製品開発者向けガイド

IPA 独立行政法人情報処理推進機構
セキュリティセンター

2026年3月

目次

はじめに

経営者の皆様へ

経営者が DX による事業価値向上を推進する上で、事業活動の基盤であるソフトウェアは欠かすことができない存在であり、今後もその重要性はより一層増大するものと考えられています。一方で、ソフトウェアの脆弱性を悪用するサイバー攻撃の発生は、後を絶たず、ひとたび被害を受けるとその影響は自組織内にとどまらず、社会インフラの停止やサプライチェーンの寸断等を通じて国民生活や経済社会活動に広範に及ぶ可能性があります。

ソフトウェアの製品開発企業における経営者は、このようなサイバー攻撃から、ソフトウェアの利用者を保護するため、以下に示す 2 つの取組に対して、一層の責任をもって対応することが必要です。

①自組織内でセキュアなソフトウェア開発を実践し、提供するソフトウェアに脆弱性を作り込まないこと

②発見された脆弱性や、脆弱性に起因して発生したインシデントに迅速に対応し、提供するソフトウェアのセキュリティを維持すること

経営者は、上記①の取組を推進することにより、ソフトウェア開発の初期段階でソフトウェアの脆弱性を把握・低減することができ、**脆弱性の修正に費やすトータルコストを低く抑えます。**上記①の取組においては、セキュア・バイ・デザイン及びセキュア・バイ・デフォルトの実践や、ソフトウェアのサプライチェーン全体でのセキュリティ対策、脆弱性検査などを推進することが経営者に求められます。

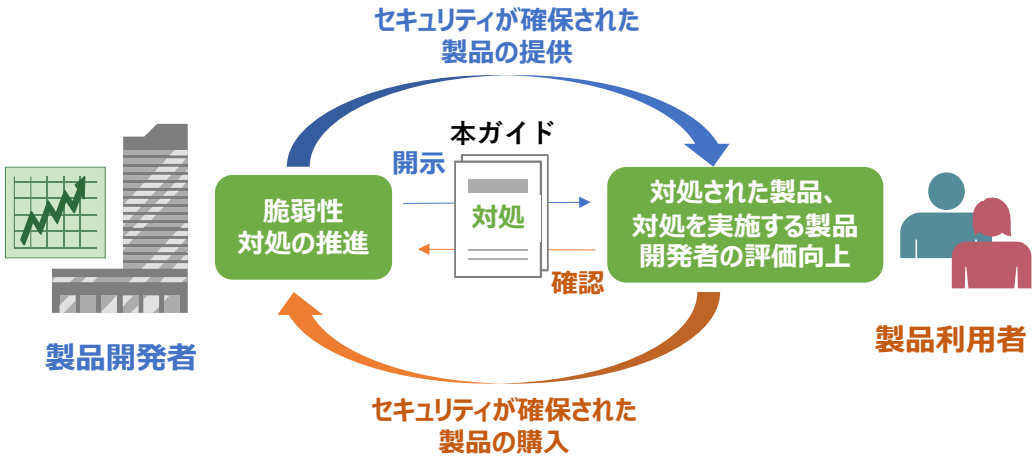
また、経営者は、上記②の取組を推進することにより、**ソフトウェアの脆弱性を悪用するサイバー攻撃による被害とその影響範囲を最小限に抑えます。**上記②の取組においては、脆弱性開示ポリシーの策定や、外部からの脆弱性報告を受け付ける対応窓口の設置、ソフトウェア部品表（SBOM）を活用した脆弱性管理などを推進することが経営者に求められます。

経営者は、自組織内にこのようなソフトウェア製品のセキュリティを確保する役割を担う**PSIRT 体制を構築します。**また PSIRT 体制を機能させるために、外部委託や組織間の連携強化を効果的に活用しつつ、開発部門や品質管理部門を含む組織内外の関係者を巻き込んで、**必要な人材・プロセス・技術の整備を推進します。**

上記を含め、脆弱性への対処（以降、脆弱性対処）については実施事項が多く、特に**リソースに限りのある製品開発者にとってはどれから着手してよいのかの判断に迷う**という課題があります。本ガイドでは、製品開発者が実施すべき対処について文献調査をもとに**重要と思われる 15 項目**をまとめました。**実施すべき対処について段階的に示して**いますので、可能なところから実施することができます。

また、製品開発者が実施する**脆弱性対処については、製品開発者が脆弱性対策にどのように取り組んでいるかが製品利用者からは分からないため、製品利用者からの製品評価向上、製品選定の優位性（顧客獲得及び売上等）につながりづらい**状況です。本ガイドでは、製品開発者が実施している脆弱性対処を製品利用者に伝えるため、実施内容の開示方法も併せて掲載しています。**本ガイドに沿って製品開発者が実施した対処を開示することで、製品利用者が対処状況を容易に確認することが可能**となります。この「製品開発者向けガイド」に対応する形で「製品利用者向けガイド」を作成しています。「製品利用者向けガイド」では、製品利用者に向けて、製品開発者が実施・開示する内容を踏まえて、ソフトウェア製品の選定・調達を行い、必要なセキュリティ設定やセキュリティ機能の活用を行うよう促しています。

これにより、顧客の安全に考慮している製品の評価の向上、製品開発者への信頼につながり、**セキュリティが確保された製品も市場原理として適い、普及することを期待**します。



図：脆弱性対処状況の開示と確認によるセキュリティが確保された製品の普及

本ガイドを参考に、組織の脆弱性対処を検討及び推進してください。

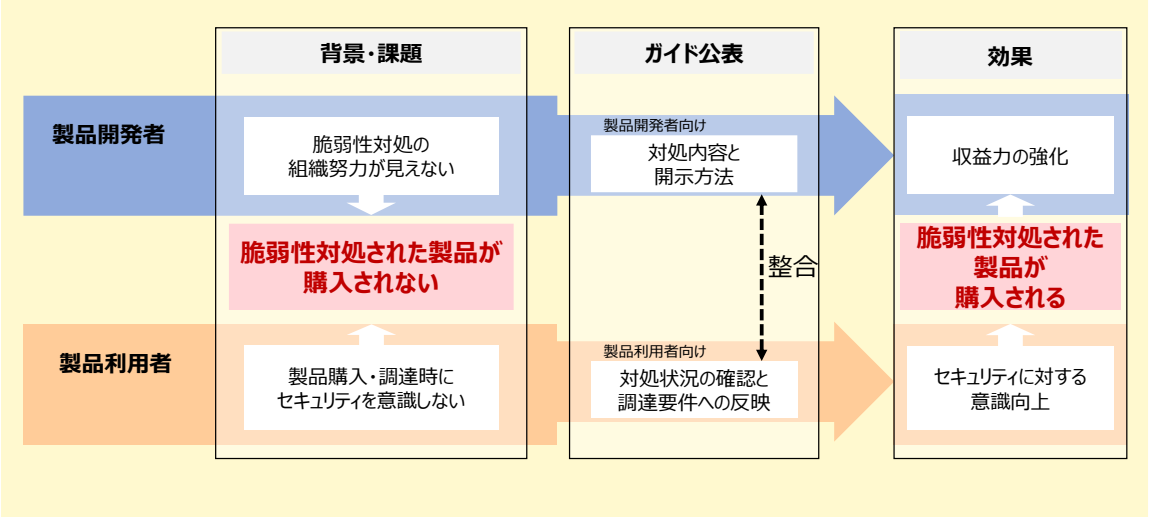
本ガイドの概要

本ガイドについて

本ガイドでは、**製品開発者において実施すべき脆弱性対処と、その開示方法を掲載**しています。**実施すべき対処は段階的に示して**いるため、リソースに限りのある製品開発者においても、可能なところから実施できるように構成しています。

なお、本ガイドと対をなし、製品利用者に対して、製品開発者が開示した脆弱性対処の状況の確認や調達要件への反映を行いつつ、セキュアな製品購入・調達を行い、さらに製品導入後や製品で構成されるシステム構築後も、脆弱性対処を行いつつ、適切に製品が利用されることを目的として、「製品利用者向けガイド」を作成しています。

本ガイドに沿って製品開発者が実施した脆弱性対処を開示することで、製品利用者は容易に確認することができ、セキュアな製品購入・調達を行うことが可能となります。開示によって対策を実施していることを示すことは、組織を守ることにもつながることに加え、製品利用者に対して、製品開発者によって開示された情報の確認を促すことで、以下に示したような効果が得られると期待できます。



図：ガイド活用により期待される効果

対象製品

本ガイドの対象製品は、主に不特定または多数の製品利用者が利用するようなインターネット等のネットワークに接続する以下のような製品を想定しています。

- 単体で販売されるソフトウェア
- 単体で販売されるパッケージソフトウェア
- 機器に組み込まれるソフトウェア、ソフトウェアを組み込んだ機器　　等

なお、特定の製品利用者向けに、受託開発されたソフトウェアについては対象外とします。

また、製品の中には、法律によって、セキュリティやセーフティに関する技術基準を満たすことが、製品開発者に課される義務となっているものも存在します。そのような法的義務について、本ガイドでは、直接的に取り扱っておりません。本ガイドは、セキュリティに関するものを含め、「対象製品」が適合すべき法的義務・要件を網羅的に解説するものではありませんので、ご留意ください。

想定読者

本ガイドの読者は、対象製品に関する製品開発者を想定しています。
本ガイドにおいて「製品開発者」とは、不特定または多数の顧客に提供するソフトウェアの開発を行っている事業者で、かつ、以下のいずれかの条件を満たす事業者を指します。

- 自組織において、上記で開発したソフトウェア自体を製品として提供している事業者（いわゆるソフトウェアの場合）
- 自組織において、上記で開発したソフトウェアを組み込んだ製品を提供している事業者（いわゆるハードウェアの場合）

ただし、特定の顧客に提供するソフトウェアの開発を行っている事業者は、製品開発者に含まれないものとします。

なお、IPA と一般社団法人 JPCERT コーディネーションセンターが運営する脆弱性届出制度「情報セキュリティ早期警戒パートナーシップ」の運用指針である「情報セキュリティ早期警戒パートナーシップガイドライン」では、「Ⅱ. 用語の定義と前提」において、製品開発者を以下のとおり定義しています。

> 製品開発者とは、次のいずれかに該当する者のことです。
> 1）ソフトウエア製品（OSS を含む）を開発した官庁、法人、個人、またはコミュニティ
> 2）ソフトウエア製品（OSS を含む）の加工、輸入、販売または頒布する官庁、法人、個人、またはコミュニティ

本ガイドでは、ソフトウェア製品を「開発した」者は対象としていますが、「加工、輸入、販売または頒布する」ことのみを実施する者については、「製品開発者」に含めないものとして整理しています。上記の「情報セキュリティ早期警戒パートナーシップガイドライン」の用語定義とは異なる用法ですので、ご留意くださいますようお願いいたします。

また、情報システムの開発・構築・運用を担うシステムインテグレータ事業者（SI 事業者）や、従来ソフトウェア製品が提供している機能をクラウドサービスとして提供する SaaS サービス事業者についても、直接的には「製品開発者」に該当しないものとして整理しています。

上述のように、特定の顧客に向けて専用にソフトウェアや情報システムを構築・カスタマイズするような業務をする SI 事業者は、「不特定または多数の顧客」への提供に当たらないため、該当しないものと整理しています。

また、SaaS サービス事業者であっても、例えば、SaaS サービスを利用するためのスマホアプリを提供しているような場合であれば、当該スマホアプリが関係する範囲においては、製品開発者に該当します。

本ガイドの構成

「Ⅰ.計画」、「Ⅱ.要件定義・設計・開発」、「Ⅲ.供給・配布」、「Ⅳ.運用」、「Ⅴ.廃棄」という 5 つの大項目に分類し、製品開発者が実施すべき望ましい脆弱性対処の 15 項目を示しました。

各項目の内容は以下のように「意義」「実施内容」「開示方法」の構成となっています。

■ 意義

対策を実施する必要性やメリット、実施しない場合に想定される影響等を記載しています。

■ 実施内容

具体的な実施内容について記述しています。項目によっては実施内容を最大 3 段階にレベル分けしているため、組織の状況に合わせて可能なレベルから実施し、徐々にレベルを高めることができます。「実施することが理想的な事項」をレベル 3 としており、その実施が困難な場合はレベル 2 を、それも難しい場合はレベル 1 の記載事項を参照してください。

レベル 1　：最低限実施すべき事項
レベル 2　：実施することが望ましい事項
レベル 3　：実施することが理想的な事項

レベル分けしていない事項は、「実施することが望ましい事項」、あるいは実施対象となる製品の性質や機能に応じて内容を取捨選択する必要があるものです。自組織の製品の機能や性質、自組織のリソースを考慮し、実施内容を選択してください。

■ 開示方法

組織の脆弱性対処の取組み状況を製品利用者に理解してもらうために、開示する情報とその例について説明しています。

製品利用者が製品を選ぶ際に必要な情報は、購入前も確認できるようパッケージ及びウェブサイト等の容易に確認できる場所に記載します。

記載内容によっては、インシデント等が発生した場合に責任を問われる等のリスクもあるため、開示内容や記載内容については、広報や法務など組織内の関連部門とも相談してください。全ての内容を開示することが必須ではありません。

活用方法

本ガイドの活用方法としては、例えば以下のように活用方法があります。

- 製品開発者がセキュリティ対策として実施すべき項目を把握できる
- 実施する対処を徐々にレベルアップできる
- 製品利用者に自組織の取組み状況をアピールするため、すべきことを把握できる

また、本ガイドの最後に対処状況を確認するためのチェックリストを付けています。チェックリストは、例えば以下のように活用できます。

- 組織の製品開発プロジェクトマネージャーが現状と課題を把握し、上長に報告
- ソフトウェアコンポーネントのサプライヤー等に発注する際、発注するソフトウェアコンポーネントのセキュリティレベルを確認・把握するために活用
- 委託先の脆弱性対処の状況確認に活用

脆弱性管理の全体像とその基本的な考え方

本ガイドの内容を理解する上で前提となる、製品開発者が提供するソフトウェア製品に対して行う脆弱性管理に関する取組の全体像とその基本的な考え方を示します。

ソフトウェア開発ライフサイクルに応じた脆弱性管理

発見された脆弱性への対処は、製品開発者が出荷・リリースしたソフトウェア製品に対して行われるものですが、このような脆弱性については、設計・開発段階での脆弱性の作り込みの回避や、運用段階での脆弱性の発見、検証、修正対策による対処を含め、ソフトウェア開発ライフサイクル全体にわたって管理することが求められています。

このため、本ガイドでは、計画段階から廃棄段階に至るまでのソフトウェア開発ライフサイクル全体における脆弱性管理を採り上げ、脆弱性管理の観点からみて各段階で必要となる実施事項やその具体的な内容・手順を取りまとめています。

各段階で必要となる実施事項は以下の通りです。また、括弧書きで、実施事項に対応するガイドの章番号を記載しています。

段階	脆弱性管理の観点からみた必要となる実施事項
Ⅰ．計画	ソフトウェア開発ライフサイクル全体を通じた人材・プロセス・技術の整備（1 章）
	セキュアなソフトウェア開発を実践するための方針・体制（2 章）
	脆弱性対処やインシデント対応を円滑に実施するための方針・体制（3 章）
Ⅱ．要件定義・設計・開発	セキュリティを確保するための設計（4 章）
	既知の脆弱性解消（5 章）
	セキュアコーディング・セキュアビルド（6 章）
	開発時の脆弱性検査（7 章）
	開発に使用する環境及びツールのセキュリティ確保（8 章）
Ⅲ．供給・配布	ソフトウェア製品のセキュアな配布（9 章）
Ⅳ．運用	脆弱性の発見（10 章）
	脆弱性の検証（脆弱性情報のトリアージと分析）（11 章）
	脆弱性の修正対策と対策情報の公表（12 章）
	ソフトウェアの脆弱性に起因したインシデントが発生した場合の対応（13 章）
	製品利用者に対する実施事項の明示（14 章）
Ⅴ．廃棄	ソフトウェアのサポート終了と廃棄対応（15 章）

脆弱性管理の基本的な考え方

ソフトウェア製品やソフトウェアを組み込んだハードウェア製品の脆弱性管理については、これまでも様々な取組・対策が進展してきました。とりわけ、このような製品について、外部から脆弱性報告を受領した場合や未修正の脆弱性が開示され悪用された場合などの緊急時

の対応を主な役割として、製品開発者において PSIRT（Product Security Incident Response Team）という体制が構築されてきました。

その一方で、近年においては、前述したように、ソフトウェア開発ライフサイクル全体にわたって脆弱施管理を行うことが求められるようになり、特に計画段階や要件定義・設計・開発段階といった上流工程でセキュリティ対策を組み込むという「セキュア・バイ・デザイン」や「シフトレフト」の考え方の重要性が、製品開発者においても再認識されています。

本ガイドでは、このような状況を踏まえ、脆弱性管理の観点から PSIRT が果たすべき役割として重点が置かれていた緊急時の対応に加えて、脆弱性の作り込みを回避するために、セキュアなソフトウェア開発を実践する場合や設計・開発するソフトウェアにセキュリティ機能を組み込む場合などの平常時の対応についても明確に位置付けています。

本ガイドを活用した、必要となる取組の選定やその段階的実施を通じて、組織の脆弱性管理の観点からの取組の充実・強化を図り、製品セキュリティの更なる向上に寄与することを期待します。

Ⅰ. 計画

1 ソフトウェア開発ライフサイクル全体を通じた人材・プロセス・技術の整備

意義

ソフトウェア開発ライフサイクル（以下「SDLC」という。）の全体を通じて脆弱性対処を組み入れ、実践している組織は、製品開発の初期段階で製品の脆弱性を把握することができ、脆弱性の修正に費やすコストを低く抑えられます。

組織レベルで SDLC 全体を通じた脆弱性対処ができるよう、人材、プロセス、技術を整備することが必要です。

実施内容

SDLC に関与する組織内外の関係者が、SDLC 全体を通じて、セキュリティを高めるために必要な役割と責務を定義し、それらを実践できるよう、人材、プロセス、技術を整備します。

またその中で、組織外の関係者に求めるべき役割と責務や、実践のための必要なプロセスは、外部委託の活用や組織間の連携強化を通じて、その実践の確実性、継続性を担保します。

実施手順
① **人材・プロセス・技術の整備** SDLC 全体が網羅されるよう、組織内外の関係者を洗い出すとともに、それぞれの関係者に求められる役割と責務を明確化し定義します。組織レベルでの SDLC の実践に対する重要性を、経営者が正しく理解することが重要です。 組織内外の関係者に求めるべき役割と責務を実践していくために必要なプロセスを整理し、それぞれの関係者が確実に実践できるように手順書を策定します。 このような必要なプロセスの整理としては、セキュアなソフトウェア開発の実践や、脆弱性対処やインシデント対応の実施に関わるプロセスの整理が特に重要になります。また、ソフトウェアのサポート終了を意味するエンドオブサービス（EOS（End of Service））や、ソフトウェアのライフサイクル終了を意味するエンドオブライフ（EOL（End of Life））となったソフトウェアの利用の終了に関わるプロセスについても、脆弱性が放置されたソフトウェアが利用されるという事態を回避するうえで重要になります。 定義された役割と責務や、実践のための必要なプロセスについては、適切に維持されていることを定期的に確認し、必要に応じて変更します。また、各プロセスにおいては、人手による対応を減らし、自動化することを目的に、使用可能なツール・ツールチェーンを確保します。とりわけ、ソフトウェアの脆弱性管理については、近年、ソフトウェアコンポーネントやそれらの依存関係の情報を、機械処理可能な形式の一覧リストとして提供するソフトウェア部品表（SBOM（Software Bill of Materials））を用いた管理手法が利用さ

れはじめており、コストや体制整備を含め、そのような SBOM の導入や活用の方針を検討することが望まれます。

さらに、ソフトウェア開発においても、プログラムのコーディングや、テストケース・テストコードの生成、コードの品質レビュー等の自動化を実現する生成 AI の活用が進んできています。今後は、脆弱性検査やセキュリティ修正プログラムの作成等の脆弱性管理の自動化への適用も期待されています。一方で、生成されたコードに対する予期せぬ脆弱性の混入や保守性の考慮不足をはじめとして、生成されたコードを精査する担当者の能力不足や、社内の利用制限や利用ルールの未整備など、品質面・教育面・ガバナンス面の課題も見られます。

定義された役割と責務や、策定された手順書は、組織内外の関係者に周知を行います。周知は、これらの内容が更新された場合にも実施します。また、SDLC のなかで脆弱性対処を担う役割と責任を持つ組織内のすべての担当者に対して、定期的な教育・訓練を実施します。

② **外部委託の活用**

組織外の関係者に求めるべき役割と責務や、実践のための必要なプロセスについては、外部委託または組織間の連携強化を活用して、確実かつ継続的に実施されるようにしてください。

また、外部委託先に対して、本ガイドの「2 セキュアなソフトウェア開発を実践するための方針・体制」において後述する製品セキュリティポリシーを開示し、当該ポリシーに沿った活動が行われるよう促します。

ソフトウェアの脆弱性検査や、ソフトウェアの脆弱性に起因したインシデントが発生した場合の対応、ソフトウェアの廃棄等のセキュリティ対処を外部委託する際には、委託先の選定基準を策定し、適切な委託先を選定します。

上記の実施内容はレベル 3 をベースにしています。これらの項目の対応が難しい場合は、以下に記載している他のレベルの実施内容を参照し対応を検討してください。

レベル	実施内容
1	組織内の関係者が、SDLC におけるセキュリティを高めるために必要な役割と責務を定義します。
2	SDLC 全体が網羅されるよう、組織内の関係者が、SDLC 全体を通じて、セキュリティを高めるために必要な役割と責務を定義し、それらを実践できるよう、人材、プロセス、技術を整備します。
3	外部委託の活用や組織間の連携強化を図りつつ、SDLC 全体が網羅されるよう、組織内外の関係者が、SDLC 全体を通じて、セキュリティを高めるために必要な役割と責務を定義し、それらを実践できるよう、人材、プロセス、技術を整備します。

開示方法

- 製品利用者や委託先・調達先に対してウェブサイトのポリシーページで開示します。
  - 実際の開示事例での開示場所は、CSR のように組織の社会的責任の説明項目の一環として開示する場合や、品質・調達等の方針と共に開示する場合等、様々な場所があります。また、開示する文書にはポリシーだけではなく、ガイドラインや指針など様々な形態があり得ます。
- 各項目の開示内容は抽象的でも、より具体的でも構いません。具体的に記載する場合は、ホワイトペーパー等に記載する方法もあります。

2 セキュアなソフトウェア開発を実践するための方針・体制

意義

セキュアなソフトウェア開発に対する組織の取組姿勢を対外的に示すことにより、組織内でのセキュリティ確保に関する意識を高めるとともに、社会的な評価を高めることが重要です。

開発するソフトウェアまたはソフトウェアの設計・開発プロセスに対して、セキュリティ要件を定義することにより、SDLC 全体にわたってセキュリティ要件を考慮・維持するというガバナンスを効かせることが必要です。

実施内容

法令の遵守や、優れた SDLC の実践、サプライチェーンの強化などの社会的な要求事項に対応して、組織がセキュアなソフトウェア開発を推進していく際の取組姿勢やセキュリティ対処の基本方針・ルールを、製品セキュリティポリシーとして策定します。

製品セキュリティポリシーに沿ってセキュリティ確保に向けた統一的な対策を組織内で実施できるようにするために、セキュリティ要件を定義します。

開発するソフトウェアがセキュリティ要件を満たしていることを監査により確認できるようにするため、セキュリティ確認基準の策定や、SDLC 全体にわたってセキュリティ確認基準への準拠を維持するというガバナンスの確立が求められます。

実施手順
① **開発・テストチームの構築・維持** ソフトウェア開発における開発上の不備・設定ミス等による脆弱性の作り込みの問題に対処できるようにするため、開発責任者とレビュー責任者、テスト責任者を配置します。各責任者は、開発やレビュー、テストに携わる担当者が、最新の技術やツールについて把握できるとともに、実践的なトレーニングを定期的に受けることができるように対応します。 ② **計画・ポリシー・手順の策定** 経営者の意向や、法令の遵守、優れた SDLC の実践、サプライチェーンの強化などの社会的な要求事項を踏まえて、組織がセキュアなソフトウェア開発を推進していく際の取組姿勢やセキュリティ対処の基本方針・ルール等を盛り込んだ製品セキュリティポリシーを策定し、法務部門や広報部門等、対外的な対応を行う部門の確認を受けた後、外部に開示します。 下記(1)～(4)の手順に従い、製品セキュリティポリシーを策定します。 (1) 組織として製品セキュリティポリシー策定の意思決定 組織として製品セキュリティポリシーを策定するにあたっては、まず経営層も含めた意思決定、製品セキュリティポリシー策定の体制作り、予算確保等を行います。製品セ

キュリティポリシーは組織としての製品セキュリティの方針・考え方を表明するものであるため、策定にあたっては経営層の意向を確認します。

(2) 製品セキュリティポリシー案の作成

経営層の意向や法令、製品セキュリティに関するガイドや基準等を踏まえて、組織面、技術面でのセキュリティ対策等を盛り込んだポリシー案を作成します。また、製品セキュリティとして製品の利用環境等で考慮すべき点がある場合は、利用者に向けて、どのような対応を組織として求めるかについても、ポリシーに含めて作成します。

なお、製品セキュリティポリシーは組織外に開示するため、法務部門や広報部門等、対外的な対応を行う部門の確認を得ることも重要です。

(3) 経営層による承認

経営層に作成した製品セキュリティポリシーの承認を得ます。

(4) 組織内への周知、組織外への開示

承認された製品セキュリティポリシーに沿った統一的な対策が組織内で実施されるように、具体的な製品セキュリティの実施事項も作成します。

その後、組織内のセキュリティに関する意識を高めるとともに、策定した製品セキュリティポリシーに沿った活動が行われるように、教育等により組織内への周知を図ります。また、組織外に策定した製品セキュリティポリシーを開示します。開示については、「開示方法」を参照してください。

このような製品セキュリティポリシーは、セキュアなソフトウェア開発に対する社会的な要求事項に応えるための対応方針を含めて策定します。対応方針の策定を行う上で考慮すべき社会的な要求事項の例としては、以下が挙げられます。

- 「セキュア・バイ・デザイン」の原則の実践
- 「セキュア・バイ・デフォルト」の原則の実践
- サプライチェーン強化に向けたセキュリティ対策

「セキュア・バイ・デザイン」及び「セキュア・バイ・デフォルト」の原則の実践において、組織は、セキュア・バイ・デザイン、セキュア・バイ・デフォルトの双方の観点から実践が求められるセキュアなソフトウェア開発の慣行をより多く取り込むためのロードマップを策定することも一案です。

また、サプライチェーン強化に向けたセキュリティ対策において、組織は、サプライチェーン全体において開発者に求められるセキュリティが担保されていることや、リスクが対応許容範囲を超えていないことを確認し、適切なリスク管理を行うためのサプライチェーンリスク管理計画を策定します。ソフトウェア開発の業務の一部を他社に委託しているような企業の場合、委託にあたって、委託先の信用状況等を確認するプロセスが既に存在している場合があります。そのような既存プロセスとの統合を図ることも方法の一つです。

③ ソフトウェア開発におけるセキュリティ要件の定義

開発するソフトウェアや、組織のソフトウェア開発に用いるインフラ及びプロセスにおいて満たすべきセキュリティ要件を定義し、文書化します。このようなセキュリティ要件には、組織内から対応が求められる要件（組織のポリシー、ビジネス目標、リスク管理戦略等）と、組織外から対応が求められる要件（適用される法規制等）の双方が含まれます。

このうち、組織内から対応が求められる要件については、組織のポリシー、ビジネス目標、リスク管理戦略等に照らして、どこまでの対応内容・範囲を、製品セキュリティにおける実施すべき事項として考慮するかを判断することが求められます。セキュリティ要件適合評価及びラベリング制度（JC-STAR）の適合基準なども参考としながら、必要となる基準を検討しつつ、セキュリティ要件を定義します。

他方、組織外から対応が求められる要件については、世界の各国・地域においてソフトウェア製品を提供している場合に、それぞれの国・地域で適用される法規制への対応が求められます。例えば、欧州連合には、サイバーレジリンス法（Cyber Resilience Act）のような製品セキュリティに関わる法規が存在します。

また、電気通信回線設備に接続して使用される端末設備や、無線機器、医療機器などのように、製品分野によっても、適用される法規が異なることがあるため、製品ごとに適用される法規制への対応も求められます。特に、IoT 機器のように、機器に組み込まれるソフトウェアについては、身体等への物理的な危害が生じる可能性もあるため、セーフティに関する法規についても確認することが重要です。

このような様々な法規が対象となる可能性があるため、特定の部門のみで対応することが難しい場合には、開発部門や品質管理部門、PSIRT などの関連部門が適切な役割分担のもとで連携して、対応が求められる法規を確認しつつ、セキュリティ要件を定義します。

また、ソフトウェア開発において、サードパーティのソフトウェアコンポーネントを利用する場合は、定義されたセキュリティ要件を満たす安全性の高いものを調達し、選定します。その際に重要となるのが、当該コンポーネントに脆弱性が発見された場合の修正対応です。コミュニティが活発で定期的なアップデート対応や脆弱性の修正対応が期待できるオープンソースソフトウェア（OSS）の選定や、サードパーティとの契約や取り決め等において修正対応に対する要求が盛り込まれるよう、依頼・調整を行うことをセキュリティ要件として定義します。このような要求については、修正対応のほかにも、外部から脆弱性情報を受け付ける窓口の設置や脆弱性発見時の報告が盛り込まれるようにします。

さらに、定めたセキュリティ要件について、「⑤監査・見直し」において準拠状況を確認できるよう、証跡情報を保存することや、開発で用いるソフトウェアツール・ツールチェーンにおいて証跡情報を記録・出力するよう設定を変更することについても、セキュリティ要件として規定することが重要です。

④ ソフトウェアセキュリティをチェックするための基準の策定

開発するソフトウェアがセキュリティ要件を満たしていることを確認できるよう、セキュリティ確認基準を定義し、当該基準への準拠状況を SDLC 全体にわたって追跡します。

⑤ 監査・見直し

組織は、セキュリティ確認基準への準拠状況を確認する際に、確認の裏付けとなる情報（以下「証跡情報」という。）を収集します。収集した証跡情報の保管にあたっては、ア

クセス権限を明確にするとともに、不正な情報の変更または削除を防止するための措置を講じます。

証跡情報を利用した監査により、SDLC 全体にわたってセキュリティ確認基準への準拠を維持するというガバナンスが、組織が意図する効果を得ていることを確認します。

なお、開発するソフトウェアにおいて、より高いレベルの脆弱性対策が求められる場合には、第三者が評価する第三者認証の取得を通じて、脆弱性検査の実施を含め、要件定義時に定めたセキュリティ要件が満たされていることを確認するという方法も考えられます。

製品セキュリティポリシーは策定するだけでなく、決められたことが適切に実行されているかを確認し、実行されない場合は是正していくことが重要です。また、製品セキュリティポリシーに関係する法令やガイド等の改定、製品セキュリティへの社会的な要求事項の変化等があった場合、製品セキュリティポリシーを見直すことも必要です。

上記の実施内容はレベル 3 をベースにしています。これらの項目の対応が難しい場合は、以下に記載している他のレベルの実施内容を参照し対応を検討してください。

レベル	実施内容
1	製品セキュリティポリシーを策定し、当該ポリシーに沿ってセキュリティ要件を定義します。
2	製品セキュリティポリシーを策定し、当該ポリシーに沿ってセキュリティ要件を定義するとともに、開発するソフトウェアがセキュリティ要件を満たしていることを確認します。
3	製品セキュリティポリシーを策定し、当該ポリシーに沿ってセキュリティ要件を定義するとともに、セキュリティ確認基準の策定や、SDLC 全体にわたってセキュリティ確認基準への準拠を維持するというガバナンスの確立を通じて、開発するソフトウェアと組織的・技術的体制がセキュリティ要件を満たしていることを監査により確認します。

開示方法

- 製品セキュリティポリシーを定め、製品利用者や委託先・調達先に対してウェブサイトのポリシーページで開示します。
  - 実際の開示事例での開示場所は、CSR のように組織の社会的責任の説明項目の一環として開示する場合や、品質・調達等の方針と共に開示する場合等、様々な場所があります。また、開示する文書にはポリシーだけではなく、ガイドラインや指針など様々な形態があり得ます。
- 各項目の開示内容は抽象的でも、より具体的でも構いません。具体的に記載する場合は、ホワイトペーパー等に記載する方法もあります。

具体的な製品セキュリティポリシーの例を以下に示します。示した開示例は「ひな型」ではないので、組織の実施状況等を踏まえた内容を記載ください。

【開示例（レベル 3）】

製品セキュリティポリシー

当社は、セキュアコーディング・セキュアビルドの実践や、開発に使用する環境のセキュリティ確保、ソフトウェア製品のセキュアな配布、ソフトウェアの脆弱性に起因したインシデントが発生した場合の対応を含む、ソフトウェア開発ライフサイクル全体にわたって脆弱性の作り込みの回避や製品セキュリティレベル維持及び改善のための活動に取り組み、お客様に安全性の高い製品を提供します。

(1)組織的対策

当社では、全社的な方針の下、セキュアコーディング・セキュアビルドの実践を通じて、製品セキュリティを確保する体制を整備し、セキュリティ対策を実施します。また、国内外のガイド等に基づいた製品セキュリティ対策基準を策定し、これに基づいた製品のセキュリティ設計・開発を行います。さらに、出荷・リリースしたソフトウェアの脆弱性に起因したインシデントが発生した場合には、迅速に体制を整え、外部からの問い合わせを受け付ける際の対応窓口の設置や、脆弱性の報告・公表、原因の究明と再発防止を実施します。

(2)技術的対策

当社では、ソフトウェア製品の開発に使用する環境や配布に使用するシステムのセキュリティ確保や、製品の出荷・リリース前における脆弱性検査を実施し、製品に脆弱性が含まれないように努めます。また、出荷・リリース後も、自社製品の脆弱性に関する情報を収集し、発見された脆弱性が、お客様への被害や製品性能に影響を及ぼす可能性があると判断した場合には、アップデートや対策ソフトウェアの提供等、当社が必要と判断した対応策等、適切な情報を提供します。

(3)情報の提供

セキュリティレベルの維持は、適切なセキュリティ対策を行った当社製品をお客様が適切に利用することで実現できます。当社は、セキュリティに関する注意喚起やセキュリティを確保した上で製品を利用するための情報等を提供します。

【開示例（レベル 2 の場合の例）】

製品セキュリティに関する方針

当社では、以下の方針の下、製品セキュリティの確保に取り組みます。

1. 製品セキュリティを確保するための体制を整備します。
2. セキュリティを考慮した設計・開発を行い、製品の出荷・リリース前は、脆弱性検査により脆弱性の解消に努めます。
3. 製品の出荷・リリース後も脆弱性情報を広く収集し、リスクがあると判断した場合は迅速に対応を行います。
4. セキュリティに関する情報や対策方法を利用者の皆様に提供します。

3 脆弱性対処やインシデント対応を円滑に実施するための方針・体制

意義

製品セキュリティを維持するために、組織内の様々な部門間だけでなくサプライチェーンや脆弱性情報の提供組織等の外部組織も含め、連携した体制を構築し、製品セキュリティを管理する必要があります。製品セキュリティを維持する体制が整備できない場合は、製品セキュリティポリシーで定めた守るべき事項や対処が適切に実施できない、必要な情報が外部組織から入手できない等、製品セキュリティを維持することが困難になります。製品セキュリティを維持する体制を構築し、設計・開発段階から製品の出荷・リリース後までセキュリティを管理することで、セキュリティが確保された製品を出荷・リリースすることが可能となり、製品の出荷・リリース後に発見された脆弱性やインシデントに迅速に対応することができます。

実施内容

製品セキュリティポリシーに基づき組織として実施すると定めたセキュリティ対処について、必要な体制を整備します。体制には、自組織内の部門はもちろん、外部組織も含みます。また、「方針決定」「設計・開発」「出荷・リリース後の対応」に必要な『平時の体制』だけでなく、外部から脆弱性報告の受領時や未修正の脆弱性が開示もしくは悪用された場合などの『緊急時の体制』についても、製品セキュリティポリシーを策定する際に整備しておく必要があります。

また、近年においては、要件定義・設計・開発段階でセキュリティ対策を組み込むという「セキュア・バイ・デザイン」や「シフトレフト」の考え方の重要性が再認識されています。脆弱性の作り込みを回避するために、セキュアなソフトウェア開発を実践し、設計・開発するソフトウェアにセキュリティ機能を組み込むことができるようにするため、平時の体制の更なる充実・強化を図る必要があります。

なお、製品に関する緊急時の体制のことを PSIRT（Product Security Incident Response Team）[1]と言いますが、PSIRT が果たすべき役割は、平時の体制の更なる充実・強化を含め、大きくなっています。本ガイドでは、「PSIRT」は緊急時の体制、平時の体制を含めた製品セキュリティ全般の体制を指すものとします。ただし、組織の内部体制や部門間の連携状況によっては、そのような平時の対応については PSIRT ではなく、別のチームの役割とすることも考えられます。連携が必要となる関係者とその役割については、「附属 主要な関係者・役割表」を参照してください。

実施手順
① **PSIRT の構築・維持** 製品セキュリティポリシーに従ってセキュリティ対処を実施する PSIRT を構築するために必要な関係者（自組織の関係部門及び外部関係者）を洗い出します。自組織で実施が

[1] PSIRT Services Framework 1.0 日本語版（一般社団法人コンピュータソフトウェア協会、JPCERT/CC）
https://www.first.org/standards/frameworks/psirts/FIRST_PSIRT_Services_Framework_v1.0_jp.pdf

難しいセキュリティ対処項目は、委託することも検討してください 。自組織の関係部門は下記の図を参照してください。

PSIRTの組織体制については、下記の図のようなPSIRT組織を設置するとともに、PSIRT組織と連携して必要となるセキュリティ対処の取組を推進する製品セキュリティ担当者を、自組織の関係部門に人員配置し、関係部門内で推進体制を構築することが求められます。このような製品セキュリティ担当者の人員配置においては、専任者または兼任者の確保やセキュリティ対処に必要な教育方法等を含め、関係部門の理解を得ることが必要になるため、経営層の協力を確保することが重要です。

PSIRT のスタッフとなる自組織の関連部門及び PSIRT と連携を図る外部組織を特定し、それぞれの役割を明確化します。自組織の部門が確実にその役割を実施できるように手順書を策定します。PSIRT と連携を図る外部組織としては、ソフトウェアやソフトウェアコンポーネント、セキュリティ機能に関わる委託先・調達先や、脆弱性報告に関わる IPA や JPCERT/CC 等のセキュリティ関係機関などについて考慮する必要があります。

一方で、多種多様な分野のソフトウェア製品を扱っている製品開発者では、ソフトウェア製品の特性の違いにより顧客が求めるセキュリティレベルが異なることから、開発部門や品質管理部門を中心とした自組織の関連部門が実施すべき手順は必ずしも一様ではありません。このため、PSIRT が自組織の関連部門向けに策定する手順書では、実施すべき手順の大まかな内容について策定するものの、その手順を実現するための具体的な手法（例えば、脆弱性情報の収集方法や、脅威・リスク分析手法、脆弱性診断手法、セキュア・バイ・デザインの実践手法など）の選定や導入、実践等については、自組織の関連部門が自部門の状況を踏まえたうえで判断して進められるような責任の委譲構造とすることも一案です。その上で、PSIRT においては、そのような自組織の関連部門に対して、必要となる情報の提供や研修、実践した結果のレビューを行い、具体的な手法の選定や導入、実践等をサポートする役割が重要になります。

外部の発見者からの脆弱性報告に対処できるようにするため、PSIRT のスタッフに対して、新たな脆弱性の動向やバグトラッキングツールの使い方のようなセキュリティ分野の幅広い知識や、組織内外の関係者とのコミュニケーションのための能力を身に付けるための実践的なトレーニングを定期的に受けることができるように教育プログラムの機会を用意します。

② PSIRT が提供する機能の検討

PSIRT の組織において、製品セキュリティポリシーに従ってセキュリティ対処を実施するために配置すべき必要な機能を検討します。必要な機能を検討する上で考慮すべき機能は以下の通りです。

- 製品セキュリティポリシーの作成
- セキュア・バイ・デザイン及びセキュア・バイ・デフォルトの実践
- ソフトウェアのサプライチェーン全体でのセキュリティ対策
- 開発時の脆弱性検査
- 脆弱性の発見
- 脆弱性情報のトリアージと分析
- 脆弱性の修正対策

- 脆弱性の開示
- インシデント対応
- 実践的なトレーニング

③ 計画・ポリシー・手順の策定

自組織内でセキュアなソフトウェア開発を実践し、提供するソフトウェアに脆弱性を作り込まないための方針を定め、必要となる体制やプロセスの整備を含む製品セキュリティポリシーやサプライチェーンリスク管理計画を策定します。製品セキュリティポリシーやサプライチェーンリスク管理計画の策定を行う上で考慮すべきプロセスは以下の通りです。

- 製品セキュリティポリシーを作成し、経営層による承認を受けるプロセス
- セキュア・バイ・デザイン及びセキュア・バイ・デフォルトを実践するプロセス
- サプライチェーン全体で求められるセキュリティが担保されているかを確認し、リスク管理を行うプロセス
- 開発時における脆弱性検査により脆弱性を発見し、適切な修正を行うプロセス

また、リリースしたソフトウェアに残存する脆弱性を特定し、速やかに対処するための方針を定め、必要となる体制やプロセスの整備を含む脆弱性対処計画を策定します。脆弱性対処計画の策定を行う上で考慮すべきプロセスは以下の通りです。

- 脆弱性情報を収集するプロセス
- 外部の発見者からの脆弱性報告を受け付けるプロセス
- 内部テストにより新しい脆弱性を発見するプロセス
- 脆弱性の開示に関する通知プロセス
- 脆弱性への対処が必要であるかを判断するプロセス
- セキュリティ修正プログラムの作成、テスト、展開を行うプロセス
- 発見された脆弱性の問題の根本原因を特定するプロセス
- インシデントを管理するプロセス

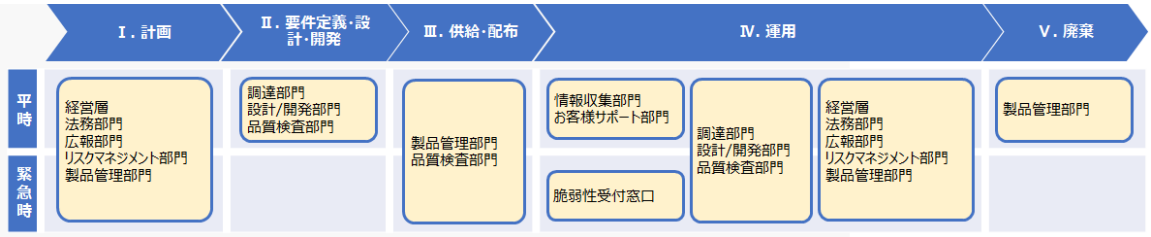
また、顧客等に対して、脆弱性対処の取組を示すとともに、外部の発見者に対しても、脆弱性報告への協力を得ることが重要であるため、外部向けの文書として脆弱性開示ポリシーを策定します。

さらに、脆弱性に起因して製品利用者においてインシデント（サイバー攻撃）が発生した場合には、事案への対処を迅速かつ円滑に行うことが重要であるため、対応業務のリストアップと手順化、関係者間の役割分担に関する取り決め等を盛り込んだインシデント対応計画を策定します。

④ ステークホルダー間の関係強化

リリースしたソフトウェアのセキュリティを改善するために、PSIRT と外部組織との間の情報連携・協力体制を強化します。外部組織には、JPCERT コーディネーションセンターや IPA のようなセキュリティ関係の組織・機関だけでなく、開発業務の委託先や、ソフトウェアに組み込んだコンポーネントの調達先が含まれます。

体制整備には、様々な部門の理解と協力及びこの体制を確保するための予算が必要です。このため、経営層による強いリーダーシップ及び支援が必要不可欠です。



図：自組織の関連部門

外部関係者も含め、連携が必要となる関係者と役割については、「附属　主要な関係者・役割表」を参照してください。

上記の実施内容はレベル 3 をベースにしています。これらの項目の対応が難しい場合は、以下に記載している他のレベルの実施内容を参照し対応を検討してください。

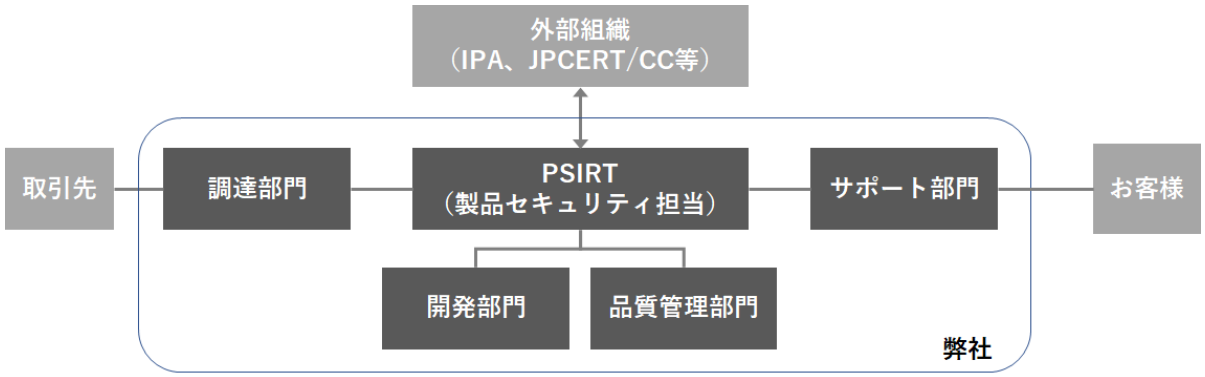
レベル	実施内容
1	PSIRT を構築・維持し、PSIRT が製品セキュリティポリシーに従ってセキュリティ対処を実施するために必要な機能を定義します。
2	製品セキュリティポリシーを策定し、当該ポリシーに沿ってセキュリティ要件を定義するとともに、PSIRT の活動上必要となる体制やプロセスを整備します。
3	PSIRT を構築・維持し、PSIRT が製品セキュリティポリシーに従ってセキュリティ対処を実施するために必要な機能を定義するとともに、脆弱性対処計画の策定やステークホルダー間の関係強化を通じて、PSIRT の活動上実効性のある体制やプロセスを整備します。

開示方法

- 製品セキュリティを維持するための体制を構築していることを製品セキュリティポリシーに含め開示します。
- 脆弱性情報の受付など、積極的に脆弱性対処を実施していることを関係者にアピールするため、緊急時の体制(PSIRT)に関する説明や体制図等を自組織のウェブサイト等で開示する場合もあります。

【開示例（体制）】

製品セキュリティ体制



II. 要件定義・設計・開発

4 セキュリティを確保するための設計

意義

ソフトウェアの設計・開発・運用に対する脅威を軽減するためには、「セキュア・バイ・デザイン（ソフトウェアの設計段階から情報セキュリティを確保するための理念または方策）」及び「セキュア・バイ・デフォルト（ソフトウェアのセキュリティ機能や設定を最初からデフォルトで組み込んだ状態にする理念または方策）」を実践し、ソフトウェアの利用用途等を想定したリスク分析の実施や、セキュリティ設計、セキュリティ機能の実装、セキュリティテスト・脆弱性検査の実施など、必要となるセキュリティ対策を実装する取組が必要になります。このうち、セキュリティ機能の実装においては、製品自体にセキュリティ機能を搭載するために、設計段階からセキュリティ機能を検討する必要があります。設計段階から製品のセキュリティ機能を考慮しておかないと、後工程で機能追加ができない場合、製品利用時の脅威が残存してしまうことになります。また、追加できたとしてもコストが増え、開発スケジュールが遅延する等の問題が発生することになります。機能搭載の検討にあたっては、製品の利用用途等を想定したリスク分析の結果を踏まえる必要があります。この分析をしないと、必要な機能が漏れる可能性があります。設計段階からセキュリティ機能を搭載することで、製品の開発工程全体のコストを抑える効果があります。

実施内容

リスク分析などを行い、分析結果をもとに必要となるセキュリティ機能を搭載します。また、「セキュア・バイ・デザイン」及び「セキュア・バイ・デフォルト」の原則を実践し、セキュリティ機能の搭載以外にも、ソフトウェアに対する脅威を軽減するために必要となるセキュリティ対策を実施します。連携が必要となる関係者とその役割については、「附属 主要な関係者・役割表」を参照してください。

実施手順
① 保護すべき機能と情報の特定 自組織の製品において守るべき対象を洗い出します。コンテンツやデータ、制御機能の動作や通信も保護対象です。 **② 脅威の特定** 上記①で保護すべき機能や情報に対するサイバー攻撃による脅威を想定します。以下の表「主な脅威と対応するセキュリティ機能例」を参考にしてください。 またその上で、想定される脅威をもとに、保護すべき機能や情報が脅かされるリスクシナリオや、それらのリスクシナリオを引き起こす攻撃ツリーを検討し、作成します。 **③ リスクアセスメント** 想定した脅威の発生頻度や想定被害から保護すべき機能や情報に対するリスクを算定し、受容するか対策が必要かを評価します。

④ 搭載するセキュリティ機能の検討

上記③の結果、受容できない脅威があればセキュリティ対策技術を検討します。適用可能な技術（製品に搭載する機能）は複数あるため、コストや効果、導入難易度などを考慮して選定します。

製品に搭載する有用な機能の１つは、製品の出荷・リリース後に脆弱性が発見された場合に、製品セキュリティを維持するために、脆弱性に対処したソフトウェアにアップデートできる機能です。出荷・リリース後も製品をアップデートできるよう、アップデート機能を設計し、搭載します。アップデートは、製品への機能の搭載だけでなく、出荷・リリース後にもアップデートファイルを作成・頒布し、運用させるまでの仕組みを維持・運用することも重要です。下記(1)～(2)の検討を踏まえて、アップデート機能を設計します。

(1) 設計するアップデート方法の選定

対象機器におけるアップデートの提供方法を検討します。アップデート方法の種類及び対象例については、表「アップデートの種類」にまとめています。例えば、製品利用者が意識せずに自動的にアップデートされる仕様の場合、常に製品が最新の状態に保たれることが期待できます。一方で、アップデートすることで製品の動作が一時的に中断し、それによって製品の利用に影響が生じる場合には、アップデートのタイミングを製品利用者に選択させる設計を検討することも必要です。

アップデートの提供方法を検討する際には、以下の点を考慮します。

- 製品が提供する機能や想定する利用環境
- 利用者によるアップデートに係る作業負荷（容易性）
- 製品のネットワークへの接続状況・接続頻度
- セキュリティ対応以外を目的としたアップデートの想定頻度
- アップデート機能搭載の費用対効果

また、アップデート方法に応じてメモリなどハードウェアの増強が必要になったり、オンラインでのアップデートであれば、アップデート配信用のサーバの構築及び運用管理も必要になったりします。ハードウェア機器に組み込まれたソフトウェアのアップデートの場合、工場への返送対応や設置場所への施工担当者の派遣などの対応を要することもあるため、アップデート方法については、そのようなコストの発生も考慮して事前に検討することが重要です。

(2) アップデート機能実装により発生する影響への対応

アップデート実施に関する処理の負荷や製品機能の停止時間などの影響を低減させる必要がある場合はその方法を検討します。例えば、アップデートによる帯域不足が生じる場合には、使用する帯域幅を制限した仕様とすれば未使用の帯域を確保できます。また、オンラインでのアップデート機能を採用する場合には、アップデートファイルの改ざんによるマルウェアの混入などを防ぐため、セキュアなアップデート方法を設計することが望まれます。具体的な内容は表「アップデート機能実装により発生する影響への対応」を参照してください。

セキュリティ機能については、製品の利用時だけではなく、利用終了時のことも考える必要があります。利用者がソフトウェアの利用を終了する場合に、当該ソフトウェアに登録・保存されているデータの消去がセキュアかつ確実に行われるようにするために、当該データを消去できる機能を搭載することも考えられます。一般的なデータ削除機能や初期化機能では十分にカバーできないハードディスクや SSD 等に残っているデータから復元されるというリスクに対応できるよう、データ消去機能を設計します。

⑤ 「セキュア・バイ・デザイン」の原則の実践

「セキュア・バイ・デザイン」の原則に則り、ソフトウェアの設計時からリスクベースのアプローチにより、ソフトウェアに対する脅威を軽減するためのセキュリティ対策を実施し、その有効性を判断します。

⑥ 「セキュア・バイ・デフォルト」の原則の実践

「セキュア・バイ・デフォルト」の原則に則り、ソフトウェアの設計時において、ソフトウェアに対する脅威から利用者を保護するように設計されたソフトウェアのセキュリティ機能や設定を最初からデフォルトの状態で組み込みます。

⑦ ソフトウェア設計のレビュー

ソフトウェア設計のレビューを通じて、全てのセキュリティ要件を満たし、必要となるセキュリティ機能の実装やセキュリティ対策の実施が確実に行われ、ソフトウェア開発・運用に対する脅威に十分に対応していることを確認します。レビューにおいては、脅威への対応状況や設計上の考慮事項に関する記録を作成し、ソフトウェアのライフサイクル全体を通じて、監査や保守の目的で使用できるようにします。

要件定義や設計の段階において、上記の分析・評価を実施し、セキュリティを確保する機能を検討します。リスク分析の結果を踏まえ、搭載する機能を選択してください。

表：主な脅威と対応するセキュリティ機能例

対応する主な脅威	セキュリティ機能
機器に保存されるデータや設定情報の漏えい	初期化機能、データ消去機能
通信データの傍受や改ざん	暗号通信機能
機器に対する不正アクセス	インシデント検知機能、ログ出力機能
機器のマルウェア感染等によるデータ消失	バックアップ機能
運用段階で検出された脆弱性	アップデート機能

どのようなセキュリティ機能を搭載するかについては、IoT 製品に関連するものではありますが、「IoT 開発におけるセキュリティ設計の手引き」の「3.2.セキュリティ対策の検討」が参考になります。また、ソフトウェアの脆弱性対処においては、修正対策としてのアップデート機能が特に重要となることから、アップデートの種類やアップデート機能実装による

影響への対応について後述します。さらに、データ消去機能についても、本ガイドの「15　ソフトウェアのサポート終了と廃棄対応」において後述します。

セキュア・バイ・デフォルトの原則に沿ってセキュリティ機能を搭載する際に、初期設定値（デフォルト値）がセキュアな設定になっているよう配慮しておくことも重要です。

ソフトウェアがハードウェアに組み込まれ、実社会において機械・機器として設置される場合、悪意のある第三者による製品への不正侵入、不正操作等により製品が誤動作をした際に、誤動作によって製品利用者に危害を及ぼさないようにするため、安全に製品を停止させる機能も考慮する必要があります。具体的な設計の考え方等については、「IoT セキュリティガイドライン ver1.0 [2]」の「要点 10. 安全安心を実現する設計の整合性をとる」や「つながる世界の開発指針 第 2 版[3]」を参照ください。

上述のとおり、脆弱性の修正の適用するためのアップデートを、どのような手段で、誰が作業して、いつ適用するのかを検討することは、製品のセキュリティを維持するために非常に重要です。アップデートの種類ごとに違いがあるため、製品に合ったアップデート方法を検討してください。採用したアップデートの方法が使えなくなった場合に備え、複数のアップデートの種類に対応することについても考慮・検討が必要です。

アップデートの種類
■ 1. 製品が自動的に実施するアップデート オンラインの自動アップデート機能の提供を指します。製品利用者の負担にならず、アップデートの実施率を上げられるため、製品セキュリティの確保が容易です。一方、利用者がアップデートのタイミングを選べないため、アップデートによる製品動作への影響がある場合には、利用者が、自動でアップデートするか自身の判断でアップデートする（以下 2.）かを、設定等で選択できるようにすることが望まれます。 対象例 常時ネットワークに接続して利用する製品、更新が頻繁である製品
■ 2. 利用者が実施時期を選択するアップデート オンラインのアップデート機能の提供を指します。ユーザインタフェース上に、ポップアップや通知ランプの点滅等でアップデートがあることを通知し、適用を促します。 対象例 常時ネットワークに接続して利用する製品、利用者自身が更新適用の時期を判断する必要がある製品
■ 3. 利用者による手動のアップデート

[2] IoT セキュリティガイドライン ver1.0（IoT 推進コンソーシアム、総務省、経済産業省）
http://www.iotac.jp/wp-content/uploads/2016/01/03-IoT セキュリティガイドライン ver1.0 別紙 1 .pdf

[3] つながる世界の開発指針 第 2 版（IPA）
https://www.ipa.go.jp/archive/publish/qv6pgp000000114a-att/000060387.pdf

インターネット経由のアップデートファイル（パッチ・ファームウェア）の提供を指します。利用者がそれをダウンロードしアップデートを適用します。アップデート時に利用者が製品を操作する必要があるため、アップデートファイルの入手方法やその適用手順を利用者へわかりやすく説明、開示する必要があります。

対象例 利用時のネットワーク接続が頻繁ではない製品、ネットワークに接続していない状態でも製品の利用が可能な製品

■ 4. オフラインでのアップデート

製品開発者が製品設置場所に訪問する、または、利用者が製品を製品開発者に発送するなどして、ネットワークを経由しない方法で、製品開発者が製品を直接アップデートする方法を指します。訪問によるアップデートの場合は、USB メモリなどの外部記憶媒体を介してアップデートファイルを製品に適用します。オフラインでのアップデートの場合、来訪の対応や発送手続きなど利用者に作業が生じます。そのため、利用者が行う作業については予め周知する必要があります。また、利用者の作業負担があるため、アップデートの適用率が他の手法より低くなる可能性があります。

対象例 インターネットに常時接続していない製品、更新が頻繁ではない製品、アップデート機能の搭載コストが製品価格に見合わない製品

アップデートの種類を比較すると、オンラインのアップデート機能を製品へ実装することはオフラインでのアップデートよりもコストがかかるように見えます。しかし、オフラインでのアップデートの場合、配送費や訪問費が発生するため、長期的にはオフラインでのアップデートの方がかえってコストがかかります。また、利用者による適用率を上げるという観点でも、適用が容易なオンラインによるアップデートを採用することが望まれます。

表：アップデート機能実装による影響への対応

影響への対応手段	目的・効果
アップデート日時の設定や帯域制御を可能とする	アップデート中の性能低下・ネットワーク帯域不足の防止
自動的にアップデート前のバージョンに戻すことを可能とする（特に自動アップデートの場合）	アップデート後に動作しなくなる可能性の低減
通信の暗号化	アップデートファイルの改ざん防止
アップデートファイルのコード署名	
アップデートの状況確認機能	・アップデート状況・バージョン情報の確認を確認可能にする ・アップデート忘れの防止
製品の異常動作検知	不正な更新が行われた場合の検知

ウイルスチェック（特にオフラインでのアップデートの場合）	USB メモリなどを経由したアップデート時のウイルス混入（感染）防止

開示方法

- セキュリティを確保するための設計に関する取り組みを製品セキュリティポリシーに含め開示します。
- 利用者がセキュリティ機能を正しく利用できるように、製品に搭載しているセキュリティ機能の利用方法や設定内容については、取扱説明書や製品情報を掲載しているウェブサイト等に記載します。
- 利用者が製品を選ぶ際に、搭載しているセキュリティ機能（アップデート機能、初期化機能等）を、パッケージや製品情報を掲載しているウェブサイト等に記載します。EC サイトなど販売形態によってはパッケージの記載を確認できない場合が想定されるため、複数個所に記載することが望ましいです。
- 利用者が製品を選ぶ際に、アップデートの種類を、パッケージや製品情報を掲載しているウェブサイト等に開示します。
- 利用者がアップデートを正しく実施するために、アップデート方法を、取扱説明書や製品情報を掲載しているウェブサイト等で掲示します。
- ソフトウェア製品に対してアップデート機能を実装していることを製品セキュリティポリシー等に含め開示します。

5 既知の脆弱性解消

意義

製品の構成要素（コンポーネント、ライブラリ等）に存在する既知の脆弱性に対しては、攻撃手法や攻撃ツール等が開示されている場合があり、サイバー攻撃を受ける可能性が高いことから、既知の脆弱性は可能な限り解消する必要があります。製品に残存した既知の脆弱性の悪用により製品利用者がサイバー攻撃の被害を受けると、対処に多くの費用が必要となることに加え、攻撃された製品がサイバー攻撃の踏み台となり、利用者に限らず社会へ影響が拡大すれば、かかる費用も比例する可能性があります。大規模な被害となった場合、たとえ外部組織が開発した構成要素が原因でも、製品や製品開発者に対する評価の低下を引き起こすことにもなりかねません。製品の設計・開発時から、構成要素の脆弱性の有無を可能な限り確認、対応することで、出荷・リリース後に脆弱性による問題が発生するリスクを低減することが可能となります。

なお、製品の設計・開発時だけではなく、出荷・リリース後も構成要素であるコンポーネント等の脆弱性情報を収集し、その影響有無を確認し続ける必要があります。出荷・リリース後の実施事項の詳細は、本ガイドの「10. 脆弱性の発見」を参照ください。

実施内容

構成管理を実施し、構成要素に含まれる脆弱性に関する情報収集を行い、必要な対処の判断と適用を実施します。連携が必要となる関係者とその役割については、「附属 主要な関係者・役割表」を参照してください。

実施手順

① 構成管理

すべての構成要素に関し構成管理を実施します。構成管理として記録すべき情報は、製品名、製品開発者名、バージョン情報等です。特に、バージョン情報がないと、脆弱性情報に対して該当製品か否かを判断できません。その結果、対処が遅れる可能性等があるため留意が必要です。構成管理を行う上でのポイントは以下の通りです。

- 委託開発したソフトウェアについては、そのソフトウェアの構成要素の情報を納品物に含むよう契約に明記します。
- 要件定義・設計・開発工程で製品に組み込む製品（コンポーネント）に変更が生じた場合は、随時記録します。また、製品の出荷・リリース時に最新情報を運用部門へ展開できるようにします。

② 脆弱性に関する情報収集

構成要素毎に関連する製品開発者、公的機関及びセキュリティコミュニティ等から脆弱性情報（対策情報含む）を収集します。

脆弱性情報の収集・対応は、設計・開発フェーズの初期段階だけではなく、出荷・リリースの直前まで定期的に実施します。

脆弱性情報及びその深刻度等の情報収集には、脆弱性対策情報のデータベースである JVN iPedia[4] を利用できます。

③ 必要な対処の判断と適用

脆弱性によって生じる被害とその大きさ、及びその発生の可能性からリスクを分析し、リスクの大きさにあわせて対応を検討し、対処します。

上記の実施内容は、レベル３をベースにしています。これらの項目の対応が難しい場合は、以下に記載している他のレベルの実施内容を参照し対応を検討してください。

レベル	実施内容
1	可能な範囲で構成管理を実施します。製品開発者及び公的機関、セキュリティコミュニティ等から脆弱性情報を収集し、深刻度の高い脆弱性から対処を行います。深刻度については、共通脆弱性評価システム（CVSS）[5] や脅威指標等を参考にしてください。
2	すべての構成管理を実施します。製品開発者及び公的機関、セキュリティコミュニティ等から脆弱性情報を収集し、深刻度の高い脆弱性から対処を行います。深刻度については、共通脆弱性評価システム（CVSS）や脅威指標等を参考にしてください。
3	すべての構成管理を実施します。製品開発者及び公的機関、セキュリティコミュニティ等から脆弱性情報を収集し、製品に関連した脆弱性情報に関するリスク分析結果に応じて対処を行います。

脆弱性情報の収集と対処については「脆弱性対策の効果的な進め方（実践編）第 2 版 [6]」を参照ください。

開示方法

- 既知の脆弱性を解消する取り組みを製品セキュリティポリシー等に含め開示します。

[4] JVN iPedia https://jvndb.jvn.jp/

[5] 共通脆弱性評価システム CVSS v3 概説（IPA） https://www.ipa.go.jp/security/vuln/scap/cvssv3.html

[6] 脆弱性対策の効果的な進め方（実践編）第 2 版（IPA）
https://www.ipa.go.jp/security/reports/technicalwatch/hjuojm0000006nd2-att/000071660.pdf

6 セキュアコーディング・セキュアビルド

意義

開発時に脆弱性を作り込まないようにするために、セキュリティに配慮したコーディング規約やビルドツールをもとに開発を行う必要があります。それを行わない場合、プログラムへのセキュリティ実装が開発者のスキルに依存することになり、組織あるいは製品のセキュリティレベルが安定しません。その結果、保守しづらくなるだけでなく、脆弱性が作り込まれる可能性が高まり、製品利用時の脅威が残存する可能性も高まります。

セキュリティに配慮したコーディング規約、コンパイラ、インタプリタ、及びビルドツールをもとにした開発を実施することで、プログラムの品質や保守性だけでなく、セキュリティを高めることが期待できます。

実施内容

セキュリティに考慮したコーディング規約を策定し、規約に則った実装が行われていることを確認します。また、実行可能なファイルのセキュリティを向上させる機能を提供するコンパイラ、インタプリタ、及びビルドツールを使用して、コードの生成・ビルドが行われていることを確認します。

連携が必要となる関係者とその役割については、「附属 主要な関係者・役割表」を参照してください。

実施手順
① **セキュアコーディング規約の策定** セキュリティに配慮したコーディング規約を定めることが重要です。公表されているコーディング規約等[7] [8] [9] [10]を参照しつつ、組織に適したコーディング規約を定めます。コーディング規約には、例えば、重要情報（特定の動作環境等のリソース、例えばパスワードや IP アドレス、暗号化鍵等）をソースコードに埋め込むことを避け、重要情報をハードコーディングしないこと等も記載します。 ② **教育** 策定したセキュアコーディング規約をプログラマに周知し、定期的に教育を実施します。 ③ **実装**

[7] セキュア・プログラミング講座（IPA）https://www.ipa.go.jp/archive/security/vuln/programming/index.html

[8] 安全なウェブサイトの作り方（IPA）https://www.ipa.go.jp/security/vuln/websecurity/about.html

[9] セキュアコーディング（JPCERT/CC）https://www.jpcert.or.jp/securecoding/

[10] SEC BOOKS: ESCR Ver.3.0:組込みソフトウェア開発向けコーディング作法ガイド［C 言語版］ESCR Ver.3.0（IPA）https://www.ipa.go.jp/archive/publish/secbooks20180629.html

策定したセキュアコーディング規約に則りコーディングを行います。

④ **レビュー担当者によるレビュー**

コーディングを担当した本人がソースコードを見直すだけでなく、別の担当者が第三者の視点でレビューを行います。複数人でレビューすることで、規約に則っていないコーディングの発見やコーディング規約についての認識のずれ等を解消でき、これにより脆弱性の作り込みの低減を期待できます。

⑤ **セキュアコーディング規約の更新**

新たな攻撃手法に対応するため、定期的に更新します。

⑥ **セキュアビルド**

コードをビルドする際には、実行可能なファイルのセキュリティを向上させる機能を提供するコンパイラ、インタプリタ、及びビルドツールを使用します。コンパイラ、インタプリタ、及びビルドツールの使用やそれぞれの構成方法については、経営層の承認を得ます。その上で承認が得られた構成を実装して使用します。

上記の実施内容は、レベル 2 をベースにしています。これらの項目の対応が難しい場合は、レベル 1 の実施内容を参照し対応を検討してください。

レベル	実施内容
1	組織のコーディング規約や、使用するコンパイラ、インタプリタ、及びビルドツールを定めて、コードの生成・ビルドを実施します。
2	組織のコーディング規約や、使用するコンパイラ、インタプリタ、及びビルドツールを定めて、コードの生成・ビルドを実施し、別に定めたレビュー担当者がレビューします。

なお、開発時にオープンソースソフトウェアなどを利用すると、ソースコードが開示されているため開発にかかる時間が短縮されるメリットがあります。しかし、ソースコードが開示されているという特性上、脆弱性が存在した場合に攻撃者にそれを発見、特定されやすいです。よって、オープンソースなどを利用する場合にも、策定したセキュアコーディング規約にそったコーディングがされているかを確認し、対処する必要があります。

開示方法

- セキュリティを配慮したコーディング規約を策定し開発を行っている点や、セキュリティを配慮したコンパイラ、インタプリタ、及びビルドツールを使用しビルドを行っている点を製品セキュリティポリシー等に含め開示します。
- 具体的に記載する場合は、ホワイトペーパー等に記載する方法もあります。

7 開発時の脆弱性検査

意義

脆弱性が残ったままで製品を出荷・リリースしないために、脆弱性検査を実施することが必要です。製品の出荷・リリース後に脆弱性が発覚すると、アップデートファイルの開発・適用や製品利用者に対する説明等に対応コストが発生します。脆弱性の内容によってはアップデートができず、製品を回収する必要が生じる恐れもあります。開発時の脆弱性検査は、出荷・リリース後に発見された脆弱性に対処するためのコスト及び脆弱性を悪用した致命的なインシデントへの対処コスト削減することが期待できます。

実施内容

設計・開発の各段階で脆弱性検査を実施し、検知された脆弱性の対応を行います。連携が必要となる関係者とその役割については、「附属 主要な関係者・役割表」を参照してください。

実施手順

① テスト方針・計画の策定

要件定義段階で、テストの目的や範囲、必要なテスト環境やその体制、スケジュール、テスト完了基準などを明確にし、テスト計画書として文書化します。

テスト計画の記載内容を組織内で合意をしておきます。テスト計画にもとづき、必要な予算や体制を確保します。

脆弱性検査に関する要員のスキル等によっては、セキュリティベンダへの脆弱性検査の依頼を検討します。

② セキュリティテストケースの作成

要件定義時に作成したセキュリティ要件にもとづき、設計の段階でセキュリティテストケースを作成します。

セキュリティ要件通り実装できているのか、また、一般的な既知の脆弱性（SQL インジェクションなど）を作りこんでいないかの確認を行います。

テストケースは、「機能にもとづくテストケース」のみでなく、「セキュアコーディング規約にもとづくテストケース」、「一般的な脆弱性（SQL インジェクションなど）に関する汎用的なテストケース」、「実装言語にもとづくテストケース」なども含めるようにします。

③ 脆弱性検査の実施

作成したセキュリティテストケースに従い、脆弱性検査を実施します。各テスト工程（単体テスト、結合テスト、システムテスト）に応じ、必要な脆弱性検査を実施することが望まれます。

また、脆弱性検査には様々な手法があり、発見できる脆弱性の種類も異なります。主な検査名は、下表のとおりです。各検査の詳細については、「脆弱性検査と脆弱性対策に関するレポート[11]」、「ファジング活用の手引き[12]」を参照してください。

検査名	概要	特徴
ソースコードセキュリティ検査	・ソースコードに作り込んでしまった脆弱性を検出する検査。 ・ソースコードの中の脆弱性を引き起こしやすい関数を見つけたり、構文解析を実施したりします。	・実装の段階で生じる脆弱性を対象に検査することが主です。よって、設計で入り込んでしまう脆弱性や未知の脆弱性は検証できません。
ウェブアプリケーションセキュリティ検査	・文字列を送付したり、ページの遷移を確認したりして、ウェブアプリケーションに特化した脆弱性の存在を検出する検査。	・深刻な被害をもたらす SQL インジェクション等の脆弱性を見つけることができる。 ・実装の段階で作り込まれる脆弱性の発見に向いている。
システムセキュリティ検査	・組み込み構成要素（コンポーネント）等に脆弱性がないか確認するためのリクエストや、バージョンを確認するためのリクエストを送付し、既知の脆弱性が残っていないか、セキュティ上問題のある設定が行われていないか等を検出する検査。	・基本的に既知の脆弱性を見つけることを目的としている。
ファジングによる検査	・脆弱性を発現させやすいデータやファイルを送り込み、脆弱性を検出する検査。	・他の検査では見つけづらい脆弱性が見つけられる。
ペネトレーションテスト	・攻撃者が実際に侵入できるかどうかという点に着目した検査。	・攻撃者がどこまで侵入できるのか、何をされてしまうのか、の検証に着目していることが特徴。 ・ソフトウェアの脆弱性だけではなく、ネットワーク上の不適切な運用についても見つけることができる。

なお、このような脆弱性検査においては、生成 AI が活用されていることがあります。生成 AI を活用した脆弱性検査では、これまで脆弱性検査に関する要員の手作業やツール利用に依存してきたテスト工程を自動化することができ、またテスト工程で見つけることができなかった脆弱性についても、生成 AI が学習してカバーし、指摘できるようになるため、検査品質の向上と検査の効率化の双方につながる効果が期待されています。

④ 検知された脆弱性の管理と対応

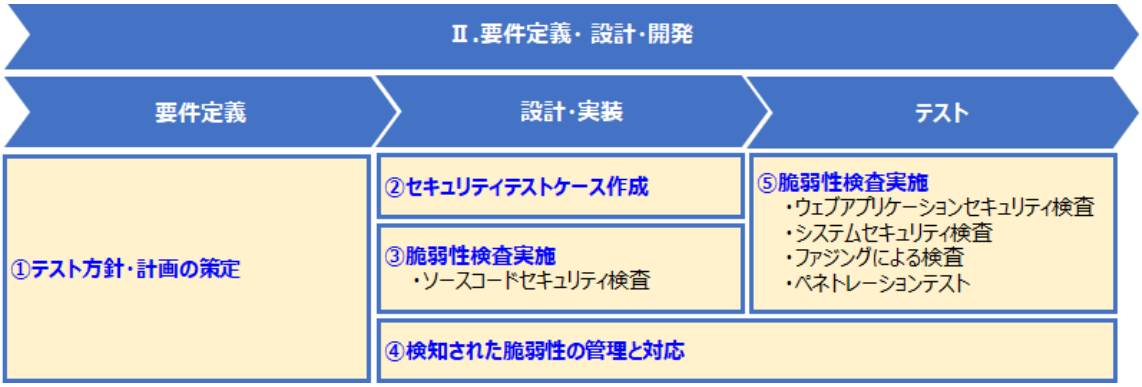
[11] 脆弱性検査と脆弱性対策に関するレポート（IPA）
https://warp.ndl.go.jp/web/20220711144627/https://www.ipa.go.jp/about/technicalwatch/20130808.html

[12] ファジング活用の手引き（IPA）https://www.ipa.go.jp/security/vuln/fuzzing/contents.html

脆弱性などの検知された問題については、対応要否を確認し適切に修正を行います。その際、問題の詳細・修正内容を記録し管理します。
原因や修正内容は関係者で共有し、発見された箇所以外にも、同様の問題がないか確認し、対応を行います。
設計の変更を伴うような修正を行うなどした場合は、リグレッションテストを行い、修正以外の箇所に新たな脆弱性を生み出していないかも検証する必要があります。

なお、脆弱性検査ツールで全ての脆弱性パターンについて調査することは不可能であることや、発見できない脆弱性もあることを認識し、不足するパターンについては、手動テストで補うことが必要です。

また、脆弱性検査を行う担当者は、脆弱性検査に関する知識・経験を備えた専門家であることが望まれます。自組織内で要員の確保が難しい場合は、必要に応じて、セキュリティベンダへ検証を依頼することも検討します。



図：開発時の脆弱性検査実施の流れ

脆弱性検査を計画・実施する際には、以下の資料も参照してください。

- 「IoT セキュリティ評価検証ガイドライン Rev1.0[13]」
- 「OWASP SAMM[14]」

開示方法

- 開発時に脆弱性検査を実施している点を製品セキュリティポリシー等に含め開示します。

[13] IoT セキュリティ評価検証ガイドライン_r1.0（CCDS）
https://www.ccds.or.jp/public/document/other/guidelines/CCDS_IoT%E3%82%BB%E3%82%AD%E3%83%A5%E3%83%AA%E3%83%86%E3%82%A3%E8%A9%95%E4%BE%A1%E6%A4%9C%E8%A8%BC%E3%82%AC%E3%82%A4%E3%83%89%E3%83%A9%E3%82%A4%E3%83%B3_rev1.0.pdf

[14] OWASP SAMM（JPCERT/CC）https://www.jpcert.or.jp/research/2010/SAMM_20100407.pdf

8 開発に使用する環境及びツールのセキュリティ確保

意義

開発にあたっては、ソフトウェア製品のセキュリティだけではなく、開発するための施設や環境、開発するために使うツールのセキュリティを確保することが重要です。開発環境に不正にアクセスされてしまうと、開発中のソフトウェア製品の情報が窃取、改ざんされる可能性があります。製品の開発に使うソフトウェアが改ざんされてしまうと、作成するソフトウェア製品にも脆弱性が作りこまれてしまうことが考えられます。また、開発時に使用していたテスト用の機能やアカウントの情報などが製品に残存していた場合、これらの情報が悪用される可能性があります。

セキュアなソフトウェア開発を実践するには、開発環境、ビルド環境や開発ツール、ビルドツールを含めた開発環境全体におけるセキュリティ確保が必要です。

実施内容

ソフトウェア製品の開発に関する情報を情報資産ととらえ、開発するための施設や環境へのセキュリティ対策を実施します。また、開発に使用するツールについては、セキュリティが確保されているものを利用します。連携が必要となる関係者とその役割については、「附属 主要な関係者・役割表」を参照してください。

実施手順

① 開発環境及び開発ツールのセキュリティ確保

下記(1)～(3)の方法により、開発環境及び開発ツールのセキュリティを確保します。

(1) 開発に使う施設・設備のセキュリティの確保

自組織の情報セキュリティポリシーや製品セキュリティポリシーに基づいて、開発に使用する施設・設備のセキュリティを確保します。開発に使用する施設・設備は、自組織で使う一般的な情報システムとは物理的にも論理的にも分離させておくことが望まれます。物理的な分離方法としては、執務フロア内にセキュリティ区画を設けて入退室管理を実施する方法があります。また論理的なものとしては、開発で使用するネットワークや開発用端末のアカウントや IP アドレスなどをもとにアクセス制限を実施する方法があります。また、開発に使用する端末では、マルウェアの感染による情報漏えいや改ざんを防止するため、業務目的での日常的なメールやインターネットブラウザの使用を制限することも検討します。

(2) 開発中のソフトウェア製品のソースコードや仕様書等のセキュリティの確保

開発中のソフトウェア製品のソースコードや仕様書などのソフトウェア製品に関連するドキュメントは、情報資産として、適切なアクセス制御のもと管理します。ソースコードは、改ざんの防止機能をもつバージョン管理システムやリポジトリを使うことで効率的に管理することができます。

また、テスト用に設けた特別な機能やアカウントの情報が記載されたドキュメントは、漏えいすると、攻撃が容易になる可能性があり、十分に注意する必要があります。

(3) 開発に使用するツールのセキュリティの確保

コードの作成を効率化するツールやノーコード・ローコード開発ツールなどの開発に使用するツールにセキュリティ上の問題がある場合、それらのツールを使って開発したソフトウェア製品にも、脆弱性が作りこまれてしまう可能性があります。開発に使用するツールは、最新のバージョンを利用することを検討します。

② ビルド環境及びビルドツールのセキュリティ確保

下記(1)の方法により、ビルド環境及びビルドツールのセキュリティを確保します。

(1) ビルドに使用するツールのセキュリティの確保

コードをビルドする際に、実行可能なファイルのセキュリティを向上させる機能を提供するコンパイラやインタプリタ、ビルドツールなどのビルドに使用するツールにセキュリティ上の問題がある場合、それらのツールを使ってビルドしたソフトウェア製品にも、脆弱性が作りこまれてしまう可能性があります。ビルドに使用するツールは、最新のバージョンを利用することを検討します。

新規のソフトウェアあるいは変更したソフトウェアを、新しいバージョンとしてリリースするための一連の手順・考え方である CI/CD パイプラインについて、継続的インテグレーション（CI（Continuous Integration）＊1）と継続的デリバリ（CD（Continuous Delivery）＊2）の 2 つの要素を内包しつつ、継続的かつ迅速なソフトウェアの改善やソフトウェア開発のライフサイクルの自動化のために構築することを検討します。

具体的には、コードレビューの完了後に、ビルドやテスト、デプロイが自動的に行われるようにすることにより、手作業による非効率性の改善や、製品・バージョン管理に係る負担の軽減につなげることができます。検討にあたっては、CI/CD パイプラインにおけるセキュリティ確保が重要となるため、デジタル庁が 2024 年 3 月 29 日に公表している「デジタル社会推進実践ガイドブック DS-202 CI/CD パイプラインにおけるセキュリティの留意点に関する技術レポート[15]」を参考にして、必要なセキュリティ対策を導入してください。

＊1：継続的インテグレーション（CI）

ソースコードに対する同時的な複数の変更を、継続的な検証によってソフトウェアの完全性を担保しながら自動的に取り込む考え方及び手順を指します。その際に、動作確認をするテストや、ソースコードから成果物の生成も合わせて行います。

＊2：継続的デリバリ（CD）

[15] デジタル社会推進実践ガイドブック DS-202 CI/CD パイプラインにおけるセキュリティの留意点に関する技術レポート（デジタル庁）https://www.digital.go.jp/assets/contents/node/basic_page/field_ref_resources/e2a06143-ed29-4f1d-9c31-0f06fca67afc/33f31336/20240329_resources_standard_guidelines_guideline_01.pdf

ソースコードあるいはソースコードを元にした成果物を実際に実行する環境に配送（Delivery）する業務を、自動的かつ継続的に実施する手順を指します。

また、CI/CD パイプラインが稼働する際には、近年、ソフトウェア開発において利用されることがあるインターネット上でソースコードの管理を行えるウェブサービスがその起点となる場合があります。そのようなウェブサービスにおいて、テスト用の機能やアカウント情報などの機密情報を誤って開示してしまうと、攻撃に悪用される可能性もあり、情報へのアクセス権限の設定には注意が必要です。

開示方法

- 開発環境、ビルド環境、検査環境や開発ツール、ビルドツール、検査ツールを含めた開発全体のセキュリティを確保していることについて、製品セキュリティポリシー等に記載します。
- 具体的に記載する場合は、ホワイトペーパー等に記載する方法もあります。

Ⅲ. 供給・配布

9 ソフトウェア製品のセキュアな配布

意義

配布システムや電子媒体を介したソフトウェアの配布については、セキュリティを確保することが重要です。配布システムに不正にアクセスされてしまうと、配布するソフトウェアが悪意のあるソフトウェアで上書きされ、改ざんされる可能性があります。また、ソフトウェアを配布する際に用いる通信路において、通信データが改ざんされ、ソフトウェアが不正な内容に変更される可能性もあります。このような配布システムだけではなく、ソフトウェアを配布する際にソフトウェアを格納した電子媒体を活用する場合にも、セキュリティに対する注意が必要です。マルウェアに感染した電子媒体に配布するソフトウェアを格納してしまうと、ソフトウェアが感染し、さらにソフトウェアを適用したシステムやサービスにまで感染被害が拡大する可能性があります。

セキュアなソフトウェア供給を行うには、配布システムや電子媒体におけるセキュリティ確保が必要です。

実施内容

ソフトウェア製品の開発に使用する環境と同様、ソフトウェア製品を配布するためのシステムや環境へのセキュリティ対策を実施します。また、配布に使用する電子媒体については、セキュリティが確保されているものを利用します。連携が必要となる関係者とその役割については、「附属 主要な関係者・役割表」を参照してください。

実施手順
① 配布システム及び電子媒体を介したソフトウェア配布のセキュリティ確保 下記(1)～(2)の方法により、配布システム及び電子媒体のセキュリティを確保します。 (1) 配布に使うシステムのセキュリティの確保 自組織の情報セキュリティポリシーや製品セキュリティポリシーに基づいて、自組織のウェブサイトやリポジトリを管理するサーバ、物理的機器が適用対象である場合のOTA（Over The Air）管理システムなどの配布に使用するシステムのセキュリティを確保します。 (2) 配布に使用する電子媒体のセキュリティの確保 配布に使用する電子媒体がマルウェアに感染している場合、当該電子媒体に配布するソフトウェアが格納されると、当該ソフトウェアにも、マルウェアが拡散してしまう可能性があります。 配布に使用する電子媒体は、事前にウイルス検疫を実施した上で利用します。 **② パッケージの配信** ソフトウェアを配布する際に用いる通信路にセキュリティ上の問題がある場合、通信デ

ータが改ざんされ、配布するソフトウェアが不正なものに変更されてしまう可能性や、既知の脆弱性を利用可能とするため、古いバージョンのソフトウェアが配布されてしまう可能性があります。

配布されるパッケージは、配信データを暗号化により保護するとともに、改ざんなどの侵害されていないことを確認し、それらの同一性を保証するための署名（コードサイニング等）を行います。

③ セキュリティ修正プログラムの配布

ソフトウェアの配布と同様、セキュリティ修正プログラムの配布においても、配布システムや電子媒体に対して、セキュリティ対策を実施します。

セキュリティ修正プログラムの配布は、ソフトウェアにおいて特定された脆弱性の問題に対処し、安全性を高めるために実施されますが、セキュリティ修正プログラムを配布経路上で改ざんされる場合には、ソフトウェアが適用されたサービス等に障害がもたらされてしまう可能性があります。

開示方法

- ソフトウェアを配布する際のセキュリティを確保していることについて、製品セキュリティポリシー等に記載します。
- 具体的に記載する場合は、ホワイトペーパー等に記載する方法もあります。

IV. 運用

10 脆弱性の発見

意義

出荷・リリース後に発見された脆弱性を早期に対処するためには、組織の製品及び構成要素（コンポーネント、ライブラリ等）に関する脆弱性情報を収集することが重要です。脆弱性は日々発見されています。出荷・リリース後の製品の脆弱性の対処が後手に回ることの無いように、日々の情報収集が必要です。遅れれば、脆弱性を悪用したサイバー攻撃を受けるリスクが高まります。積極的に脆弱性情報を収集することで、より迅速に脆弱性に対処し、被害を最小限に抑えることができます。

また、第三者によって発見された脆弱性は、組織として適切に報告を受付・対処する必要があります。第三者からの脆弱性報告を受け付ける窓口を設置し、それを判りやすく公開しないと、第三者は脆弱性を発見しても報告できません。脆弱性が放置された結果、こうげき者によって悪用されてしまうかもしれません。第三者からの脆弱性報告を適切に受付・対処することは、脆弱性の放置を未然に防ぐことにつながります。

脆弱性情報の収集や第三者による脆弱性報告によって発見されていない新しい脆弱性を発見するために、内部テストを能動的かつ定期的に実施することも重要です。

実施内容

出荷・リリース後の製品について、関連する脆弱性情報の収集、第三者によって発見された脆弱性の報告受付、新しい脆弱性を発見するための内部テストを行います。連携が必要となる関係者とその役割については、「附属 主要な関係者・役割表」を参照してください。

実施手順
① 製品と構成要素の脆弱性監視 出荷・リリースした製品と、製品に組み込んだ構成要素に関する脆弱性情報を収集します。 出荷・リリースした製品の脆弱性情報は、セキュリティに関連する組織やコミュニティによって開示されることがあります。またソーシャルメディアなどで開示されることもあります。構成管理の情報をもとにして、製品や組織の体力に応じて定期的なパブリックモニタリングをすることが有用です。 また、利用者からの不具合に関する問い合わせの中には、原因が脆弱性であると判明する場合もあります。利用者からの問い合わせも情報源として扱います。 構成要素については、「5 既知の脆弱性解消」において作成した構成管理の資料をもとに、製品の出荷・リリース後も継続して脆弱性情報を収集します。詳細については、「5 既知の脆弱性解消」を参照してください。脆弱性修正や機能改善のために構成要素であるコンポーネント等のバージョンが変更された場合には、構成管理の資料を適切に更新することが必要です。脆弱性の修正対応時の構成管理の資料の更新については「12 脆弱性の修正対策と対策情報の公表」も参照してください。

② 脆弱性報告の受付

下記(1)～(2)の方法により、第三者によって発見された脆弱性について、組織として適切に報告を受付・対処します。

(1) 受付窓口の設置

脆弱性情報の受付窓口を設置し、脆弱性の発見者から情報を受け付けられるようにします。なお、受付窓口は脆弱性情報専用である必要はありませんが、脆弱性情報を受け付けていることが分かるように明示します。

また、第三者によって発見された脆弱性の報告を受け付ける際には、報告の品質を担保し、報告内容の迅速な評価に繋げるため、発見者に対して、ガイドラインや報告用フォーマットの提示等を通じて報告内容に最低限盛り込むべき情報を周知します。

さらに、脆弱性情報の報告があった場合に適切な対応が取れるように、組織内での報告ルートや対応方法を事前に組織内に周知します。一般的な問い合わせ窓口へ脆弱性情報が報告される場合もあるため、脆弱性の報告があった場合の対応を定め、周知します。

(2) 脆弱性情報の受付

窓口に脆弱性情報の報告があった場合、(1)で決めた報告ルートに従い、適切な部門に連絡します。脆弱性情報を受領した旨や対応状況を報告者へ連絡します。

また、発見者がどのような人物か（セキュリティ研究者、製品の利用顧客、関係するSI 事業者等）についても確認し、関係部門に参考情報として展開することで、報告内容が前提とする状況や、再現に関する信頼性を判断しやすくなります。

③ 内部テストによる脆弱性の発見

脆弱性情報の収集や第三者による脆弱性報告によって発見されていない新しい脆弱性を発見するため、内部テストを定期的に実施します。

内部テストでは、ソフトウェアの設計時に実施したリスクアセスメントの結果や、脅威への対応状況、設計上の考慮事項等をレビューし、搭載したセキュリティ機能によるリスクへの対処が適切であるかを確認します。また、ソフトウェアの開発時に実施した脆弱性検査を改めて実施することも望まれます。

さらに、実際のソフトウェア構成に照らし合わせて、事前に把握しているコンポーネント間の関係性に変化が生じていないかを解析します。このような解析は、ソフトウェア構成解析（SCA（Software Composition Analysis））と呼ばれています。ソフトウェア構成解析は、コンポーネントやライブラリの脆弱性管理においても有用です。リリース後、ソフトウェア構成解析を活用して、リリースされたソフトウェア製品が利用しているコンポーネントやライブラリとそのバージョンを把握し、コンポーネントやライブラリ内に含まれている脆弱性のあるプログラムを検出できるようにします。

コンポーネント間の関係性については、リリースされたソフトウェアの全てのコンポーネントの出所データや関係性等を管理するソフトウェア部品表（SBOM（Software Bill of Materials））を作成・更新しておくことが望まれます。

脆弱性を発見した第三者が、IPA と JPCERT/CC が運営する「情報セキュリティ早期警戒パートナーシップ」に報告[16]する場合があります。その場合の製品開発者への脆弱性情報の報告は JPCERT/CC から来ることになり、直接の報告よりも時間を要します。そのため、開発者は脆弱性情報を迅速に入手できるよう、JPCERT/CC の製品開発者リストに予め登録しておいてください[17]。パートナーシップの詳細は「情報セキュリティ早期警戒パートナーシップガイドライン[18]」を参照してください。

また、上述のように、脆弱性の発見のきっかけとなるのは、製品利用者からの問い合わせであることもあります。そのような場合に備え、日常より製品利用者との間で良好な関係を構築し、脆弱性の調査に必要となるログファイルの提供などの協力を依頼できるようにしておくことも重要な対策の一つといえます。

製品利用者がソフトウェア製品の使い方などの質問を自由に投稿でき、それに対して製品開発者が解決策などを投稿することで必要な情報の迅速な入手を可能とする技術フォーラムなどのコミュニティ運営を利用することも、製品利用者との関係を築く 1 つの方法論です。

開示方法

- 出荷・リリース後に脆弱性情報を収集している点や新しい脆弱性を発見するための内部テストを実施している点を製品セキュリティポリシー等に含め開示します。
- 製品に関する脆弱性情報について、利用者に対して製品サポートのウェブページ等で開示します。
- 外部からの問合せや脆弱性報告を受け付けるために、製品利用者及び脆弱性発見者からの問合せ窓口を設置し、ウェブなどで告知しておきます。製品利用者向けの問合せ窓口（電話、メールアドレス等）は、ウェブサイトの製品ページや製品パッケージ等、容易に確認可能な場所に記載します。
- 利用者が迅速に対策を適用できるよう、報告された脆弱性情報について、製品サポートのウェブページ等で公表します。

【開示例（問合せ窓口）】

サポートに関するお問合せ

電話：0120-XXX-XXX　**メールでのお問合せ** 

（9:00-18:00　平日）　**（フォームが開きます）**

[16] 脆弱性関連情報の届出受付業務における取扱いプロセス（IPA）
https://www.ipa.go.jp/security/todokede/vuln/process.html

[17] 製品開発者登録（JPCERT/CC）https://www.jpcert.or.jp/vh/register.html

[18] 情報セキュリティ早期警戒パートナーシップガイドライン（IPA）
https://www.ipa.go.jp/security/guide/vuln/partnership_guide.html

11 脆弱性の検証（脆弱性情報のトリアージと分析）

意義

日々発見され、報告される脆弱性情報は、情報量が多く、また脆弱性の対象や深刻度もさまざまであるため、脆弱性情報の見落としや対処の遅れが発生しないよう、脆脆弱性情報を適切に管理します。また、報告される脆脆弱性情報をもとに、脆弱性の認定や対処の優先順位付け（トリアージ）を行い、自組織にとって対処が必要な脆弱性であるかの判断を迅速かつ効率的に行うことが重要です。

また、脆弱性が認定された後に、脆弱性の修正対策の検討を進めるにあたり、報告された脆弱性を再現し、脆弱性が作り込まれた原因や、脆弱な状態が成立し得る際の条件を把握することが必要です。報告された脆弱性が確実に再現可能であることを確認することで、ソフトウェアの影響を受ける範囲の特定や、他の脆弱性が作り込まれるリスクの判断を行うことができます。

実施内容

報告された脆弱性情報について、自組織にとって対処が必要な脆弱性であるかの判断を行うために、脆弱性の認定を行います。また、脆弱性を認定した後に、脆弱性の再現、根本原因の特定を行います。連携が必要となる関係者とその役割については、「附属 主要な関係者・役割表」を参照してください。

実施手順
① 脆弱性の認定 自組織において対処が必要な脆弱性であるかどうかを適切に判断できるようにするため、対処の優先順位付けの考え方を明確にした上で脆弱性の認定基準を策定します。 脆弱性の認定基準は、その後の運用を通じて得られたフィードバックを反映しつつ、定期的に見直し、改善することが必要です。 **② 脆弱性の再現** 脆弱性が認定された後に、脆弱性が作り込まれた原因や、脆弱な状態が成立し得る際の条件を検証し、報告された脆弱性が確実に再現可能であることを確認します。 **③ 根本原因の特定** 脆弱性の再発を防ぐために、脆弱性を分析して根本原因を特定します。脆弱性の分析では、脆弱性の問題に繋がるコーディングパターンがセキュアコーディングの実践において実施されていないかの確認や、ソフトウェアに類似する脆弱性が存在しないかの確認、ソフトウェア開発ライフサイクルのプロセスに問題がないかの確認等を実施します。

脆弱性の再現を確認するにあたっては、再現検証用の環境を準備するだけではなく、報告対象となっているバージョンのソフトウェアを、いつでもすぐに利用できるような環境にし

ておくことが重要です。そのため、ソフトウェアのバージョン管理を適切に実施したうえで、過去のバージョンのソフトウェアを保管し、管理・整理しておくことが望まれます。

上記の実施内容は、レベル 2 をベースにしています。これらの項目の対応が難しい場合は、レベル 1 の実施内容を参照し対応を検討してください。

レベル	実施内容
1	報告された脆弱性情報について、自組織にとって対処が必要な脆弱性であるかの判断を行うために、脆弱性の認定・再現を実施します。
2	レベル 1 を満たしたうえで、脆弱性の再発を防ぐために、脆弱性を分析して根本原因を特定し、改善を図ります。

開示方法

- 報告される脆脆弱性情報をもとに、脆弱性の認定や対処の優先順位付け（トリアージ）等を行い、自組織にとって対処が必要な脆弱性であるかを判断している点を製品セキュリティポリシー等に含め開示します。
- 脆弱性が認定された後に、発見者との関係構築、脆弱性の再現、根本原因の特定を行っている点を製品セキュリティポリシー等に含め開示します。

12 脆弱性の修正対策と対策情報の公表

意義

存在することを把握した脆弱性については、組織として適切に対処し、最終的には、対策情報を公表する必要があります。脆弱性対策を作成してもその対策情報が適切に公表されない場合、製品の利用者が対策の必要性を認識できず、対策を適用しないと被害に遭う恐れがあります。その結果、製品や製品開発者への信頼低下に繋がりかねません。

また、適切に脆弱性情報を公表すれば、利用者へ脆弱性対策を促すとともに、脆弱性への対応を怠っていない企業の姿勢を潜在的な製品利用者に訴求でき、信頼や製品満足度の向上につながります。

実施内容

収集した情報や第三者からの報告、内部テストから脆弱性の存在が判明した場合には、対策の策定、対策の開示を行います。連携が必要となる関係者とその役割については、「附属 主要な関係者・役割表」を参照してください。

実施手順
① 修正対策の策定 脆弱性であると確認がとれた場合、脆弱性によって生じる被害とその大きさ及び発生の可能性からリスクを分析し、リスクの大きさにあわせて対応を検討し、対処します（詳しくは「5 既知の脆弱性解消」を参照ください）。このフェーズは製品の出荷・リリース後であり、脆弱性による被害を受ける利用者や第三者が現に存在しているため、可能な限り早急な対応が望まれます。 また、このような対処では、セキュリティ修正プログラムを作成し、配布する場合があります。なお、これらのプログラムに不具合が存在する場合には、ソフトウェアが適用されたサービス等に障害がもたらされてしまう可能性があります。このため、特定された脆弱性の問題に関わるリスク評価の結果やその結果に基づく対応の優先順位を考慮し、要否を判断しつつ作成するとともに、修正対策の配布前には、不具合を確認するためのテストを行います。特に、ソフトウェアの更新プログラムの配布においては、近年、更新プログラムの不具合が原因となり、ソフトウェアが適用されたサービス等に障害がもたらされる事例が発生しています。配布前の不具合を確認するためのテストについては、必ず実施するようにしてください。 さらに、これらのプログラムにおいて、ソフトウェアコンポーネントの入れ替えが必要になる場合があります。このような場合には、ソフトウェアとソフトウェアコンポーネント間の関係性を十分に把握したうえで、ソフトウェアコンポーネントの入れ替え等がもたらす影響について考慮することが求められるため、ソフトウェア部品表（SBOM（Software Bill of Materials））を活用することが望まれます。 その他に、これらのプログラムの適用による脆弱性の修正が困難な場合は、回避策や代替策で影響の発生を防げる場合があります。回避策や代替策は、セキュリティ修正プログラムやソフトウェアの更新プログラムの作成に時間を要する場合にも有用であるため、修正対策の公表の前に、回避策・代替策を提供することも選択肢として検討してください。

その上で回避策や代替策を展開できない場合は、利用者に対して使用停止を促すことの検討も重要です。

② 対策情報の公表

策定した対策を利用者に適用してもらうため、脆弱性対策情報を製品サポートのページ等で公表します。対策情報の公表方法については「ソフトウエア製品開発者による脆弱性対策情報の公表マニュアル[19]」を参照してください。また、製品登録やライセンス登録の過程で、製品開発者が製品利用者の連絡先を把握している場合や、SI 事業者を通じて、製品利用者への伝達が可能である場合には、公表と併用するかたちで、製品利用者に向けて電子メール等で対策情報を個別通知することも考えられます。

③ 構成管理資料の更新

脆弱性対策を適用した状態の製品の構成情報を、構成管理の資料に反映します。製品の構成は、脆弱性対策だけでなく、機能強化などによっても変更になることがあるため、製品のアップデートごとに構成情報を資料に反映します。反映した構成管理の資料は「10 脆弱性の発見」の「①製品と構成要素の脆弱性監視」で利用します。

開示方法

- 利用者が迅速に対策を適用できるよう、報告された脆弱性情報とその対策について、製品サポートのウェブページ等で公表します。
- 脆弱性の対応・情報提供を行っていることについて、製品セキュリティポリシーに含め開示します。
- 報告された脆弱性がどのように取り扱われるかを脆弱性の発見者が理解できるように、脆弱性の対応・情報提供に関して、対応フローや判断基準の詳細をウェブサイトに開示する場合もあります。

【開示例（対策情報の例）】

☆☆☆☆株式会社 ＞ セキュリティ脆弱性情報 ＞ ○○○○製品

緊急

○○○○製品における××××の脆弱性

公開日 20XX 年 12 月 4 日

最終更新日 20XX 年 12 月 9 日

[19] ソフトウエア製品開発者による脆弱性対策情報の公表マニュアル 第 6 版（IPA）
https://www.ipa.go.jp/security/reports/vuln/nq6ept000000ldxx-att/publication_manual.pdf

■概要

○○○○のバージョン△△以前に××××の脆弱性が存在することが判明しました。この脆弱性を悪用された場合、悪意ある第三者の攻撃により、○○○○が動作しているコンピュータ上で□□□□が実行されてしまう危険性があります。

この問題の影響を受ける○○○○のバージョンを以下に示しますので、以下の修正プログラムを適用してください。

該当製品をご利用の場合、今後被害が拡大するおそれがあるため、至急、修正プログラムをインストールしてください。

（※既に攻撃が確認されている場合） 【重要】本脆弱性を利用した攻撃の発生が既に確認されています。至急、修正プログラ ムを適用して下さい。

■該当製品の確認方法

影響を受ける製品は以下の製品です。

製品名称 ○○○○
該当バージョン
1.5.4 (Windows 版) 以前の全てのバージョン
1.5.4 (Linux 版) 以前の全てのバージョン

使用しているバージョン番号の確認方法は以下の通りです。
1, ○○○○を起動し、「ヘルプ」メニューから「バージョン情報」を選択する。
2. 現れたウィンドウの下記の部分が起動している○○○○のバージョン番号です。

バージョン表示ウィンドウの図 （省略）

■脆弱性の説明

○○○○製品は、ファイルの■■■■のために▽▽▽▽の機能を搭載しています。◎◎◎◎データの一部として提供され▲▲▲▲で配布された▽▽▽▽の機能に、××××の脆弱性が存在するため、外部の第三者からインターネット越しに□□□□を実行される脆弱性が存在します。

※その他の設定および条件

▽▽▽▽の機能が搭載されていないバージョン 1.5.4 以前（Windows 版） を利用している場合、または、この機能が無効化されている場合には、外部の第三者からインターネット越しに□□□□を実行されることはありません。

・CWE-20 不適切な入力確認

■脆弱性がもたらす脅威

システム管理者権限でログインして本ソフトウエアを利用している場合、攻撃が成功すると、悪意のある第三者によってコンピュータを完全に制御されてしまう可能性があります。これにより、悪意のある第三者は、不正プログラムのインストール、データの変更や削除など、システム管理者の権限でコンピュータを任意に操作する可能性があります。

・CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/BS9.8 緊急
・○○製品における技術詳細情報

■対策方法

バージョン 1.5.4.より前の製品を利用されているお客様は、一度製品をアンインストールしてから対策版製品をインストールしてください。バージョン 1.1 以降の製品を利用されているお客様は、修正プログラムをインストールしてください。
各プログラムのインストール方法に関しては同梱の readme.txt を参照してください。

対象製品名称　○○○○
修正プログラムのダウンロード
1.5.5 patch.zip (Windows 版) 20XX.12.4
1.5.5 patch.tgz (Linux 版) 20XX.12.4

・修正プログラムによって置き換えられる設定ファイル
xxxxx.cfg、yyyyy.dif

■回避策

この脆弱性は、次の手順で影響を緩和できる場合があります。
○○○○で使用する管理用ポート番号宛ての通信を、信頼できる IP アドレスのみに限定するよう、ルータ等にてフィルタリング設定を行うことで、攻撃元の範囲を限定することができます。

■関連情報

CVE-20XX-12345678
JVN#12345678 ○○○○製品における××××の脆弱性

■謝辞

□□□の□□□氏よりこの問題をご報告いただき（略）

■更新履歴

20XX.12.4　この脆弱性情報ページを公開しました。
20XX.12.9　脆弱性がもたらす脅威に、システム管理者の権限でコンピュータを任意に操作する際の技術詳細情報を追加しました。

■連絡先

本件に関するお問い合わせはこちら
脆弱性連絡窓口
電話　：03-xxxxx-xxxx (平日 10:00 – 17:00)
メール：example@example.co.jp

13 ソフトウェアの脆弱性に起因したインシデントが発生した場合の対応

意義

ソフトウェアの脆弱性に起因してインシデントが生じる場合、ソフトウェアの製品利用者においては、不正アクセスやマルウェア感染等による情報の漏えいや改ざん、破壊・消失、情報システムの機能停止など、さまざまな事象が起こり得ます。このような事象が及ぼす被害によって製品利用者のみならず、製品開発者の経営や事業に重大な影響がもたらされる可能性があります。

ソフトウェアの脆弱性に対して、セキュリティ修正プログラムの作成・配布や、脆弱性情報の公表等を速やかに行うことにより、製品利用者側でのインシデント発生による被害とその影響範囲を最小限に抑えられるようにすることが重要です。

実施内容

インシデント発生時には、「検知・初動対応」、「報告・公表」、「再発防止と対応改善」の 3 つの段階における対応を実施します。連携が必要となる関係者とその役割については、「附属 主要な関係者・役割表」を参照してください。

実施手順
① 検知・初動対応 下記(1)～(3)の方法により、インシデントを早期把握し、迅速に対応することで、インシデント発生による被害とその影響範囲を最小限に抑えます。 (1) 検知と連絡受付 製品に起因してインシデントが生じた可能性があるとの連絡を製品利用者から受けた場合には、PSIRT に報告します。 また、製品利用者だけではなく、第三者（特にセキュリティベンダ）からの脆弱性の報告のなかで、製品利用者でのインシデントについて記載があるケースもあります。そのような場合にも PSIRT に報告します。 (2) 対応体制の構築 PSIRT は、製品の脆弱性に起因したインシデントであって、製品開発者側での対応が必要となると判断した場合に、速やかに経営者に報告します。 経営者は、速やかにインシデント対応のための体制（情報セキュリティ委員会や緊急対策会議等）を立ち上げ、あらかじめ策定している対応方針に従い、責任者と担当者を定めて、役割分担を明確にします。 (3) 初動対応 当初の報告と関係する製品利用者以外の製品利用者においてもインシデントが生じている懸念がある場合には、初動対応として、外部からの問い合わせを受け付ける際の対応窓口を設置することを検討します。また、インシデントによる被害を受けたソフトウ

ェアの製品利用者等に対して、要員を派遣し、必要となる情報収集や、原因調査の支援を行うことも検討します。

なお、原因調査の支援では、不用意な操作により保全されるべき証拠が失われないようにします。

② 報告・公表

下記(1)～(2)の方法により、ソフトウェアの製品利用者に対して適切に必要となる情報を提供することで、インシデント発生による被害とその影響範囲を最小限に抑えます。

(1) ソフトウェアの製品利用者に対する脆弱性の通知

インシデントによる被害を受けたソフトウェアの製品利用者以外の製品利用者に対して、脆弱性の通知を行います。初動対応の過程等で入手出来ている場合には、いわゆるIoC（Indicator of Compromise）のような脅威指標も併せて提供し、製品利用者において、サイバー攻撃を受けたか否かを調査できるようにすることも重要です。

(2) ウェブサイトやメディアを通じた脆弱性の公表

すべての製品利用者への通知が困難である場合は、ウェブサイトやメディアを通じて脆弱性の概要や対策を公表します。インシデントが生じた製品利用者とは十分に協議したうえで、その脆弱性を悪用する形でインシデントが生じていることにも触れることが重要です。単に修正対策を適用するだけでなく、侵害有無の調査や初期化対応など、通常の脆弱性の対策とは異なる対応が製品利用者に必要となることを示すことが求められます。また、公表によって被害の拡大を招かないよう、時期、内容、対象などを考慮します。

③ 再発防止と対応改善

脆弱性を再発させないために根本原因を分析し、新たな対策の導入、ルールの策定、教育の徹底、体制整備、運用の改善等、抜本的な再発防止策を検討し、実施します。「11 脆弱性の検証（脆弱性情報のトリアージと分析）」の実施手順「③ 根本原因の特定」も参考としてください。

また、インシデントが生じた問題である場合、個別の脆弱性の再発防止だけではなく、製品利用者との連携対応や、PSIRT を中心とした内部報告体制の在り方といった製品インシデントの対応プロセスについても、改善点がないか検討することが重要です。

開示方法

- ソフトウェアの脆弱性に起因したインシデントが発生した場合に、「検知・初動対応」、「報告・公表」、「再発防止と対応改善」の 3 つの段階における対応を行っている点を製品セキュリティポリシー等に含め開示します。

14 製品利用者に対する実施事項の明示

意義

製品セキュリティを維持するためには、製品開発者が脆弱性対処を実施することに加え、利用者に対し、セキュリティを確保するための設定やセキュリティ機能の利用方法等を説明し、正しく利用するよう促すことが重要です。利用者がセキュリティ機能を正しく利用できない場合、セキュリティが維持されていない状態で製品を利用し続けることになり、サイバー攻撃により被害を受けるリスクが高まります。利用者が実施するべき事項と実施しない場合にどんな問題が起こるかを明示することで、利用者がセキュリティ確保の必要性を認識することが期待できます。

実施内容

製品の利用開始時、利用中及び利用終了時に利用者が実施すべき事項をまとめ、利用者に開示します。また、実施しない場合のリスクも明示します。

利用者の実施事項
■ **利用開始前に実施することの明示** ➢ パスワード設定・ネットワーク接続設定等の初期設定を正しく実施すること ➢ 利用開始時に工場出荷時・利用開始時点の製品共通のアカウント・パスワードを変更すること ■ **利用中に実施することの明示** ➢ 製品のアップデート情報や脆弱性対策情報を確認し、必要な対策を実施すること ➢ 製品が提供するセキュリティ機能を利用すること ➢ バックアップや設定内容の記録を行うこと ➢ セキュリティサポートが終了した製品の利用は中止するか、サポート対応中の製品に変更すること ■ **利用終了時に実施することの明示** ➢ 製品を利用しない場合は、ネットワークから切り離すこと ➢ 製品を廃棄する際に登録・保存されているデータを消去すること（初期化等） このような明示については、ソフトウェア製品に付帯して、利用者向けに作成・提供する利用規約やユーザーマニュアルの中で記載します。「安全な製品の使い方」や「利用上の注意」といった文書がある場合には、そのような文書に併せて記載することも一案です。 ライセンス登録や製品登録の過程で製品利用者から連絡先情報を取得しているような場合には、製品利用者に向けて直接伝達することも可能ですが、そのような伝達手段がない場合には、ソフトウェア製品に同梱されるユーザーマニュアルが、利用者への説明責任を果たす上で重要となります。製品のアップデート情報や脆弱性対策情報を確認し、必要な

対策を実施することを始めとして、設定内容の確認や変更などを含め、セキュリティ対処の観点から利用者側に実施してもらいたい事項をリストアップし、それらをユーザーマニュアルに盛り込むとともに、利用者に対して、ユーザーマニュアルの周知徹底を図ります。

上記記載の実施依頼事項を実施しない場合の、利用者が受けるリスクも併せて明示してください。

また、利用者が実施するべき事項を明示した上で、利用者がソフトウェア製品を利用する際のリスクに備え、万が一、インシデントが発生した場合に補償範囲を巡って混乱が生じないよう利用者との契約や取り決め等において、製品開発者の責任範囲を明確化しておくことも考えられます。さらに、このような契約や取り決め等においては、製品開発者の故意や過失が問題となる事項についても明確化しておくことも求められます。

なお、実施事項を明示する先である製品利用者は、「セキュリティを確保するための設定」を行ったり、「セキュリティ機能の利用」を行ったりする者のことを念頭に置いています。消費者や一般向けのソフトウェア製品の場合には、ソフトウェア製品に紙媒体の取扱説明書を同梱したり、ウェブサイト上でマニュアル等を掲載したりすることを通じて、必要な実施事項を伝達することが考えられます。BtoB のソフトウェア製品の場合、導入時に設定を行うのは、利用者本人、利用企業本体ではなく、導入作業や保守運用を担う SI 事業者であることも考えられます。そのような場合、そのような SI 事業者に向けて、必要な実施事項を伝達する必要があります。

また、「利用者が実施するべき事項」を、製品開発者から直接利用者に提示しない形で情報展開する場合もあります。例えば、ソフトウェア製品を調達し、構築・導入を行う SI 事業者を経由して、利用企業に情報展開するような方法も考えられます。また、どのようなセキュリティ機能が搭載されているか、サポート期限がいつまでであるかといった内容は、ソフトウェア製品の商流に応じて、卸売事業者や小売販売事業者が、その販売にあたって顧客に提示することもあります。そのような場合、製品開発者は、卸売事業者や小売販売事業者に対しても、わかりやすく説明することが重要です。

開示方法

- 利用者がセキュリティ機能を正しく利用できるように、利用者自身が実施すること及び実施しなかった場合に発生しうる被害等のリスクについて、製品の取扱説明書や製品情報を掲載しているウェブサイト等で開示します。

V. 廃棄

15 ソフトウェアのサポート終了と廃棄対応

意義

開発者がソフトウェアのサポート終了を決定した場合や、利用者がソフトウェアの使用終了を決定した場合には、当該ソフトウェアがセキュアに廃棄される前に、不正アクセス等により保存された重要なデータが不正に窃取されることがないよう、速やかに安全性の高い方法で廃棄を行うことが重要です。ソフトウェアが適用された機器・システムを廃棄する場合を含め、廃棄予定のソフトウェアが残置されている場合、時間の経過に伴って管理が行き届かなくなり、サイバー攻撃により被害を受けるリスクが高まります。

実施内容

ソフトウェアのセキュアな廃棄について、契約時に開発者と利用者の間でサポート終了時期の事前通知や利用者への対応依頼事項等に関する取り決めを行います。また、開発者において、ソフトウェアのサポート終了を決定した場合には、利用者が当該ソフトウェアのセキュアな廃棄に必要な情報を継続的に利用できるようにします。連携が必要となる関係者とその役割については、「附属 主要な関係者・役割表」を参照してください。

実施手順

① 開発者がソフトウェアのサポートを終了する際の対応

開発者は、ソフトウェアを利用し続けることによるリスクについて、十分に考慮しつつ、ソフトウェアのサポート終了を決定します。サポート終了を決定する要因には、自組織の運用体制の維持管理状況などのほかにも、製品に組み込まれたコンポーネント等の構成要素のサポート終了も挙げられます。構成要素のサポート終了期限も把握するにあたっては、ソフトウェア部品表（SBOM（Software Bill of Materials））を活用することも一案です。

開発者がソフトウェアのサポート終了を決定した場合には、契約時に取り決めた期日までに、利用者に対して、サポート終了時期等を事前通知するとともに、当該ソフトウェアのセキュアな廃棄に必要な情報を継続的に利用できるようにします。

また、ソフトウェアのサポート終了後においても、利用者が当該ソフトウェアの利用を継続している場合は、製品開発者として取り得る対応責任について十分に考慮しつつ、過去に提供したセキュリティ修正プログラムの継続的な提供や、延長サポートサービスの提供などの実施についても、検討することが期待されます。

② 利用者がソフトウェアの利用を終了する際の対応

利用者がソフトウェアの使用終了を決定した場合には、利用者に対して、利用規約や契約時の取り決めに従い、当該ソフトウェアに登録・保存されているデータの消去がセキュアに行われるよう依頼します。またその際、開発者によりデータ消去機能が提供されていれば、それを活用することを依頼します。

なお、データの消去は、利用者において、データの機密性のレベルやデータが保存されている環境に応じた適切な消去方法が選択され、消去が確実に履行されるよう、依頼してください。このような消去方法としては、下記(1)～(2)の方法について考慮します。

(1) ソフトウェアが適用された機器の記憶媒体に保存されているデータを消去する場合

高いレベルの機密性が求められるデータについては、記憶媒体の分解・粉砕・溶解・焼却・細断などによって物理的に破壊し、確実に復元不可能とします。また、そのような破壊を委託事業者に依頼する場合は、当該破壊の完了証明書により、消去が確実に履行されたことを確認します。それ以外のデータについては、ソフトウェアからアクセス可能なすべてのストレージ領域を、データ消去ソフトウェア等を用いて上書きして消去する上書き消去や、磁気的な消去、ブロック消去のうち、いずれかの消去を行い、確実に復元不可能とします。

(2) ソフトウェアと連携するクラウド環境に保存されているデータを消去する場合

共用を前提とするクラウド環境に保存されているデータについては、記録媒体の物理的な破壊や磁気的な消去ができないため、データを暗号化し、暗号化したデータを消去するとともに、データの復号を不可能とするため、鍵自体を消去する暗号化消去を行い、確実に復元不可能とします。

このようなデータの消去に関わる完了証明書の発行については、一般社団法人ソフトウェア協会における「データ適正消去実行証明書発行事業[20]」の情報を参考にしてください。

開示方法

- 契約時に開発者と利用者の間で、サポート終了時期の事前通知や利用者への対応依頼事項等に関する取り決めを行っている点を製品セキュリティポリシー等に含め開示します。

[20] データ適正消去実行証明書発行事業（一般社団法人ソフトウェア協会）
https://www.saj.or.jp/data_erase

用語集

■ CSR

Corporate Social Responsibility の略称であり、企業が社会に与える影響について責任を持ち、社会の持続的発展のために貢献すべきとする考え方。また、そのような考え方に基づいて実践される諸活動。

■ PSIRT

Product Security Incident Response Team の略称であり、組織が提供する製品の脆弱性に起因するリスクに対応するための組織内機能。

■ 脅威指標

脆弱性対策情報を提供している製品開発者が独自で定め公表している脅威に関する指標。

■ サプライチェーン

複数の開発者間でリンクされたリソース・プロセスで、製品とサービスについて、調達に始まり設計・開発・製造・加工・販売及び購入者への配送に至る一連の流れ。

■ 脆弱性

ソフトウェア製品やウェブアプリケーション等におけるセキュリティ上の問題箇所。コンピュータへの不正アクセスやコンピュータウイルス等により、この問題の箇所が攻撃されることで、そのソフトウェア製品やウェブアプリケーションの本来の機能や性能を損なう原因となり得るもの。

■ セキュリティポリシー

トップマネジメントによって正式に表明された組織のセキュリティに係る意図や方向付け及びそのような意図や方向付けに基づいてセキュリティ対策を行うために組織が定めた規定。

■ ハードコーディング

ソフトウェア開発の際に、特定の動作環境を確定し、その環境を前提とした処理やデータをソースコードの中に直に記述すること。

■ パブリックモニタリング

ソフトウェアの脆弱性や攻撃手法等に関して、公開情報から継続的に情報収集すること。

■ リスク分析

リスクの特質を理解し、リスクレベル（ある事象の結果とその起こりやすさとの組合せとして表現される、リスクの大きさ）を決定するプロセス。

附属：主要な関係者・役割表

凡例	説明
○	当該部門/関係者の関与が必須の場合
△	当該部門/関係者の関与が状況次第の場合

部門/組織		計画			要件定義・設計・開発					供給・配布	運用					廃棄	役割の説明
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
自組織の関連部門	経営層	○	○	○							△	△	△	○			・製品セキュリティポリシー、脆弱性対処計画、インシデント対応計画に関して、策定時・変更時の承認・決定を行う。 ・必要な予算の確保、及び、体制構築のため各部門へ指示を行う。 ・脆弱性対策情報の公表は、製品ブランドイメージや販売計画に影響を及ぼす場合があるため、公表内容の承認を行う場合がある。 ・外部から脆弱性の報告を受付した場合は、不適切な対応をすると未修正の脆弱性情報が公開されるなどのインシデントに繋がるため、脆弱性の対処状況を把握する。
	法務部門	○	○	○									△		△		・製品セキュリティポリシー、脆弱性対処計画、インシデント対応計画に関して、策定時・変更時の内容確認を行う。 ・脆弱性が悪用され世の中に被害を及ぼした場合、製品ブランドイメージや販売計画に影響を及ぼす場合があるため、脆弱性対策情報の公表内容の確認を行う場合がある。 ・セキュリティ対処をアウトソースしている場合、契約締結時に契約内容の内容確認を行う。 ・自社の脆弱性対応や外部組織の対応に関して、契約内容との齟齬が発生した場合に、確認や折衝を行う場合がある。
	広報部門		○	○									○	○	○		・対外的に情報を開示、公表を行う場合には内容の確認を行う。 特に製品セキュリティポリシー、脆弱性対処計画、インシデント対応計画、脆弱性対策情報の公表内容など
	リスクマネジメント部門	○	○	○							△	△	○	○	○		・製品セキュリティポリシー、脆弱性対処計画、インシデント対応計画の策定・変更を行う際に、リスクマネジメントの観点で確認を行う。 ・関係部門において手順書に則った対応が実施されているか定期的な監査を行う。 ・脆弱性が悪用され世の中に被害を及ぼした場合、製品ブランドイメージや販売計画に影響を及ぼす場合があるため、脆弱性対策情報の公表内容の確認を行う。
	製品管理部門	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	・製品管理部門は、全工程において状況を把握、管理を行う。 ・製品セキュリティポリシー、脆弱性対処計画、インシデント対応計画および手順書の策定を行う。 ・手順書の周知・教育・訓練を主導する。 ・セキュリティ対処をアウトソースする場合、委託先の選定基準の策定を行う。 ・外部の情報共有組織との連携や所轄官庁への報告を行う。
	調達部門	○			○	○			○		△		△				・設計/開発時に、構成要素(ソフトウェアコンポーネントやライブラリ等)の比較検討や調達を行い、構成要素に関する脆弱性情報の収集と構成管理表の管理を行う。 ・セキュリティ対処をアウトソースする場合、委託先との調整や管理を行う。 ・製品出荷後に、構成要素(ソフトウェアコンポーネントやライブラリ等)に脆弱性が発覚した際に、サプライヤーへ問い合わせや調整を実施する場合がある。
	設計/開発部門				○	○	○	△	○		○	○	○	△			・製品の設計/開発を行う。 ・脆弱性検査の結果、脆弱性が検出された場合は、修正を行う。 ・製品出荷後に脆弱性の改修を行う。
	品質検査部門							○			○	○	○	△			・開発時の脆弱性検査、製品出荷後に機能改修や脆弱性修正パッチ公開前に脆弱性検査等を行う。
	情報収集部門										○						・出荷した製品の脆弱性情報が開示されてないかSNSやセキュリティコミュニティを監視する。 ・調達部門から構成管理表を引き継ぎ、構成要素の脆弱性監視を行う。
	製品サポート窓口										○		○				・製品出荷後、製品利用者からの問い合わせ受付を行う。脆弱性に関する報告を受付ける場合もある。 ・自社製品の脆弱性対策情報を公開する際、代理店などへ事前に説明などを行う場合がある。 ・自社製品の脆弱性対策情報を公開後、利用者から対策適用に関して問い合わせ受付を行う。
	脆弱性受付窓口										○	○					脆弱性報告を受付、及び、報告者とのコミュニケーションを行う。

部門/組織		計画			要件定義・設計・開発					供給・配布	運用					廃棄	役割の説明
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
外部関係者	サプライヤー	△		△	△	○	△				○						構成要素(コンポーネントやライブラリ)の供給元。
	セキュリティベンダ	△		△	△	△	△	△	△		△			△			脆弱性検査など高い専門性が求められる対処をアウトソースする場合がある。
	開発の外部委託先	△	△	△	△	△	△	△	△								製品開発者から委託を受け、ソフトウェア開発の業務の一部を実施する場合がある。
	SI事業者												○		○		製品利用者から委託を受け、製品の機能設定や脆弱性修正の適用などの作業を行う場合がある。
	情報共有組織			△													日本シーサート協議会(NCA)、Software ISACなどに加盟することで、製品セキュリティに関する組織体制やセキュリティ対処に関する情報共有や意見交換を行う。
	脆弱性情報の報告者										○	○					脆弱性の報告は、IPAおよびJPCERT/CCからされる場合がある。
	代理店														○		製品に搭載されているセキュリティ機能の説明を製品利用者に実施する場合がある。
	製品利用者														○	○	製品利用者が購入した製品に対して、セキュリティ設定やセキュリティ機能の利用、廃棄を行う。

別紙：製品開発者向けガイド チェックリスト

製品開発者向けガイドの項目に沿って作成したチェックリストです。

未実施	/15	実施済 レベルなし	/9	実施済 （レベル 1）	/6	実施済 （レベル 2）	/6	実施済 （レベル 3）	/4

カテゴリ	No	項目			チェック	備考
I. 計画	1	ソフトウェア開発ライフサイクル全体を通じた人材・プロセス・技術の整備	レベル 1	組織内の関係者が、SDLC におけるセキュリティを高めるために必要な役割と責務を定義します。		
			レベル 2	SDLC 全体が網羅されるよう、組織内の関係者が、SDLC 全体を通じて、セキュリティを高めるために必要な役割と責務を定義し、それらを実践できるよう、人材、プロセス、技術を整備します。		
			レベル 3	外部委託の活用や組織間の連携強化を図りつつ、SDLC 全体が網羅されるよう、組織内外の関係者が、SDLC 全体を通じて、セキュリティを高めるために必要な役割と責務を定義し、それらを実践できるよう、人材、プロセス、技術を整備します。		
	2	セキュアなソフトウェア開発を実践するための方針・体制	レベル 1	製品セキュリティポリシーを策定し、当該ポリシーに沿ってセキュリティ要件を定義します。		
			レベル 2	製品セキュリティポリシーを策定し、当該ポリシーに沿ってセキュリティ要件を定義するとともに、開発するソフトウェアがセキュリティ要件を満たしていることを確認します。		
			レベル 3	製品セキュリティポリシーを策定し、当該ポリシーに沿ってセキュリティ要件を定義するとともに、セキュリティ確認基準の策定や、SDLC 全体にわたってセキュリティ確認基準への準拠を維持するというガバナンスの確立を通じて、開発するソフトウェアがセキュリティ要件を満たしていることを監査により確認します。		
	3	脆弱性対処やインシデント対応を円滑に実施するための方針・体制	レベル 1	PSIRT を構築・維持し、PSIRT が製品セキュリティポリシーに従ってセキュリティ対処を実施するために必要な機能を定義します。		
			レベル 2	製品セキュリティポリシーを策定し、当該ポリシーに沿ってセキュリティ要件を定義するとともに、PSIRT の活動上必要となる体制やプロセスを整備します。		
			レベル 3	PSIRT を構築・維持し、PSIRT が製品セキュリティポリシーに従ってセキュリティ対処を実施するために必要な機能を定義するとと		

				もに、脆弱性対処計画の策定やステークホルダー間の関係強化を通じて、PSIRT の活動上実効性のある体制やプロセスを整備します。		
II. 要件定義・設計・開発	4	セキュリティを確保するための設計	–			
	5	既知の脆弱性解消	レベル 1	可能な範囲で構成管理を実施します。製品開発者及び公的機関、セキュリティコミュニティ等から脆弱性情報を収集し、深刻度の高い脆弱性から対処を行います。深刻度については、共通脆弱性評価システム（CVSS） や脅威指標等を参考にしてください。		
			レベル 2	すべての構成管理を実施します。製品開発者及び公的機関、セキュリティコミュニティ等から脆弱性情報を収集し、深刻度の高い脆弱性から対処を行います。深刻度については、共通脆弱性評価システム（CVSS）や脅威指標等を参考にしてください。		
			レベル 3	すべての構成管理を実施します。製品開発者及び公的機関、セキュリティコミュニティ等から脆弱性情報を収集し、製品に関連した脆弱性情報に関するリスク分析結果に応じて対処を行います。		
	6	セキュアコーディング・セキュアビルド	レベル 1	組織のコーディング規約や、使用するコンパイラ、インタプリタ、及びビルドツールを定めて、コードの生成・ビルドを実施します。		
			レベル 2	組織のコーディング規約や、使用するコンパイラ、インタプリタ、及びビルドツールを定めて、コードの生成・ビルドを実施し、別に定めたレビュー担当者がレビューします。		
	7	開発時の脆弱性検査	–			
	8	開発に使用する環境及びツールのセキュリティ確保	–			
III. 供給・配布	9	ソフトウェア製品のセキュアな配布	–			

Ⅳ. 運用	10	脆弱性の発見	−			
	11	脆弱性の検証（脆弱性情報のトリアージと分析）	レベル 1	報告された脆弱性情報について、自組織にとって対処が必要な脆弱性であるかの判断を行うために、脆弱性の認定・再現を実施します。		
			レベル 2	レベル 1 を満たしたうえで、脆弱性の再発を防ぐために、脆弱性を分析して根本原因を特定し、改善を図ります。		
	12	脆弱性の修正対策と対策情報の公表	−			
	13	ソフトウェアの脆弱性に起因したインシデントが発生した場合の対応	−			
	14	製品利用者に対する実施事項の明示	−			
Ⅴ. 廃棄	15	ソフトウェアのサポート終了と廃棄対応	−			

製品開発者向けガイド

2026 年 3 月　第 1 版発行

独立行政法人情報処理推進機構

〒113-6591

東京都文京区本駒込２丁目２８番８号　文京グリーンコートセンターオフィス１６階

URL　https://www.ipa.go.jp/security/

電話　03-5978-7527　　FAX　03-5978-7552