, thông tin xuất xứ) của mỗi phần mềm đã được đưa vào sử dụng.	• Thiết lập quyền truy cập cho các đối tượng có thể sử dụng dữ liệu với mục đích kiểm tra, đánh giá. • Sao lưu và bảo đảm tính toàn vẹn của phần mềm đã được đưa vào sử dụng. • Mã hóa các tệp dữ liệu nhạy cảm bằng thuật toán mã hóa mạnh.
	PS.3.2: Thu thập, bảo vệ, duy trì và chia sẻ thông tin xuất xứ cho tất cả các thành phần của mỗi bản phát hành phần mềm (ví dụ: thông tin về ngày phát hành, năm phát hành, số phiên bản,… trong bản phát hành của mỗi phần mềm)	• Cung cấp thông tin xuất xứ cho cá nhân, tổ chức mua phần mềm để phù hợp với chính sách của tổ chức. • Cung cấp thông tin xuất xứ cho các nhóm ứng cứu sự cố để hỗ trợ trong việc giảm thiểu lỗ hổng phần mềm. • Cập nhật thường xuyên và liên tục thông tin xuất xứ để theo dõi sự thay đổi của phần mềm khi cập nhật.
3.Sản xuất phần mềm đáp ứng yêu cầu bảo mật tốt (PW)		
3.1. Thiết kế phần mềm để đáp ứng các yêu cầu bảo mật và giảm thiểu rủi ro bảo mật (PW.1): Xác định và đánh giá các yêu cầu bảo mật cho	**PW.1.1:** Áp dụng phương pháp mô hình hóa rủi ro để đánh giá tính bảo mật phần mềm (VD: mô hình hóa mối đe dọa, mô hình hóa cuộc tấn công; mô hình hóa lược đồ tấn công).	• Đào tạo, nâng cao nhận thức về bảo mật, an toàn thông tin cho nhóm phát triển hoặc cộng tác với cá nhân, tổ chức chuyên về mô hình rủi ro để có biện pháp xử lý, giảm thiểu. • Thực hiện các đánh giá chuyên sâu hơn với các khu vực có nguy cơ rủi ro về bảo mật cao. • Rà soát các báo cáo và số liệu thống kê về lỗ hổng bảo mật phần mềm.

Quy định	Nhiệm vụ	Ví dụ triển khai
phần mềm; xác định những rủi ro bảo mật mà phần mềm có thể gặp phải trong quá trình hoạt động và cách thiết kế của phần mềm nhằm giảm thiểu những rủi ro.		• Phân loại dữ liệu để xác định từng loại dữ liệu mà phần mềm sẽ tương tác.
	PW.1.2: Sao lưu các yêu cầu về bảo mật phần mềm và các rủi ro để đưa ra các phương án thiết kế phù hợp.	• Ghi lại tác động đối với từng rủi ro, bao gồm các phương án giảm thiểu và lý do của bất kỳ trường hợp ngoại lệ nào đã được phê duyệt đối với các yêu cầu bảo mật.
	PW.1.3: Xây dựng các yêu cầu về tính năng và dịch vụ bảo mật. (ví dụ: tích hợp với hệ thống quản lý nhật ký, quản lý danh tính, kiểm soát truy cập và quản lý lỗ hổng bảo mật hiện có).	• Xây dựng thư viện phần mềm gồm các mô-đun để hỗ trợ các tính năng và dịch vụ bảo mật được tiêu chuẩn hóa. • Đưa ra các yêu cầu về tính năng và dịch vụ bảo mật trong quá trình phát triển phần mềm.
3.2. Rà soát, đánh giá thiết kế phần mềm để xác minh sự tuân thủ với các yêu cầu bảo mật và các yếu tố rủi ro (PW.2): Đảm bảo phần mềm đáp ứng các yêu cầu bảo mật và xử lý rủi ro.	**PW.2.1:** Yêu cầu cá nhân, đơn vị độc lập với đơn vị thiết kế kiểm tra đánh giá rủi ro và các yêu cầu bảo mật.	• Rà soát từ khâu thiết kế phần mềm bảo đảm các yêu cầu, tính năng bảo mật. • Rà soát, đánh giá các yếu tố rủi ro trong quá trình thiết kế phần mềm. • Rà soát, đánh giá các phương án xử lý rủi ro. • Yêu cầu nhà thiết kế phần mềm sửa lỗi để đáp ứng các yêu cầu. • Thay đổi thiết kế hoặc chiến lược xử lý rủi ro nếu không thể đáp ứng các yêu cầu bảo mật.

Quy định	Nhiệm vụ	Ví dụ triển khai
3.3. Kế thừa, phát triển tính năng bảo mật hiện có trên phần mềm thay vì đầu tư, mua sắm phần mềm mới bảo mật mới. **(PW.3):** Giảm chi phí phát triển phần mềm, đẩy nhanh quá trình phát triển phần mềm và giảm khả năng gia tăng lỗ hổng bảo mật vào phần mềm bằng cách sử dụng lại các mô-đun và dịch vụ phần mềm đã được kiểm tra tình trạng bảo mật.	**PW.3.1:** Mua sắm, đầu tư các thành phần phần mềm được bảo mật tốt (ví dụ: thư viện phần mềm, mô-đun, phần mềm trung gian…) từ các nhà phát triển và bên thứ ba.	• Rà soát, đánh giá các thành phần phần mềm trước khi đưa vào sử dụng. • Kiểm thử mã nguồn cho cho từng thành phần của phần mềm và đánh giá rủi ro có thể gây ra. • Xây dựng thư viện phần mềm để lưu trữ các thành phần mã nguồn mở đã được kiểm định và kiểm duyệt. • Xác định những thành phần phải có trong phát triển phần mềm.
	PW.3.2: Xây dựng các thành phần phần mềm được bảo mật tuân theo SDLC để đáp ứng các nhu cầu phát triển mà phần mềm của bên thứ ba không thể đáp ứng.	• Tuân thủ các yêu cầu bảo mật để phát triển phần mềm an toàn. • Xác định những thành phần phải có trong phát triển phần mềm.
	PW.3.3: Đã chuyển lên mục **PW.1.3**	
	PW.3.4: Kiểm tra, đánh giá tính tuân thủ các yêu cầu bảo mật ở các thành phần phần mềm, mã nguồn mở.	• Rà soát, đánh giá lỗ hổng bảo mật chưa được sửa chữa, khắc phục.
	PW.3.5: Rà soát, kiểm tra tính toàn vẹn của các thành phần phần mềm.	• Đảm bảo việc rà soát các thành phần của phần mềm được diễn ra định kì, thường xuyên; Phát hiện kịp thời, hạn chế xuất hiện lỗ hổng.

Quy định	Nhiệm vụ	Ví dụ triển khai
		• Lên phương án xử lý phần mềm không còn khả dụng. • Xác nhận tính toàn vẹn của các thành phần phần mềm thông qua chữ ký số hoặc các cơ chế khác.
3.4. Phát triển phần mềm bằng dựa trên các quy tắc bảo mật (PW.4): Giảm số lượng lỗ hổng bảo mật, giảm chi phí trong phần mềm với việc loại bỏ các lỗ hổng trong quá trình tạo mã nguồn, bằng cách tuân theo các tiêu chí về mức độ nghiêm trọng của lỗ hổng bảo mật do tổ chức xác định.	**PW.4.1:** Tuân thủ tất cả các phương pháp mã hóa an toàn phù hợp với ngôn ngữ và môi trường phát triển để đáp ứng các yêu cầu của tổ chức.	• Xác thực đầu vào, xác thực và mã hóa đầu ra. • Tránh sử dụng các chức năng và cuộc gọi không an toàn. • Cung cấp khả năng ghi nhật ký và truy vết. • Sử dụng các IDE có khả năng bảo mật tốt. • Kiểm tra các lỗ hổng bảo mật khác, thường gặp đối với môi trường và ngôn ngữ phát triển. • Yêu cầu nhà phát triển xem xét mã nguồn phần mềm để bổ sung (không thay thế), việc đánh giá mã do người hoặc công cụ khác thực hiện.
3.5. Cấu hình môi trường phát triển tích hợp, biên dịch, trình thông dịch và xây dựng các quy trình để cải thiện khả năng bảo mật (PW.5): Giảm số lượng lỗ hổng bảo mật trong phần mềm,	**PW.5.1:** Sử dụng trình biên dịch, trình thông dịch và xây dựng các công cụ cung cấp các tính năng cải thiện tính bảo mật.	• Sử dụng các bản cập nhật của công cụ biên dịch, công cụ thông dịch và công cụ thiết lập. • Tuân thủ các quy trình quản lý thay đổi khi triển khai hoặc cập nhật các công cụ biên dịch, công cụ thông dịch và công cụ thiết lập, đồng thời kiểm tra tất cả các thay đổi không hợp lệ đối với các công cụ. • Xác nhận thường xuyên tính xác thực và tính toàn vẹn của các công cụ biên dịch, công cụ thông dịch và công cụ thiết lập.
	PW.5.2: Xác định các tính năng của trình biên dịch, trình	• Bật các tính năng của trình biên dịch, tạo ra các cảnh báo cho mã bảo mật kém an toàn trong quá trình biên dịch.

Quy định	Nhiệm vụ	Ví dụ triển khai
loại bỏ các lỗ hổng trước khi kiểm thử để giảm chi phí.	thông dịch, công cụ thiết lập và cách cấu hình từng tính năng, sau đó triển khai và sử dụng các cấu hình đã được phê duyệt.	• Tiến hành kiểm thử để đảm bảo các tính năng hoạt động tốt, tránh gây ra sự cố vận hành hoặc các sự cố khác. • Xác minh liên tục các cấu hình đã phê duyệt và đang sử dụng. • Cung cấp thông tin về trình biên dịch, trình thông dịch và cấu hình công cụ thiết lập trong cơ sở kiến thức của các nhà phát triển có thể truy cập, tìm kiếm và tái tạo trong môi trường phát triển cục bộ.
3.6. Đánh giá và phân tích mã Human-Readable để xác định các lỗ hổng và xác minh sự tuân thủ với các yêu cầu bảo mật **(PW.6):** Xác định, khắc phục các lỗ hổng trước khi phần mềm được đưa vào sử dụng nhằm ngăn chặn việc khai thác lỗ hổng bảo mật. Sử dụng các công cụ, tính năng phát hiện lỗ hổng có sẵn để giảm tải nguồn lực, chi phí, thời gian.	**PW.6.1:** Xác định hai phương án: Thứ nhất, đánh giá mã nguồn được thực hiện trực tiếp bởi một nhân sự chuyên trách, thứ hai, sử dụng các công cụ tự động. Sẽ tùy vào tình hình thực tế của cơ quan tổ chức.	• Tuân thủ các chính sách và hướng dẫn của tổ chức về thời điểm nên thực hiện kiểm tra mã nguồn và cách thức tiến hành. • Lựa chọn phương pháp đánh giá hoặc phân tích mã nguồn dựa trên giai đoạn của phần mềm.
	PW.6.2: Kiểm tra, đánh giá, phân tích mã dựa trên các tiêu chuẩn an toàn thông tin. Phân	• Thực hiện kiểm tra, đánh giá ngang hàng về mã nguồn • Yêu cầu chuyên gia thực hiện đánh giá để kiểm tra mã cho backdoor và mã độc khác.

Quy định	Nhiệm vụ	Ví dụ triển khai
	loại và đưa ra phương án khắc phục lỗ hổng bảo mật.	• Áp dụng công cụ phân tích tĩnh để tự động kiểm tra lỗ hổng bảo mật và đánh giá sự tuân thủ các tiêu chuẩn mã hóa an toàn. • Xác minh tính tuân thủ của mã thông qua các tiêu chí kiểm tra đánh giá. • Sử dụng các công cụ tự động để xác định và khắc phục các hoạt động phần mềm không an toàn. • Xác định và ghi lại nguyên nhân của mỗi vấn đề đã phát hiện.
3.7. Kiểm tra thực thi luật để xác định lỗ hổng và xác minh sự tuân thủ với yêu cầu bảo mật (PW.7): Xác định các lỗ hổng có thể xử lý trước khi phần mềm được đưa vào sử dụng nhằm tái khai thác. Sử dụng các phương pháp tự động làm giảm nguồn lực trong việc phát hiện các lỗ hổng.	**PW.7.1:** Xác định việc thực hiện kiểm tra mã thực thi để xác định và loại bỏ các lớp lỗ hổng không được đề cập trong các bài đánh giá, phân tích hoặc thử nghiệm trước đó.	• Tuân thủ các chính sách hoặc nguyên tắc của tổ chức về thời điểm nên thực hiện kiểm tra mã và cách thức tiến hành.
	PW.7.2: Triển khai thực hiện phạm vi, thiết kế, thực hiện kiểm tra đánh giá và ghi lại kết quả, bao gồm ghi lại và xử lý tất cả các vấn đề đã phát hiện và các biện pháp khắc phục được	• Thực hiện kiểm tra chức năng của các tính năng bảo mật • Tích hợp kiểm tra lỗ hổng động vào bộ kiểm thử tự động của dự án. • Kết hợp các bài đánh giá lỗ hổng bảo mật trước đây vào bài đánh giá của dự án để đảm bảo các lỗi không xuất hiện lại.

Quy định	Nhiệm vụ	Ví dụ triển khai
	đề xuất trong quy trình làm việc của nhóm phát triển hoặc qua hệ thống theo dõi sự cố.	• Xem xét cơ sở hạ tầng và công nghệ phần mềm sẽ sử dụng trong khi phát triển các kế hoạch thử nghiệm. • Sử dụng các công cụ kiểm tra fuzz testing để tìm các vấn đề với việc xử lý đầu vào. • Sử dụng phương pháp kiểm thử xâm nhập trong trường hợp có sẵn mã nguồn nhằm nâng cao khả năng đánh giá. • Xác định và ghi lại nguyên nhân của từng vấn đề được phát hiện. • Ghi lại kinh nghiệm qua việc phân tích nguyên nhân lỗ hổng bảo mật để đưa ra hướng dẫn, khuyến nghị cho lập trình viên phát triển phần mềm có thể truy cập và tìm kiếm.
3.8. Cấu hình phần mềm cài đặt bảo mật theo mặc định (PW.8): Cải thiện tính bảo mật của phần mềm tại thời điểm cài đặt giảm khả năng phần mềm được triển khai với cài đặt bảo mật yếu.	**PW.8.1:** Xác lập chính sách bảo mật chuẩn để cài đặt, cấu hình bảo mật. Giúp các cài đặt mặc định được an toàn, không làm tác động tới các chức năng bảo mật được cung cấp bởi các nền tảng, cơ sở hạ tầng mạng.	• Lên kế hoạch kiểm tra để đảm bảo các cài đặt cấu hình mặc định hoạt động một cách tốt nhất, không gây ra điểm yếu và ảnh hưởng đến những hoạt động khác của hệ thống.
	PW.8.2: Triển khai cài đặt mặc định (hoặc nhóm cài đặt mặc định, nếu có), ghi lại từng cài đặt cho quản trị viên phần mềm.	• Xác minh cấu hình đã được phê duyệt có sẵn cho phần mềm. • Sao lưu mục đích của từng cài đặt, các tùy chọn, giá trị mặc định, mức độ liên quan đến bảo mật, tác động hoạt động tiềm năng và mối quan hệ với các cơ sở khác. • Sử dụng các cơ chế kỹ thuật để ghi lại cách mỗi cài đặt có thể được thực hiện và đánh giá bởi quản trị viên phần mềm.

Quy định	Nhiệm vụ	Ví dụ triển khai
		• Lưu trữ cấu hình mặc định ở định dạng có thể sử dụng được và tuân theo kiểm soát thay đổi thực hành để sửa đổi nó (ví dụ: cấu hình dưới dạng mã).
4. Ứng phó với các lỗ hổng bảo mật (RV)		
4.1. Xác định và xác nhận các lỗ hổng trên một nền tảng đang triển khai (RV.1): Đảm bảo các lỗ hổng được xác định sớm để có phương án điều chỉnh kịp thời, giảm khả năng bị tấn công.	**RV.1.1:** Thu thập thông tin từ người sử dụng, các nguồn công khai về các lỗ hổng tiềm năng trong phần mềm và các mô-đun phần mềm bên thứ ba sử dụng.	• Thiết lập chương trình báo cáo về các lỗ hổng bảo mật, giúp các chuyên gia bảo mật dễ dàng trong việc tiếp cận, tìm hiểu mã nguồn và báo cáo các lỗ hổng có thể gặp. • Giám sát cơ sở dữ liệu về lỗ hổng bảo mật, danh sách gửi thư bảo mật và các nguồn khác về báo cáo lỗ hổng thông qua các phương tiện thủ công hoặc tự động. • Sử dụng các nguồn thông tin tình báo về mối đe dọa để hiểu rõ hơn về cách các lỗ hổng đang bị khai thác. • Thường xuyên kiểm tra nguồn gốc và phần mềm dữ liệu cho mỗi bản phát hành phần mềm đang được sử dụng để xác định các lỗ hổng mới tiềm ẩn trong các ứng dụng. • Quản lý danh sách các thư viện, các thành phần, framework có liên quan bao gồm tên và phiên bản đang sử dụng
	RV.1.2: Đánh giá, phân tích và kiểm tra mã nguồn phần mềm để xác định việc phần mềm không tồn tại những lỗ hổng bảo mật.	• Cấu hình chuỗi công cụ để thực hiện phân tích và kiểm tra mã tự động một cách thường xuyên hoặc liên tục. • Chủ động truy xuất nguồn gốc và dữ liệu của phần mềm nhằm xác định kịp thời các lỗ hổng mới của phần mềm.
	RV.1.3: Có chính sách giải quyết việc khắc phục lỗ hổng bảo mật, đồng thời thực hiện các vai trò, trách nhiệm và quy	• Có Nhóm ứng phó sự cố về bảo mật sản phẩm (PSIRT) và các quy trình để xử lý các phản hồi đối với các báo cáo và sự cố về lỗ hổng bảo mật.

Quy định	Nhiệm vụ	Ví dụ triển khai
	trình cần thiết để hỗ trợ chính sách.	• Có nhật kí ghi lại những đánh giá, phản hồi về xử lý lỗ hổng, zero-days, lỗ hổng đang bị khai thác và những sự cố nghiêm trọng liên quan đến cộng đồng và ứng dụng.
4.2. Đánh giá và loại bỏ các lỗ hổng (RV.2): Để đảm bảo các lỗ hổng bảo mật được khắc phục kịp thời, tránh được những rủi ro bị tấn công	**RV.2.1:** Phân tích từng lỗ hổng để thu thập đầy đủ thông tin, đưa ra kế hoạch khắc phục.	• Sử dụng phần mềm theo dõi sự cố để ghi lại sự kiện rủi ro. • Dự trù kế hoạch để khắc phục các lỗ hổng • Tính toán ảnh hưởng của việc khai thác lỗ hổng. • Tính toán mọi phương án cần thiết để lập kế hoạch khắc phục lỗ hổng bảo mật.
	RV.2.2: Phát triển và thực hiện kế hoạch khắc phục cho từng lỗ hổng.	• Lập kế hoạch phản ứng nhanh với các lỗ hổng trước mắt, trước khi có những biện pháp khắc phục lâu dài. • Cung cấp biện pháp khắc phục cho người sử dụng sản phẩm thông qua thiết bị tự động và cơ chế phân phối đáng tin cậy. • Phát triển và ban hành các khuyến cáo bảo mật cho người sử dụng, bao gồm những mô tả về thay đổi trong phần mềm, cấu hình, cài đặt .
4.3. Phân tích chuyên sâu các lỗ hổng (RV.3): Giảm thiểu	**RV.3.1:** Phân tích tất cả các lỗ hổng đã xác định để tìm ra nguyên nhân ban đầu.	• Ghi lại nguyên nhân của mỗi vấn đề khi được phát hiện. • Ghi lại kinh nghiệm qua việc phân tích nguyên nhân lỗ hổng để đưa ra hướng dẫn, khuyến nghị cho lập trình viên phát triển phần mềm có thể truy cập và tìm kiếm.

Quy định	Nhiệm vụ	Ví dụ triển khai
các lỗ hổng trong tương lai.		
	RV.3.2: Phân tích nguyên nhân lỗ hổng theo từng thời điểm và từng giai đoạn để đưa ra những hướng dẫn chính xác.	• Ghi lại kinh nghiệm qua việc phân tích nguyên nhân lỗ hổng để đưa ra hướng dẫn, khuyến nghị cho lập trình viên phát triển phần mềm có thể truy cập và tìm kiếm. • Tích hợp thêm cơ chế tự động phát hiện vào các chuỗi công cụ để tự động phát hiện các nguyên nhân lỗ hổng trong tương lai.
	RV.3.3: Chủ động lên kế hoach rà quét lỗ hổng trong các phần mềm, và đưa ra cách khắc phục.	Xem PW.6 và PW.7.
	RV.3.4: Rà soát lại SDLC để chủ động đưa ra những cập nhật thích hợp nhằm ngăn chặn, giảm khả năng của lỗ hổng trong các bản cập nhật phần mềm hoặc trong phần mềm mới được tạo.	• Ghi lại kinh nghiệm qua việc phân tích nguyên nhân lỗ hổng để đưa ra hướng dẫn, khuyến nghị cho lập trình viên phát triển phần mềm có thể truy cập và tìm kiếm. • Lập kế hoạch và chủ động lên phương án thực hiện chỉnh sửa thay đổi đối với các hướng dẫn của SSDF.

CỤC AN TOÀN THÔNG TIN./.