2021 年度～2022 年度調査報告書

戦略的イノベーション創造プログラム（ＳＩＰ）第２期／ＩｏＴ社会に対応したサイバー・フィジカル・セキュリティ/ＩｏＴ社会に対応したサイバー・フィジカル・セキュリティに係るＯＳＳの技術検証のあり方等に関する調査

2022 年 7 月

国立研究開発法人新エネルギー・産業技術総合開発機構

委託先　　一般社団法人重要生活機器連携セキュリティ協議会

目次

１．研究開発の成果と達成状況 4
1.1　要約 4
1.1.1　和文要約 4
1.1.2　英文要約 6
1.2　本文 9
1.2.1　用語集 9
1.2.2 ＜調査項目１＞各業界において、OSS を安全に活用する取組の調査 13
1.2.2.1　調査の背景、目的 13
1.2.2.2　調査の方針、前提 13
1.2.2.3　業界における OSS を安全に活用する取組の実態調査結果 15
1.2.2.3.1 業界別の OSS 利用状況、取組の調査結果（一覧） 15
1.2.2.3.2　業界における取組の事例、業界分野別の特徴 28
1.2.2.3.3　OSS の利用や管理、セキュリティ検証における課題 42
1.2.2.3.4　OSS（ソフトウェア）利用上の課題及び、認証制度への要望の整理 .. 45
1.2.3 ＜調査項目２＞OSS の安全な活用の技術検証活動のあるべき姿及び技術検証項目、検証のためのルール調査 48
1.2.3.1　調査の背景、目的 48
1.2.3.2　OSS 起因を含むソフトウェアのインシデント事例調査 48
1.2.3.2.1　インシデント事例の調査 48
1.2.3.2.2　インシデント事例に対する検証手法の調査 55
1.2.3.2.3　ソフトウェアコンポーネントに求められる検証手法 65
1.2.3.3　国内外の関連ガイドライン調査結果 70
1.2.3.3.1　調査対象のガイドライン一覧 70
1.2.3.3.2　ガイドラインからまとめた体系的なセキュリティプラクティス 73
1.2.3.3.3　技術検証に関する項目 95
1.2.3.3.4　組織マネジメントに関する要求事項 98
1.2.3.4　OSS 検証ツールの実態調査 103
1.2.3.4.1　OSS 検証ツールの調査結果一覧 103
1.2.3.4.2　各 OSS 検証ツールの特徴 107
1.2.3.5　技術検証項目と実施ルール 109
1.2.3.5.1　必要な技術検証項目 109
1.2.3.5.2　組織マネジメントにおける対策 128
1.2.4 ＜調査項目３＞検証のためのルール作りや認証等を行う検証機関のあり方調査 134

1.2.4.1　調査の背景、目的 134
1.2.4.2　検証機関の認定基準の調査 134
1.2.4.2.1　ISO/IEC17065：2012 に定義された認証を行う機関への要求事項 135
1.2.4.2.2　ISO/IEC17025：2017 に定義された試験所・校正機関に関する要求事項 135
1.2.4.2.3　IT セキュリティ評価及び認証制度における要求事項 136
1.2.4.2.4　参考情報）システム・ソフトウェア品質標準 SQuaRE シリーズ 137
1.2.4.2.5　ソフトウェア（OSS）に対する認証機関、検証機関への要求事項の考察 139
1.2.4.3　認証制度の調査結果 142
1.2.4.3.1　既存の認証制度の調査結果 142
1.2.4.3.2　認証制度の調査結果の整理 177
1.2.4.4　実現可能かつ実効的な認証制度や検証機関のプロセス・ルール提言 179
1.2.4.4.1　認証制度の開始までに必要とされる段階 179
1.2.4.4.2　認証制度や検証機関のプロセス・ルール提言 181
1.2.4.5　認証及び検証のプロセス・ルール 186
1.2.4.6　認証制度の普及啓発 190
1.2.4.7　参考情報）民間主導による認証制度の対応状況 191
２．研究発表・講演、文献、特許等の状況 194

１．研究開発の成果と達成状況

1.1 要約

1.1.1 和文要約

OSS を含むソフトウェアを安全に活用する取り組みについて、各業界の企業にアンケート及びヒアリング調査を行った。その結果、分野を問わず、87%の企業で OSS（商用サポートなし、無償）を利用していることが分かった。OSS や他社製ソフトウェアを利用しておらず、今後も利用予定のない企業は 8%であり、サプライチェーンの観点から、ソフトウェアセキュリティの認証制度を確立する意義があると言える。ソフトウェアのセキュリティ検証の実施状況について、自社開発ソフトウェアに対しては、必ず何らかの検証はなされており、その中でも静的解析やコードレビューの比率が高いことが分かった。OSS や他社製のソフトウェア単体に対しては検証の実施比率は少し下がり、その中でなされている検証はペネトレーションテストや動的解析の比率が高く、ソースコードを対象とせず、バイナリデータを対象とした検証を実施している比率が高かった。そして、組織上の管理状況について、特に社内基準がなく、開発担当者や社内調達者の裁量で選定しているところが多かった。ただし、開発チーム内のレビューなどで利用実績や脆弱性への対応状況、ライセンスの状況などを見て判断していることが確認された。認証制度への要望については、自社または委託先で適合性評価を行い、第三者認証機関が認証する方式が望まれていることが分かった。さらに、取得にかかる費用が安く、国や業界団体によって取得が推奨されていることがよいとの声が多かった。

OSS を含むソフトウェアの安全な活用の技術検証活動の手法を導くため、OSS に起因するものを含むソフトウェアのインシデント事例を調査した。さらに参考文献をもとに、原因を事前に検出するための検証手法及びその脆弱性が公開された後にその脆弱性を持つコンポーネントがないかを検出するための手法を整理した。その結果、脆弱性が公開される前に脆弱性がないかを検証する手法としては、あくまで可能性ではあるが、ソースコード解析、既知脆弱性の診断、ファジングなどが挙げられた。脆弱性が公開された後では、脆弱性のあるソフトウェア情報と製品の SBOM の照合が有効であることが分かった。

技術検証活動の手法を導くためのもう一つのアプローチとして、国内外で現在公開されているセキュリティガイドラインを調査し、そこで推奨されている技術的な検証に関わるセキュリティプラクティスと、組織マネジメント上の管理項目に関わるセキュリティプラクティスを抜き出し、類似プラクティスをまとめて表に整理した。ソフトウェア開発および IoT サプライチェーンに関連する欧米の 4 ガイドラインからすべてのプラクティス、さらに包括的なセキュリティフレームワークである 2 ガイドラインからサプライチェーンのセキュリティに関わるプラクティスを抽出した。技術的な検証に関わるセキュリティプラクティスは、ソフトウェアアップデートや実行コードの改ざん検知機能といった機能に関するもの、脅威分析やセキュアコーディングといったセキュア開発に関するもの、静的解析や動的解析といったテストに関するものに分けられた。組織マネジメント上の管理項目に関

わるセキュリティプラクティスは、製品のセキュリティ要件定義や開発したソフトウェアの構成管理といった組織・体制に関するもの、セキュリティに取り組んでいるサプライヤの採用やサードパーティソフトウェアの構成管理といったサプライチェーンに関するもの、継続的な脆弱性情報の収集や一般向けへのセキュリティ/脆弱性情報の公開といった脆弱性対応に関するものに分けられた。

次に OSS およびサードパーティのソフトウェアコンポーネントを対象に、セキュリティ上の安全性を検証するツールを調査した。調査対象とした国内でサービス展開されている商用の 3 ツールを対象として調査した結果、調査対象としたツールすべてが、ソフトウェアのインシデント事例の調査結果や、セキュリティガイドラインのプラクティスで求められる SBOM の表示/作成機能や脆弱性診断機能を持つことが分かった。

前述のセキュリティガイドライン調査で整理したセキュリティプラクティスから、ソフトウェアに関わるものを抜き出して技術検証項目としてまとめた。それぞれ想定される検証の実施内容例も併せて記載した。開発上流工程における脅威分析やその対策の検討、実装における実行コードの改ざん検知機能、強力な暗号技術の導入やセキュアコーディングの実施、運用中におけるソフトウェアのアップデートに至るまで、ここでまとめた技術検証項目はソフトウェアライフサイクル全体をカバーしている。

認証制度を提案するにあたっての前提情報として、認証機関および検証機関の国際規格の認定基準を調査した。認証機関への要求事項は ISO/IEC 17065：2012 に、検証機関への要求事項は ISO/IEC 17025：2017 に規定され、どちらも公平性の担保や組織のマネジメント、資源の確保といった要求事項が定義されている。さらにこの調査結果と企業にヒアリングした結果とを比較して考察を行った。その結果、認証機関に対しては、高度なマネジメントシステムの保持の要求事項が、認証コストを下げられるような緩和が求められると考えた。また、検証機関に対しては、いくつか挙げられるが、例えば、自社での検証（自己適合性評価）を可能となるように公平性の担保の要求事項に緩和が求められると考えた。

さらに、国内でサービスが行われている認証制度を調査した。国際規格による認証制度と独自標準の認証制度から 3 制度を選んで調査対象とし、認証対象や認証制度の特徴、認証スキームといった制度の調査結果をまとめた。また、その制度内容とヒアリングによる要望との比較考察を行った。ヒアリングによる要望では認証にかかる費用や負担をなるべく減らしつつ、適度なハードルで自社や第三者機関への委託で検証を実施できる制度が望まれていたが、国際規格による認証制度は厳密であるが故、要望とはマッチしていなかった。一方、国や業界団体によって取得が推奨されているなど、認証のコストを製品に付加しても顧客や消費者に説明しやすい制度が望まれているが、調査対象の制度は政府調達の要件になっていたり法令で対応が求められていたりするなど顧客や消費者に説明しやすい制度となっており要望とマッチしていた。

最後に、各調査結果及びヒアリング結果の要望事項を踏まえた、実現可能かつ実効的な認証制度や検証機関のプロセス・ルールとして、2 種類の認証スキームを提言している。一つ

は認定された検証機関が検証を行う制度であり、もう一つは自社での検証を可能とする制度である。前者では、ISO 規格に準拠した検証機関を認定機関が認定し、検証は認定された検証機関のみが実施可能となる。後者では、認証に対応する資格制度を認証機関が整備し、必要資格を備えた要員を有する組織であれば、自社検証、第三者検証のいずれも実施可能とするものであり、この点でより要望を取り入れた制度となっている。そのほかの観点でも、認証プロセスは複雑化せずに明確なフローとすることや、申請にあたって機密情報の取り扱いに配慮すること、認証にかかる費用を製品やサービスの単価に応じた妥当な額に設定することなど要望に沿った認証制度を提言している。

1.1.2 英文要約

We conducted questionnaires and interviews with companies in various industries regarding their efforts to utilize software, including open-source software （OSS） securely. The results showed that 87% of companies in all sectors use OSS （no commercial support, free of charge）, and only one company does not use OSS or other third-party software, which indicates the significance of establishing a certification system for software security from the supply chain perspective. Any company conducts some form of verification for in-house developed software, with a high percentage of static analysis and code review. They conduct less security verification for OSS and other third-party software alone, and the type of verification conducted are penetration testing and dynamic analysis, which do not require source code. Regarding the status of organizational management, there were no specific internal criteria for OSS/third-party software adoption, and in many cases, the development staff or internal procurers decided which should be adopted. It was confirmed that decisions were made based on usage, vulnerability response status, license conditions, and other factors through reviews within the development team. As for requests for certification systems, it was found that a scheme was desired in which conformity assessment is conducted by the company or contractor and certified by a third-party certification organization. They also desired that the cost of certification should be low and that certification should be recommended by the government or industry associations.

To derive a methodology for technical verification activities for secure use of software including OSS, we investigated software incidents including those caused by OSS. Furthermore, based on the references, we organized the verification methods for possibly detecting the cause of the vulnerability in advance and the methods for detecting components with a vulnerability after the vulnerability is disclosed to the public. As a result, the methods for detecting a vulnerability, if possible, before the vulnerability is disclosed included source code analysis, vulnerability assessment, and fuzzing. After the vulnerability is disclosed, matching vulnerable software information with the product's SBOM was found to be effective.

As another approach to deriving a methodology of technical verification activities, we surveyed current security guidelines published in Japan and overseas, extracted security practices related to technical verification recommended in the guidelines as well as security practices related to organizational management control items, and organized them in tables, summarizing similar practices. We extracted all practices from the four European and US guidelines related to software development and the IoT supply chain, as well as practices related to supply chain security from two guidelines that are comprehensive security frameworks. Security practices related to technical verification were divided into those related to functions such as software updates and executable code tamper detection, those related to secure development such as threat analysis and secure coding, and those related to testing such as static analysis and dynamic analysis. The security practices related to organizational management items included those related to organization and structure such as defining product security requirements and managing the configuration of developed software, those related to the supply chain such as adopting security-conscious suppliers and managing the configuration of third-party software, and those related to ongoing vulnerability management.

Next, we surveyed tools to verify the security of OSS and third-party software components. We surveyed three commercial tools that are in service in Japan and found that all of the tools surveyed could display/create SBOMs and assess vulnerabilities as required by software incident case findings and security guideline practices.

From the security practices organized in the aforementioned security guideline survey, we extracted and summarized those related to software as technical items for verification. Examples of the practices of each verification are also listed. The technical verification items summarized here cover the entire software lifecycle, from threat analysis and consideration of countermeasures in the upper process of the development phase, through tamper detection of executable code in implementation, the introduction of strong cryptography, and secure coding, and software updates during operation.

As prerequisite information for proposing a certification system, we investigated the accreditation criteria of international standards for certification bodies and verification bodies. Requirements for certification bodies are specified in ISO/IEC 17065：2012, and requirements for verification bodies are specified in ISO/IEC 17025：2017, both of which define requirements such as assurance of impartiality, and organizational management, and resources. Further discussion was made by comparing the results of this survey with the results of interviews conducted with companies. As a result, we considered that the requirements for certification bodies to maintain advanced

management systems should be eased so that the cost of certification can be lowered. For verification bodies, the requirements for assurance of impartiality should be eased to allow in-house verification （self-conformity assessment）, for example.

We also surveyed certification systems in service in Japan. We selected three systems from certification systems based on international standards and a certification system based on original standards, and we summarized the results of the survey in terms of the target of certification, characteristics of the certification systems, and certification schemes. Furthermore, we compared the certification systems to the requests made in the interviews. The interviewees desired a system that would allow them to reduce the cost and burden of certification as much as possible, allowing them to implement verification by themselves or by entrusting it to a third-party organization with a moderate level of hurdles. The certification systems based on international standards, however, did not match the desire because of their strictness. On the other hand, interviewees also desired a system that is easy to explain to customers and consumers even if the cost of certification is added to the product, such as a system recommended by the government or an industry organization, which the certification systems are based on international standards matched because certification is mandated for government procurement or required by law.

Finally, we propose two types of certification schemes as feasible and effective certification systems and processes/rules for verification bodies, based on the results of each survey and the requests from the interview results. One is a system in which accredited verification bodies conduct verification, and the other is a system that allows in-house verification. In the former, the accreditation body accredits verification bodies that comply with ISO standards, and only accredited verification bodies can perform verification. In the latter system, the certification body establishes a qualification system for certification, and any organization that has personnel with the necessary qualifications can conduct both in-house and third-party verification, making this system more responsive to the desire. In addition, we recommend a certification system that meets the requirements for a straightforward and uncomplicated certification process flow, consideration for the handling of confidential information when applying for certification, and a reasonable certification fee by the unit price of the product or service.

1.2 本文

本報告書に記載した会社名、製品名などは、一般に各社の登録商標または商標となる。

1.2.1 用語集

用語	説明
BSD（Berkeley Software Distribution）	カリフォルニア大学バークレー校（UCB）の開発者グループが開発・配布していた、UNIX 系 OS および関連ソフトウェア群。多くの商用あるいはフリーの派生 OS を生み出し、「BSD 系 OS」と総称される。
CAPEC（Common Attack Pattern Enumeration and Classification）	セキュリティ攻撃パターンを網羅的に分類・カタログ化したものである。2007 年に米国国土安全保障省によってリリースされ、現在は Mitre Corporation によって管理される。
CCRA（Common Criteria Recognition Arrangement）	CC 承認アレンジメント。Common Criteria Recognition Arrangement の略。国内に認証制度をもつ認証国は 17 カ国、認証制度は持たないが認証された製品を受け入れる受入国は 14 カ国（2022 年 6 月時点）
CERT	Computer Emergency Response Team の略称。脆弱性や不正アクセスなどのセキュリティインシデントに対応するチーム、組織。
CI（Continuous Integration）/ CD（Continuous Delivery）	CI/CD は「Continuous Integration（継続的インテグレーション）/ Continuous Delivery（継続的デリバリー）」の略称のこと。ソフトウェアの変更を定期的にテストし、自動で本番環境に適用できるような状態にしておく開発手法を指す。
CPE（Common Platform Enumeration）	情報システムを構成する、ハードウェア、ソフトウェアなどを識別するための共通の名称基準
CSIRT（Computer Security Incident Response Team）	情報システムにおけるセキュリティインシデント対応を行う組織のこと。コンピュータセキュリティインシデントに関する報告を受け取り、調査し、対応活動を行う。
CVSS（Common Vulnerability Scoring System）	FIRST （Forum of Incident Response and Security Teams） が公開するリスク評価手法であり、脆弱性の深刻度を同一の基準の下で定量的に比較できる。
EO 14028	2021 年 5 月 12 日に出された米国の国家のサイバーセキュリティの向上に関する大統領令で、この中で NIST に対して、消費者向けの IoT や消費者向けソフトウェアについてのラベリングプログラムのための基準の明確化などを求めている。
GPL（GNU General Public License）	ソフトウェアの利用許諾条件などを定めたライセンスの一種。GNU GP とも呼称され、GNU プロジェクトのために作成されたフリーソフトウェアのライセンスを指す。

KVS（Key-Value Store）	データ管理システムの種類の一つで、保存したいデータ（value：値）に対し、対応する一意の標識（key：キー）を設定し、これらをペアで格納する方式。
NVD（National Vulnerability Database）	米国 NIST が管理する脆弱性情報の公開データベース
ODM（Original Design Manufacturing）	委託者のブランドとして、設計、生産までを委託して製品を製造すること。
OEM（Original Equipment Manufacturing）	委託者のブランドとして、生産を委託し、製品を製造すること。
PoC（Proof of Concept）	新しい概念や理論、原理などの実現可能性や、それによって得られる効果などについて検証すること。（概念実証）
PSIRT（Product Security Incident Response Team）	製造または販売する製品に含まれている脆弱性やセキュリティインシデントに対応するために活動する組織のこと。
RDBMS（Relational DataBase Management System）	「リレーショナル・データベース・マネジメント・システム」の略称。リレーショナルデータベースを管理するためのソフトウェアのこと。
RTOS（Real Time Operating System）	リアルタイムシステムのためのオペレーティングシステム （OS） のこと。汎用 OS とは異なり、常に厳格な時間制約のもとで反復タスクが実行されるという特徴がある。
SBOM（Software Bill Of Materials）	特定のソフトウェアに含まれるコンポーネント、ライセンスの種類、依存関係を一覧化したもの。（ソフトウェア部品表）
SDK（Software Development Kit）	アプリケーション開発に必要なプログラム、API、サンプルコード、技術文書などをパッケージにしたもの。（ソフトウェア開発キット）
SDLC	Software Development Life Cycle の略、ソフトウェア開発ライフサイクルのこと。ソフトウェア開発において、計画やコーディング、テスト、運用といったソフトウェアのライフサイクルにわたって実施すべきプロセスを規定するフレームワーク。
SOC（Security Operation Center）	サイバー攻撃の検知や分析を行い、その対応策の立案やアドバイスを専門とする組織のこと。
TOE	Target Of Evaluation。評価対象のこと。
インシデント（Incident）	望まない単独若しくは一連の情報セキュリティ事象，又は予期しない単独若しくは一連の情報セキュリティ事象であって、事業運営を危うくする確率及び情報セキュリティを脅かす確率が高いもの。（情報セキュリティインシデント）

エドテック（EdTech）	Education（教育）と Technology（技術）を組み合わせた造語で、テクノロジーを用いて教育を支援する仕組みやサービスのこと。
監査証跡	情報システムにおいて、誰が、いつ、何を行ったかなどを記録したデータのこと。監査やインシデント発生時の調査における証拠となる。
脅威インテリジェンス	攻撃のメカニズムや攻撃者の動機、防御方法などを整理した情報。
サイバーセキュリティ	電子データの漏えい・改ざん等や、期待されていた機器、IT システム、制御システム等の機能が果たされないといった不具合が生じないようにすること。
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、調達にはじまり設計・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。
真正性	対象が主張する通りのものであること。改ざんやなりすましがされておらず、正しく本物であること。
脆弱性	一つ以上の脅威によって付け込まれる可能性のある、資産又は管理策の弱点。
脆弱性検証	脆弱性の存在を確認するアクティブなセキュリティ検証手法。脆弱性を洗い出すことを目的とする。
脆弱性スキャン	ネットワークを通してサーバやルータ、ファイアウォールなどの機器のソフトウェアの脆弱性を発見する手法。エージェントを検証対象にインストールすることで、内部からスキャンする方法もある
静的解析	ソフトウェア解析手法のうち、プログラムを実行せずに解析、検証を行うもの。
セキュアコーディング	攻撃者やマルウェアなどからの攻撃に耐えられるような、脆弱性を排除した堅牢なプログラムを書くこと。
セキュアブート	システムのブート時に、信頼できるソフトウェアのみ実行を許可する方式。
セキュリティ検証	機器、システム、組織における脅威に対するセキュリティ対策の妥当性や脆弱性の有無を確認する手法。本手引きでは、特に機器に対するセキュリティ検証について記載している。
セキュリティ・バイ・デザイン（Security by Design）	情報セキュリティを企画・設計段階から確保するための方策。IoT 機器等においても、製品の企画・設計のフェーズからセキュリティ対策を組み込み、サイバーセキュリティ対策を確保しておく概念として、適用することができる。
ゼロトラストアーキテクチャ	組織内のネットワークであっても信頼できる領域はない(ゼロトラスト)という前提を置き、すべてのアクセスを都度検証する方式。

ツールチェーン	ソフトウェアを開発するためのツールセット。１つのツールの出力が別のツールの入力となるというように連携して利用されることからチェーンと呼ばれる。
動的解析	ソフトウェア解析手法のうち、プログラムを実行して解析、検証を行うもの。
バックドア	機器に設けられた、正規のログイン方法ではない非公表のアクセス方法。潜在的なセキュリティリスクとなりうる。
ファジングテスト	検証対象の機器やソフトウェアに脆弱性を引き起こしうるデータ（ファズデータ）を送り込み、その挙動を確認することで脆弱性を検出する手法。
プロプライエタリソフトウェア	ソフトウェアの配布者が知的財産権を持ち、その改変や複製が制限されているソフトウェアを指す。一般的にはソースコードが公開されず、ソフトウェア使用許諾契約による法的な利用上の制限がある。
ベアメタル	OS を介さず、ソフトウェア（ファームウェア）が直接ハードウェアを制御して動作する機器のこと。
ペネトレーションテスト	組織が有するすべてのシステムや、指定されたシステム全体を対象とし、明確な意図を持った攻撃者によって、その目的が達成されうるかを確認するセキュリティ検証手法。
マルウェア	許可されていないプロセスの実施を試みることによって、情報システムの機密性・完全性・可用性に悪影響をもたらすソフトウェア又はファームウェア。
リスク	目的に対する不確かさの影響。
リバースエンジニアリング	実行形式のソフトウェアを解析することにより、そのソフトウェアの機能や挙動を明らかにすること。
リポジトリ	アプリケーション開発の際に、システムを構成するデータやプログラムの情報が納められたデータベースのこと。構成管理に使われる。

1.2.2 ＜調査項目１＞各業界において、OSS を安全に活用する取組の調査

1.2.2.1 調査の背景、目的

「戦略的イノベーション創造プログラム（SIP）第 2 期/IoT 社会に対応したサイバー・フィジカル・セキュリティ」（以下「本プロジェクト」という。）においては、セキュアな Society5.0 の実現に向け、様々な IoT 機器を守り、社会全体の安全・安心を確立するため、IoT システム・サービス及び中小企業を含む大規模サプラチェーン全体を守ることに活用できる『サイバー・フィジカル・セキュリティ対策基盤』の開発と実証に取り組んでいる。

近年、企業においても Open Source Software（以下「OSS」という。）の活用が進む中、安全な OSS の選定や脆弱性の管理など、OSS の利活用に起因するサプライチェーンセキュリティリスク対策の必要性が顕在化してきている。本調査では、OSS のセキュリティ確保に関して技術検証の実態事例を調査・分析することで、その課題と最適化された技術検証のあり方を提言することを目的とする。

本調査項目（調査項目１）では、IoT 機器を開発・利用している各業界において、OSS を安全に活用する取組の実態を調査し、具体的な事例を含めてどのように活用されているかを、その背景や今後の見通しも含めて分析し報告する。

1.2.2.2 調査の方針、前提

本章の業界の取組実態調査は、アンケートシートの回収による一次調査と、各社へのヒアリングによる二次調査の 2 段階で実施した。ヒアリング調査では、アンケート回答結果の不明点や、より詳細な導入事例などを個別に確認している。調査は、6 製品分野より計 10 社（計 15 部門）を対象に実施（表 1-1）し、アンケート項目は下記の A～D の 4 分類、計 39 問で構成した（表 1-2）。

表 1-1 調査対象の製品分野及び企業・部門数

製品分野	調査対象企業・部門数
決済端末分野	1 社（計 1 部門）
金融端末分野	2 社（計 4 部門）
住設機器分野	2 社（計 3 部門）
情報システム機器分野	2 社（計 2 部門）
情報家電分野	1 社（計 3 部門）
ロボット分野	2 社（計 2 部門）
合計	10 社（15 部門）

表 1-2　アンケート項目一覧

分類		アンケート項目		策定項目数
A	OSS の利用状況	A-1	製品ごとの OSS 利用状況、製品への利用度（割合）	全 7 問
		A-2	OSS を利用する製品の種類	
		A-3	利用する OSS の機能や利用目的	
		A-4	今後の OSS 利用予定	
B	OSS 利用上の対策①～検証技術	B-1	製品構成（ハードウェア、ソフトウェア）の管理方法	全 15 問
		B-2	SBOM の利活用状況と採用している SBOM の規格	
		B-3	委託先に対する OSS 管理の状況・実施方法の例	
		B-4	OSS の脆弱性検査の状況・実施方法の例	
C	OSS 利用上の対策②～組織における管理方針	C-1	（安全な）OSS の選定に関する対応状況や実施方法	全 9 問
		C-2	委託先への（安全な）OSS の選定に関する対応状況や実施方法	
		C-3	製品構成（ハードウェア、ソフトウェア）の管理、更新状況や実施方法	
		C-4	販売済み製品への OSS 対策状況・方法	
		C-5	サポートが修了した OSS への対策状況や実施方法	
		C-6	廃棄、利用終了時の OSS への対策状況や実施方法	
		C-7	OSS ライセンスの管理状況や実施方法	
		C-8	今後に向けた OSS の安全性に関する対策の予定	
D	セキュリティ認証制度等に関する調査	D-1	認証制度へのニーズ	全 8 問
		D-2	認証制度、セキュリティ検証・適合性評価の主体について	
		D-3	セキュリティ検証（適合性評価）の内容について	
		D-4	認証制度への要望・課題について	

1.2.2.3　業界における OSS を安全に活用する取組の実態調査結果

1.2.2.3.1 業界別の OSS 利用状況、取組の調査結果（一覧）

本項では、業界別に実施した一次調査、二次調査の結果を一覧として示す。以下の結果一覧では、全体の傾向を分かりやすく可視化することを目的に、過半数を越える回答及び、同じ項目内で比較的割合が高い回答を色付けして区別を行っている。

２．

1.2.2.3.1.1　OSS の利用状況

調査対象企業に対して実施した「OSS の利用状況」に関する調査結果を表 1-3 に示す。

表 1-3　OSS の利用状況の調査結果一覧

ID	アンケート項目	回答項目	回答数	割合
Q1-1	製品で利用している OS	Windows[1]系 OS	8	53%
		Linux, BSD 系 OS（商用サポートなし、無償）	6	40%
		Linux, BSD 系 OS（商用サポートあり、有償）	3	20%
		RTOS 等の組み込み系 OS（商用サポートなし、無償）	5	33%
		RTOS 等の組み込み系 OS（商用サポートあり、有償）	2	13%
		ベアメタル（OS なし）	4	27%
		その他	1	7%
		[その他回答内容] ・T-kernel、μTron		
Q1-2	製品で利用しているソフトウェアの種類	OSS（商用サポートなし、無償）	13	87%
		OSS（商用サポートあり、有償）	5	33%
		他社製、プロプライエタリソフトウェア（無償）	5	33%
		他社製、プロプライエタリソフトウェア（有償）	7	47%
		OSS や他社製のソフトウェアは利用していない	2	13%
Q1-3	OSS を利用している製品の種類	インターネット/ソフトウェア、インフラストラクチャ	6	46%
		IoT	5	38%
		サイバーセキュリティ	2	15%
		エドテック	0	0%
		マーケティングテック	0	0%
		金融サービス、フィンテック	3	23%
		エネルギー、クリーンテック	0	0%

[1] Windows の正式名称は、Microsoft Windows Operating System となる。また Microsoft、MS、Windows、Windows Server、Windows Vista、Excel 及び関連する名称並びにそれぞれのロゴは、米国 Microsoft Corporation の米国およびその他の国における登録商標である。

ID	アンケート項目	回答項目	回答数	割合
		仮想現実（VR）、ゲーム、エンターテインメント、メディア	0	0%
		航空宇宙、航空、自動車、物流、運輸	1	8%
		エンタープライズ・ソフトウェア/SaaS	0	0%
		コンピュータ・ハードウェア、半導体	4	31%
		インターネット/モバイルアプリ	4	31%
		医療、ヘルステック、生命科学	1	8%
		リテール、e コマース	0	0%
		製造、産業、ロボット工学	3	23%
		テレコミュニケーション、ワイヤレス	0	0%
		スマートホーム等住設機器	3	23%
		情報家電	0	0%
Q1-4	製品における OSS の利用の割合	多くの製品に OSS が組み込まれている	6	40%
		一部の製品に OSS が組み込まれている	7	47%
		OSS を利用していない	2	13%
		状況を把握していない	0	0%
Q1-5	製品で利用している OSS	Web サーバ	6	46%
		アプリケーションサーバ	3	23%
		RDBMS	5	38%
		KVS	1	8%
		OS	7	54%
		開発言語	9	69%
		ライブラリ	9	69%
		UI フレームワーク	7	54%
		開発フレームワーク	5	38%
		SSL/VPN/SSH	10	77%
		プロキシ・ファイアウォール	4	31%
		認証・アクセス管理	3	23%
		DNS/DHCP	2	15%
		FTP サーバ	0	0%
		ファイルサーバ	2	15%
		メールサーバ	1	8%
		その他サーバ	2	15%

ID	アンケート項目	回答項目	回答数	割合
		メッセージング	0	0%
		IoT	3	23%
Q1-6	今後 OSS を利用する上での方針	現在利用している OSS を今後も利用、または、採用する OSS を増やすことを検討している	13	100%
		セキュリティやコスト等を考慮し、機能を代替できるソフトウェアを一部商用のものに切り替える	2	15%
		必要に応じて OSS コミュニティの活動に参加・貢献していく	2	15%
		現在 OSS を利用していないが、今後利用することを検討する	3	23%
		現在 OSS を利用しておらず、今後も利用する予定はない	1	8%
		その他	0	0%
Q1-7	OSS の利用における課題	OSS の利用状況を網羅的に把握することが難しい	7	54%
		OSS の脆弱性などセキュリティ上の懸念がある	10	77%
		継続的なアップデートやメンテナンスに懸念がある	10	77%
		OSS のバージョンアップに伴う仕様変更や互換性に懸念がある	12	92%
		できれば OSS は利用したくないが、代替手段がない	2	15%
		その他	0	0%

1.2.2.3.1.2　OSS の利用時の対策、セキュリティ検証の実施状況

調査対象企業に対して実施した「OSS の利用時の対策、セキュリティ検証の実施状況」に関する調査結果を表 1-4 に示す。

表 1-4　OSS の利用時の対策、セキュリティ検証の調査結果一覧

ID	アンケート項目	回答項目	回答数	割合
Q2-1	ソフトウェアコンポーネント情報の管理体制	社内で一元管理している	0	0%
		開発部門やチームごとに管理している	15	100%
		その他	0	0%
Q2-2	ソフトウェアコンポーネント情報の管理方法	文書ベース（Excel、Word 等）で管理している	13	87%
		SBOM の管理システムを利用して管理している	10	67%
		SPDX や SWID 等の標準化された SBOM 管理システムを利用して管理している	0	0%

ID	アンケート項目	回答項目	回答数	割合
		特に管理していない	0	0%
Q2-3	ソフトウェアコンポーネントの管理対象となる情報	バージョン情報	14	93%
Q2-3		ライセンス情報	12	80%
Q2-3		OSS・商用等の分類	9	60%
Q2-3		ソフトウェアの機能情報	6	40%
Q2-3		取得先情報（購入元情報、URL 等）	6	40%
Q2-3		その他	0	0%
Q2-4	自動解析ツール等のシステムの導入状況	開発部門等において人力・手動で管理・集計している	12	80%
		ソフトウェアコンポーネントの解析ツールを利用している（自動解析）	1	7%
		その他	1	7%
		[その他回答内容] ・社内環境として、設計要件や仕様書の管理、社内用 Git サーバの構築など、対象の文書によって異なるが、システムを構築し、様々な文書を管理している。		
Q2-5	ソフトウェアコンポーネントのバージョン管理方法（自社開発、プロプライエタリ）	バージョン管理システムを利用して管理している 例）git, mercurial, subversion 等	10	67%
		ファイルベースで管理している 例）ファイル名に日付やバージョンをつけて管理	6	40%
		その他	1	7%
		[その他回答内容] ・自社開発コンポーネントのリソースやバイナリ埋め込みでバージョン情報を内部に埋め込み。Excel 等のバージョン管理表で管理。ソース管理は一部バージョン管理システムを使用。		
Q2-6	ソフトウェアコンポーネントのバージョン管理方法（OSS,他社製）	製品で利用するすべてのソフトウェアコンポーネントを 自社開発のソフトウェアと同様に管理している	10	67%
		有償のソフトウェアのみ、自社開発のソフトウェアと同様に管理している	2	13%
		github 等のリポジトリを直接参照・ダウンロードするようにしており、特に管理していない	3	20%
		その他	1	7%
Q2-7	セキュリティ検証の実施状況（自社開発ソフトウェア）	脅威モデリング、脅威分析	6	40%
		静的解析	9	60%
		動的解析	7	47%

ID	アンケート項目	回答項目	回答数	割合
		コードレビュー	9	60%
		ペネトレーションテスト	7	47%
		ファジングテスト	1	7%
		その他	0	0%
		特に実施していない	0	0%
Q2-8	セキュリティ検証の実施状況（OSS、他社製）	脅威モデリング、脅威分析	4	27%
		静的解析	4	27%
		動的解析	5	33%
		コードレビュー	2	13%
		ペネトレーションテスト	6	40%
		ファジングテスト	1	7%
		その他	2	13%
		特に実施していない	3	20%
		[その他回答内容] ・製品全体としては、セキュリティ検査（振る舞い検査）を実施しているが、OSS 単体として検査を実施していることはない。		
Q2-9	セキュリティ検証の実施担当部門	自社の開発部門で行っている	11	73%
		自社の品質保証部門で行っている	1	7%
		開発の委託先で行っている	7	47%
		セキュリティ検証ソリューションのベンダへ委託している	1	7%
		その他	0	0%
Q2-10	セキュリティ検証の実施期間	1 製品あたり 1 ヶ月未満	10	67%
		1 製品あたり 1 -2 ヶ月	3	20%
		1 製品あたり 3-4 ヶ月	1	7%
		1 製品あたり 5-6 ヶ月	0	0%
		1 製品あたり半年以上	0	0%
Q2-11	セキュリティ検証の費用（開発後、テスト段階）	1 製品あたりの開発費用の 5%未満	12	80%
		1 製品あたりの開発費用の 5%以上 10%未満	1	7%
		1 製品あたりの開発費用の 10%以上	1	7%
Q2-12	セキュリティ検証で利用しているツールの種別	脅威モデリング、脅威分析	1	7%
		静的解析	7	47%
		動的解析	4	27%

ID	アンケート項目	回答項目	回答数	割合
		コードレビュー	2	13%
		ペネトレーションテスト	1	7%
		ファジングテスト	1	7%
		その他	3	20%
		[その他回答内容] ・製品全体としては、セキュリティ検査（振る舞い検査）を実施しているが、OSS 単体として検査を実施していることはない。また、セキュリティ検査の対応は、QA のチームが対応している（ツール名は不明）		
Q2-13	ソフトウェアコンポーネントの情報管理上の課題	ソフトウェアコンポーネント情報の管理に関するノウハウが少ない	8	53%
		ソフトウェアコンポーネント情報の管理方法が整備されていない	8	53%
		ソフトウェアコンポーネント情報の更新や正確さ・網羅性に懸念がある	10	67%
		SBOM の構築、管理のソリューション・システムがない	5	33%
		ソフトウェアコンポーネント情報の更新や情報の正確さ・網羅性に懸念がある	6	40%
		その他	0	0%
Q2-14	ソフトウェアコンポーネントの構成管理上の課題	バイナリデータの管理に懸念がある 例）サイズが大きい、差分管理ができない等	3	20%
		OSS や他社製のソフトウェアの保持・バージョン管理が難しい	7	47%
		OSS 等の外部リポジトリの参照に懸念がある 例）リポジトリの公開停止等	3	20%
		その他	3	20%
		[その他回答内容] ・管理をより簡易に、効率的に行いたい。 ・特に課題なし（2 回答）		
Q2-15	セキュリティ検証上の課題	セキュリティ検証にかける時間や予算が不足している	11	73%
		セキュリティ検証の委託、または、ツールの導入に費用がかかる	7	47%
		どのようなセキュリティ検証を実施すべきか、基準が明確になっていない	9	60%

ID	アンケート項目	回答項目	回答数	割合
		セキュリティ検証の費用対効果に懸念がある	9	60%
		セキュリティ検証ツールの性能に懸念がある 例）検出可能な脆弱性が少ない	4	27%
		セキュリティ検証を担当できる人材が不足している	9	60%
		その他	1	7%
		・特に課題なし		

1.2.2.3.1.3　OSS に関する組織上の管理対策状況

調査対象企業に対して実施した「OSS の利用時の対策、セキュリティ検証の実施状況」に関する調査結果を表 1-5 に示す。

表 1-5　OSS に関する組織上の管理対策状況の調査結果一覧

ID	アンケート項目	回答項目	回答数	割合
Q3-1	ソフトウェアコンポーネントの選定基準	ソフトウェアの選定に関する社内基準に準拠し、基準を満たしたソフトウェアのみ利用する	6	40%
		特に社内基準がなく、開発担当者や社内調達者の裁量で選定している	10	67%
		国や業界団体が発行するガイドライン等を参考にソフトウェアを選定する	1	7%
		開発コスト軽減を優先し、価格や性能要件に見合ったものを選定している	2	13%
		特に選定基準を整備していない	2	13%
		その他	2	13%
		[その他回答内容] ・過去の導入実績や脆弱性への対応、ライセンス、汎用性や将来性など、多角的な視点で、開発チームでレビューを行い選定している。開発者や調達者個人の裁量ではなく、チームレビューを通じて決定している。		
Q3-1（追加）	社内基準の更新状況	定期的に内容を精査し、見直している	4	57%
		国内外のガイドラインや法制度に合わせて更新している	0	0%
		更新に関する決まりはない	3	43%
Q3-2	開発委託先に求めるソフトウェアコンポーネントの選定方針	自社と同じ基準を要求している	8	53%
		委託先の社内基準の有無を確認し、協議の上で基準を策定している	3	20%
		委託先の基準や裁量にゆだねているが、選定したソフトウェアの報告を求めている	5	33%
		委託先の基準や裁量にゆだねており、報告等は求めていない	1	7%
		その他	1	7%
		[その他回答内容] ・委託先が従っていないことが後で判明したケースがあった。		

ID	アンケート項目	回答項目	回答数	割合
Q3-3	開発の委託先での情報管理体制	契約締結時に、仕様文書として利用するソフトウェアや脆弱性対応を含む管理方針を委託先に明示している	6	40%
		開発中、または、納品時などで、委託先でのソフトウェアの管理方針の実施状況を把握（監査）している	6	40%
		情報管理のため、ISO27001（ISMS）等に準拠した管理体制を求めている	2	13%
		特に委託先での管理は求めていない	3	20%
		その他	3	20%
		[その他回答内容] ・自社と同じ基準に準ずるはずだが、管理体制や監査は無く、十分伝達・認識されていないケースがある。 ・具体的には ISO27000 に準じた社内のガイドラインとチェックシートがあり、社内ガイドライン基準への適合を求めている。　ただし ISO27000 への厳密な対応はソフトウェアベンダにはかなり負担になるので、やや柔軟な対応を求めている。		
Q3-4	製品リリース後のソフトウェアコンポーネント管理状況	製品に含まれるコンポーネントのメンテナンス状況や最新バージョンの確認を定期的に行っている	4	27%
		製品に含まれるコンポーネントの脆弱性情報を定期的に確認している	6	40%
		脆弱性の報告や顧客からの問い合わせがあった場合に確認している	9	60%
		特に管理はしていない	0	0%
Q3-5	サポートが終了し、メンテナンスがされていないソフトウェアコンポーネントへの対応方針	当該コンポーネントのセキュリティ検証を行い、リスク評価の結果で利用を判断する	4	27%
		OSS などのメンテナンス可能なコンポーネントは、自社または委託先でメンテナンスを行う	4	27%
		同じ機能を代替するコンポーネントに切り替える	7	47%
		その他	2	13%
		[その他回答内容] ・対応できない場合は製品の使用中止をお客様に案内する（自社 HP 等） ・問題の原因や影響度を調査、分析した上で、影響が少なければ利用を継続する場合もある。		
Q3-6		製品のセキュリティに関する Web ページ等の相談窓口を設けている	8	53%

ID	アンケート項目	回答項目	回答数	割合
	製品のセキュリティに関するサポート体制	品質保証部門等にセキュリティを担当するチーム、担当者がいる	11	73%
		社内に CSIRT、PSIRT 等のセキュリティインシデント担当部門がある	9	60%
		SOC（セキュリティオペレーションセンター）等のセキュリティに関する監視の担当部門がある	2	13%
Q3-7	出荷後、販売済み製品におけるソフトウェアのセキュリティ対策状況、方法	ソフトウェアの最新の脆弱性情報を定期的にキャッチアップし、脆弱性が発覚した場合は、システムアップデート等で対応する	7	47%
		ユーザや外部機関から脆弱性やインシデントが報告された場合は、調査、システムアップデート等で対応する	12	80%
		製品のソフトウェアコンポーネントに対して継続的にセキュリティ検証テストを行う	1	7%
		その他	1	7%
		[その他回答内容] ・客からの問い合わせに応じて方針を検討した上で、対応を行う。		
Q3-8	サポート終了製品におけるソフトウェアコンポーネントへの対策状況、方法	重大な脆弱性が判明した場合には対応する	9	60%
		有償またはユーザからの依頼で対応する	2	13%
		対応に関するルールは特に決まっていない	9	60%
		その他	2	13%
		[その他回答内容] ・対応できない場合は製品の使用中止をお客様に案内する（自社 HP 等） ・問題の原因や影響度を調査、分析した上で、影響が少なければ利用を継続する。過去の事例では OSS の脆弱性が製品のセキュリティに影響を与えるケースは殆どない。		
Q3-9	OSS 等の利用や管理上の課題	ソフトウェアの選定方針や管理体制の構築等で参考になる情報が少ない	10	67%
		自社でのソフトウェアコンポーネントの管理体制の構築が難しい（予算や人員など）	11	73%
		社内や開発の委託先での情報管理やセキュリティ対策の実施状況の把握が難しい	8	53%
		OSS の管理に関するノウハウが不十分 例）ライセンスに関する知識がないなど	9	60%

ID	アンケート項目	回答項目	回答数	割合
		脆弱性などの情報収集にまでコストをかけることが難しい	9	60%
		その他	1	7%
		[その他回答内容] ・OSS としては機能が不足しているケースや、バージョンアップにより互換性が失われてしまうという問題が過去にあり、課題となっている。バージョンアップにより互換性が失われてしまう場合も、製品側として対応していく必要があり、コストが掛かる。		

1.2.2.3.1.4　ソフトウェアのセキュリティに対する検証機関、認証制度への要望

調査対象企業に対して実施した「ソフトウェアのセキュリティに対する検証機関、認証制度への要望」に関する調査結果を表 1-6 に示す。

表 1-6　セキュリティに対する検証機関、認証制度に対する要望の調査結果一覧

ID	アンケート項目	回答項目	回答数	割合
Q4-1	ソフトウェアコンポーネントのセキュリティに関する認証制度のニーズ	自社で認証を取得したい	3	20%
		製品で利用するソフトウェアの選定基準の 1 つとして活用したい	12	80%
		認証制度は必要ない	0	0%
		その他	1	7%
		[その他回答内容] ・国内向けの認証制度が望ましい。		
Q4-2	認証制度の基準への適合性評価（セキュリティ検証等）の実施機関	認定された第三者認証機関が適合性評価を行い、認証する（第三者認証方式）	2	13%
		認定された検証機関が適合性評価を行い、第三者認証機関が認証する（第三者の適合性評価による第三者認証方式）	1	7%
		自社または委託先で適合性評価を行い、第三者認証機関が認証する（自己適合性評価による第三者認証方式）	9	60%
		自社または委託先で適合性評価を行い、自社で結果を公表する（自己適合宣言方式、自己認証方式）	2	13%
Q4-3		当該認証制度に合わせた資格などを設け、資格を保有する技術者が在籍する機関を選定する	4	27%

ID	アンケート項目	回答項目	回答数	割合
	適合性評価の時実施機関・事業者として望ましい基準	既存制度のセキュリティ資格を保有する技術者が在籍する機関を選定する　例）情報処理安全確保支援士など	1	7%
		認定基準や資格などをあまり厳格にせず、多くの事業者が評価を実施できるようにする	5	33%
		事業者認定の制度を設ける 例）ISO27001 取得事業者等	2	13%
		その他	2	13%
		[その他回答内容] ・対象製品によって考え方が異なる。人命や安全、重要な資産に関わるソフトウェア（製品）であれば、第三者認証機関による認証の取得が望ましいが、そうでないソフトウェア製品については、もっと簡易的な適合性評価までで（自己適合宣言もしない）良いのではないか。 ・制度の内容によって異なるため現時点では判断が難しい。基本的には、テストツールによる自動実施や実施内容が明確に定義されているなど、実施者によるブレが生じないことが望ましい。		
Q4-4	第三者認証において、認証機関へ提出する提出可能な文書	認証に関連する仕様書や設計書等の文書	8	53%
		ソフトウェアのバイナリデータ	7	47%
		ソフトウェアのソースコード	2	13%
		セキュリティ検証のテスト結果の文書 例）ログデータやスクリーンショット等	11	73%
		ソフトウェアコンポーネント一覧、SBOM 等	3	20%
		その他	2	13%
		[その他回答内容] ・設計文書についてはボリュームに依存する。バイナリデータに関しては暗号化対応を行った状態で可能と想定している。 ・検証ツールを公開し、そのツールの検査結果を Output として提出する形であれば、基準の明確化/徹底、作業の省力化が図れるのではないか。		
Q4-5	申請手続きや認証付与の仕組みへの要望	申請の進捗状況を把握できるようにしてほしい	7	47%
		申請はオンライン手続きが望ましい	10	67%
		ソースコードや設計書等のデータを提出せず、申請書と検証結果のみで認証できる形が望ましい	11	73%
		認証の証書を発行してほしい	4	27%
		その他	2	13%
		[その他回答内容]		

ID	アンケート項目	回答項目	回答数	割合
		・現時点においてどのような仕組みが適切かは、判断が難しい。 ・申請時に機密事項の書類を取り扱う場合は、メールでの送付ではなく、暗号化された文書を SaaS でアップロードする形が良い。		
Q4-6	各社の製品に関係する認証制度	ISO27000 シリーズ	3	20%
		ISO/IEC 15408 （Common Criteria）	1	7%
		IEC 62443 （EDSA）	1	7%
		関係するものは特にない	11	73%
		その他	1	7%
		[その他回答内容] ・ISO27000 シリーズは関連するが、取得を義務化するのでなく、ISO27000 シリーズに準じた対応を想定している（ISO27000 シリーズ相当）		
Q4-7	海外のセキュリティ認証制度に関する課題、懸念事項	認証の取得に関する事例や情報が少ない	8	53%
		海外機関への機密情報の漏えいの恐れがある	5	33%
		制度が多数あり、取得にコストがかかる	4	27%
		基準が厳しく取得のハードルが高い	5	33%
		取得に対する費用対効果が低い	8	53%
		課題や懸念は特にない	1	7%
		その他	4	27%
		[その他回答内容] ・現状、海外向け製品の予定が無いため回答が困難。 ・認証制度についての知見がないため、判断ができない ・海外の認証制度では自国製品が優遇されるのではないかという懸念がある。		
Q4-8	セキュリティ認証制度について要望（３つまで）	国や業界団体によって取得が推奨されている	6	40%
		業界団体等ではなく、公的な機関が運用、認証している	4	27%
		政府調達や税制上の優遇がある	3	20%
		取得するための費用が安く、金額が明確に提示されている	8	53%
		認証申請から、完了までのプロセスが分かりやすい	3	20%
		対応すべきセキュリティの基準が明確であり、信頼できる。 例）有識者レビュー等を経て策定されているなど	8	53%

ID	アンケート項目	回答項目	回答数	割合
		米国や欧州等のガイドラインのセキュリティ要件に互換性がある（網羅的な要件とする）	4	27%
		認証を取得した製品に関連した脆弱性の情報等が共有される	3	20%
		その他	1	7%
		[その他回答内容] ・やはり対応するためのセキュリティコストが課題となるため、セキュリティコストを捻出、負担するための目的やストーリーが購入者や利用者に正しく理解してもらえる仕組みが必要。		

1.2.2.3.2　業界における取組の事例、業界分野別の特徴

本項では、調査結果により確認された調査対象企業における OSS の利用状況、セキュリティ対策について、特徴的な傾向の分析や対策の事例を示す。

1.2.2.3.2.1　OSS の利用状況

OSS は業界分野を問わず、87%の企業で「OSS（商用サポート無し、無償）」が利用されており、次いで「他社製のプロプライエタリソフトウェア（有償）」が 47%の企業で利用されている。OSS を利用しておらず、今後も導入予定がない企業は全体の 8%に留まり、理由としては、製品の基幹モジュールに対する品質保証及び、セキュリティ上の懸念が挙げられていた。また対象製品によっては利用可能な OSS が現状では市場に普及していないため、OSS を導入しておらず、今後有益な OSS が公開されれば、利用していくという回答もあった。結果として、今回の調査対象企業は、殆どの企業で既に OSS の利用をしており、今後の導入についても継続していく傾向が確認された（図 1-1、図 1-2）。

製品の OS（オペレーティングシステム）に対する OSS の利用状況については、今回の対象企業では、「Windows 系 OS」が 53%、「Linux, BSD 系 OS（商用サポートなし、無償）」が 40%の比率で利用されているが、どちらの製品 OS でも OSS は利用されており、明確な　製品 OS の違いによる差異は、確認されなかった。

業界分野別の傾向についても、今回の調査対象企業では、分野による OSS の利用状況の差異は確認されておらず、幅広い業界で OSS が導入されている結果となった。

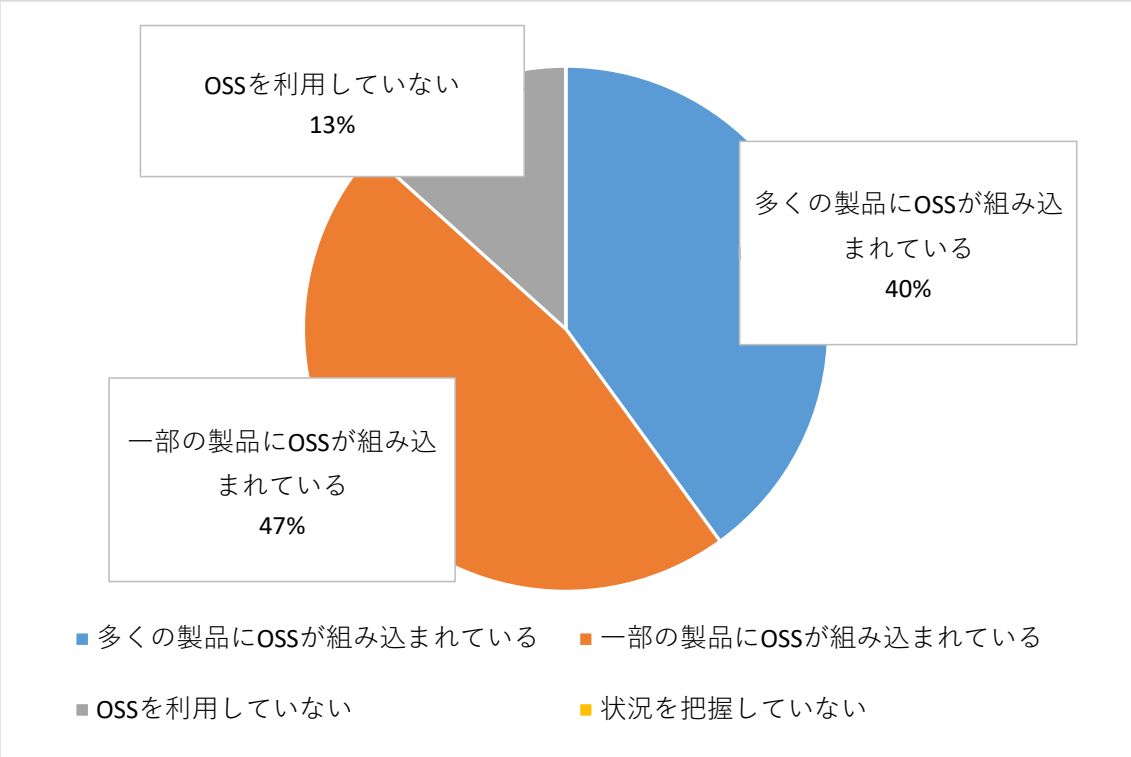


図 1-1　製品の OSS 利用割合

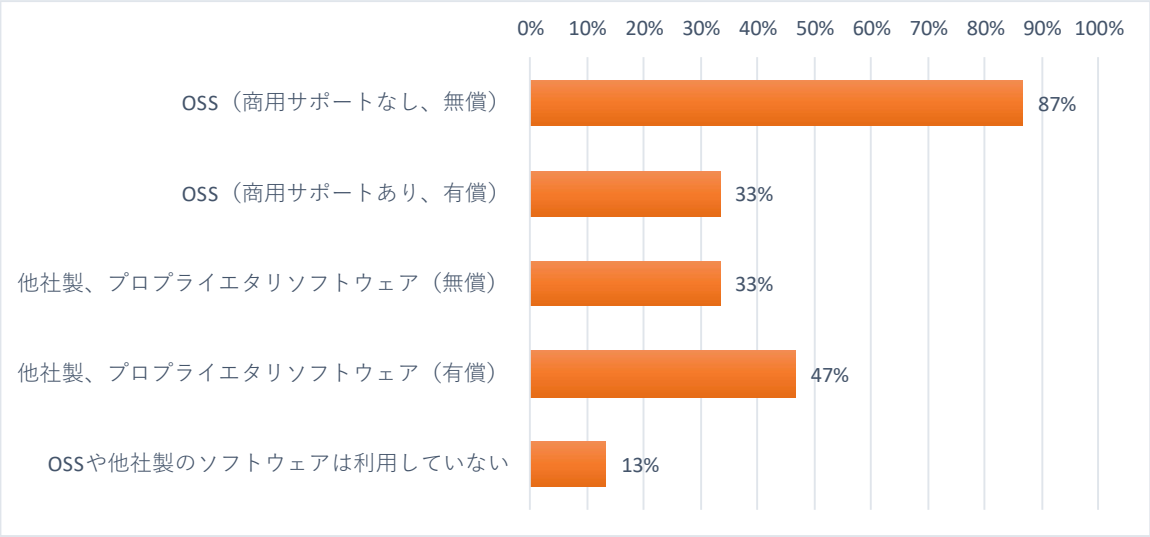


図 1-2　製品で利用しているソフトウェアの種類

各業界別の製品で利用されている OSS の種類としては、「開発言語」、「ライブラリ」が全体の 69%、「SSL/VPN/SSH」が全体の 77%で利用されている（図 1-3）。次いで「OS」、「UI フレームワーク」54%、「Web サーバ」が 46%の利用率となった。各社へのヒアリング調査では、こうした OSS は、自社で開発を行う場合と比較し、開発効率や安全性を高める上で

有益であることが導入の理由として回答されている。またウェブや、サーバに関連する OSS については、比較的早期から普及が進み、成熟した OSS が多く公開されていることが導入の理由として挙げられている。

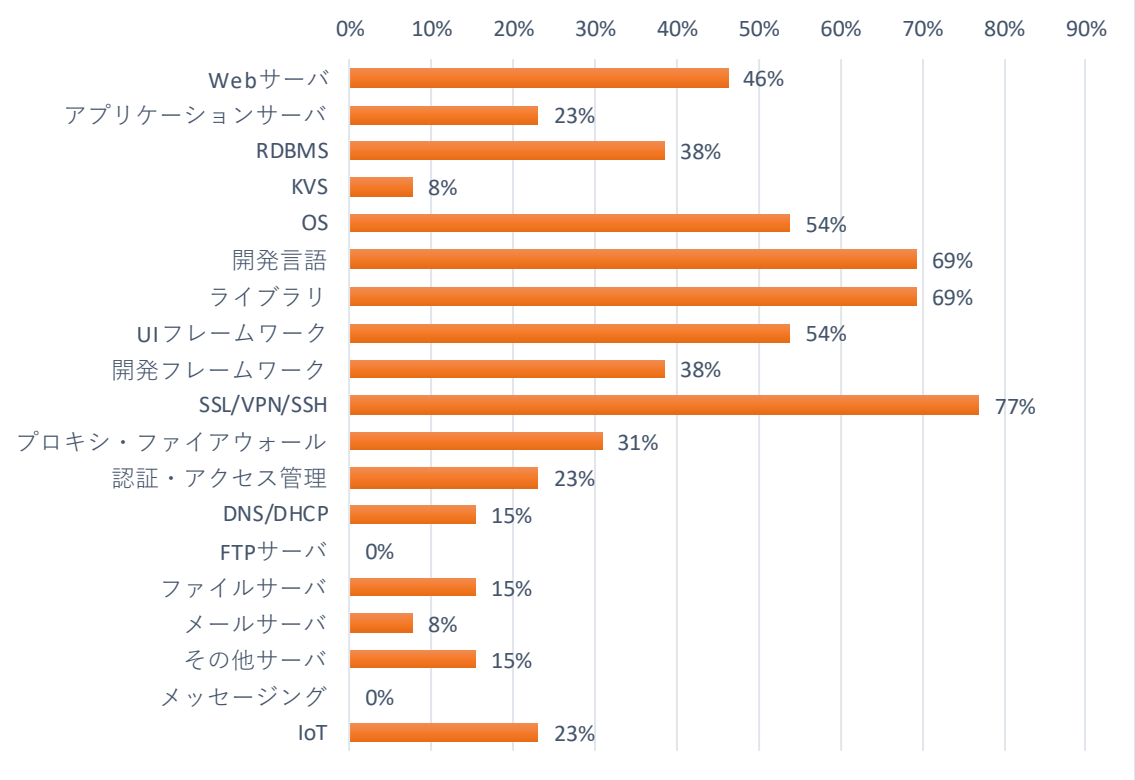


図 1-3　製品で利用されている OSS の状況

1.2.2.3.2.2　OSS の利用時の対策、セキュリティ検証の実施状況

①ソフトウェアコンポーネントの管理体制、管理方法

OSS を含むソフトウェアコンポーネントの情報管理については、全ての対象企業が「開発部門やチームごとに管理している」という回答しており、全社的に統一されたフォーマットや、SPDX や SWID 等の標準規格に基づく情報管理を行っている企業はなかった（図 1-4）。理由としては、管理コストの問題から、必要な情報を取捨選択した上で管理を行う必要があり、全社的な統一フォーマットを使用した場合、不要な情報まで管理項目に含まれることで、管理コストが増えることを懸念している。また、SPDX や SWID 等の SBOM（ソフトウェア構成表）の標準規格については、まだ国内において情報の公開が少なく、今回の調査対象企業においても広く認知されているとは言えない状況であった。調査対象企業のうち 2 社は、P-SIRT 機能強化の一環として全社的な SBOM フォーマットの統一を検討しており、標準規格が効率的に利用できるものではあれば導入を検討する旨、回答があった。

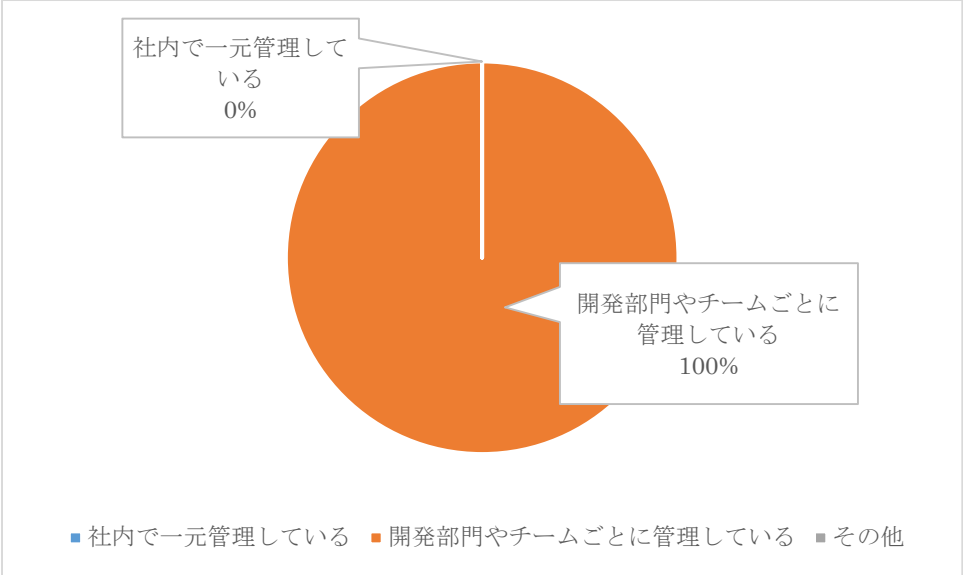


図 1-4　ソフトウェアコンポーネント情報の管理体制

OSS を含むソフトウェアコンポーネントの情報管理の方法については、調査対象の 87%が「文書ベース（Excel、Word 等）」による手動での管理を行っている状況である。ただし、全ての情報を手動で管理しているのではなく、全体の 60%は CI（Continuous Integration＝継続的インテグレーション）環境や GIT サーバによる設計文書管理などの導入により、ソフトウェアコンポーネントのバージョン情報やライセンス情報をシステム的に管理していることが確認された（図 1-5）。また 1 社ではあるが、ソフトウェアコンポーネントの解析診断ツールを導入し、対応している部門もあったが、全社的な導入ではなく、プロジェクト単位での試験的な利用に留まっており、現状では広く普及されているとは言えない状況が確認された。また OSS のソースコード管理については、ヒアリング調査の結果、OSS を利用している全ての企業が、公開されているレポジトリではなく、自社の GIT サーバ等の環境で管理を実施しており、提供元による公開停止への対処が行われていた。

ヒアリング調査による業界分野別の傾向としては、情報システム機器分野が、比較的 OSS を早期に導入しており、OSS の管理やセキュリティ検証についても対応プロセスが確立されている印象であった。

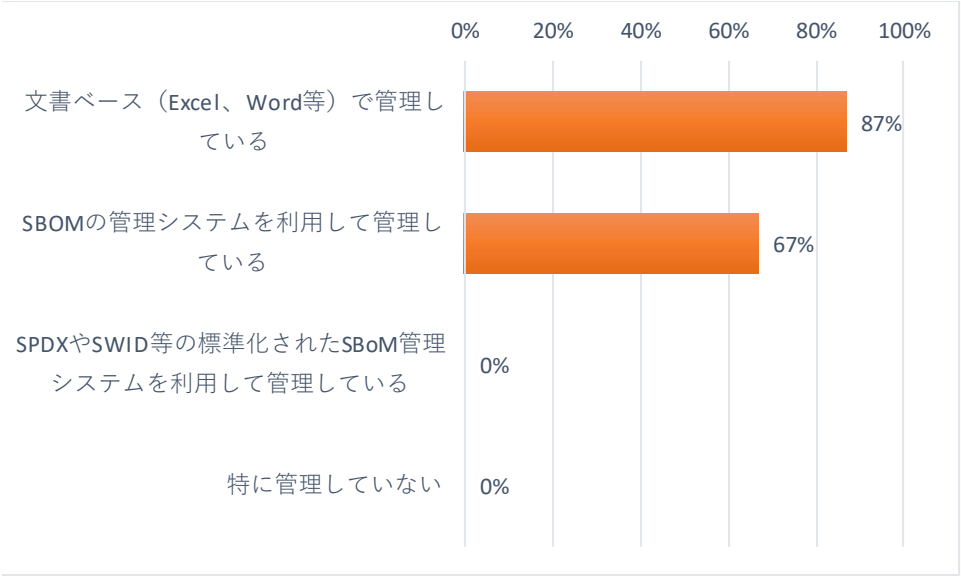


図 1-5　ソフトウェアコンポーネント情報の管理方法

管理対象となるソフトウェアコンポーネントの情報としては、バージョン情報が全体の93%、ライセンス情報が 80%の回答なっており、管理情報として重要視されている事が確認された。特にライセンス情報は、SQL のように利用中にライセンスの形態が変更された例も過去にあり、ビジネス的には訴訟リスクに関わることから、多くの企業が管理対象としていることが確認された。また OSS のバージョン管理方法としては、67%が「バージョン管理システムを利用して管理している」と回答しており、自社開発やプロプライエタリのソフトウェアと同様に git, mercurial, subversion などの環境を利用して管理を行なっている。

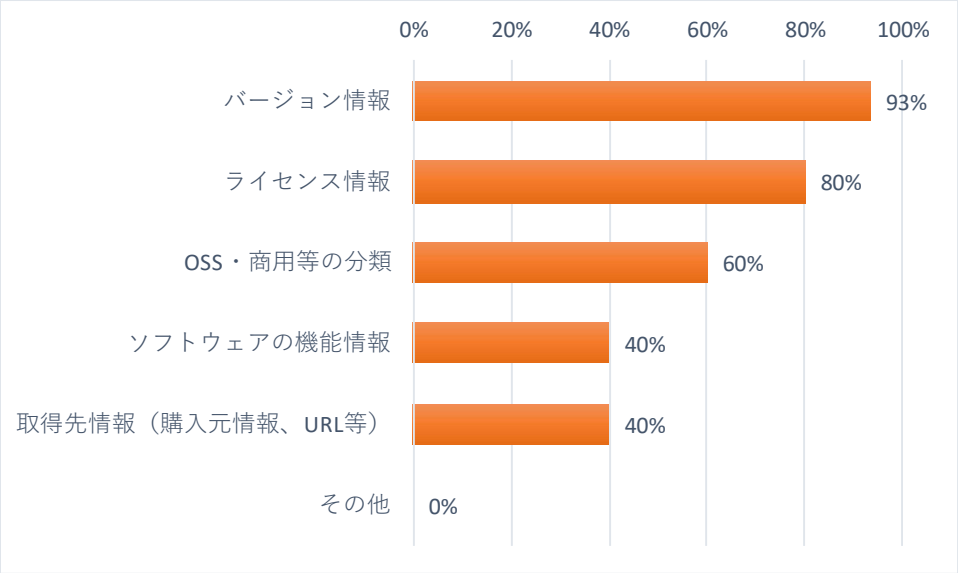


図 1-6　ソフトウェアコンポーネントの管理対象となる情報

②ソフトウェアのセキュリティ検証の実施状況

ソフトウェアに対するセキュリティ検証については、自社開発のソフトウェアを対象とした場合、何らかの検証が実施されており、特に静的解析（60%）や、コードレビュー（60%）の比率が高い結果が確認された（図 1-7）。一方で、OSS や他社製のソフトウェアを対象とした場合、製品全体としてのセキュリティ検査は実施しているものの、OSS や他社製のソフトウェア単体を対象とした検査の比率はいずれも低い結果となった（図 1-8）。検証に対する自動化ツール等の導入状況については、静的解析が 47%と最も高く、多くは商用ツールが利用されているが、一部オープンソースのツールも利用されていた。また比率的には少数となるが、ソフトウェアコンポーネントの解析と脆弱性診断が可能な商用ツールを利用し、OSS を含めた静的解析を実施している企業も確認された。

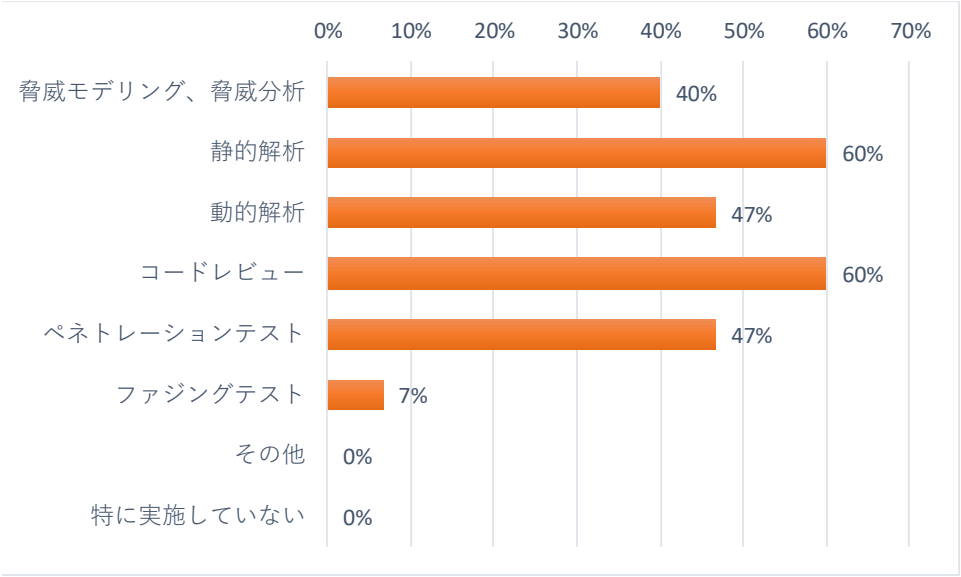


図 1-7 セキュリティ検証の実施状況（自社開発ソフトウェア）

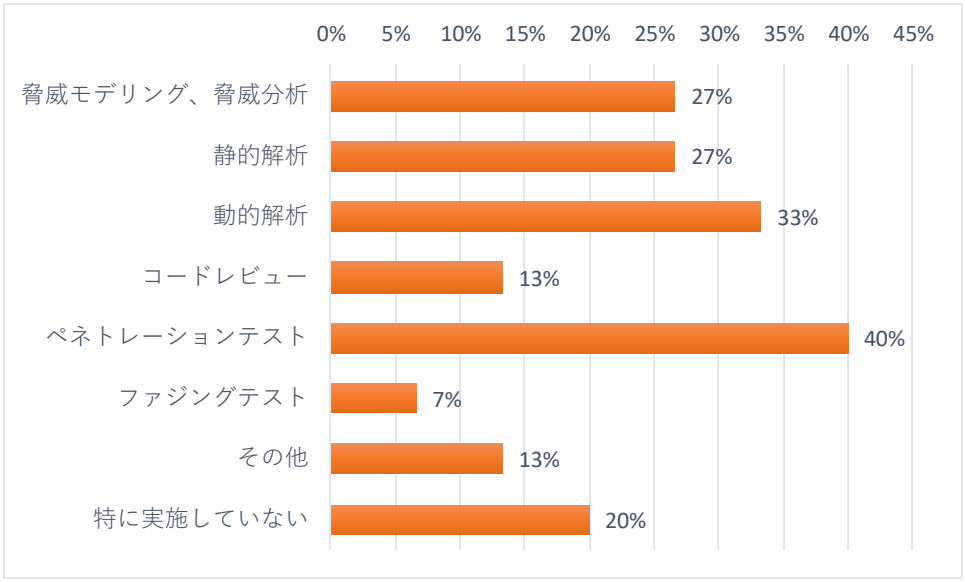


図 1-8 セキュリティ検証の実施状況（OSS、他社製）

セキュリティ検証に掛けられている費用としては、開発費用の 5%未満という回答が最も多く（80%）、検証の実施期間については 67%の回答が 1 ヵ月未満であることが確認された。（図 1-9）

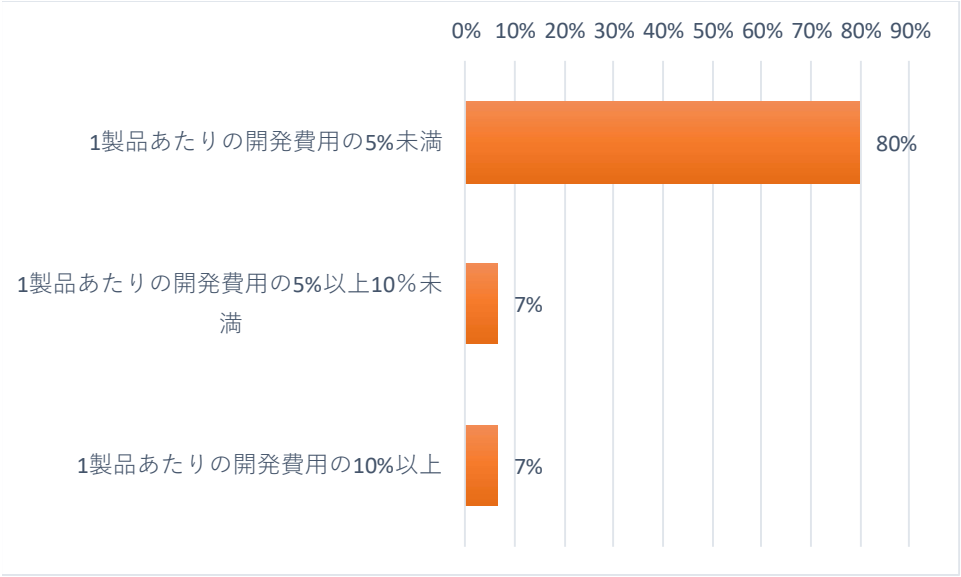


図 1-9 セキュリティ検証の費用（開発後、テスト段階）

1.2.2.3.2.3 OSS に関する組織上の管理対策状況

①ソフトウェアコンポーネントの選定基準、開発委託先への情報管理について

OSS を含めた製品に導入するソフトウェアコンポーネントの選定基準としては、67%の調査対象より「特に社内基準がなく、開発担当者や社内調達者の裁量で選定している」と回答があり、40%の調査対象より「ソフトウェアの選定に関する社内基準に準拠し、基準を満たしたソフトウェアのみ利用する」という回答が得られた。ヒアリング調査では、社内基準の例として、明確に利用可能な OSS のリストを策定している企業も確認された。また、明確に社内基準がない場合であっても、開発チーム内のレビューにて、採用するソフトウェアの利用実績や脆弱性への対応状況、ライセンスの状況など、セキュリティを含め多角的に確認している企業が殆どであることが確認されている。

開発委託先に求めるソフトウェアコンポーネントの選定基準については、53%より「自社と同じ基準を要求している」と回答があり、社内基準を策定している対象の殆どが、委託先にも自社と同様の基準を求めていることが確認された（図 1-10）。また開発委託先の情報管理体制については、40%から「契約締結時に、仕様文書として利用するソフトウェアや脆弱性対応を含む管理方針を委託先に明示している」と回答があり、対象企業の多くが契約段階で情報セキュリティ管理方針を明確にしている。文書として明確化していない場合も、40%が「開発中、または、納品時などで、委託先のソフトウェアの管理方針の実施状況を把握（監査）している」と回答しており、仕様提示段階あるいは開発中のミーティング等でソフトウェアコンポーネントに対する脆弱性対応や情報セキュリティ管理の遵守状況について確認を行なっている（図 1-11）。

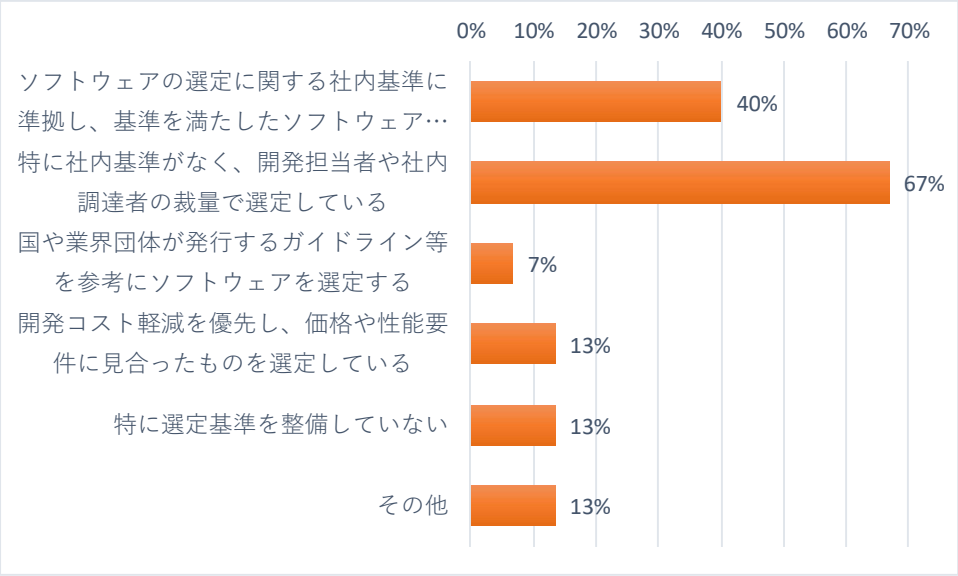


図 1-10　ソフトウェアコンポーネントの選定基準

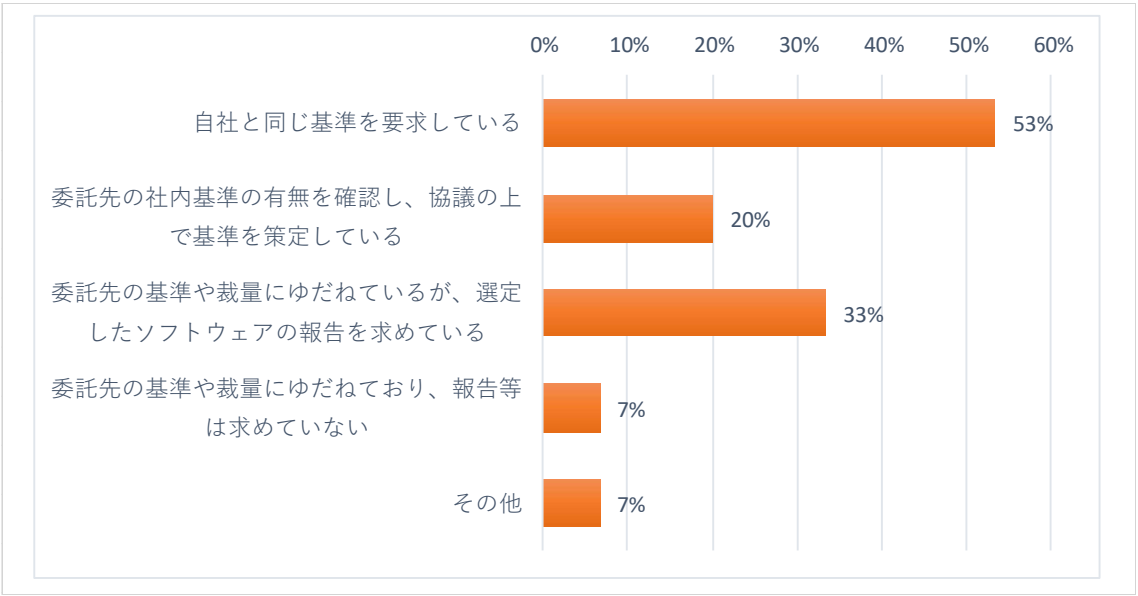


図 1-11　開発委託先に求めるソフトウェアコンポーネントの選定方針

②製品リリース後のソフトウェアコンポーネント管理、インシデントレスポンス対応

製品リリース後のソフトウェアコンポーネントの管理状況については、60%の調査対象企業が「脆弱性の報告や顧客からの問い合わせがあった場合に確認している」と回答しており、脆弱性情報を定期的に確認している企業は 40%、メンテナンスや最新バージョンの状況までを行っている企業は 27%と、いずれも半数以下の比率であることが確認された（図

1-12)。

製品に対するセキュリティサポートや脆弱性への対応については、73%が品質保証部門等のチームが兼任でセキュリティも含めて対応を行っており、60%が社内に CSIRT、PSIRT 等のインシデント対応の担当部門が設置されているという回答であった（図 1-13)。またヒアリング調査では、PSIRT という呼称ではないが、社内の開発メンバーが集うコミュニティが社内形成され、脆弱性情報の共有やインシデント対応方針を協議するという仮想的な PSIRT 体制が構築されている企業もあった。今回の調査では CSIRT や PSIRT に選任の担当者を配置することが、コスト的にも厳しいという回答が多い中、こうした柔軟性のある試みは、今後のインシデントレスポンス対応の対応事例として有益であると考える。

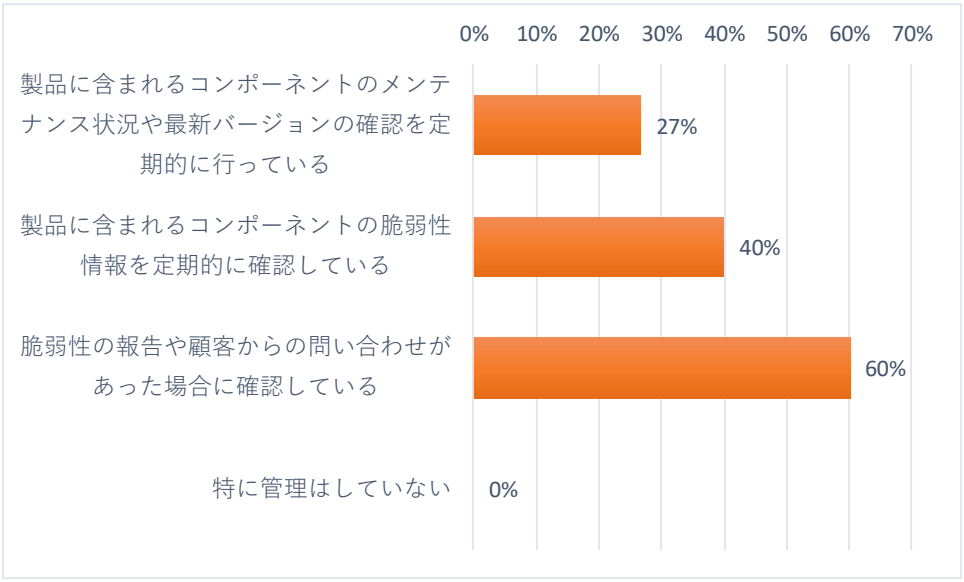


図 1-12　製品リリース後のソフトウェアコンポーネント管理状況

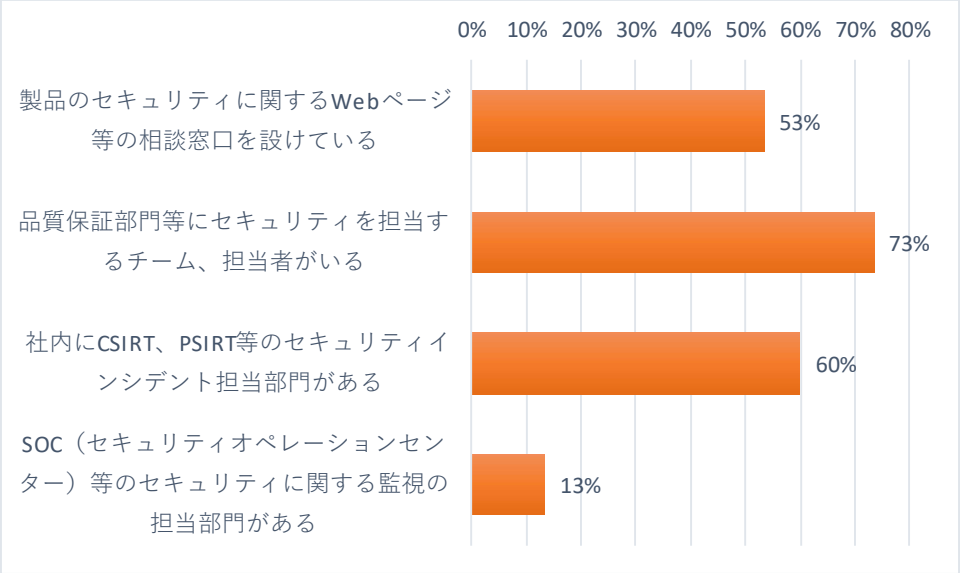


図 1-13　製品のセキュリティに関するサポート体制

③サポートが終了したソフトウェアコンポーネントへの対応

調査対象企業の製品において、OSS を含めサポートが終了し、メンテナンスがされていないソフトウェアコンポーネントへの対応については、47%が「同じ機能を代替するコンポーネントへ切り替える」という回答であった。他の回答では「当該コンポーネントのセキュリティ検証を行い、リスク評価の結果で利用を判断する」、「自社または委託先でメンテナンスを行う」という回答がそれぞれ 27%となっており、製品に組み込まれているソフトウェアコンポーネントの置換はコストも含めて影響が大きく、他のコンポーネントへの置換が困難であることが確認された。(図 1-14)。製品分野別の傾向では、情報システム機器分野が、比較的早期に OSS を導入しており、OSS の利用についても知見やノウハウを他分野に先行して有している印象であった。OSS に起因する脆弱性への対応についても、OSS 提供元でのアップデートが困難な場合は、自社の製品側で対応することが、通常の対応として定着していることが確認された。

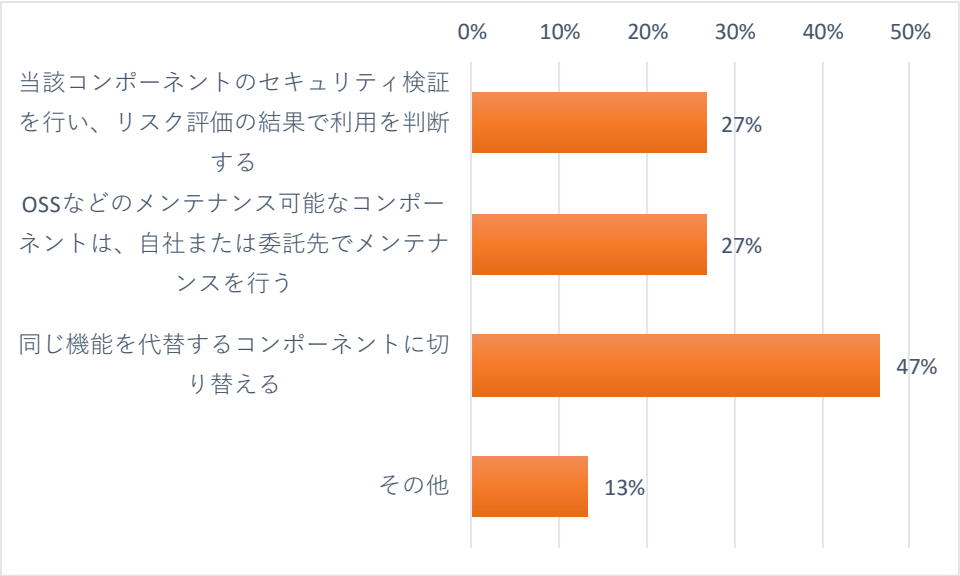


図 1-14　サポートが終了しメンテナンスがされていないソフトウェアコンポーネントへの対応方針

1.2.2.3.2.4　製造システムや製品管理システムについての状況

ヒアリングにて、製造システムや製品管理システムへの OSS 導入状況や、OSS によるセキュリティリスクについても追加調査を実施した。製造システムについては、調査を依頼した部門とは担当部門が異なるため、明確な回答は得られなかったものの、原則として殆どの企業で外部ネットワークからの隔離や、クリーンルームが徹底されており、OSS は利用されていないという回答が確認されている。また製品管理システムについては、より広義では開発システムも対象となり、Redmine 等のチケット管理システムなど、オープンソースのシステムが一部で利用されているが、基本的には提供元によるアップデートは適用されており、過去にオープンソースのシステムを利用することで、トラブルが発生した事例はないという回答が確認された。

1.2.2.3.2.5　ソフトウェアのセキュリティに対する検証機関、認証制度への要望

①認証制度や適合性検査実施機関へのニーズ

ソフトウェアを対象とした認証制度のニーズとしては、80%が「自社で利用するソフトウェアの選定基準として活用したい」という回答であった（図 1-15）。今回の調査対象企業がソフトウェアを利用するユーザ企業であるため、調達基準として活用できる制度への期待値が高いこと、認証制度自体に対するニーズは非常に高いことが確認されている。

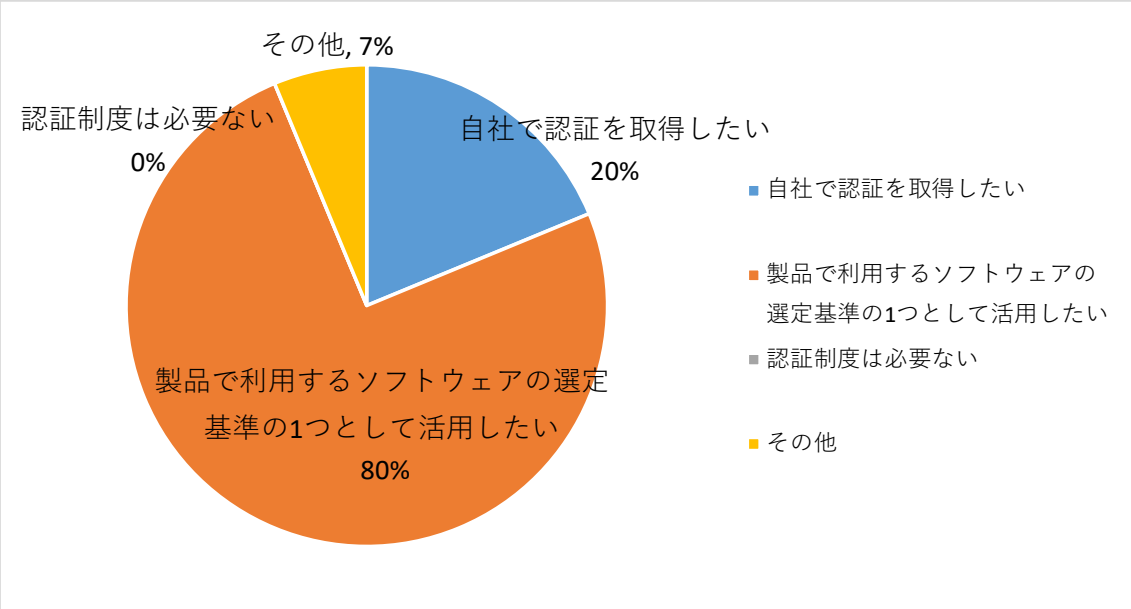


図 1-15　ソフトウェアコンポーネントのセキュリティに関する認証制度のニーズ

認証制度の適合性評価については、60%の回答が「自社または委託先で適合性評価を行い、第三者認証機関が認証する（自己適合性評価による第三者認証方式)」であり、ユーザ企業としては、第三者評価機関による信頼性が担保された認証制度へのニーズが高いことが確認された（図 1-16)。また、認証制度の要望としては、「認定基準や資格などをあまり厳格にせず、多くの事業者が評価を実施できるようにする」が 33%、「当該認証制度に合わせた資格などを設け、資格を保有する技術者が在籍する機関を選定する」が 27%と、相対的に高い比率で回答されており、あまり適合性評価の実施機関については、あまり厳格な基準でない方が望ましいが、一定のスキルレベルは要望されていることが確認された（図 1-17)。また、その他の回答としては、テストツールによる自動実施や実施内容が明確に定義されているなど、実施者による差異が生じないことが望ましいという意見も見られた。

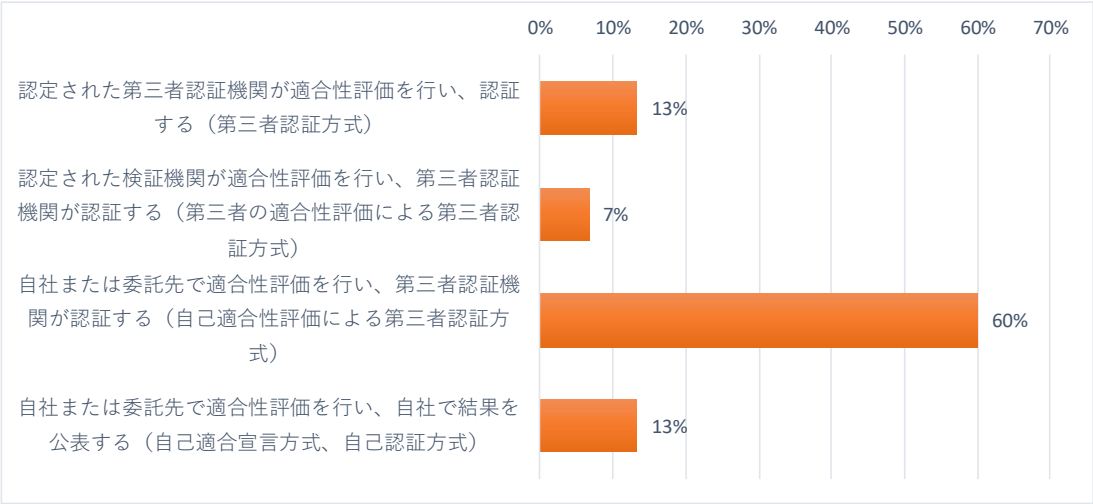


図 1-16　認証制度の基準への適合性評価（セキュリティ検証等）の実施機関

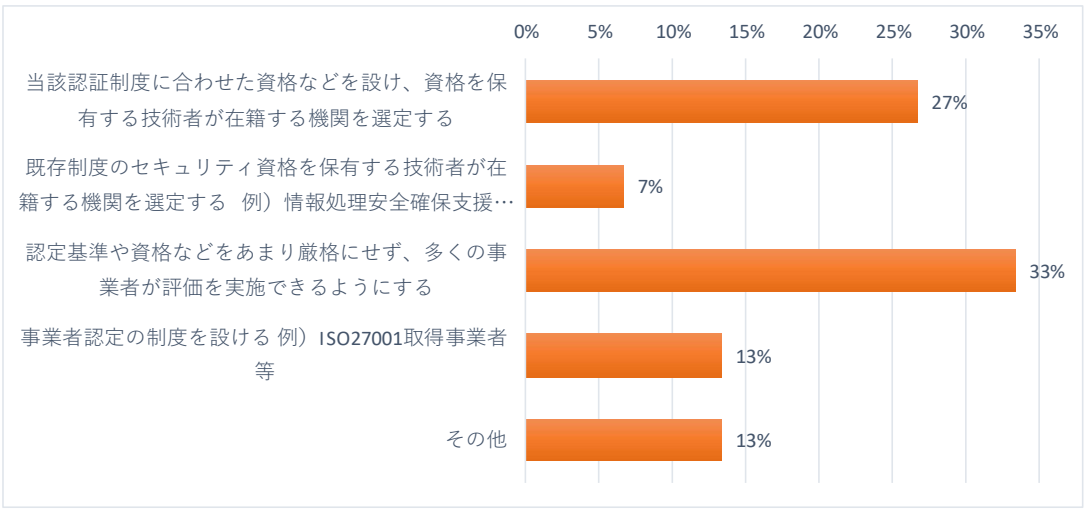


図 1-17　適合性評価の実施機関・事業者として望ましい基準

②認証制度に対する要望事項

認証機関への要望事項としては、「取得するための費用が安く、金額が明確に提示されている」が最も高く、53%の回答であった。次いで「国や業界団体によって取得が推奨されている」が 40%の回答となっている（図 1-18）。ヒアリング調査においては、多くの企業から寄せられた回答としては、やはり認証制度ではセキュリティコストが課題となり、取得コストを負担する目的が、製品利用者や顧客に明確に伝わる制度が必要であるという意見が非常に多かった。最終的には認証取得のコストは製品単価に転嫁せざるを得ず、普及啓発の仕組みに加え、認証制度の目的が製品利用者や顧客に分かりやすく説明できるような運用を含めた制度構築が求められている。

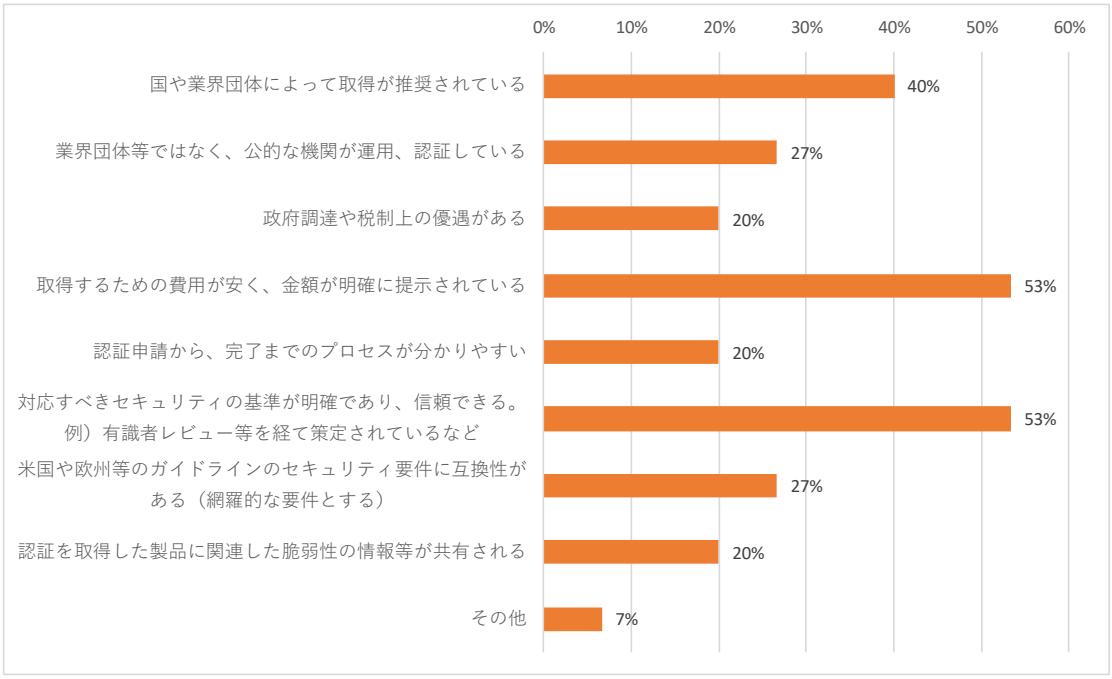


図 1-18　セキュリティ認証制度について要望

1.2.2.3.3　OSS の利用や管理、セキュリティ検証における課題

本章の最後に、調査において確認された OSS の利用や管理運用上の課題、セキュリティ検証の課題を示す。

1.2.2.3.3.1　OSS 利用上の課題

OSS 利用上の課題としては、92%より「OSS のバージョンアップに伴う仕様変更や互換性に懸念がある」という回答があった（図 1-9）。特に製品のライフサイクルが長期にわたる機器では、OSS を含むソフトウェアコンポーネントのライフサイクルと製品自体のライフサイクルが一致しておらず、更新されたソフトウェアコンポーネントを適用するために、製品側でも品質検査や修正対応が必要となる。また、OSS やソフトウェアコンポーネントの更新により、製品との互換性が失われ、同様に製品側で対応が求められるという回答も確認されている。

その他の課題としては「継続的なアップデートやメンテナンスに懸念がある」、「OSS の脆弱性などセキュリティ上の懸念がある」が共に 77%と多くの企業で課題となっている（図 1-9）。ただし、多くの企業では、ソフトウェアコンポーネントの選定時に脆弱性への対応実績や、品質上の安定性などをレビューした上で導入されており、過去に OSS に起因する問題により製品トラブルが発生した事例は、ごく少数であった。

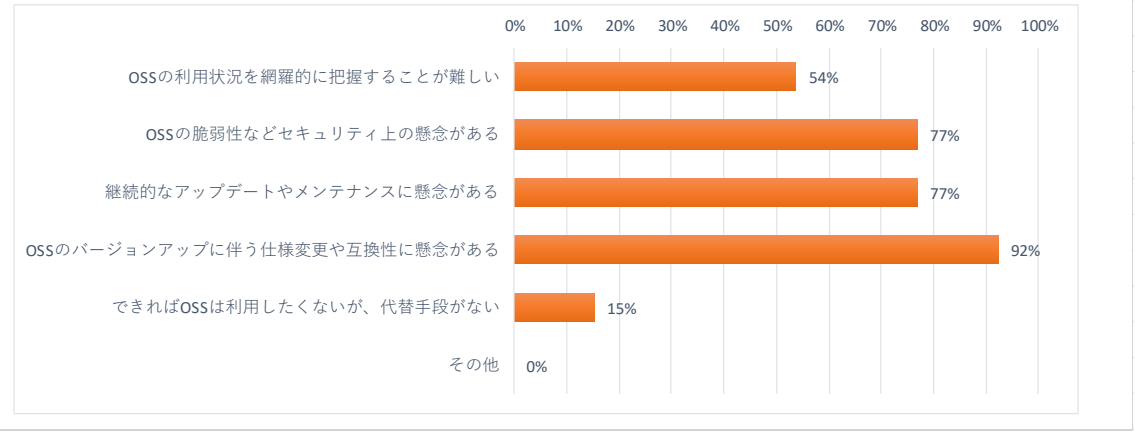


図 1-19　OSS 利用上の課題

1.2.2.3.3.2　ソフトウェアコンポーネントの管理上の課題

ソフトウェアコンポーネントの情報管理における課題としては、「ソフトウェアコンポーネント情報の更新や正確さ・網羅性に懸念がある」という回答が 67%で最も比率が多い。次いで「ソフトウェアコンポーネント情報の管理に関するノウハウが少ない」、「ソフトウェアコンポーネント情報の管理方法が整備されていない」という回答がそれぞれ 53%となり、過半数を占める比率となっている（図 1-20）。ソフトウェアコンポーネントの情報管理については、調査対象企業の殆どが開発部門ごとに構成表のフォーマットを定義し、運用している傾向が確認されているが、より効率的な運用を目指して、継続的に試行錯誤が行われている印象である。

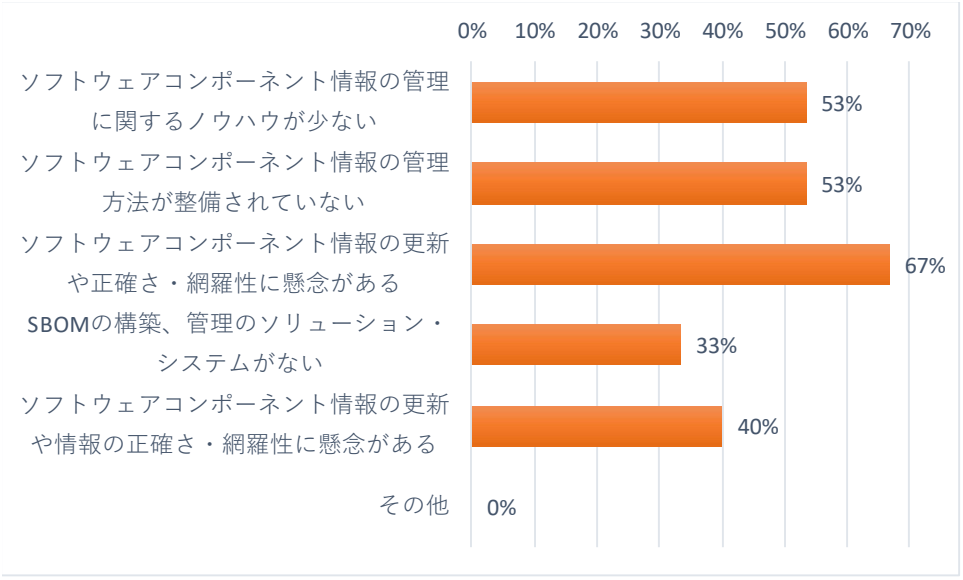


図 1-20　ソフトウェアコンポーネントの情報管理上の課題

1.2.2.3.3.3　セキュリティ検証における課題

ソフトウェアコンポーネントを含む製品のセキュリティ検証の課題としては、「セキュリティ検証にかける時間や予算が不足している」が最も比率が高く 73%の調査対象企業より回答があった（図 1-21）。セキュリティ検証を含む対策コストは、製品単価に転嫁せざるを得ず、セキュリティ対策に関する利用者や顧客の理解が必要である。製品購入時に、価格だけでなく安心、安全な製品であることが選定の基準として浸透するよう、国の施策や業界団体の活動を通じて、普及啓発を継続していくことが期待されている。

その他の回答としては、「どのようなセキュリティ検証を実施すべきか、基準が明確になっていない」、「セキュリティ検証の費用対効果に懸念がある」、「セキュリティ検証を担当できる人材が不足している」がそれぞれ 60%と高い比率で回答されている（図 1-21）。セキュリティ検証については、製品全体と実施しているケースが殆どであり、組み込まれている OSS を対象としてセキュリティ検証を実施しているケースは少数の回答であった。セキュリティ検証の手法自体も多くの企業では、対費用効果として充分な効果を実感しているとは言えない状況であった。また、ソフトウェアコンポーネントを対象とした場合、セキュリティの検証手法や手順が、まだ確立されておらず、対応にあたっての情報が少ない点も多くの対象企業から回答されている。

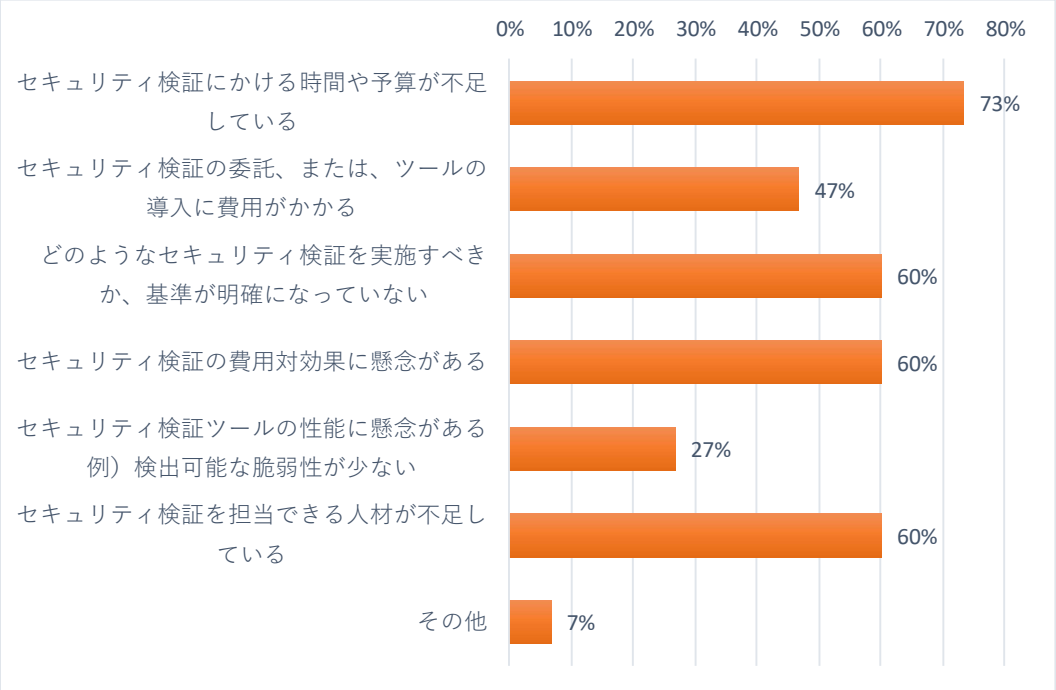


図 1-21　セキュリティ検証上の課題

1.2.2.3.4　OSS（ソフトウェア）利用上の課題及び、認証制度への要望の整理

本章の調査結果のまとめとして、OSS（ソフトウェア）利用上の課題及び、認証制度への要望事項を以下（表 1-7、表 1-8）に整理する。

表 1-7　OSS を含むソフトウェアの管理や対策、セキュリティ検証における課題

【ソフトウェアの更新や置換における課題】		
ID	**課題**	**想定される対応**
T1	更新されたソフトウェアコンポーネントを適用するためには、製品側でも品質検査や修正対応が必要となり、コスト負担が大きい。	顧客、一般消費者にセキュリティ対策の必要性を理解してもらうための普及啓発や情報公開。 ⇒本調査報告を含め、継続的な普及啓発活動が求められる
T2	ソフトウェアコンポーネントの置換による品質への影響懸念、他のコンポーネントへの置換の困難さ。	ソフトウェアの選定段階でのレビュー徹底（市場での利用実績や、高い保守性、脆弱性への対応実績など）。 ⇒本書 1.2.3.5 項「技術検証項目と実施ルール」にて提言する
【ソフトウェアコンポーネント管理上の課題】		
ID	**課題**	**想定される対応**
T3	ソフトウェアコンポーネント情報の管理方法、手順について、情報が不足し、正確さや網羅性に懸念がある。	SBOM の標準管理の方式、手順に関する情報の整備と普及啓発。 ⇒本書 1.2.3.3.1.2 項に SBOM の標準に関する参考情報を記載。また 1.2.3.5 項「技術検証項目と実施ルール」にて提言
【セキュリティ対策や検証の課題】		
ID	**課題**	**想定される対応**
T4	セキュリティ検証を含む対策コスト負担は、製品単価に反映されるが、セキュリティ品質が製品の購買に直結しない。	顧客、一般消費者にセキュリティ対策の必要性を理解してもらうための普及啓発や情報公開。 ⇒本調査報告を含め、継続的な普及啓発活動が求められる
T5	明確なソフトウェアに対するセキュリティ検証の基準や方法が定められていない。※各企業は独自の基準、方法で検証を実施しているが、対費用効果への懸念がある。	セキュリティ検証の基準や方法、手順の開発及び、情報の公開。 ⇒本書 1.2.3.5 項「技術検証項目と実施ルール」にて提言

表 1-8　認証制度における要望事項

【認証制度における検証の方式】		
ID	**要望事項**	**要望を踏まえた制度のポイント**
R1	第三者認証機関が検証を行う場合は、比較的対応負荷や費用（コスト負担）が大きい、自社での検証は対応負荷や費用は小さいが信頼度に課題。	自社又は委託先で適合性評価を行い、第三者認証機関が認証する方式 ⇒本書 1.2.4.4.2 項「認証制度や検証機関のプロセス・ルール提言」に反映
R2	第三者認証機関が検証を行う方式及び、自社で認証が完結する方式よりも、自社や委託先で検証を行ったものを第三者が認証する方式が望まれている。	
【自社や委託先で検証を行う場合の条件】		
ID	**要望事項**	**要望を踏まえた制度のポイント**
R3	検証機関の認定基準は、第三者機関への委託もしくは自社での検証実施が可能となるような、適度なハードルである事が望まれる。	・検証機関の認定を資格者の在籍等、一定の信頼性を得られる制度とする。 ⇒本書 1.2.4.4.2 項「認証制度や検証機関のプロセス・ルール提言」に反映
R4	一定の信頼性を確保するために、当該認証制度に合わせた資格制度は要望が高い。	・認証制度に対応する資格制度の検討。 ⇒本書 1.2.4.4.2 項「認証制度や検証機関のプロセス・ルール提言」」に反映
【第三者認証にあたり認証機関に提出可能な文書		
ID	**要望事項**	**要望を踏まえた制度のポイント**
R5	ソフトウェアコンポーネント一覧やソースコードは機密性が高く、認証機関であっても情報提出のハードルは高い。	認証機関への情報提出は、認証に関連する設計文書やセキュリティ検証結果やエビデンスまでが容認可能。 ⇒本書 1.2.4.4.2 項「認証制度や検証機関のプロセス・ルール提言」に反映
【認証を受ける際の手続き・仕組み】		
ID	**要望事項**	**要望を踏まえた制度のポイント**
R6	認証の申請、認証付与のプロセスが明確であり、申請のオンライン化や申請の進捗が把握できるようにすることが望まれている。	・認証の申請、認証付与のプロセスの分かりやすさ及び、情報開示が必要。またオンライン手続き等を活用した進捗状況の透明性の確保。 ⇒本書 1.2.4.4.2 項「認証制度や検証機関のプロセス・ルール提言」に反映

【セキュリティ認証制度全般に対する要望】		
ID	**要望事項**	**要望を踏まえた制度のポイント**
R7	取得費用が安くその金額が明確なことや、対応すべきセキュリティ基準が明確で信頼できることが望まれている。	・認証を取得するための費用が製品単価に照らして現実的であり、費用が明確に提示されている。対応すべきセキュリティの基準が明確であり、信頼できる。 ⇒本書 1.2.4.4.2 項「認証制度や検証機関のプロセス・ルール提言」」に反映
R8	国や業界団体によって取得が推奨されているなど、認証取得や検証のコストを顧客や消費者に説明しやすく、理解を得やすい制度運用が望まれる。	・資格取得が国や業界団体に推奨され、顧客や消費者の理解を得られやすい制度の検討。 ⇒本書 1.2.4.4.2 項「認証制度や検証機関のプロセス・ルール提言」に反映

1.2.3 ＜調査項目２＞OSS の安全な活用の技術検証活動のあるべき姿及び技術検証項目、検証のためのルール調査

1.2.3.1　調査の背景、目的

本調査項目（調査項目２）では、サプライチェーンセキュリティリスクに対処するために OSS を安全に活用するための技術検証活動のあるべき姿及び技術検証の項目、検証のためのルール等を調査分析して報告する。

1.2.3.2　OSS 起因を含むソフトウェアのインシデント事例調査

本項では、過去のインシデント事例の調査結果及び、その結果から必要とされるソフトウェア検証手法を示す。

調査では過去に発生したインシデント事例をもとに、発生したインシデントの原因を調査し、未然にインシデントの原因を検出するための検証手法及び、脆弱性情報が公開された後に、製品に組み込まれたソフトウェアコンポーネントに脆弱性が含まれていないかを検証する手法を導出している。

1.2.3.2.1　インシデント事例の調査

1.2.3.2.1.1　インシデント事例の調査対象文献

インシデント事例の調査は、国内、米国、欧州のセキュリティ関連省庁からの報告資料を中心に調査を行い、2014 年から 2021 年までの事例について、ソフトウェアの脆弱性や改ざんに関連する事例を 31 件抽出した。また 2019 年以前の事例については代表的な事例のみを抽出した。図 2-1 として、本調査で抽出したインシデント事例の報告年代別の内訳を示す。

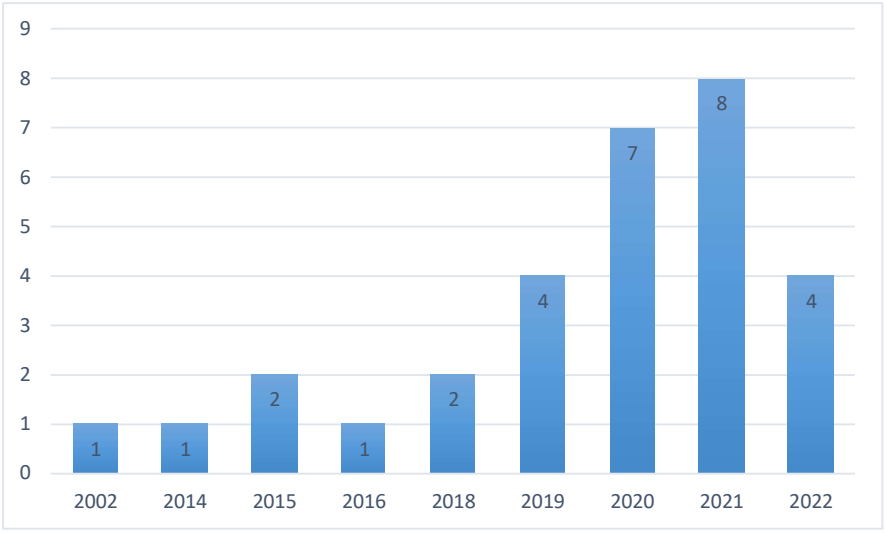


図 2-1　抽出したインシデント事例の年代別内訳

調査の対象とした主要な対象文書を以下に示す。

[主な調査対象文書]

- 経済産業省：

「サイバー・フィジカル・セキュリティ確保に向けたソフトウェア管理手法等検討タスクフォース報告資料）[2]」

- CISA（米国サイバーセキュリティ・インフラセキュリティ庁）：

“Defending Against Software Supply Chain Attacks[3]”

- ENISA（欧州 ネットワーク情報セキュリティ庁）：

“THREAT LANDSCAPE FOR SUPPLY CHAIN ATTACKS[4]”

- Linux Foundation

“Open Source Software Supply Chain Security[5]”

1.2.3.2.1.2 インシデント事例の調査結果一覧

本業務で調査を実施したインシデント事例を下記表 2-1 に示す。それぞれのインシデント事例については、脆弱性の概要と発生原因を調査し、一覧として整理している。

表 2-1 インシデント事例と脆弱性の概要と発生原因

No	事例タイトル	報告時期	脆弱性の概要・発生原因
1	依存パッケージの参照を悪用した攻撃方法の研究成果	2021 年 2 月	[概要] ・プライベートリポジトリに登録されているパッケージと同じ名称のものをパブリックパッケージに登録することで不正なパッケージをダウンロードさせる攻撃が可能である事が報告された。 [原因] ・プライベートリポジトリ内のソフトウェアと同名の不正なソフトウェアをパブリックリポジトリ（Github 等）へ登録したため、不正なソフトウェアをダウンロードさせる攻撃につながった。

[2] 経済産業省： 「サイバー・フィジカル・セキュリティ確保に向けたソフトウェア管理手法等検討タスクフォース報告資料）
https：//www.meti.go.jp/shingikai/mono_info_service/sangyo_cyber/wg_seido/wg_bunyaodan/software/index.html

[3] CISA：“Defending Against Software Supply Chain Attacks ”
https：//www.cisa.gov/publication/software-supply-chain-attacks

[4] ENISA：“THREAT LANDSCAPE FOR SUPPLY CHAIN ATTACKS ”
https：//www.enisa.europa.eu/news/enisa-news/understanding-the-increase-in-supply-chain-security-attacks

[5] Linux Foundation：“Open Source Software Supply Chain Security ”
https：//www.linuxfoundation.jp/wp-content/uploads/2020/02/oss_supply_chain_security.pdf

No	事例タイトル	報告時期	脆弱性の概要・発生原因
2	Linux,UNIX 系権限管理プログラム Polkit におけるローカル権限昇格の脆弱性：PwnKit	2022 年 1 月	[概要] ・2009 年に追加された機能に脆弱性があり、12 年以上報告されないまま対策が行われていなかった。比較的エクスプロイトがしやすい脆弱性であり、PoC も公開。
			[原因] ・メモリ破損につながる脆弱性
3	JavaScript のライブラリ開発者が自らリポジトリを改ざん：color.js, faker.js	2022 年 1 月	[概要] ・JavaScript ライブラリ colors.js, Faker.js において、作者自らが意図的に欠陥（アメリカ国旗が出力される）のあるバージョンを公開した。
			[原因] ・パブリックリポジトリの OSS の公開停止、改ざん
4	PyPI 管理者がリポジトリ上のマルウェアを発見し削除	2021 年 12 月	[概要] ・Python 系パッケージ管理リポジトリ PyPI にマルウェアが登録された。
			[原因] ・第三者による不正なパッケージの登録
5	Apache Log4j における任意のコードが実行可能な脆弱性：Log4shell	2021 年 11 月	[概要] ・Java ベースのロギングライブラリ Log4j の JNDI 機能に外部入力値の検証不備があり、任意の Java コードを実行可能な脆弱性があり、ランサムウェア等で悪用され、実害が発生
			[原因] ・標準で有効な機能（JNDI 機能）があり、Log4j を利用する多くのアプリケーションが同機能を呼び出す制御コードに対する入力確認が不十分
6	Realtek 社製の無線機器用 SDK にバッファオーバーフロー等の脆弱性	2021 年 8 月	[概要] ・Realtek 社製無線機器用 SDK に潜在する脆弱性であり、無線 LAN ルータや IoT 機器など多数の機器に影響する。
			[原因] ・バッファオーバーフロー脆弱性と、その脆弱性に起因するコマンドインジェクション脆弱性
7		2021 年 8 月	[概要]

No	事例タイトル	報告時期	脆弱性の概要・発生原因
	Arcadyan 製ソフトウェアベースの WiFi ルータにパストラバーサル脆弱性		・Arcadyan 社製の Web インターフェイスソフトウェアに潜在する脆弱性。 [原因] ・Web 管理機能におけるパストラバーサルの脆弱性
8	EOL の Broadcom 社製 SDK を利用しているルータ機器にリモートコード実行の脆弱性	2021 年 7 月	[概要] ・Cisco 社や Linksys 社等のルータ製品の UPnP 機能にリモートコード実行の脆弱性となる。 [原因] ・スタックベースのバッファオーバーフロー脆弱性 ・既知の脆弱性のあるソフトウェアの利用
9	リモート監視ソフト Kaseya VSA の脆弱性を悪用したランサムウェア攻撃	2021 年 7 月	[概要] ・米 Kaseya 社製のリモート監視ソフト VSA に当時のゼロディ脆弱性があり、悪用されるとランサムウェア REvil がダウンロード、実行される可能性がある。 [原因] ・認証バイパスの脆弱性
10	CI 環境の認証情報の流出による情報漏えい：Codecov	2021 年 5 月	[概要] ・コードカバレッジ計測サービス Codecov を利用している CI 環境において、個人情報やソースコードの漏洩のインシデントが発生した。 [原因] ・CI テストサービス提供の管理サーバへの不正アクセス
11	iOS 用ライブラリパッケージのリポジトリサーバにリモートコード実行の脆弱性	2021 年 4 月	[概要] ・iOS 用ライブラリパッケージソフト CocoaPods のリポジトリにリモートコード実行脆弱性があることが報告された。 [原因] ・リモートコード実行脆弱性があるリポジトリサーバのシステムに登録しているパッケージの改ざん
12	メンテナンス終了ソフトを含む TCP/IP の OSS ライブラリに複数の脆弱性：Amnesia33	2020 年 12 月	[概要] ・組み込み系機器に利用されているライブラリであり、広範かつ多数の機器に影響がある。 [原因] ・メモリ管理の不備に起因する複数の脆弱性

No	事例タイトル	報告時期	脆弱性の概要・発生原因
13	SolarWinds 社のソフトウェアアップデートに仕込まれたバックドア経由の情報漏えい	2020 年 12 月	[概要] ・SolarWinds 社のネットワーク監視ソフトウェア Orion Platform にマルウェアが仕込まれていた。
			[原因] ・ソフトウェアアップデート管理サーバへの不正アクセスに起因したマルウェアの拡散
14	IoT 機器のファームウェアに含まれる OSS の脆弱性の調査結果	2020 年 10 月	[概要] ・2000 年から 2019 年までに公開された国内外 13 社、1,510 種の IoT 機器の 5,712 個のファームウェアを解析し、2,379 種の OSS のバージョン情報とソースコードから脆弱性を調査。広範な OSS に脆弱性が潜在していることを報告。
			[原因] ・既知の脆弱性のあるソフトウェアの利用
15	Netlogon の特権昇格の脆弱性：Zerologon	2020 年 9 月	[概要] ・Netlogon の特権昇格の脆弱性「Zerologon」が報告された。攻撃によりドメイン管理者の権限が奪取され、ドメインに参加する全ての端末が制御下に置かれる可能性がある。
			[原因] ・不十分なランダム値の使用の脆弱性
16	Java 製 Web アプリケーションフレームワーク Apache Struts 2 の脆弱性	2020 年 8 月	[概要] ・Web アプリケーションフレームワーク Apache Struts 2 の 2 件の脆弱性に関する情報を公開された。
			[原因] ・制御コードの入力確認の不備（OGNL 機能）
17	ブートローダーGRUB2 の脆弱性：BootHole	2020 年 7 月	[概要] ・Linux 等で用いられるブートローダー「GRUB2」に脆弱性があり、「セキュアブート機能」を回避できることが確認された。
			[原因] ・バッファオーバーフロー脆弱性
18		2020 年 7 月	[概要]

No	事例タイトル	報告時期	脆弱性の概要・発生原因
	Windows DNS サーバの脆弱性：SIGRed		・Windows DNS Server の脆弱性として「SIGRed」が報告された。攻撃者が不正な DNS レスポンスを送信することで、DNS サーバ上で任意のコードが実行される可能性がある。
			[原因] ・バッファオーバーフロー脆弱性
19	組み込み系機器に利用される商用 TCP/IP ライブラリの脆弱性：Ripple20	2020 年 6 月	[概要] ・米 Treck 社製 TCP/IP ライブラリに 19 個の脆弱性があり、悪用により任意コード実行や情報の窃取、サービス不能状態につながる可能性がある。
			[原因] ・バッファオーバーフロー脆弱性 ・メモリの二重開放の脆弱性 ・不十分なランダム値の使用の脆弱性
20	航空機のエンタメシステムの脆弱性事例	2019 年 8 月	[概要] ・IOActive 社が、特定の脆弱性を用いて、機内エンターテイメントシステムから機体の操作に関わるネットワークに到達できることを発見してメーカに報告した。
			[原因] ・ソフトウェアの脆弱性（詳細は不明）
21	リアルタイム OS VxWorks 等における脆弱性（URGENT/11）	2019 年 7 月	[概要] ・WindRiver 社の VxWorks に 11 個の脆弱性があることを報告された。（非常に広範な産業や機器で影響）
			[原因] ・バッファオーバーフロー脆弱性
22	ruby 製 OSS ライブラリリポジトリの権限ハイジャックによるバックドア混入：strong_password	2019 年 7 月	[概要] ・ruby 製パスワード生成ライブラリの strong_password にバックドアが混入され、第三者によるリモートコード実行の脆弱性の挙動と当該 URL の攻撃者サイトへの送信の可能性があることが報告された。
			[原因] ・パブリックリポジトリの乗っ取りによる不正なコードの混入

No	事例タイトル	報告時期	脆弱性の概要・発生原因
23	ASUS 社端末におけるアップデート機能を悪用した攻撃	2019 年 3 月	[概要] ・正規のアップデートサーバが攻撃を受け、当該サーバから端末向けに配布されたアップデートファイルを介し、数十万の同社端末がマルウェアに感染するインシデントが発生した。
			[原因] ・ソフトウェアアップデート管理サーバへの不正アクセスに起因したマルウェアの拡散
24	JavaScript 製 OSS ライブラリリポジトリの権限ハイジャックによるバックドア混入：event-stream	2018 年 9 月	[概要] ・JavaScript 製 OSS ライブラリ event-stream の依存関係に不正なパッケージ flatmap-stream が混入され、ビットコインウォレットアプリ Copay で動作した場合にビットコインを攻撃者のウォレットに移動する挙動をするよう組み込まれていたことが報告された。
			[原因] ・パブリックリポジトリの乗っ取りによる不正なコードの混入
25	自動車内システムの脆弱性により制御が乗っ取られる恐れ	2018 年 2 月	[概要] ・中国 Tencent 社の Keen Security lab が、大手自動車メーカの自動車の脆弱性を検証してメーカに報告した。
			[原因] ・署名保護機能の不備により認証がバイパスされる脆弱性
26	Android 向けマルウェアにより悪意のあるコードが混入される恐れ：Triada	2016 年 4 月	[概要] ・Android デバイスに悪意のあるアプリをインストールしたり、スパム広告を表示させたりするマルウェア「Triada」が発見された。
			[原因] ・デバイスメーカの外注先のソフトウェアベンダによる不正コードの混入
27	多くの JavaScript 系ソフトウェアが依存して	2016 年 3 月	[概要] ・JavaScript 系のソフトウェアが依存していた left-pad パッケージの公開が停止され、left-pad に依存している

No	事例タイトル	報告時期	脆弱性の概要・発生原因
	いた left-pad の公開停止		パッケージのビルドやインストールでエラーが発生した。
			[原因] ・パブリックリポジトリの OSS の公開停止
28	iOS アプリの海賊版開発環境におけるアプリへのバックドア注入：XcodeGhost	2015 年 9 月	[概要] ・iOS アプリの開発環境の海賊版 Xcode で開発されたアプリに XcodeGhost と呼ばれる不正なコードが混入。
			[原因] ・不正な開発用ソフトウェアの利用
29	Linux の標準 C ライブラリ glibc におけるバッファオーバーフロー脆弱性：GHOST：2015	2015 年 1 月	[概要] ・Linux の標準 C ライブラリ glibc の gethostbyname の関数にバッファオーバーフローの脆弱性が潜在。
			[原因] ・バッファオーバーフロー脆弱性
30	OpenSSL 死活監視機能における脆弱性：Heartbleed	2014 年 12 月	[概要] ・OSS の暗号ライブラリ OpenSSL の死活監視機能に脆弱性があり、第三者によって秘密鍵などの重要な情報を取得される可能性がある。
			[原因] ・バッファオーバーフロー脆弱性
31	Linux 系プリンタドライバにおける GPL 違反事例	2002 年	[概要] ・リリースした Linux 用プリンタドライバ及びスキャナドライバについて、第三者から GPL 違反の指摘を受けて対応を行った
			[原因] ・OSS ライセンスの管理不備

1.2.3.2.2　インシデント事例に対する検証手法の調査

1.2.3.2.2.1　検証手法の整理

インシデント事例に対応する検証手法の検討にあたり、まずはガイドライン文書や事例調査の出典文書を参考に検証手法の整理を行った。具体的な検証手法のうち「技術検証（V-1～V-4）」に該当する項目については、経済産業省より公開された「機器のサイバーセキュ

リティ確保のためのセキュリティ検証の手引き[6]」及び、別冊文書を参考に一部記載を本調査にて修正した。「組織運用（O-1～O-3）」に該当する項目は、各事例調査の出典資料に記載されている対策内容を参考に、本書で追加を行った。

表 2-2　技術検証項目と組織運用項目[7]

ID	対策分類	名称	概要
V-1	技術検証	設計文書レビュー	機器の設計書を確認し、不適切なサービスや不適切な設定が存在しないか、適切なセキュリティ対策が組み込まれているかどうかを確認する。
V-2	技術検証	ファームウェア解析	機器のファームウェアを抽出する。脆弱性が含まれてないかを確認する。ファームウェア解析の手法は大きく以下に分類される。 ①ソースコード解析 ソースコードを確認し、要求を満たすか、環境固有値やエラーが存在しないか、処理フローに問題が無いか、規約違反が存在しないかを確認する。 ②バイナリ解析 ファームウェア等のバイナリコードについて、実行パスに異常は無いか、不正なアドレス命令が無いかを静的に確認する。
V-3	技術検証	既知脆弱性の診断	既知の脆弱性が機器に内在しうるかを調べ、実際に悪用可能かを確認する。（ペネトレーションテストを含む） ※既知の脆弱性を調査する過程で以下の手法が利用される。 ①ネットワークスキャン どのポートに対して通信可能か、接続が許可されていない機器やサービスが存在しないかを確認する。 ②ネットワークキャプチャ 機器やサービスのネットワークパケットを取得し、不審なパケットが無いかを確認する。
V-4	技術検証	ファジング	極端に長い文字列や記号の組み合わせ等、問題が起こりそうなデータや改変したデータを挿入し、その挙動を確認する。

[6] 経済産業省：「機器のサイバーセキュリティ確保のためのセキュリティ検証の手引き」を参考に一部記載を変更
https：//www.meti.go.jp/press/2021/04/20210419003/20210419003.html

ID	対策分類	名称	概要
O-1	組織運用	SBOM の管理	製品のソフトウェアの SBOM を管理し、ライセンス管理等に役立てる。
O-2	組織運用	コンプライアンス遵守	正規のソフトウェアを利用するよう組織運用する。
O-3	組織運用	リポジトリのベンダ化	リポジトリの公開停止による影響に対応するため、リポジトリの独自コピーを作成して管理する。

1.2.3.2.2.2　各インシデント事例の検証手法の調査

前項の検証手法を各インシデント事例に対応させた調査結果を表 2-3 に示す。発生原因を検出するための「A：インシデントを未然に防ぐための検証手法」と、公開された脆弱性情報に基づきソフトウェアの脆弱性を検出するための「B：ソフトウェアの脆弱性の検証手法」に区分し、それぞれに対して効果的な検証手法を一覧として整理している。

表 2-3 インシデント事例と検証方法の対応一覧

No	事例タイトル	A：インシデントを未然に防ぐための検証手法	B：ソフトウェアの脆弱性の検証手法
1	依存パッケージの参照を悪用した攻撃方法の研究成果	（組織運用 O-1：SBOM の管理） 不正なパッケージの参照を防ぐため、プライベートリポジトリとパブリックリポジトリのパケットを分別できるよう SBOM 上で管理する	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する
2	Linux,UNIX 系権限管理プログラム Polkit におけるローカル権限昇格の脆弱性：PwnKit	（技術検証 V-4：ファジング） メモリ破損等の脆弱性はファジングにより検出可能	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する
3	JavaScript のライブラリ開発者が自らリポジトリを改ざん：color.js, faker.js	（組織運用 O-3：リポジトリのベンダ化） リポジトリで管理されている OSS を利用する場合に、リポジトリの独自コピーを作成して管理する	（組織運用 O-3：リポジトリのベンダ化） リポジトリで管理されている OSS を利用する場合に、リポジトリの独自コピーを作成して管理する
4	PyPI 管理者がリポジトリ上のマルウェアを発見し削除	（技術検証 V-2：ファームウェア解析　※ソースコード解析） 不正なコードが混入されているかを確認するため、ソースコードの静的解析ツールにより、ソフトウェアをチェックし、不正コードを検出する	（技術検証 V-3：既知脆弱性の診断） 不正なパッケージのソフトウェアを利用しているか確認するため、該当するソフトウェアのバージョン情報を、製品の SBOM と照合する

No	事例タイトル	A：インシデントを未然に防ぐための検証手法	B：ソフトウェアの脆弱性の検証手法
5	Apache Log4j における任意のコードが実行可能な脆弱性：Log4shell	本脆弱性の検出にはファジングやソースコード解析を組み合わせた高度な解析手法が必要であり、一般的な単独の技術検証では検出困難 ※本脆弱性は制御コードに対する入力確認テストにより検出可能であるが、JNDI 機能用のテストデータが必要であり、このような特殊なテストデータを脆弱性が報告される前に用意することは難しい	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する
6	Realtek 社製の無線機器用 SDK にバッファオーバーフロー等の脆弱性	（技術検証 V-4：ファジング） バッファオーバーフロー系の脆弱性のため、ファジングにより検出できる可能性がある	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する
7	Arcadyan 製ソフトウェアベースの WiFi ルータにパストラバーサル脆弱性	（技術検証 V-3：既知脆弱性の診断） 事例のパストラバーサル脆弱性は、Tenable 社により発見、報告されており、Web アプリケーションを対象とする脆弱性スキャンで検出可能	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する
8	EOL の Broadcom 社製 SDK を利用しているルータ機器にリモートコード実行の脆弱性	（技術検証 V-4：ファジング） バッファオーバーフロー等の脆弱性はファジングにより検出できる可能性がある （技術検証 V-3：既知脆弱性の診断） 事例の脆弱性は、2011 年に Broadcom 社によってパッチが適用された脆弱性と同様であり、脆弱性情報と照合するツールにより検出できた可能性があった。	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する

No	事例タイトル	A：インシデントを未然に防ぐための検証手法	B：ソフトウェアの脆弱性の検証手法
9	リモート監視ソフト Kaseya VSA の脆弱性を悪用したランサムウェア攻撃	本事例の原因はゼロディの脆弱性であり、本脆弱性の検出にはファームウェア解析や他の手法を組み合わせた高度な解析手法が必要であり、一般的な単独の技術検証では検出困難	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する
10	CI 環境の認証情報の流出による情報漏えい：Codecov	本事例の Codecov のサービスは、開発用リポジトリへのアクセス権限を与えて利用するサービスであり、ユーザ企業による未然防止は困難	情報漏えいに関連する事例のため、技術検証等の該当なし
11	iOS 用ライブラリパッケージのリポジトリサーバにリモートコード実行の脆弱性	（組織運用 0-3：リポジトリのベンダ化） リポジトリで管理されている OSS を利用する場合に、リポジトリの独自コピーを作成して管理する	（組織運用 0-3：リポジトリのベンダ化） リポジトリで管理されている OSS を利用する場合に、リポジトリの独自コピーを作成して管理する
12	メンテナンス終了ソフトを含む TCP/IP の OSS ライブラリに複数の脆弱性：Amnesia33	（技術検証 V-4：ファジング） バッファオーバーフロー等の脆弱性はファジングにより検出できる可能性がある	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する
13	SolarWinds 社のソフトウェアアップデートに仕込まれたバックドア経由の情報漏えい	（技術検証 V-2：ファームウェア解析　※ソースコード解析） バックドアにつながる不正なコードが混入されているかを確認するため、ソースコードの静的解析ツールにより、ソフトウェアをチェックし、不正コードを検出する	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する

No	事例タイトル	A：インシデントを未然に防ぐための検証手法	B：ソフトウェアの脆弱性の検証手法
14	IoT 機器のファームウェアに含まれる OSS の脆弱性の調査結果	（技術検証 V-2：ファームウェア解析） 出典元の報告にある脆弱性は、ファームウェアの解析により、検出されている。	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する
15	Netlogon の特権昇格の脆弱性：Zerologon	本脆弱性の検出にはネットワークパケットキャプチャやバイナリ解析を組み合わせた高度な解析手法が必要であり、一般的な単独の技術検証では検出困難	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する
16	Java 製 Web アプリケーションフレームワーク Apache Struts 2 の脆弱性	本脆弱性の検出にはファジングやソースコード解析を組み合わせた高度な解析手法が必要であり、一般的な単独の技術検証では検出困難 ※本脆弱性は制御コードに対する入力確認テストにより検出可能であるが OGNL 機能のテストデータが必要であり、このような特殊なテストデータを脆弱性が報告される前に用意することは難しい	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する
17	ブートローダーGRUB2 の脆弱性：BootHole	（技術検証 V-4：ファジング） バッファオーバーフロー系の脆弱性のため、ファジングにより検出できる可能性がある	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する
18	Windows DNS サーバの脆弱性：SIGRed	本脆弱性の検出にはファジングやバイナリ解析を組み合わせた高度な解析手法が必要であり、一般的な単独の技術検証では検出困難	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する

No	事例タイトル	A：インシデントを未然に防ぐための検証手法	B：ソフトウェアの脆弱性の検証手法
19	組み込み系機器に利用される商用 TCP/IP ライブラリの脆弱性: Ripple20	（技術検証 V-4：ファジング） バッファオーバーフロー系の脆弱性のため、ファジングにより検出できる可能性がある	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する
20	航空機のエンタメシステムの脆弱性事例	詳細が不明のため検討不可	詳細が不明のため検討不可
21	リアルタイム OS VxWorks 等における脆弱性（URGENT/11）	（技術検証 V-4：ファジング） バッファオーバーフロー系の脆弱性のため、ファジングにより検出できる可能性がある	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する
22	ruby 製 OSS ライブラリリポジトリの権限ハイジャックによるバックドア混入：strong_password	（技術検証 V-2：ファームウェア解析　※ソースコード解析） 不正なコードが混入されているかを確認するため、ソースコードの静的解析ツールにより、ソフトウェアをチェックし、不正コードを検出する	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する
23	ASUS 社端末におけるアップデート機能を悪用した攻撃	（技術検証 V-2：ファームウェア解析　※ソースコード解析） バックドアにつながる不正なコードが混入されているかを確認するため、ソースコードの静的解析ツールにより、ソフトウェアをチェックし、不正コードを検出する	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する

No	事例タイトル	A：インシデントを未然に防ぐための検証手法	B：ソフトウェアの脆弱性の検証手法
24	JavaScript 製 OSS ライブラリリポジトリの権限ハイジャックによるバックドア混入：event-stream	（技術検証 V-2：ファームウェア解析　※ソースコード解析） 不正なコードが混入されているかを確認するため、ソースコードの静的解析ツールにより、ソフトウェアをチェックし、不正コードを検出する	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する
25	自動車内システムの脆弱性により制御が乗っ取られる恐れ	本脆弱性の検出にはネットワークパケットキャプチャやバイナリ解析を組み合わせた高度な解析手法が必要であり、一般的な単独の技術検証では検出困難	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する
26	Android 向けマルウェアにより悪意のあるコードが混入される恐れ：Triada	（技術検証 V-2：ファームウェア解析　※ソースコード解析） 不正なコードが混入されているかを確認するため、ソースコードの静的解析ツールにより、ソフトウェアをチェックし、不正コードを検出する	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する
27	多くの JavaScript 系ソフトウェアが依存していた left-pad の公開停止	（組織運用 O-3：リポジトリのベンダ化） リポジトリで管理されている OSS を利用する場合に、リポジトリの独自コピーを作成して管理する	（組織運用 O-3：リポジトリのベンダ化） リポジトリで管理されている OSS を利用する場合に、リポジトリの独自コピーを作成して管理する
28	iOS アプリの海賊版開発環境におけるアプリへのバックドア注入：XcodeGhost	（組織運用 O-2：コンプライアンス遵守） 正規のソフトウェアを利用するようルールを規定して組織運用する	（技術検証 V-2：ファームウェア解析　※ソースコード解析） 不正なコードが混入されているかを確認するため、ソースコードの静的解析ツールにより、ソフトウェアをチェックし、不正コードを検出する

No	事例タイトル	A：インシデントを未然に防ぐための検証手法	B：ソフトウェアの脆弱性の検証手法
29	Linux の標準 C ライブラリ glibc におけるバッファオーバーフロー脆弱性：GHOST：2015	（技術検証 V-4：ファジング） バッファオーバーフロー系の脆弱性のため、ファジングにより検出できる可能性がある	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する
30	OpenSSL 死活監視機能における脆弱性：Heartbleed	（技術検証 V-4：ファジング） バッファオーバーフロー系の脆弱性のため、ファジングにより検出可能（当該インシデントはファジングツールにより検出）	（技術検証 V-3：既知脆弱性の診断） 脆弱性のあるソフトウェアのバージョン情報を、製品の SBOM と照合する
31	Linux 系プリンタドライバにおける GPL 違反事例	（組織運用 O-1：SBOM の管理） 製品のソフトウェアの SBOM を管理し、利用する OSS ライセンスに準拠するよう運用する	ライセンス違反発覚後の対応となるため、技術検証は該当なし

1.2.3.2.3　ソフトウェアコンポーネントに求められる検証手法

本章のまとめとして、前項の表 2-3 を踏まえて、各インシデント事例の発生原因を５つの典型（①～⑤）に分類（表 2-4、図 2-1）し、それぞれの発生原因の典型において求められるソフトウェアコンポーネントの検証手法を整理した（表 2-5）。特に①「製品・サービスに組み込まれているソフトウェアの脆弱性」については、多様な機器やサービスに問題のあるソフトウェアが組み込まれているケースがあり、サプライチェーンにおいて特に影響が大きい脆弱性であることが想定される（例えば、Java ベースの製品の多くに組み込まれている Log4j など）。

表 2-4 インシデント事例の発生原因の分類一覧

インシデントの発生原因の分類	件数	該当事例 No（表 2-1 参照）
①製品・サービスに組み込まれているソフトウェアの脆弱性	15	2, 5, 6, 7, 8, 12, 14, 16, 17, 19, 21, 26, 28, 29, 30
②企業・ユーザが利用しているソフトウェアの脆弱性	7	9, 13, 15, 18, 20, 23, 25
③リポジトリ環境の脆弱性	2	10, 11
④リポジトリに不正な OSS パッケージが登録	6	1, 3, 4, 22, 24, 31
⑤OSS パッケージが公開停止	1	27

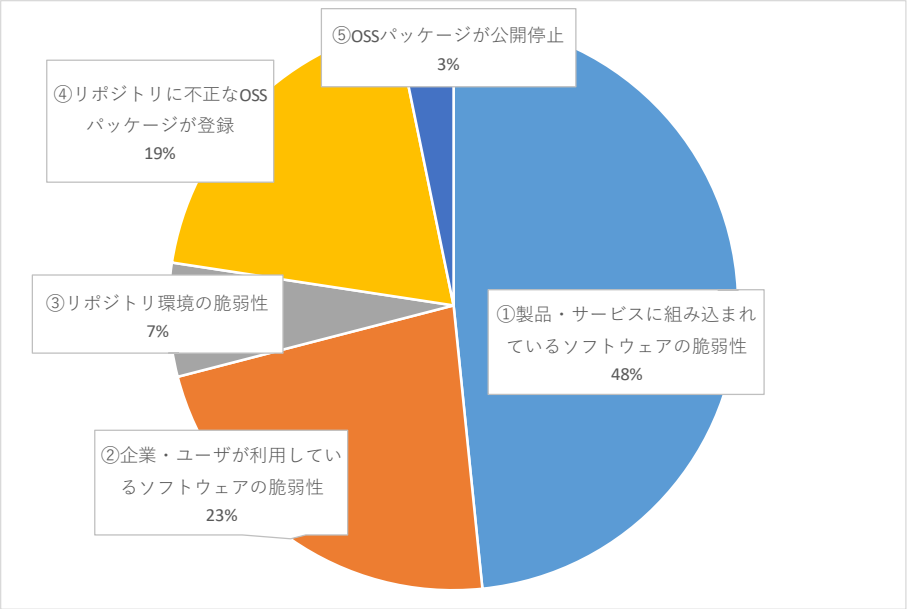


図 2-2　インシデント事例の発生原因の分類（グラフ）

表 2-5　ソフトウェアコンポーネントに求められる検証手法

インシデント発生原因の分類	A：インシデントを未然に防ぐための検証手法	B：ソフトウェアの脆弱性の検証手法
①製品・サービスに組み込まれているソフトウェアの脆弱性	● （技術検証 V-3：既知脆弱性の診断）脆弱性のあるソフトウェア情報と製品の SBOM の照合 ● （技術検証 V-3：既知脆弱性の診断）既知の脆弱性情報から生成したテストデータによる入力確認テスト ● （技術検証 V-4：ファジング）ファジングによる脆弱性の検出 ● （技術検証 V-2：ファームウェア解析 ※ソースコード解析）ソースコード解析による不正なコードの検出 ● （組織運用 O-2：コンプライアンス遵守）正規ソフトウェアの利用等のコンプライアンス遵守	● （技術検証 V-3：既知脆弱性の診断）脆弱性のあるソフトウェア情報と製品の SBOM の照合
②企業・ユーザが利用しているソフトウェアの脆弱性	● （技術検証 V-2：ファームウェア解析 ※ソースコード解析）ソースコード解析による不正なコードの検出 ●	● （技術検証 V-3：既知脆弱性の診断）脆弱性のあるソフトウェア情報と製品の SBOM の照合
③リポジトリ環境の脆弱性	● （組織運用 O-3：リポジトリのベンダ化）リポジトリの停止や不正なリポジトリ参照等に対するベンダ化（独自コピー）による管理	● （組織運用 O-3：リポジトリのベンダ化）リポジトリの停止や不正なリポジトリ参照等に対するベンダ化（独自コピー）による管理
④リポジトリに不正な OSS パッケージが登録	● （技術検証 V-2：ファームウェア解析 ※ソースコード解析）ソースコード解析による不正なコードの検出 ● （組織運用 O-1：SBOM の管理）SBOM によるライセンス管理とライセンス準拠対応 ● （組織運用 O-3：リポジトリのベンダ化）リポジトリの停止や不正なリポジトリ参照等に対するベンダ化（独自コピー）による管理	● （技術検証 V-3：既知脆弱性の診断）脆弱性のあるソフトウェア情報と製品の SBOM の照合
⑤OSS パッケージが公開停止	● （組織運用 O-3：リポジトリのベンダ化）リポジトリの停止や不正なリポジトリ参照等に対するベンダ化（独自コピー）による管理	● （組織運用 O-3：リポジトリのベンダ化）リポジトリの停止や不正なリポジトリ参照等に対するベンダ化（独自コピー）による管理

①製品・サービスに組み込まれているソフトウェアの脆弱性

【概要】

製品やサービスに組み込まれている OSS や SDK 等のソフトウェアの脆弱性に起因する事例は 15 件が該当する。主な事例として、「Apache Log4j における任意のコードが実行可能な脆弱性」や「メンテナンス終了ソフトを含む TCP/IP の OSS ライブラリに複数の脆弱性：Amnesia33」等がある。これらの事例の発生原因として、ソフトウェアの開発時に組み込まれた脆弱性や、メンテナンスが終了しているソフトウェアの利用、既知の脆弱性が指摘されているソフトウェアの利用が要因となっている。

【A：インシデントを未然に防ぐための検証手法】

まず、入力確認の不備による脆弱性に対して、制御コードのテスト等、過去の脆弱性の事例を参考に生成したテストデータによる入力確認テストを実施することで検出可能な事例がある。また、バッファオー

バーフロー脆弱性については、ファジングにより検出された事例もあり、ファジングテストによる脆弱性の検出が期待できる。

Broadcom 社製 SDK を利用しているルータ機器の事例では、過去に脆弱性が修正されたパッチが提供されているのにも関わらず、脆弱性のある SDK が利用されている事例があった。このような既知の脆弱性に対しては、SBOM と脆弱性情報を照合によるソフトウェアの脆弱性検出が有効な手法となる。また、こうした検証手法は一般的に既知の脆弱性を検出するスキャンツールに組み込まれている。

【B：ソフトウェアの脆弱性の検証手法】

脆弱性のあるソフトウェアが利用されているか検出する手法として、該当するソフトウェアの脆弱性情報と製品の SBOM の照合が有効となる。

②企業・ユーザが利用しているソフトウェアの脆弱性

【概要】

企業・ユーザが利用しているソフトウェアの脆弱性に起因する事例は 7 件が該当する。「Windows DNS サーバの脆弱性：SIGRed」や「Netlogon の特権昇格の脆弱性：Zerologon」の事例のように、企業やユーザが利用しているソフトウェア・サービスの脆弱性が悪用され、この脆弱性を起点としてマルウェアの拡散や情報漏えいの被害が発生している。また、「SolarWinds 社のソフトウェアアップデートに仕込まれたバックドア経由の情報漏えい」や「リモート監視ソフト Kaseya VSA の脆弱性を悪用したランサムウェア攻撃」の事例のように、ソフトウェアアップデートを提供するサーバが不正アクセスされた結果、当該サーバを踏み台としてマルウェアの拡散や不正アクセスが行われ、被害が伝播する事例も報告されている。

【A：インシデントを未然に防ぐための検証手法】

企業・ユーザが利用しているソフトウェアの脆弱性については、ソースコードや設計書等が必要なものもあり、高度な解析なため脆弱性が発覚する前に検出することは困難である。また、ASUS 社端末におけるアップデート機能を悪用した攻撃の事例では、バックドアにつながる不正なコードが混入されているかを確認するため、ソースコードの静的解析ツールにより、ソフトウェアをチェックし、不正コードの検出を行う。

【B：ソフトウェアの脆弱性の検証手法】

脆弱性のあるソフトウェアが利用されているかを検出する手法として、該当するソフトウェアの脆弱性情報と製品の SBOM の照合が有効である。

③リポジトリ環境の脆弱性

【概要】

リポジトリ環境の脆弱性に関連する事例は 2 件が該当する。本事例は、OSS を管理しているリポジトリに対する攻撃・脅威事例となっている。一般的に OSS の管理は Github や Sourceforge などのリポジトリ管理サービスで運用されており、リポジトリの編集等の権限が設定されている。これらの事例の中に

は、リポジトリの管理ユーザのアカウントの乗っ取りや、OSS コミュニティへの参加等の権限の窃取による攻撃の事例がある。

【A：インシデントを未然に防ぐための検証手法】

該当する事例はリポジトリ環境の脆弱性が原因であり、ソフトウェアコンポーネントを利用する企業側が検証等によって対策できるものではない。企業側で可能な対策としては、リポジトリの独自にコピーして管理するベンダ化（Vendoring）[8]と呼ばれる管理方法がある。リポジトリのバックアップを用意することで、リポジトリの停止に対して特に有効となる。

【B：ソフトウェアの脆弱性の検証手法】

上記 A と同様にソフトウェアコンポーネントを利用する企業側が検証等によって対策できるものではない。（対策についても上記 A と同様）

④リポジトリに不正な OSS パッケージが登録

【概要】

リポジトリに不正な OSS パッケージが登録された事例は 6 件が該当する。該当する事例としては、登録されている OSS に不正なコードやバックドアが混入しているケースや、既存 OSS パッケージ名に類似した名称の OSS を登録し、OSS を利用する開発者の打ち間違いを狙ったタイポスクワッティングと呼ばれる攻撃の事例も確認されている。

また、OSS パッケージが GPL 違反であることを気づかずに登録していた事例もあり、製品に組み込んだソフトウェアがライセンス標記を明確にされないまま二次配布を行った結果、訴訟リスクにつながった。

【A：インシデントを未然に防ぐための検証手法】

リポジトリに不正なコードが混入された事例への対策として、ソースコード解析ツールによるスキャンを実施することで、不正なコードの検出につながる。また、上記③同様にリポジトリを独自にコピーして管理するベンダ化の対策も有効である。タイポスクワッティング攻撃については、類似のパッケージ名をチェックするツールが公開されているが[9]、一般的にタイポスクワッティング攻撃は厳密な検出が難しいとされている[10]。

GPL 違反事例については、製品のソフトウェアの SBOM を管理し、利用する OSS ライセンスに準拠するよう運用することが対策として想定される。

---

[8] Hacking 3,000,000 apps at once through CocoaPods https：//justi.cz/security/2021/04/20/cocoapods-rce.html

[9] Nexus Repository Manager dependency/namespace confusion checker https：//github.com/sonatype-nexus-community/repo-diff

[10] Cloudsmith Supports OpenSSF's Efforts to Secure OSS https：//cloudsmith.com/resources/blog/cloudsmith-supports-openssfs-efforts-to-secure-oss

【Ｂ：ソフトウェアの脆弱性の検証手法】

リポジトリへの不正コードの混入については、該当するソフトウェアコンポーネントのバージョン情報（脆弱性情報）を製品の SBOM と照合することで検出することが可能である。

⑤OSS パッケージが公開停止

【概要】

OSS パッケージが公開停止された事例は 1 件が該当する。本事例では JavaScript 系のソフトウェアが依存していた left-pad パッケージの公開が開発者自身により停止され、left-pad に依存しているパッケージのビルドやインストール時にエラーが発生した。

【A：インシデントを未然に防ぐための検証手法】

該当する事例は開発者自身がパッケージの公開を停止したことに起因しており、検証等の対策によって企業側で対応できるものではない。企業側で可能な対策としては、上記③と同様に、リポジトリの独自にコピーして管理するベンダ化が想定され、リポジトリのバックアップを用意することで、突然の公開停止に対しては特に有効となる。

【B：ソフトウェアの脆弱性の検証手法】

上記 A と同様に、開発者自身がパッケージの公開を停止したことに起因しており、検証等の対策によって企業側で対応できるものではない。（対策についても上記 A と同様）

1.2.3.3　国内外の関連ガイドライン調査結果

本項では国内外の IoT に関係するガイドライン文書や規格の調査を行い、OSS を含むソフトウェアのセキュリティに関する技術検証項目及び、組織マネジメント上の管理項目に類する事項を選定する。

1.2.3.3.1　調査対象のガイドライン一覧

1.2.3.3.1.1　要件調査

サプライチェーンを構成している OSS は、サードパーティソフトウェアであるという観点から、OSS に関する検証技術に関連する項目があるセキュリティガイドラインとして、主にソフトウェア開発および IoT サプライチェーンに関連するガイドラインを調査対象としての選定基準に設定した。その結果、本調査で選定した 4 つのガイドラインを表 2-6 に示す。

表 2-6　ソフトウェア開発およびサプライチェーンに関わるセキュリティガイドライン

対象文書	発行機関	発行日	文書の概要
NIST Special Publication 800-218 Secure Software Development Framework （SSDF） Version 1.1	米国商務省、米国国立標準技術研究所（以下 NIST）	2022 年 2 月	ソフトウェア開発プロセスに関して、これまでの一般的なソフトウェア開発ライフサイクル（SDLC）モデルではあまり扱われてこなかったセキュリティの観点をまとめた、セキュアソフトウェア開発フレームワーク（SSDF）。
Recommended Criteria for Cybersecurity Labeling for Consumer Internet of Things （IoT） Products	NIST	2022 年 2 月	大統領令 14028 に基づく、特にコンシューマ向けの IoT 製品のラベリングのためのセキュリティ基準。IoT のサプライチェーンを考慮したセキュリティ対策が記載されている。
Recommended Criteria for Cybersecurity Labeling of Consumer Software	NIST	2022 年 2 月	大統領令 14028 に基づく、コンシューマ向けソフトウェアのラベリングのためのセキュリティ基準。NIST SP 800-218 （SSDF）と整合を取る形でソフトウェアのセキュリティ対策が記載されている。
Guidelines for Securing the Internet of Things	欧州 ネットワーク情報セキュリティ庁（以下 ENISA）	2020 年 11 月	IoT 製品のサプライチェーンを保護するためのガイドライン。要件定義、設計から運用、保守、廃棄に至るまでの製品ライフサイクルにわたるセキュリティ対策が記載されている。

さらに、表 2-7 に示した包括的なセキュリティフレームワークである 2 つのガイドラインについても、サプライチェーンのセキュリティの観点で参考となる項目があり、これらも調査対象とした。

表 2-7　包括的なセキュリティフレームワーク

対象文書	発行機関	発行日	文書の概要
NIST Special Publications 800-53 Rev.5 / NIST Special Publications 800-53B	NIST	2020 年 10 月	情報システム・組織向けのセキュリティ・プライバシー管理策カタログ。第 4 版より、サプライチェーンリスク管理（SCRM）に関する管理策ファミリが新設。
サイバー・フィジカル・セキュリティ対策フレームワーク（CPSF）	経済産業省	2019 年 4 月	サプライチェーン全体のサイバーセキュリティ確保を目的として、産業に求められるセキュリティ対策の全体像をフレームワークとして整理している。

1.2.3.3.1.2　その他の参考ガイドライン

前項にて調査対象としたガイドラインの他にも一部参考にしたガイドラインを以下表 2-8 に示す。

表 2-8 その他の参考ガイドライン

対象文書	発行機関	発行日	文書の概要
機器のサイバーセキュリティ確保のためのセキュリティ検証の手引き（別冊文書含む）	経済産業省	2021 年 4 月	セキュリティ検証サービスの高度化を目的とした IoT 機器に対する検証手法の解説書。別冊 1、2 には SBOM の生成と脆弱性検証に対応したツールが紹介されている。
NTIA Software Component Transparency	米国 電気通信情報局（NTIA）	2021 年 10 月	SBOM の具体的な項目や、サプライチェーンにおける運用方法について例を交えて整理している。
IoT 機器セキュリティ要件ガイドライン 2021 年版：CCDS-GR01-2021 Ver. 2.0	重要生活機器連携セキュリティ協議会（CCDS）	2021 年 6 月	本ガイドラインは、つながる機器における最低限守るべき要件（対策レベル：★星一つ）を定義している。つながる機器を用いた IoT 機器、及びシステムにおける最低限守るべき要件としての適用を想定している。

■SBOM の具体的な項目

NTIA Software Component Transparency に記載されている、SBOM として必要な具体的な項目を表 2-9 に掲載する。

表 2-9 NTIA による SBOM の具体的な項目の定義

SBOM の属性	意味	特記事項
Author Name **作者名**	SBOM の作者	必ずサプライヤであるとは限らない。その場合、その SBOM はサプライヤが作成したものではないということを示す。
Timestamp **タイムスタンプ**	SBOM の最終更新日時	SBOM エントリが変更されたときに必ず更新する。
Supplier Name **サプライヤ名**	SBOM エントリ内のコンポーネントのサプライヤの名前またはその他の識別子	複数の名前を扱えること。作者名とサプライヤ名が同じ場合、サプライヤが自身のコンポーネントについて SBOM を作成したということを示す。異なる場合は、コンポーネントのサプライヤではない者がそのコンポーネントについての主張を行っているということを示す。
Component Name **コンポーネント名**	コンポーネントの名前またはその他の識別子	複数の名前を扱えること。作者名かサプライヤがコンポーネント名を決める。コンポーネント名はサプライヤ名を伝えることができる。
Version String **バージョン文字列**	コンポーネントのバージョン	サプライヤと作者はバージョンスキームを自由に選択してよい。
Component Hash **コンポーネントハッシュ**	コンポーネントの暗号化ハッシュ	ハッシュの代わりにデジタル署名を使うことも可能だがキー管理や署名検証などが必要となる。複数のハッシュを提供することが可能であり、ソースコンポーネントのハッシュ、バイナリコンポーネントのハッシュ、およびコンポーネントのコレクションのハッシュを含めてもよい。
Unique Identifier **ユニーク ID**	コンポーネントを一意に定義するのに役立つ追加情報	グローバルに一意の階層または名前空間に関連して生成することも、既存のグローバル座標系を参照することも可能。Common Platform Enumeration（CPE）、Package URL（PURL）、Universal Unique Identifier（UUID）（Globally Unique Identifier [GUID]）、Software Heritage ID（SWHID）などが使用可能。
Relationship **関係性**	SBOM コンポーネント間の関連付け	デフォルトの関係性タイプは includes であり、これはある別の上流コンポーネントを含む、あるいはそれに依存していることを示す。関係性タイプ primary はそのコンポーネントの上流に依存がないことを示す。

1.2.3.3.2　ガイドラインからまとめた体系的なセキュリティプラクティス

前項 1.2.3.3.1 で調査対象とした各ガイドラインから、セキュリティプラクティスを抽出して整理した。OSS の観点に関わらず、表 2-6 のガイドラインについては全項目を対象、表 2-7、表 2-8 のガイドラインについては IoT 製品提供企業におけるセキュリティ対策の参考となるプラクティスを抽出した。抽出したプラクティスから同様の項目をまとめて、体系立てて整理したものを表 2-10～表 2-12 に示す。

技術に関するものを大きく「機能」「セキュア開発」「テスト」に分類し、それぞれ表 2-10～表 2-12 に示している。また、組織運用に関するものを大きく「組織・体制」「サプライチェーン」「脆弱性対応」「その他」に分類し、それぞれ表 2-13～表 2-16 に示している。それぞれの分類の中を、さらに大項目、中項目、小項目のレベルに細分化した。小項目に記載しているものがセキュリティプラクティスとなる。

小項目のセキュリティプラクティスごとに、その項目を検証するとした場合にその対象が何であるかを検証対象欄に記載している。検証対象として、「IoT 製品」「ソフトウェア」「自社開発ソフトウェア」「サードパーティソフトウェア」「商用ソフトウェア」「OSS」「ハードウェア」「組織運用」に区分した。それらの簡単な包含関係を図 2-3 に示す。

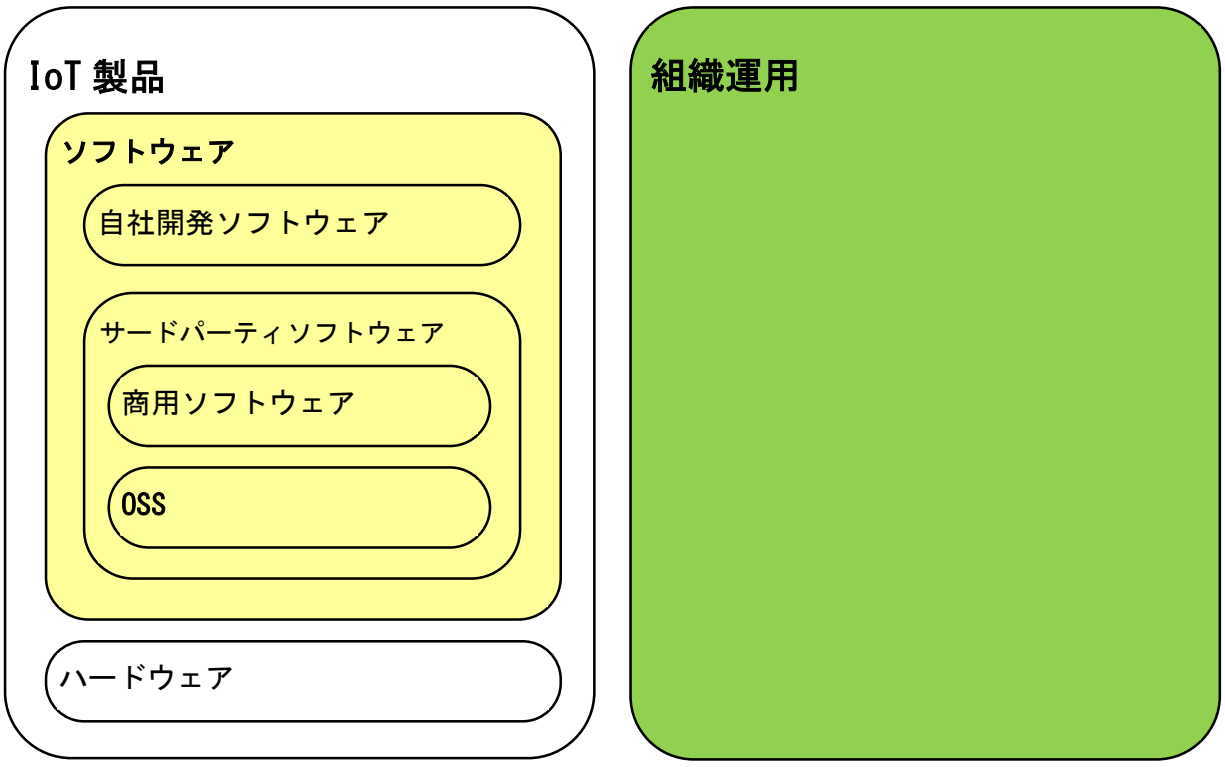


図 2-3　検証対象とその関係

なお、調査したガイドラインでは「商用ソフトウェア」および「OSS」を個別に対象とするようなセキュリティプラクティスはなかったため、表 2-10～表 2-15 にはこれらの区分は出てこない。

1.2.3.3.2.1　セキュリティプラクティス（技術項目）：表 2-10～表 2-12

表 2-10 セキュリティプラクティス（技術） ― 機能

ID	大項目	中項目	小項目	検証対象
FNC-01	インタフェース制御	不要な I/F は削除/無効化する。	システム運用上、開放が不要な TCP/UDP ポートは停止しておく。	IoT 製品
FNC-02			Wi-Fi、Bluetooth、USB などの I/F は必ず製品機能に必要なもののみ装備する。	IoT 製品
FNC-03		必要な I/F は認証を行う。	システム運用上、開放が必要なポートについては、脆弱性検査により、所定の合格基準を満たしていることを提示すること。	IoT 製品
FNC-04			・Wi-Fi について、最新の認証方式が実装されていること。 ・Bluetooth について、最新のペアリング方式が実装されていること。 ・USB について、適切なアクセス制御およびアクセス権限の制限を行っていること。	IoT 製品
FNC-05		通信データフォーマットを検証する。	・Wi-Fi について、最新の認証方式が実装されていること。 ・Bluetooth について、最新のペアリング方式が実装されていること。 ・USB について、適切なアクセス制御およびアクセス権限の制限を行っていること。	IoT 製品
FNC-06	データ保護	保存データを保護する。	機器本体のメモリ領域へ保存される情報資産については、不正なアクセスや変更から保護することができること。（SD カード等の、ストレージメディアに格納するデータについても同様）	IoT 製品
FNC-07			機器内に認証情報（パスワード、秘密鍵など）を保存する場合、ネットワーク経由での不正アクセス（改ざん、盗聴など）から保護された領域で管理すること。	IoT 製品

ID	大項目	中項目	小項目	検証対象
FNC-08		通信データを保護する。	他の IoT 機器やサーバ（クラウド上のサーバを含む）へ送信される情報資産について、情報の漏えいや変更から保護することができること。	IoT 製品
FNC-09		不要になったデータは安全に削除する。	ユーザデータおよび設定データを安全に消去する機能を持つこと。	IoT 製品
FNC-10	セキュリティの維持	一意な製品識別を行う。	製品を一意に識別できるようにすること。	IoT 製品
FNC-11			製品のすべてのコンポーネントのインベントリを作成する。	IoT 製品
FNC-12		設定変更をセキュアに行える。	設定変更機能は、特権的ユーザ以外による機能の実行を制限すること。	IoT 製品
FNC-13			セキュアなデフォルト設定に復元できること。	IoT 製品
FNC-14		ソフトウェアアップデートの手段を持つ。	ソフトウェアを最新の状態に保つ手段を持つこと。	IoT 製品/ソフトウェア
FNC-15			アップデートは検証してから適用すること。	IoT 製品/ソフトウェア
FNC-16		ログを記録する。	監査証跡の取得機能、蓄積機能を実装し、特権的ユーザによる監査証跡の読み出しを可能とすること。	IoT 製品
FNC-17			・監査証跡については監査に必要な容量を確保※し、監査証跡の保存容量を超過した場合には、古い記録から順次上書きするなど、管理上の対策を行うこと。 ・監査証跡は不正な情報削除や変更を防止する対策を行うこと。	IoT 製品
FNC-18			セキュリティイベントの監査証跡の発生日時を記録するため、時間管理機能を有すること。	IoT 製品

ID	大項目	中項目	小項目	検証対象
FNC-19	セキュアな実装	ユニークな認証情報を設定する。	・認証情報の設定変更を可能とすること。 ・工場出荷後の初回起動時には、強制的にパスワードを変更する機能を実装すること。	IoT 製品
FNC-20			暗号化、パスワード、またはその他の認証方法に使用されるシークレットをソフトウェア内にハードコーディングしないこと。	IoT 製品
FNC-21		実行コードの改ざん検知/防止機能を実装する。	ハッシュやコード署名などを使用して、実行可能ファイルの完全性を保護する。	IoT 製品/ ソフトウェア
FNC-22			コードを難読化する。	IoT 製品/ ソフトウェア
FNC-23		強力な暗号を使用する。	NIST などの標準に従った強力な暗号を用いること。	IoT 製品/ ソフトウェア
FNC-24		ハードウェアレベルの認証メカニズムを実装する。		IoT 製品/ ハードウェア

表 2-11 セキュリティプラクティス（技術） ― セキュア開発

ID	大項目	中項目	小項目	検証対象
SDL-01	リスクアセスメント	脅威分析を行う。	・システムに対する脅威および脆弱性を特定する。 ・システム、およびそのシステムが処理/保存/伝送する情報、およびなんらか関連する情報への認可されていないアクセス、使用、開示、中断、改変、または破壊による被害の可能性と規模を判断する。 ・脅威分析結果を文書化する。 ・脅威分析結果を定期的にレビューする 。	IoT 製品/ ソフトウェア
SDL-02		対策の検討を行う。	・ソフトウェア設計が、適用可能なセキュリティ要件に対応していることを確認する。 ・ソフトウェア設計が、リスクモデルによって特定されたリスクに十分に対処していることを確認する。 ・レビューの結果に応じて、要件を満たすために障害を修正する。 ・セキュリティ要件を満たせない場合に、設計および/またはリスク対応戦略を変更する。	IoT 製品/ ソフトウェア
SDL-03	実装	セキュアコーディングを実施する。	・すべての入力を検証し、すべての出力を検証し、適切にエンコードする。 ・安全でない関数と呼び出しの使用を避ける。 ・エラーを検出し、それらを適切に処理する。 ・ログ機能とトレース機能を提供する。 ・開発言語と環境に共通する他の脆弱性を確認する。	ソフトウェア
SDL-04			・セキュアコーディングプラクティスを使用することをサポートする自動化された機能を備えた開発環境を使用する。	組織運用

表 2-12 セキュリティプラクティス（技術） ― テスト

ID	大項目	中項目	小項目	検証対象
TST-01	静的解析	ツールを使った静的解析を行う。	・静的分析ツールを使用して、コードの脆弱性をチェックする。ツールによって報告された問題は人手でレビューし、必要に応じて修正する。 ・ソースコードのスタイルとフォーマットを標準化するためにツール（例えば、リンタ、フォーマッタ）を使用する。	ソフトウェア
TST-02			・自動化ツールを使用して、ソースコードがリポジトリにチェックインされる際に、安全でないコーディングを特定し、修復できるようにする。	組織運用
TST-03		ピアレビューを行う。	・コードのピアレビューを実行し、ピアレビューの一部として既存のコードレビュー、分析、またはテスト結果を確認する。 ・ピアレビュープロセスを容易にするピアレビューツールを使用し、すべてのディスカッションおよびその他のフィードバックを文書化する。 ・レビューチェックリストを使用して、コードが要件に準拠していることを確認する。	自社開発ソフトウェア
TST-04			・エキスパートによって、バックドアやその他の悪意のあるコンテンツのコードをチェックする。	自社開発ソフトウェア
TST-05		レビュー実施後	・検出された問題の根本原因を特定し、記録する。 ・コードレビューから得られた教訓を、開発者がアクセスして検索できるよう wiki に残す。	組織運用

ID	大項目	中項目	小項目	検証対象
TST-06	動的解析	動的解析を実施する。	・セキュリティ機能についての機能テストを実行する。	IoT 製品/ ソフトウェア
TST-07			・ファジングテストツールを使用して、入力処理に関する問題を見つける。	IoT 製品/ ソフトウェア
TST-08			・脆弱性スキャンを実施する。	IoT 製品/ ソフトウェア
TST-09			・ペネトレーションテストを実施する。(攻撃者が危険度の高いシナリオでソフトウェアを侵害しようとする方法をシミュレートする)。	IoT 製品/ ソフトウェア
TST-10			・動的な脆弱性テストをプロジェクトの自動テストスイートに統合する。	組織運用
TST-11		動的解析の実施後	・検出された問題の根本原因を特定し、記録する。 ・コードテストから得られた教訓を、開発者がアクセスして検索できるよう wiki に残す。	組織運用

1.2.3.3.2.2　セキュリティプラクティス（組織・運用）：表 2-13～表 2-16

表 2-13 セキュリティプラクティス（組織・運用） ― 組織・体制

ID	大項目	中項目	小項目	検証対象
ORG-01	開発体制・プロセス定義	関係者に役割と責任を割り当て、教育を行い、全社的にセキュリティに取り組む。	・SDLC 関連の役割と責任を定義する。 ・役割ごとに教育計画を作成する。 ・教育を実施する。 ・効率的な改善のため、教育の成果のパフォーマンスを測定する。 ・一定期間や、特定のイベントの後に教育コンテンツを更新する。	組織運用
ORG-02		ソフトウェア開発環境とプロセスに関するセキュリティ要件を定義し、文書化する。	・ソフトウェアを作成および保守する際に従うべきセキュリティプラクティスを確立する。 ・開発エンドポイントを含むソフトウェア開発インフラストラクチャとそのコンポーネントを SDLC 全体でセキュリティ保護し、そのセキュリティを維持するためのポリシーを定義する。 ・SDLC 全体でソフトウェア開発プロセスをセキュリティ保護し、開発中のソフトウェアが利用するオープンソースやその他のサードパーティ製ソフトウェア コンポーネントを含む、そのセキュリティを維持するためのポリシーを定義する。 ・セキュリティ要件を少なくとも毎年見直し、更新する場合、または内部または外部のソースから新しい要件が発生した場合、またはソフトウェア開発インフラストラクチャを対象とする大規模なセキュリティ インシデントが発生した場合。 ・OSS の使用に関する規則を定め、派生使用に関するライセンスなどの問題を考慮する。	組織運用

ID	大項目	中項目	小項目	検証対象
ORG-03		定量的なセキュリティ基準を設定する。	・主要業績評価指標 （KPI)、主要リスク指標 （KRI)、脆弱性重大度スコア、およびソフトウェア セキュリティに関するその他の対策を定義する。 ・既存のチェックにソフトウェアセキュリティ基準を追加する（例： アジャイル SDLC の方法論で完了の定義)。 ・ソフトウェア開発ワークフロー システムの一部として生成された成果物を確認して、それらが基準を満たしているかどうかを判断する。 ・ワークフローおよび追跡システムの一部として、セキュリティ チェックの承認、拒否、および例外要求を記録する。 ・各開発プロジェクトのセキュリティの成功と失敗のコンテキストで収集されたデータを分析し、その結果を使用して SDLC を改善する。	組織運用
ORG-04		テスト計画	・コードテストを実行するタイミングと、コードテストの実施方法（サンドボックス環境内で行うなど） に関するポリシーまたはガイドラインに従う。 ・ソフトウェアのステージに基づいてテスト方法を選択する。 ・エラーが再導入されることがないよう、過去にあった脆弱性に関するテストをプロジェクトのテスト項目に組み込む。	組織運用

ID	大項目	中項目	小項目	検証対象
ORG-05	セキュリティ要件定義	製品/ソフトウェアのセキュリティ要件を定義し、文書化する。	以下について定義し、文書化すること。 ・開発プロセスの中で行われた仮定や製品に関係する想定など。 ⇒想定ユースケースや設置環境、ネットワークアクセス要件、想定される入出力データ、想定されるセキュリティ要件、関連法規、想定される寿命、など ・製品の設計関連 ⇒ハードウェアおよびソフトウェア（OSS、サードパーティソフト、内作)、セキュリティ要素（セキュアブートなど)、セキュアなソフトウェア開発プラクティス、セキュリティ認証結果、など ・セキュアなメンテナンスのための要件 ・リリース時、使用中、サポート終了後などのライフサイクルにわたって考慮されたセキュリティ ・脆弱性管理ポリシー	IoT 製品/ソフトウェア

ID	大項目	中項目	小項目	検証対象
ORG-06		製品/ソフトウェアのセキュアな設定/構成を決定し文書化する。	・ソフトウェア コンポーネントの安全な構成を決定し、開発者が構成を簡単に使用できるように、これらを使用可能にする （例えば、コードとしての構成として)。 ・ソフトウェアに対して承認された構成が正しく行われていることを確認する。 ・各設定の目的、オプション、デフォルト値、セキュリティの関連性、潜在的な操作上の影響、およびその他の設定との関係を文書化する。 ・プログラムによる技術的メカニズムを使用して、ソフトウェア管理者が各設定を実装および評価する方法を記録する。 ・デフォルトの構成を使用できる形式で保存し、変更管理の手法に従って変更する（例えば、コードとしての構成)。	IoT 製品/ソフトウェア
ORG-07		設定した定量的なセキュリティ基準に基づいて測定する。	・ツールチェーンを使用して、セキュリティの意思決定を知らせる情報を自動的に収集する。 ・条件をサポートする情報の生成と収集をサポートするために必要な場合は、追加のツールを展開する。 ・条件を使用して意思決定プロセスを自動化し、これらのプロセスを定期的に見直す。 ・権限を持つ担当者のみが収集した情報にアクセスできるようにし、情報の変更や削除を防止する。	IoT 製品/ソフトウェア

ID	大項目	中項目	小項目	検証対象
ORG-08	開発環境	開発環境のネットワークを分離し、アクセス制御する。	・ネットワークセグメンテーションとアクセス制御を使用して、環境を相互に分離し、運用環境以外の環境内でコンポーネントを互いに分離して、攻撃対象領域や攻撃者の横移動やアクセス権/アクセスのエスカレーションを削減する。 ・認証を強制し、必要な情報のみにインターネットへのアクセスを最小限に抑えるなど、各ソフトウェア開発環境に出入りする接続を厳しく制限する。 ・ビルド サービスなどのツールチェーン システムへの直接アクセスを最小限にする。すべてのアクセス試行と特権アクセスのすべての使用を継続的に監視および監査する。 ・環境間、および各環境内のコンポーネント間の認可およびアクセスに関する信頼関係を定期的にログ、監視、および監査する。	組織運用
ORG-09			・ゼロ信頼アーキテクチャーに従って、環境のホスティング・インフラストラクチャを保護するための手段を構成および実装する。	組織運用
ORG-10		開発に使うツールを適切に導入し、設定する。	・ツールチェーンのカテゴリを定義し、各カテゴリに使用する必須のツールまたはツールタイプを指定する。 ・ツールを評価、選択、および取得し、各ツールのセキュリティを評価する。 ・ツールの脆弱性に対処したり、新しいツール機能を追加したりするために、必要に応じてツールを更新、アップグレード、または置換する。	組織運用

ID	大項目	中項目	小項目	検証対象
ORG-11		適切なコンパイラを適切な設定で使用する。	・コンパイラ、インタプリタ、およびビルドツールの最新バージョンを使用する。 ・コンパイラ、インタプリタ、およびビルドツールの信頼性と整合性を定期的に検証する。 ・コンパイルプロセス中に、セキュリティで保護されていないコードに対して警告を生成するコンパイラ機能を有効にする。 ・すべてのコンパイラ警告がエラーとして扱われ、誤検出または無関係であると判断されたものを除き、削除される「クリーン ビルド」の概念を実装する。 ・メモリ位置の使用状況など、実行特性をランダム化または難読化するコンパイラ機能を有効にする。 ・機能が期待どおりに動作し、不注意で操作上の問題やその他の問題が発生していないことを確認するテストを行う。	組織運用
ORG-12			・コンパイラ、インタプリタ、およびビルドツールを配備または更新する際の変更管理プロセスに従い、ツールに対する予期しない変更をすべて監査する。 ・承認された構成が使用されていることを継続的に確認する。 ・承認されたツール構成をコードとして構成できるようにして、開発者が容易に使用できるようにする。	組織運用
ORG-13		構成管理を行う。	・ソフトウェア構成管理のためのリポジトリを保守する。 ・出自データを組織のポリシーに従ってソフトウェアの取得者が利用できるようにする。 ・出自データを組織の運用チームおよび対応チームが利用できるよう	組織運用

ID	大項目	中項目	小項目	検証対象
			にして、ソフトウェアの脆弱性を軽減する支援を行う。 ・ソフトウェアのコンポーネントのいずれかが更新されるたびに、出自データを更新する。 ・フィールドに展開済みの古いバージョンのソフトウェアについて、新しいバージョンへの移行が正常に完了するまで、古いバージョンのソフトウェアを保持する。	
ORG-14			・出自データの整合性を保護し、受信者が出自データの整合性を検証する方法を提供する。	組織運用
ORG-15			・出自データは標準形式を使用する。	組織運用
ORG-16		構成管理により、ソースコードの改ざん防止策を講じる。	・すべてのソースコードとコードの構成をコードリポジトリに格納し、コードの性質に基づいてアクセスを制限する。 ・リポジトリのバージョン管理機能を使用して、個々のアカウントに対する説明責任を持ってコードに加えられたすべての変更を追跡する。 ・コード所有者に、他のユーザがコードに加えたすべての変更を確認し、承認する。 ・リリース・ファイル、関連イメージなどを、組織の確立されたポリシーに従ってリポジトリに保管する。必要な担当者が読み取り専用でアクセスできるようにし、他のユーザによるアクセスを許可しない。	組織運用
ORG-17			・暗号化 （暗号化のハッシュなど） を使用して、ファイルの整合性を保護する。 ・リリース整合性検証情報を、リリース ファイルとは別の場所に保管したり、データに署名したりして保管し、保護する。	組織運用

表 2-14 セキュリティプラクティス（組織・運用） ― サプライチェーン

ID	大項目	中項目	小項目	検証対象
SPC-01	サプライチェーンのセキュリティモデルの構築	サプライチェーンを考慮した標準、モデルを開発する。	IoT サプライチェーンのための標準を作る、もしくは既存の標準を IoT サプライチェーン適用する。	組織運用
SPC-02		サプライチェーンの脅威/信頼モデルを開発する。	サプライチェーンにおいて、自組織が担う役割を特定し関係者と共有する。	組織運用
SPC-03			あらかじめ定められた自組織の優先事業、優先業務と整合したセキュリティポリシー・対策基準を明確化し、自組織の取引に関係する者（サプライヤー、第三者プロバイダ等を含む）に共有する。	組織運用
SPC-04			リスクアセスメント結果及びサプライチェーンにおける自組織の役割から自組織におけるリスク許容度を決定する。	組織運用
SPC-05	契約	セキュリティに取り組んでいるサプライヤを採用する。	何らかの包括的なセキュリティ対策を実装しているサプライヤを採用する。	組織運用
SPC-06			一般的なセキュリティレコメンデーション（例：NISTIR 8259.10）を実装している、あるいはセキュリティ標準（例：ISO27036、ISO28000）に準拠しているサプライヤを採用する。	組織運用
SPC-07		セキュリティに関する契約を結ぶ。	セキュリティの監査を契約に含める。	組織運用
SPC-08			ソフトウェアに対して、セキュアブートやファームウェア署名を実装させるような契約を結ぶ。	組織運用
SPC-09	サードパーティソフトウェア	組み込むにあたって評価を行う。	・サードパーティのソフトウェアを、想定されるユースケースでレビュー、評価する。（ユースケースが変わったらそのときに再びレビュー、評価を行う） ・ソフトウェアのセキュアな構成を決定する。	サードパーティソフトウェア

ID	大項目	中項目	小項目	検証対象
SPC-10			・自社ソフト同様のコードレビュー、分析、テストを行う。	サードパーティソフトウェア
SPC-11			・デジタル署名またはその他のメカニズムを使用して、ソフトウェアコンポーネントの整合性を確認する。	サードパーティソフトウェア
SPC-12		構成管理を行う。	・各ソフトウェアコンポーネントの出自情報（SBOM、ソース構成分析、バイナリソフトウェア構成分析など）を分析して、脅威や OSS のライセンスなどのリスクを評価する。 ・認可した OSS をホストするソフトウェアリポジトリを構築する。 ・認可した商用ソフトウェアとバージョンのリストを、出自データと共に管理する。 ・フィールドに展開済みの古いバージョンのソフトウェアについて、新しいバージョンへの移行が正常に完了するまで、古いバージョンのソフトウェアを保持する。 ・バイナリで取得したソフトウェアの完全性または出自を確認する。確認できない場合は、ソースコードの完全性と出自を確認した後、ソースコードからバイナリをビルドする。	組織運用

ID	大項目	中項目	小項目	検証対象
SPC-13		定期的に脆弱性などをチェックし、必要なアクションを取る。	・ソフトウェア/サービスに、ベンダがまだ修正していない既知の脆弱性があるかどうかを定期的に確認する。 ・ソフトウェアコンポーネントの既知の脆弱性を自動検出する機能をツールチェーンに組み込む。 ・各ソフトウェアコンポーネントが積極的に保守されており、サポートが切れていないことを確認する。 ・保守が終了しているか、近い将来メンテナンスが終了するソフトウェアコンポーネントに対して、アクションプランを決定する。	組織運用

表 2-15 セキュリティプラクティス（組織・運用） ― 脆弱性対応

ID	大項目	中項目	小項目	検証対象
VUL-01	脆弱性対応プロセスの確立	脆弱性開示/修正するための体制を確立する。	・PSIRT を設置する。 ・一般に報告された脆弱性や、ゼロディのレポート、実際に悪用された脆弱性、および複数の関係者と OSS が関わる大規模な進行中のインシデントに対処するため、セキュリティレスポンスのプレイブックを準備する。 ・製品セキュリティに関するインシデントレスポンスの演習を定期的に実施する。	組織運用
VUL-02		継続的にセキュリティ情報を収集する。	・脆弱性データベース、セキュリティメーリングリスト、およびその他の脆弱性レポートなどからセキュリティ情報を得る。	組織運用
VUL-03			・脅威インテリジェンスを使用して、一般的に脆弱性がどのように悪用されているかを理解する。	組織運用
VUL-04		継続的にソフトウェアコードのテストを実施する。	・サポート対象であるすべてのリリースについて、定期的にセキュリティテストを行う。	組織運用
VUL-05			・ツールチェーンを構成し、サポート対象であるすべてのリリースについて、コード分析とテストを自動的に、定期的/継続的に実行する。	組織運用

ID	大項目	中項目	小項目	検証対象
VUL-06		脆弱性に対処しパッチを提供する。	・脆弱性の悪用可能性、悪用された場合の潜在的な影響、およびその他の関連する特性の推定に基づいて、各脆弱性に対してリスクを計算する。 ・各脆弱性を修復するか、またはその他の手段 （リスクの受け入れ、リスクの移転など） を通じてリスクに対処するか、リスクベースの決定を行い、実行するアクションに優先順位をつける。 ・脆弱性に対する恒久的な緩和策がまだ利用できない場合は、恒久的なソリューションが利用可能になるまで、この脆弱性を一時的に緩和する方法を決定し、その一時的な改善策を計画に追加する。 ・脆弱性の内容、脆弱なソフトウェアがあるかを調べる方法、およびその対処方法 （パッチの入手場所、ソフトウェア内で修正プログラムの変更、変更が必要な構成設定、一時的な回避策の実装方法など）を含む、ソフトウェア取得者に必要な情報を提供するセキュリティアドバイザリを開発およびリリースする。 ・自動化された信頼できる配信メカニズムを使用して、顧客にパッチを提供する。	組織運用
VUL-07		脆弱性の根本原因を分析し、再発を防止する。	・特定された脆弱性を分析し、その根本原因を特定する。 ・脆弱性のパターンを特定するために、時間の経過とともに根本原因を分析する。 ・同種の脆弱性について確認し、横展開する。 ・根本原因の再発を防止するよう、SDLC プロセスを更新する。	組織運用

ID	大項目	中項目	小項目	検証対象
VUL-08	情報を受け取る窓口の用意	IoT 製品の脆弱性情報を受け取るための窓口を用意する。	・メンテナンスおよび脆弱性情報を受け取る窓口を用意する。 ・IoT 製品開発者が、顧客や他の関係者（機器保守員など）からの情報を受け取る窓口と連携すること。 ・IoT 製品開発者が、製品エコシステム内の顧客やその他の人々から、IoT 製品のサイバーセキュリティについての問い合わせを受信して応答できること。	組織運用
VUL-09	情報の提供	一般向け（顧客を含む）にセキュリティおよび脆弱性情報を公開する。	以下を一般向けに公開する。 ・サポート条件（更新の頻度やアプリケーションのメカニズムなど）が変更された場合 ・ソフトウェアのアップデートが利用可能であること ・IoT 製品のサポートまたは機能の期間の終了時期 ・必要なメンテナンス作業 ・製品の脆弱性情報、および必要な対応	組織運用
VUL-10			・対象の IoT 製品に関連のあるセキュリティ侵害情報、および必要な対応（存在する場合）	組織運用
VUL-11		CERT などの関係者にセキュリティおよび脆弱性情報を提供する。	以下を関係者に提供する。 ・IoT 製品の設計/開発で作成された関連ドキュメント ・IoT 製品開発者が使用する、情報セキュリティ関連のプラクティスと予防策の概要。 ・IoT 製品開発者のビジネス環境のリスク姿勢に関するリスク評価レポートまたは要約。	組織運用

ID	大項目	中項目	小項目	検証対象
VUL-12			・脆弱性の開示プログラムを確立し、セキュリティ研究者がそのプログラムについて簡単に知り、可能性のある脆弱性を報告できるようにする。 ・サイバーセキュリティおよび脆弱性に関するアラート、および脆弱性の解決に関する情報 ・サイバーセキュリティ関連のプラクティスに対する認証や評価結果。	組織運用
VUL-13		顧客に対してセキュリティ意識を高めるための情報を提供する。	以下を顧客に対して開示する。 ・製品のセキュリティ機能 / 設定変更について / アクセス制御機能について / ソフトウェアアップデートについて / 製品におけるデータ管理（削除方法など）について ・製品のセキュリティの維持方法 ・製品で利用可能なセキュリティ関連のオプションに関する情報	組織運用
VUL-14			・製品購入の参考にするための追加情報（製品のサポート期間、サポート範囲など）	組織運用

表 2-16 セキュリティプラクティス（組織・運用） ― その他

ID	大項目	中項目	小項目	検証対象
MSC-01	可能であれば、独自実装せず、標準機能や OSS を利用する。	―	―	組織運用
MSC-02	試作品などを安全に廃棄する。	―	―	組織運用
MSC-03	新しい技術を取り入れる	―	―	組織運用
MSC-04	保守において部品交換に関連するサプライチェーンのリスクを考慮する。	―	―	組織運用
MSC-05	システムの実装において、採用する技術の多様性を高める。	―	―	組織運用

1.2.3.3.3　技術検証に関する項目

表 2-10～表 2-12 にまとめた技術に関するセキュリティプラクティスのうち、検証対象がソフトウェア（自社開発ソフトウェア、サードパーティソフトウェア、商用ソフトウェア、OSS を含む）であるセキュリティプラクティスに言及しているガイドラインを逆引きし、そのガイドラインが推奨している関連プラクティスを 1.2.3.3.3.1 以降に示す。

タイトルの横に括弧書きで示したものは、ガイドラインの中の該当するプラクティス項目を示している。

1.2.3.3.3.1　NIST Special Publication 800-218

■実行コードの改ざん防止（PS.1.1/PS.2.1）

不注意および意図的な実行コードに対する変更を防止するため、ソフトウェアを改ざんから保護することが重要である。ソフトウェアのハッシュを取る、あるいは電子署名を付与したり、ソフトウェアを暗号化したりすることで、ソフトウェアの完全性を保護する。

■脅威分析・対策立案（PW.1.1/PW.2.1）

運用中にソフトウェアが直面する可能性のあるセキュリティリスクを特定し、リスクへの対策方法（軽減方法など）を検討・評価する。

脅威分析には脅威モデリングや攻撃モデリングなどの手法が用いられる。機密性の高いデータの保護や、認証に関わる管理といった高リスク領域はより厳格に検討する。次に、脅威分析の結果に応じて対策を行う。設計に関与しない第三者によるレビューや、可能であればツールによる自動解析によって、個々の対策が適切かどうかを確認する。

■セキュアコーディング（PW.5.1）

脆弱性の数を減らし、脆弱性対応コストを削減するために、ソースコードの作成においてセキュアコーディングに従うことが重要である。開発言語および実行環境に応じたセキュアコーディングルールを遵守する。セキュアコーディングルールには、例えば外部からのすべての入力値を検証する、安全でない関数の使用を避ける、エラーを検出し適切に対処する、などがある。

■静的解析・ピアレビュー（PW.7.1/7.2）

脆弱性の悪用を防ぐため、および、セキュリティ要件に準拠していることを確認するため、人間が読めるコードを分析する。自動化された方法を使用すると、脆弱性の検出に必要な労力とリソースが削減できる。

静的分析ツールあるいはピアレビューによってコードが組織のセキュアコーディングルールに従っていることを確認する。専門家によって、バックドアや悪意のあるコンテンツがないかも確認する。

■動的解析（PW.8.1/8.2）

それまでに実施した静的解析などではまだあぶりだされていない脆弱性を見つけ出すため、およびセキュリティ要件に遵守していることを確かめるため、実行可能コードをテストする。新規開発したソフトウェアだけではなく、サードパーティや社内で流用するソフトウェアを対象に入れる場合もある。動的解析には、入力処理に関する問題を見つけるためのファジングテストや、攻撃者がソフトウェアの脆弱性を突くことをシミュレートするペネトレーションテストなどがある。

1.2.3.3.3.2 Labeling for Consumer Internet of Things （IoT） Products

■ソフトウェア更新（5-1/5-2）

IoT 製品で実行されるソフトウェアは、セキュアかつ設定可能なメカニズムによって更新できることが重要である。IoT 製品は正しい更新ソフトウェアを受け取り、それが正しいことを検証した上で適用する。常にソフトウェアを最新の状態に維持できるメカニズムを実装することが必要である。

1.2.3.3.3.3 Labeling of Consumer Software

■実行コードの改ざん防止（2-4）

ソフトウェアは、初期およびその後のアップデートの真正性を担保し、悪意のある攻撃者による改ざんや偽造を防止することが重要である。プログラムのハッシュを取る、あるいは電子署名を付与することで、プログラムの完全性が保護されていることを確認する。電子署名を使う場合、その有効性が確認できるよう、確立された認証局を使うことや、証明書の更新、失効管理といったコード署名プロセスの定期的な見直しも有効である。

■強力な暗号の使用（2-7）

ソフトウェアがセキュリティ目的で使用するすべての暗号アルゴリズムについて、最低限 NIST の暗号標準/ガイドラインに準拠することが求められる。認証スキームによって、他の暗号アルゴリズムが追加定義されることもある。

■静的解析・ピアレビュー（2-2）

人間が読めるコードをチェックし、脆弱性を特定し、セキュリティ要件に準拠していることを確認する。

■動的解析（2-2）

実行可能コードをテストして脆弱性を特定し、セキュリティ要件に準拠していることを確認する。

1.2.3.3.3.4　Guidelines for Securing the Internet of Things

■ソフトウェア更新（TEC-01/07）

様々な世代のデバイスとソフトウェアが共存する IoT ソリューションでは、異なるレベルのセキュリティや安全性を扱うことになるのを避けるためにも更新していく必要がある。製品がリリースされた後も開発が続くこともあり、サプライチェーンシステムに影響を与える脆弱性が、リリース後に発見される可能性もある。環境の変化に俊敏に対応して更新プログラムを展開する機能は、設計の初期段階から考慮されなければならない。ただし、ソフトウェア更新機能はミスやマルウェアの注入を防ぐようになってなければならない。

■セキュアコーディングおよび静的・動的解析（PRO-01/06）

適切なセキュリティ機能を実装しそれを検証するために、セキュアコーディングやセキュリティに焦点を当てたテスト（ペネトレーションテストや脆弱性スキャンなど）を IoT サプライチェーンの適切な段階で実施する必要がある。

受け入れテストでは、サプライチェーンの前段階までに行われた可能性のあるテストとは独立して行う必要がある。デバイスの（全数ではなく）一部を製造の最後の部分で検査するとともに、サイバーセキュリティテストを実施して、設定誤りやエラーを検出する必要がある。

1.2.3.3.3.5　NIST Special Publications 800-53 Rev.5

■コードの難読化（SR-9）

ソフトウェアをリバースエンジニアリングや改ざんなどの脅威から守るため、難読化などの耐タンパー技術を取り入れることが重要である。難読化にセルフチェックを併せて使用することで、攻撃者にとってリバースエンジニアリングや改ざんをより時間がかかるものにすることができる。

■脅威分析（RA-3/SA-15）

サプライチェーンを考慮に入れたリスクアセスメントを実施することが重要である。サプライチェーン関連のリスクイベントとしては、サプライチェーンが途絶えたり、使用するコンポーネントに欠陥があったり、偽物が挿入されたり、悪意のある開発行為が行われたり、コンポーネントの配信方法が適切でなかったり、悪意のあるコードが挿入されたりなどがある。これらは、システムとその情報の機密性、完全性、または可用性に重大なインパクトを与える可能性があるため、組織の運営 （ミッション、機能、イメージ、または評判を含む)、組織の資産、個人、他の組織、ひいては国にも有害なインパクトを与える可能性がある。サプライチェーン関連のリスクイベントは、意図的でないことも、悪意のある場合もあり、システムのライフサイクルの任意の時点で発生する可能性がある。サプライチェーンリ

スクの分析は、組織が追加のサプライチェーンリスク軽減が必要なシステムまたはコンポーネントを特定するのに役立つ。

1.2.3.3.4　組織マネジメントに関する要求事項

表 2-13～表 2-16 にまとめた組織運用に関するセキュリティプラクティスのうち、検証対象がソフトウェア（自社開発ソフトウェア、サードパーティソフトウェア、商用ソフトウェア、OSS を含む）であるセキュリティプラクティス、表 2-2 の「組織運用（O-1～O-3）」に該当するセキュリティプラクティス、およびサプライチェーンに関わるセキュリティプラクティスに言及しているガイドラインを逆引きし、そのガイドラインが推奨している関連プラクティスを 1.2..3.4.1 以降に示す。

タイトルの横に括弧書きで示したものは、ガイドラインの中の該当するプラクティス項目を示している。

1.2.3.3.4.1　NIST Special Publication 800-218

■セキュリティ要件の定義・文書化（PO.1/PW.1.2/4.2）

組織が開発したソフトウェアが満たすべきすべてのセキュリティ要件を特定して文書化し、その要件を長期にわたって維持する。リスクベースのソフトウェアアーキテクチャと設計要件を規定するポリシーを定義することや、組織のソフトウェアのセキュリティ要件を規定するポリシーを定義することなどが含まれる。また、ポリシーがソフトウェアのライフサイクル全体をカバーしていることも確認する。

すべてのセキュリティ要件は少なくとも年に 1 回はレビューを行う。新しい要件が出てきたり、大きな脆弱性が発見されたり、組織が開発したソフトウェアに大きなセキュリティインシデントが発生したりした場合はもっと早くレビューを行う。

脆弱性の軽減をどう実現するのか、および、セキュリティ要件に対する承認された例外の根拠が何であるかなど、各リスクへの対応を記録する。また、ソフトウェアのライフサイクルが終了するまで監査および保守の目的で使用できる、設計上の決定、リスク対応、および承認済みの例外の記録を保持する。

■セキュアな設定・構成の定義・文書化（PW.4.2/9.1/9.2）

デフォルト設定がセキュアかつ、プラットフォーム、ネットワークインフラストラクチャ、またはサービスによって提供されるセキュリティ機能を弱めることがないように、セキュリティまたはセキュリティ関連の設定に影響を与える各設定を構成する方法を決定する。これによってセキュアなベースラインを定義する。デフォルト設定を実装し、ソフトウェア管理者向けに各設定を文書化する。

テストを実施して、デフォルト設定を含む設定が期待どおりに機能し、セキュリティ上の弱点、運用上の問題、またはその他の問題を不注意によって引き起こしていないことを確認

する。承認された構成がソフトウェアに対して適切に設定されていることを確認する。各設定の目的、オプション、デフォルト値、セキュリティの関連性、潜在的な運用上の影響、および他の設定との関係を文書化する。また、デフォルトの構成を使用可能な形式で保存し、変更する場合は変更管理に従う（例えば、configuration-as-code）。

■定量的なセキュリティ基準に基づいた測定（PO.4）

開発中にソフトウェアのセキュリティをチェックするための基準を定義して使用することにより、SDLC に従って生み出されるソフトウェアが組織の期待に確実に応えるようにする。

ツールチェーンを使用して、セキュリティの意思決定に情報を提供する情報を自動的に収集する。基準をサポートする情報の生成と収集をサポートするために、必要に応じて追加のツールを展開する。基準を利用して意思決定プロセスを自動化するとともに、これらのプロセスを定期的に確認するようにする。許可された担当者のみが収集した情報にアクセスできるようにし、情報の変更または削除を防止する。

1.2.3.3.4.2 Labeling for Consumer Internet of Things （IoT） Products

■文書化（Documentation）

IoT 製品開発者は、顧客が購入する前、および製品の開発とその後のライフサイクル全体にわたって、IoT 製品とその製品コンポーネントのサイバーセキュリティに関連する情報を作成、収集、および保存する。これには、開発時に行った想定、例えば、IoT 製品の設置場所のセキュリティを含む IoT 製品の物理的な使用方法や IoT 製品の開発者が想定する IoT 製品のサイバーセキュリティ要件が含まれる。また、IoT 製品の設計およびサポートに関する検討事項、例えば、IoT 製品の作成に使用されたすべてのハードウェア/ソフトウェアコンポーネントの構成要素（オープンソース、サードパーティ、または内作）が含まれる。他には、IoT 製品のメンテナンス要件、例えば、脆弱性管理計画といったサイバーセキュリティのメンテナンスについての想定と関連手順も含まれる。

1.2.3.3.4.3 Labeling of Consumer Software

■セキュリティ要件の定義（Practices Secure Design and Vulnerability Remediation）

脆弱性の発生と影響を最小限に抑えるために、ソフトウェア開発で使用される多くのベストプラクティスがある。まず、ソフトウェアのセキュリティ要件、リスク、および設計上の決定を追跡および維持する必要がある。

1.2.3.3.4.4 Guidelines for Securing the Internet of Things

■セキュリティ要件の定義と文書化（PRO-01/PRO-11）

セキュリティモジュールは、優先度の高いコンポーネントと見なされ、サプライチェーン

全体の最初の段階から設計プロセスに組み込まれる必要がある。ヒューマンエラーを防止するための包括的なドキュメントを用意する。構成管理と障害後の復旧の観点は特に重要である。ドキュメントがないことは問題であるが、あったとしても標準レベルに達しないものであるなら有害である可能性が高い。

■サプライチェーンの完全性指標の作成（PRO-03）

IoT サプライチェーンにおける完全性の概念は非常に広いが、たいていは、偽物やマルウェア、または可視性と説明責任を低下させる可能性のあるその他の影響（ファームウェアの更新を適切に追跡できないなど）がないといった、混じりけのない形で稼働するサプライチェーンの状態に関するものと言える。指標を作成して継続的に監視することで、サプライチェーンの状態を可視化できる。現在フィールドに展開されているファームウェアバージョンの分布、といったものがこのようなメトリックの例として挙げられる。

■SBOM の作成（PRO-13）

SBOM は、オープンソースと商用の両方のパッケージまたはライブラリを含む、ある製品の構成要素として使用されるソフトウェアコンポーネントを網羅的に記述したものである。これらのリストにより、製品の可視性が高まり、ベンダと外部ユーザの両方が既知の脆弱性をチェックし、セキュリティの観点からデバイスを検証できるようになり、攻撃者がある脆弱性をうまく利用することを可能にしてしまう脆弱性のギャップを減らすことができる。

SBOM は、商業的に配布されているかどうかに関係なく、特定の組織のすべての IoT 製品に対して利用できるのが理想的である。

■サプライチェーンに対する新しいセキュリティモデルの開発（ACT-03/PRO-04）

サプライチェーンにおける信頼モデルとは、さまざまな関係者のふるまいに関する正式な保証を提供し、セキュリティを強化するためのフレームワークを定義するものである。サプライチェーン専用の信頼モデルを開発するか既存の信頼モデルをサプライチェーンに適合させることが重要となってくる。

また、サプライチェーンにおける脅威モデルは、サイバーフィジカルシステムに固有の物理的セーフティとデジタルセキュリティの両方の概念を統合する必要がある。さらに、サプライチェーンを機能ブロックに分割し、それらのブロック内の資産を一覧表示して、後で重要な資産とブロックを検出できるようにする。攻撃の脅威に加えて、IoT の追加に起因するシステムの複雑さの増大を管理する際のエラーに起因するセキュリティ、安全性、およびパフォーマンスにも影響を与える可能性のある意図しないインシデントも含める必要がある。

■サプライチェーンに対する新しい標準の開発（PRO-12）

現在、すべての業界で IoT のサプライチェーンを保護するという目的に完全に適合する標準は存在しない。一部の IT セキュリティ標準を適用可能ではあるが、業界によっては制限がある。ISO27001 や最近の NERCCIP-013-1 などの一部の規格は、オープンすぎるあるいは汎用的すぎるかもしれない。特定のドメインまたは業界では、一部の標準は抽象的すぎて、コンテキストで理解および適用するのが難しい。さらに、標準化団体と開発コミュニティの間には間違いなくギャップがある。新しい標準を開発すること、または既存の標準を適用させることは、IoT のグローバルサプライチェーンのセキュリティ管理に一貫性を与え、すべての関係者間のセキュリティ情報の統合を改善することに貢献する。

■セキュリティを保証するサプライヤの採用（ACT-01）

外部のサプライヤと協業するということには、セキュリティ対策の管理が行き届かないことによる固有の脅威があるが、これは通常避けられないビジネスの現実である。この脅威は、ISO27036 や ISO28000 などの標準、または NISTIR 8259.10 などの推奨事項を実装する企業を優先することで最小限に抑えることができる。認証を受けることを求める企業は、通常、サプライチェーンのセキュリティの向上に真剣に取り組んでいるということを示している。認証は通常、すべての組織に適しているわけではない、コストのかかるプロセスである。標準の認証を取ってはいないが、包括的なセキュリティ対策が実施されていて、それらについて透過的である組織（監査の権利、契約上のセキュリティ要件など）も信頼できると見なせるはずである。

■透明性の確保（ACT-02/PRO-05）

サプライチェーンのセキュリティを管理するには、透明性が不可欠である。利害関係者、特にサプライヤは透明性があり、供給された製品の操作と通常の動作に関する明確で詳細な情報を提供する必要がある。そして、すべての関連情報をチェーンの次のステップに伝達する。透明性のレベルを上げると、サプライチェーンの参加者間の信頼を強化するという望ましい副作用も生じる。

1.2.3.3.4.5　NIST Special Publications 800-53 Rev.5

■サプライチェーンを考慮した SDLC（SA-3）

情報セキュリティとプライバシーに関する考慮事項を組み込んだ、組織が定めるシステム開発ライフサイクルを使用して、システムを取得、開発、管理する。システム開発ライフサイクル全体を通じて、情報セキュリティとプライバシーの役割と責任を規定し、文書化する。情報セキュリティおよびプライバシーの役割と責任を有する個人を特定する。組織の情報セキュリティおよびプライバシーリスクマネジメントプロセスをシステム開発ライフサイクル活動に統合する。

■サプライチェーンを考慮したリスクアセスメント（RA-3（1））

組織が定めるシステム、システムコンポーネント、およびシステム・サービスに関連するサプライチェーンリスクをアセスメントする。組織が定める頻度で、または、関連するサプライチェーンに大幅な変更がある場合や、システム運用環境、またはその他の条件の変更によりサプライチェーンの変更が必要になる場合にサプライチェーンのリスクアセスメントを更新する。

1.2.3.3.4.6　サイバー・フィジカル・セキュリティ対策フレームワーク（CPSF）

■サプライチェーンにおける役割の特定（CPS.BE-1）

サプライチェーンにおいて、自組織が担う役割を特定し共有する。

■自組織のセキュリティポリシーの共有（CPS.BE-2）

あらかじめ定められた自組織の優先事業、優先業務と整合したセキュリティポリシー・対策基準を明確化し、自組織の取引に関係する者（サプライヤ、第三者プロバイダ等を含む）に共有する。

■自組織のリスク許容度の決定（CPS.RM-2）

リスクアセスメント結果及びサプライチェーンにおける自組織の役割から自組織におけるリスク許容度を決定する。

1.2.3.4　OSS 検証ツールの実態調査

本項では OSS 及びサードパーティのソフトコンポーネントを対象に、セキュリティ上の安全性を検証するツールの調査結果を示す。対象ツールは、ソフトウェア・コンポジション解析機能及び、各コンポーネントの脆弱性診断機能を有するものを対象に 2 ツールの調査を行った。いずれも国内でサービス展開されている商用ツールを対象としている。

1.2.3.4.1　OSS 検証ツールの調査結果一覧

本調査では以下の項目について、各ツールの公開資料（スペックシートやカタログ等）及び、提供ソリューションベンダへのヒアリングにより調査を行った。

調査結果については、表 2-17 に一覧を示す。

１）基本スペック

・1-1）検査可能なファームウェアの最大サイズ

・1-2）スキャン方法（バイナリ解析/ソースコード解析）

・1-3）ライセンスプラン（ツール利用時のライセンス形態）

２）SBOM/OSS 管理

・2-1）内在するソフトウェアコンポーネント（SBOM）の表示機能

・2-2）ソフトウェアコンポーネントの識別方法

・2-3）対応コンポーネントのアップデート機能

・2-4）規格に準拠した SBOM の作成機能

・2-5）作成可能な SBOM の標準規格（SWID、SPDX など）

・2-6）OSS のライセンス検査（管理）機能

３）脆弱性診断

・3-1）脆弱性診断機能（CVE-ID ベースの脆弱性検出）

・3-2）脆弱性のリスクレベル値表示

・3-3）診断レポート機能

・3-4）脆弱性データベースのアップデート機能

・3-5）国内外 IoT セキュリティ要件との比較分析機能

４）その他　ツール個別の付加機能

・4-1）ダッシュボード画面

・4-2）CI/CD ツール・環境との連携機能

・4-3）その他ツール固有の機能

表 2-17　OSS 検証ツールの調査結果一覧

※項目について以下の基準でセル色の区別を行っている。

：必須機能　1.2.3.3 項の調査結果から、機能の実装が必須と想定される項目

：推奨機能　1.2.3.3 項の調査結果から、ツール以外の運用や環境で代替が可能であるが、実装が推奨される項目

項目分類	機能項目	検証ツール	
		ツール A	ツール B ※機能の一部はオプション提供
1）基本スペック	1-1）検査可能なファームウェア最大サイズ	4GB	ライセンスにより変動
	1-2）スキャン方法	バイナリスキャン ※手動の情報入力により、バイナリファイルがなくとも検査が可能	ソースコードスキャン、バイナリスキャン
	1-3）ライセンスプラン	提供プランは下記 3 プランあり ・回数制（1 回ごとの課金） ・1 日の容量制限制 年間の総容量サイズ制 ※価格は代理店により異なるため非公開	機能、容量に応じたライセンス形態によるサブスクリプションプラン
2）SBOM/OSS 管理	2-1）内在するソフトウェアコンポーネント（SBOM）の表示機能	○ 下記 3 種類の表示が可能 ・ビジュアル化された統計情報の表示	○ 検出した各サードパーティ・コンポーネントについて、バージョン、ロケーション、

項目分類	機能項目	検証ツール	
		ツール A	ツール B ※機能の一部はオプション提供
		・XML 形式でのプレビュー表示 ・JSON 形式でのプレビュー表示	ライセンス取得状況、既知の脆弱性など、詳細な情報を提示
	2-2）ソフトウェアコンポーネントの識別方法	NVD が公開する CPE リストに基づき、ソフトウェアコンポーネントを識別する	独自の OSS 情報データベースを使用
	2-3）対応コンポーネントのアップデート機能	○ ※NVD の情報更新に準ずる	○
	2-4）規格に準拠した SBOM の作成機能	○	○
	2-5）作成可能な SBOM の標準規格	SWID ※SPDX も対応予定	SPDX 等
	2-6）OSS のライセンス検査（管理）機能	○ ※バイナリから検出した情報と Github のリポジトリ情報を照合して表示。	○ 独自の OSS 情報データベースにより、ライセンス競合の可能性を含め診断
3）脆弱性診断	3-1）コンポーネントの脆弱性診断機能 （CVE-ID ベースの課題検出）	○ CVE-ID ベースの課題検出が可能 ※脆弱性データベースは NVD を参照	○ 脆弱性データは NVD の公開情報に加え、独自の脆弱性情報を保持
	3-2）脆弱性のリスク評価レベル値表示	○ CVSS 2/CVSS 3（NVD）、ioXt likelihood	○ CVSS 2/CVSS 3 など

項目分類	機能項目	検証ツール	
		ツール A	ツール B ※機能の一部はオプション提供
	3-3）診断レポート機能	○ ※NVD データに記載された対策内容を参考 URL として表示	○ ※対策手引きのレポート機能を有する
	3-4）脆弱性データベースのアップデート機能	○ ※NVD の情報更新に準ずる	○ ※NVD の情報更新に準ずる
	3-6）国内外 IoT セキュリティ要件との比較分析機能	○ ※下記要件の適合状況について比較レポートを生成 ・CCDS IoT 機器セキュリティ要件 ・ioXt セキュリティ要件	不明
４）その他付加機能	4-1）ダッシュボード画面	○	○
	4-2）CI/CD ツール・環境との連携機能	○ ※「Mantis」（不具合管理システム）との連携	○ ※統合開発環境（IDE）や CI ツール等との連携が可能
	4-4）その他の固有機能	・サプライチェーン管理（複数企業間連携機能） ・セキュリティニュースレポートの配信機能 ・製品に含まれる CPE の新規脆弱性情報の配信 ・CVE-ID、CPE-ID による情報検索機能	・スキャン、手動登録された OSS は、継続的に監視され、アラートが通知される

1.2.3.4.2　各 OSS 検証ツールの特徴

今回調査対象としたツールについて、それぞれの特徴を本項で示す。総合的な調査結果としては、1.2.3.3 項の調査結果から求められる必須機能、推奨機能については、いずれのツールも機能を満たすものであることが確認された。またツール A はクラウド環境により SaaS 型として利用可能であり、ツール B については利用者のクライアント環境と連携する必要があるが統合開発環境（IDE）や CI ツール等との連携が可能であり、利用者側のニーズに合わせて、ツールを選定することができる。

以下の項では、各ツール別にそれぞれの特徴を示す。

1.2.3.4.2.1　ツール A の特徴

１）SBOM/OSS 管理機能の特徴

ソフトウェアコンポーネントの検出については、米国の NVD[11]に登録されている CPE[12]リストを参照し、内部的な OSS やサードパーティのソフトウェアコンポーネントの検出を行っている。検出可能なソフトウェアコンポーネントについては、NVD の更新により、都度最新のものが適用される。

SBOM の表示については、ツールの UI 上で統合情報（名称、バージョン、脆弱性の数やリスク値）が表示されると共に、標準規格の SWID（SPDX についても今後対応予定）に準じた XML 形式、JSON 形式でのプレビュー表示や出力が可能となっている。

OSS のライセンス管理については、Permissive（GPL 等利用許諾されているライセンス）か、Protective（利用に制限のあるライセンス）を区別し、各ソフトウェアのライセンス情報（AGPL、GPL、LGPL、MIT License など）についても表示される。また、ライセンス情報の検出は、バイナリ解析結果と Github 上の公開データを照合しており、両者のデータに差異がある場合は、Github 上の公開データを優先する。

２）脆弱性診断機能の特徴

ツール A はバイナリスキャンに対応したツールであり、対象製品のバイナリファイルをクラウド環境へアップロードすることで詳細な解析が可能となる。またバイナリファイルをアップロードできない場合は、CPE やソフトウェアコンポーネントの情報（名称やバージョン）を手動入力することでバイナリ解析を利用しなくとも、簡易的な脆弱性の検査が可能なところに特徴がある。

脆弱性情報の検索は、NVD を参照しており、NVD の更新により、脆弱性の情報も最新のものが適用される。リスクレベル値の表示については、NVD に登録されている CVSS2、CVSS3 の値を標準とし、ioXt アライアンスが定義する likelihood に照らした表示も可能となっている。

脆弱性診断レポートについては、NVD に登録されている情報表示を行い、対策内容については、参考情報の URL を参照可能となっている（ただし、対策内容は NVD の登録情報に準じており、対策内容の記載がない脆弱性については対象外となる）

３）その他付加機能の特徴

ツール A の付加機能の特徴としては、admin 権限管理及び、グループ内でのサブアクセス権限の設定

[11] NVD（National Vulnerability Database）：米国 NIST が管理する脆弱性情報の公開データベース
[12] CPE（Common Platform Enumeration）：情報システムを構成する、ハードウェア、ソフトウェアなどを識別するための共通の名称基準

が可能であり、業務提携している ODM、OEM の企業と連携したソフトウェアコンポーネント情報の共有、管理が可能となっている（サプライチェーン対応）。

CI/CD 環境との連携については、「Mantis（不具合データ管理システム）」と連携し、脆弱性情報の統合的な管理が可能となる。

また解析を行った製品については、関連するセキュリティニュースの配信機能（外部のニュースサイトより情報を参照）や、利用するソフトウェアコンポーンに新規脆弱性が報告された場合の情報配信機能についても実装している。

1.2.3.4.2.2　ツール B の特徴

１）SBOM/OSS 管理機能の特徴

ソフトウェアコンポーネントの検出については、独自の OSS 情報ベースを参照している。検出した OSS やサードパーティのソフトウェアコンポーネントについては、バージョン、ロケーション、ライセンス取得状況、既知の脆弱性などの、詳細な情報表示を可能としている。検出されたソフトウェアコンポーネントについては、一覧を通知ファイルや CSV、SPDX 等の形式で生成可能となっている。

OSS のライセンス管理については、各ソフトウェアのライセンス情報（AGPL、GPL、LGPL、MIT License など）に加え、ライセンス競合の可能性を含めて診断を行う。

２）脆弱性診断機能の特徴

ツール B は、ソースコードスキャン、バイナリスキャンの両形式をサポートしている。脆弱性情報については、NVD の公開情報に加え、独自の解析用データベースを参照している。独自のデータベースを有することで、報告から公開までにタイムラグが想定される脆弱性情報についても、比較的早期に検出することが可能となる。

脆弱性情報のリスクレベル値については、CVSS2/CVSS3 に加え、NVD には情報登録されていない現状値の参照が可能である。また CAPEC によるセキュリティ攻撃パターンの分類についても情報として参照が可能となる。

脆弱性診断レポートについては、独自の解析用データベースを参照することで NVD では公開されていない脆弱性の回避方法や、修正方法などに関する情報も提示可能となる。

３）その他付加機能の特徴

ツール B の付加機能の特徴としては、統合開発環境（IDE）や CI ツールとの連携機能が多用な点にある。REST API によって外部ソフトウェアと連携することも可能としている。また事前に設定したポリシーやアラートに応じて、解析を行ったソフトウェアコンポーネント（OSS）に新規の脆弱性が報告された場合には、通知や追跡が可能な機能を有している。

1.2.3.5　技術検証項目と実施ルール

本項では、1.2.3.3 項の調査結果を踏まえ、ソフトウェアのセキュリティに求められる技術検証項目、組織マネジメントにおける対策を整理し、一覧として提示する。また技術検証項目については、検証実施内容（ルール）や検証ツール例についても、本項にて記載を行う。

1.2.3.5.1　必要な技術検証項目

1.2.3.3.2 項の表 2-10～表 2-12 にまとめた技術に関するセキュリティプラクティスのうち、検証対象がソフトウェア（自社開発ソフトウェア、サードパーティソフトウェア、商用ソフトウェア、OSS を含む）であるプラクティスを抜粋したものを表 2-18 に示す。小項目ごとに、想定される検証の実施内容例を記載している。

対応レベルは、すべてのソフトウェアに最低限必要な技術検証項目を「必須」（必須要件：Mandatory Requirement）、生命、身体や財産に直接的な影響を与えるような高信頼性が求められるソフトウェアに必要な技術検証項目を「高度」（高度要件：Advanced Requirement）とした。

対象ソフトウェアは、図 2-4 に示すようなソフトウェアの種類によって「OS」「デバイスドライバ」「ミドルウェア／ライブラリ」「アプリケーション」に分けて、それぞれ 1 ～ 4 の番号を割り当てたものである。認証対象のソフトウェアの種別によって対応が求められる検証項目が分かるようにした。認証対象のソフトウェア単体ではなく、そのソフトウェアが組み込まれた最終製品として構成されたソフトウェア全体が対象となる場合は 5 の番号を割り当てた。なお、OS のないベアメタルなど、製品によっては図 2-4 に示したもの以外のソフトウェア構成をとるものもあり、あくまで代表的な一例を示したものである。

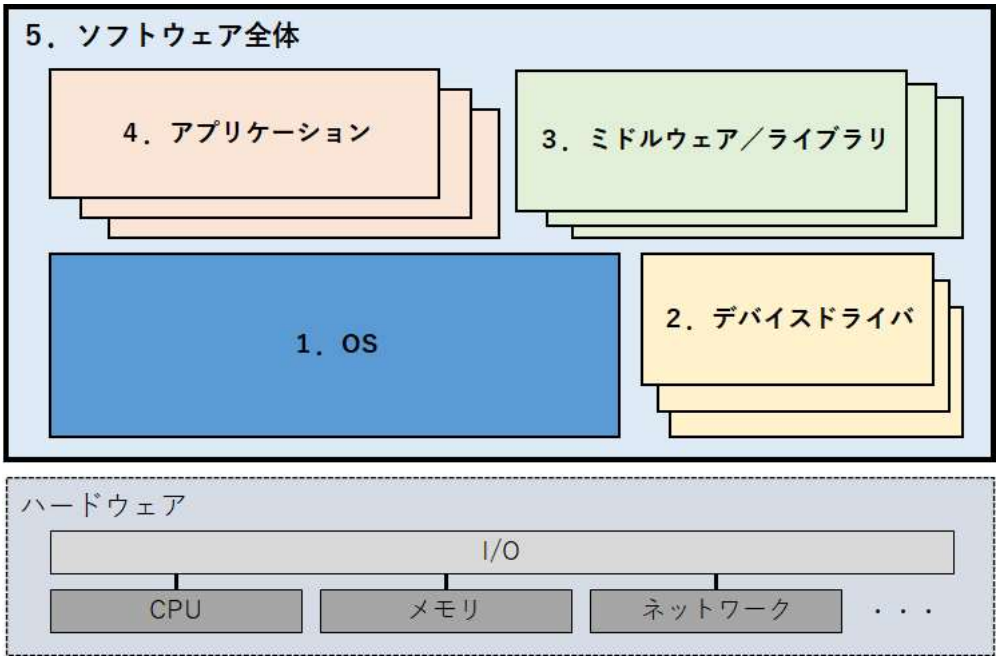


図 2-4　IoT 機器におけるソフトウェア構成例

表 2-18 セキュリティプラクティスおよび検証項目

ID	大項目	中項目	小項目	検証実施内容例	対応項目（表 2-10～2-12）	検証手法（表 2-2 の ID）	対応レベル	対象のソフトウェア構成
TEC-S01	リスクアセスメント	脅威分析を行う。	・システムに対する脅威および脆弱性を特定する。 ・システム、およびそのシステムが処理/保存/伝送する情報、およびなんらか関連する情報への認可されていないアクセス、使用、開示、中断、改変、または破壊による被害の可能性と規模を判断する。 ・脅威分析結果を文書化する。 ・脅威分析結果を定期的にレビューする 。	1．ソフトウェアの脅威と弱点を継続的に特定、評価、監視するためのプロセスが存在することを確認する。 2．OSS を使用している場合、OSS に脆弱性がないかを分析し、脆弱性があれば対策し、その OSS を含むソフトウェアを更新するためのプロセスが確立されていることを確認する。 3．開発中に作成されたドキュメントを参照し、ソフトウェア、ソフトウェアを含む製品、製品が使われるシステム信頼境界[13]を明確にした上で、脅威分析が行われていること、およびその妥当性を確認する。 4．ソフトウェアの入出力、データフロー、信頼境界について攻撃者からの悪用の可能性を考慮していることを確認する。 5．適切な知識および権限を持つ者により、見つかった脅威や設計上の欠点に関して、記録され、承認されていることを確認する。 6．上記の確認で問題がないこと。	SDL-01	V-1	必須	1 2 3 4

[13] ソフトウェアの開発者が制御できる範囲と制御できない範囲の境界。一般的には自社開発ソフトウェアの内外を境界とすることが多い。

ID	大項目	中項目	小項目	検証実施内容例	対応項目（表 2-10～2-12）	検証手法（表 2-2 の ID）	対応レベル	対象のソフトウェア構成
TEC-S02		対策の検討を行う。	・ソフトウェア設計が、適用可能なセキュリティ要件に対応していることを確認する。 ・ソフトウェア設計が、リスクモデルによって特定されたリスクに十分に対処していることを確認する。 ・レビューの結果に応じて、要件を満たすために障害を修正する。 ・セキュリティ要件を満たせない場合に、設計および/またはリスク対応戦略を変更する。	1．脅威分析（#TEC-S01 にて検証）で見つかった脅威に対して、対策が必要とみなしたものはすべて対策されていることを確認する。（対応しないと判断したものは妥当な理由が説明されていること） 2．対策が妥当であることを確認する。 3．残存リスク（例外含む）に関しても、記録され、正当化され、承認されていることを確認する。 4．上記の確認で問題がないこと。 ※セキュリティ対策に関する優先度の判断基準や実施タイミングについては、以下の文書が参考となる。 ・経済産業省「機器のサイバーセキュリティ確保のためのセキュリティ検証の手引き 別冊 2 機器メーカに向けた脅威分析及びセキュリティ検証の解説書」[14] 5.2.2 項　検出課題を含むリスク対応方針の検討	SDL-02	V-1	必須	1 2 3 4

[14] 経済産業省「機器のサイバーセキュリティ確保のためのセキュリティ検証の手引き
別冊 2 機器メーカに向けた脅威分析及びセキュリティ検証の解説書」
https：//www.meti.go.jp/press/2021/04/20210419003/20210419003-3.pdf

ID	大項目	中項目	小項目	検証実施内容例	対応項目（表 2-10～2-12）	検証手法（表 2-2 の ID）	対応レベル	対象のソフトウェア構成
TEC-S03	セキュアな実装	実行コードの改ざん検知/防止機能を実装する。	ハッシュやコード署名などを使用して、実行可能ファイルの完全性を保護する。	1．仕様書などのドキュメントを参照し、ソフトウェアの完全性を保護する方法が暗号的（#TEC-S05 にて検証）にセキュアであることを確認する。確認できた場合は３へ。 2．ソフトウェアの完全性を保護する方法が暗号的にセキュアでない場合、脅威分析（#TEC-S01 にて検証）の結果を確認し、実装されている代替手段による対応でセキュリティが十分保たれることを確認する。 3．実機テストを行い、改ざん検知/防止機能が働くことを確認する。 ＜実機テストの例＞ ・正規のもののバイナリの一部を書き換えたソフトウェア用意する。実行環境のソフトウェアを用意したソフトウェアで置き換えて実行し、実行が失敗することを確認する。 4．上記の確認で問題がないこと。	FNC-21	V-1/実機テスト	必須	3 4

ID	大項目	中項目	小項目	検証実施内容例	対応項目（表 2-10～2-12）	検証手法（表 2-2 の ID）	対応レベル	対象のソフトウェア構成
TEC-S04			コードを難読化する。	１．仕様書などのドキュメントを参照し、ソフトウェアを難読化する方法がセキュアであることを確認する。 ２．実機テストを行い、ソフトウェアの難読化が行われていることを確認する。 ＜実機テストの例＞ ・逆アセンブラを使用して難読化されていることを確認する。 ・フリーで入手可能な packer などのツールを用いて、ソフトウェアに適用している難読化が解除できないことを確認する。 ３．上記の確認で問題がないこと。	FUNC-22	V-1/ 実機テスト	高度	3 4

ID	大項目	中項目	小項目	検証実施内容例	対応項目（表 2-10～2-12）	検証手法（表 2-2 の ID）	対応レベル	対象のソフトウェア構成
TEC-S05		強力な暗号を使用する。	CRYPTREC などの標準に従った強力な暗号を用いること。	1．仕様書などのドキュメントを参照し、ソフトウェアが#02、#03 にて検証するセキュリティに暗号を用いている、あるいはソフトウェアが保存データや通信データの保護、認証、その他のセキュリティ機能に暗号を使用している、あるいはソフトウェアがライブラリなどの形態であって外部に暗号機能を提供するものであるかを確認する。いずれにも該当しない場合はこの検証項目は対象外である。該当する場合は次項を確認する。 2．仕様書などのドキュメントを参照し、ソフトウェアが使用する暗号アルゴリズムが次のようなガイドラインに準拠した暗号アルゴリズムを用いていることを確認する。 － CRYPTREC「電子政府における調達のために参照すべき暗号のリスト」「CRYPTREC 暗号技術ガイドライン（軽量暗号）」 － IPA「TLS 暗号設定ガイドライン」「SSL/TLS 暗号設定ガイドライン改定及び鍵管理ガイドライン作成のための調査・検討－調査報告書－」 － NIST「SP 800-57 Part1」「SP 800-52」など	FUNC-23	V-1	高度	1 3 4

ID	大項目	中項目	小項目	検証実施内容例	対応項目（表 2-10～2-12）	検証手法（表 2-2 の ID）	対応レベル	対象のソフトウェア構成
TEC-S06	実装	セキュアコーディングを実施する。		1．#TEC-S07 および#TEC-S08 記載の検証項目により、ソフトウェアに対してセキュアコーディングが実施されていることを確認する。 ＜確認観点の例＞ ・すべての入力を検証し、すべての出力を検証し、適切にエンコードする。 ・安全でない関数と呼び出しの使用を避ける。 ・ログ機能とトレース機能を提供する。 ・開発言語と環境に共通する他の脆弱性を確認する。	SDL-03	V-2	必須	1 2 3 4

TEC-S07	静的解析	ツールを使った静的解析を行う。	・ソースコードのスタイルとフォーマットを標準化するためにツール(例えば、リンタ、フォーマッタ)を使用する。 ・静的分析ツールを使用して、コードの脆弱性をチェックする。ツールによって報告された問題は必要に応じて修正する。	1．リンタ[15]やフォーマッタ[16]を用いて、ソースコードのフォーマットや記述レベルが統一されていることを確認する。 2．使用するソースコード解析ツール（※）のバージョンが最新であることを確認し、そのツール用いてソフトウェアを解析する。その結果としてツールが出した指摘について、妥当性を確認したうえで、修正が必要なものの指摘レポートを作成する。指摘がなければ５へ。 （※）ここでは、セキュアコーディングチェックツールおよびソフトウェアコンポジションアナリシスツールを想定。 3．前項での指摘に対応したソースコードを再びチェックする場合は、修正が必要な指摘が残っていないことを確認する。 4．意図的に修正しないものがある場合は開発者にその理由を確認し、妥当であることを確認する。 5．ソフトウェアコンポジションアナリシスツールのバージョンが最新であることを確認し、そのツールを用いて、以下を確認する。 ・ソフトウェアコンポーネントの構成が意図した通りであること ・OSS が含まれている場合、ライセンスが意図した通りであること 上記の確認で問題がないこと。 （注）フォールスポジティブな指摘を適切に扱う必要があるため、代表的な脆弱性に関する知見と、脆弱性が生じうるソースコード上のメカニズムに関する知見を有すること	TST-01	V-2	必須	1 2 3 4

ID	大項目	中項目	小項目	検証実施内容例	対応項目（表 2-10～2-12）	検証手法（表 2-2 の ID）	対応レベル	対象のソフトウェア構成
TEC-S08		ピアレビューを行う。	・コードのピアレビューを実行し、ピアレビューの一部として既存のコードレビュー、分析、またはテスト結果を確認する。 ・ピアレビュープロセスを容易にするピアレビューツールを使用し、すべてのディスカッションおよびその他のフィードバックを文書化する。 ・レビューチェックリストを使用して、コードが要件に準拠していることを確認する。	1．仕様書を確認し、セキュリティ仕様に問題がないことを確認する。 2．次の観点で分析したレビューチェックリストを作成し適宜更新する。 － 使用しているソースコード解析ツールでは検知しづらい観点の脆弱性を分析する。（※） － 前回のレビュー結果があれば、その結果から傾向を分析する。 （※）本項目の検証は#TEC-S07 の検証と併用することが前提。例えば、ソースコード解析ツールでは、セキュリティ機能が仕様通り実装されているかという観点でのチェックは基本的にできないため本項目でカバーする必要がある。 3．上記２のチェックリストをもとにコードレビューを行い、修正が必要なものを指摘する。なければ検証終了。 4．上記の指摘に対応したソースコードを再びチェックする場合は、修正が必要な指摘が残っていないことを確認する。 5．意図的に修正しないものがある場合は開発者にその理由を確認し、妥当であることを確認する。 6．上記の確認で問題がないこと。 7．	TST-03	V-1 V-2	必須	1 2 3 4

[15] 静的解析ツールの一種。ソースコードの記述に対して、プログラミング言語として構文上問題ないが、バグの要因となる可能性があるあいまいな記述に対して警告を出すツール。統合開発環境においてコンパイルと同時に実行されるものもある。

[16] インデントなどソースコードの整形を行うツール。統合開発環境やエディタによっては機能が組み込まれていて、自動で整形してくれるものもある。

ID	大項目	中項目	小項目	検証実施内容例	対応項目（表 2-10～2-12）	検証手法（表 2-2 の ID）	対応レベル	対象のソフトウェア構成
TEC-S09			・エキスパートによって、バックドアやその他の悪意のあるコンテンツのコードをチェックする。	1．仕様書などを併せてチェックし、仕様書に書かれていないバックドア機能などが含まれていないか、といった観点でソースコードを確認する。 2．確認した内容に基づき、指摘レポートを作成する。指摘がなければ検証終了。 3．上記の指摘に対応したソースコードを再びチェックする場合は、修正が必要な指摘が残っていないことを確認する。 4．意図的に修正しないものがある場合は開発者にその理由を確認し、妥当であることを確認する。 5．上記の確認で問題がないこと。	TST-04	V-2	高度	1 2 3 4

ID	大項目	中項目	小項目	検証実施内容例	対応項目（表 2-10～2-12）	検証手法（表 2-2 の ID）	対応レベル	対象のソフトウェア構成
TEC-S10	動的解析	動的解析を実施する。	・セキュリティ機能についての機能テストを実行する。	1．仕様書などのドキュメントを確認し、ソフトウェアが実現しているセキュリティ機能を特定する。 2．ソフトウェアを実行し、セキュリティ機能に応じてテストを行い、セキュリティ機能が仕様通りであることを確認する。 3．それらのセキュリティ機能が回避可能でないことを確認する。 4．確認した内容に基づき、指摘レポートを作成する。指摘がなければ検証終了。 5．上記の指摘に対応したソフトウェアを再びチェックする場合は、修正が必要な指摘が残っていないことを確認する。 6．意図的に修正しないものがある場合は開発者にその理由を確認し、妥当であることを確認する。 7．上記の確認で問題がないこと。 （注）#TEC-S03、#TEC-S04、#TEC-S05、#TEC-S14、#TEC-S15 での検証項目を含む	TST-06	V-1/実機テスト	必須	1 2 3 4 ※セキュリティ機能を持っている場合

ID	大項目	中項目	小項目	検証実施内容例	対応項目（表 2-10～2-12）	検証手法（表 2-2 の ID）	対応レベル	対象のソフトウェア構成
TEC-S11			・ファジング[17]テストツールを使用して、入力処理に関する問題を見つける。	1．仕様書などのドキュメントを確認し、ソフトウェアが外部から受け付けるエントリポイントを特定する。 2．特定した入力に対して、動作に問題を起こす可能性のあるデータを送り込み、その応答や挙動を確認する。 3．問題が起きた場合、原因解析を行い、挙動が仕様通りであることを確認する。問題なければ検証終了。 4．上記の指摘に対応したソフトウェアを再びチェックする場合は、再テストを行い、指摘事項が修正されていることを確認する。 5．意図的に修正しないものがある場合は開発者にその理由を確認し、妥当であることを確認する。 6．上記の確認で問題がないこと。	TST-07	V-4	高度	1 2 3 4 あるいは 5 ※1～4を個別に行うケースとソフトウェア全体を対象に行う場合の両者が想定される

[17] IPA：「脆弱性対策：ファジング」
https：//www.ipa.go.jp/security/vuln/fuzzing.html

ID	大項目	中項目	小項目	検証実施内容例	対応項目（表 2-10～2-12）	検証手法（表 2-2 の ID）	対応レベル	対象のソフトウェア構成
TEC-S12			・脆弱性スキャンを実施する。	1．脆弱性スキャンツールを使用し、既知の脆弱性がソフトウェアに内在しうるかを調べる。 2．内在する脆弱性が検出された場合、それぞれの脆弱性のスコアリング（例えば CVSS v3.1 など）とともに指摘レポートを作成する。問題なければ検証終了。 3．上記の指摘に対応したソフトウェアを再びチェックする場合は、再テストを行い、指摘事項が修正されていることを確認する。 4．意図的に修正しないものがある場合は開発者にその理由を確認し、妥当であることを確認する。 5．上記の確認で問題がないこと。	TST-08	V-3	必須	5

ID	大項目	中項目	小項目	検証実施内容例	対応項目（表 2-10～2-12）	検証手法（表 2-2 の ID）	対応レベル	対象のソフトウェア構成
TEC-S13			・ペネトレーションテスト[18]を実施する（攻撃者が危険度の高いシナリオでソフトウェアを侵害しようとする方法をシミュレートする）。	1．ソフトウェアが使用されるシステムに対する攻撃シナリオを検討した後、既知脆弱性の診断等で明らかになったシステムの脆弱性を悪用して攻撃者の目的を達成できるかどうかの確認を行う。 2．見つかった攻撃を、攻撃難易度、攻撃時間、影響などの指標（例えば OWASP 等）を用いてスコアリングし、指摘レポートを作成する。問題なければ検証終了。 3．上記の指摘に対応したソフトウェアを再びチェックする場合は、再テストを行い、指摘事項が修正されていることを確認する。 4．意図的に修正しないものがある場合は開発者にその理由を確認し、妥当であることを確認する。 5．上記の確認で問題がないこと。 （注）ここで実施する内容は、ソフトウェア、およびソフトウェアが使用される環境、#TEC-S10～#TEC-S12 での結果などに応じて異なる。	TST-09	V-3	高度	5

[18] IPA：「脆弱性検査と脆弱性対策に関するレポート」 https：//www.ipa.go.jp/about/technicalwatch/20130808.html

ID	大項目	中項目	小項目	検証実施内容例	対応項目（表 2-10～2-12）	検証手法（表 2-2 の ID）	対応レベル	対象のソフトウェア構成
TEC-S14	セキュリティの維持	ソフトウェアアップデートの手段を持つ。	ソフトウェアを最新の状態に保つ手段を持つこと。	1．ユーザマニュアルなどのドキュメントを確認し、ソフトウェアをアップデートする機能があること、およびアップデートする方法を確認する。 2．開発者に検査対象と更新用の最新ソフトウェアを用意してもらう。必要に応じて更新用の環境も用意してもらう（ローカルアップデート用の SD カードや、リモートアップデート用のサーバなど） 3．SBOM を参照するとともに、更新用ソフトウェアに対してソフトウェアコンポジションアナリシスツールを適用し、組み込まれているソフトウェアが最新であることを確認する。 4．ユーザマニュアルなどのドキュメントの手順に従い、ソフトウェアのアップデートを行う。ソフトウェアバージョン確認画面などで、最新のバージョンにアップデートされたことを確認する。 5．SBOM を参照し、ソフトウェアコンポーネントに OSS などのサードパーティソフトウェアを含む場合、現在サポートされているものであることを確認する。サポートされていない場合、最新の状態であることを確認する。 6．上記の確認で問題がないこと。	FNC-14	V-1/ V-2/ 実機テスト	必須	1 2 3 4

ID	大項目	中項目	小項目	検証実施内容例	対応項目（表 2-10～2-12）	検証手法（表 2-2 の ID）	対応レベル	対象のソフトウェア構成
TEC-S15			アップデートは検証してから適用すること。	1．社内の責任者が、ソフトウェアを構成するソフトウェアコンポーネントについて改ざん等がなく正当なものであることを保証していることを確認する。 2．仕様書などのドキュメントを参照し、ソフトウェアのアップデートを検証する方法が暗号的（#TEC-S05 にて検証）にセキュアであることを確認する。確認できた場合は 4 へ。 3．アップデートの検証方法が暗号的にセキュアでない場合、脅威分析（#TEC-S01 にて検証）の結果を確認し、実装されている代替手段による対応でセキュリティが十分保たれることを確認する。 4．実機テストを行い、ソフトウェアのアップデートが検証してから適用されていることを確認する。 ＜実機テストの例＞ 正規のもののバイナリの一部を書き換えたソフトウェア用意する。ユーザマニュアルなどのドキュメントの手順に従い、用意したソフトウェアを使ってアップデートを行い、アップデートが失敗することを確認する。 5．上記の確認で問題がないこと。	FNC-15	V-1	必須	1 2 3 4 ※自社から提供するソフトウェアに関して適用される

なお、表 2-18 の検証項目には、自社開発ソフトウェア、商用ソフトウェア、OSS によって検証の実施有無や対応が異なるものがある。表 2-19 にソフトウェアの種類に応じた対応について記載する。

表 2-19　ソフトウェア分類による実施要否

#	自社開発ソフトウェア	商用ソフトウェア	OSS
TEC-S01	○	○	○
TEC-S02	○	○ 対策が必要な脅威が存在する場合、商用ソフトウェアを提供しているサードパーティに対策を求めるか、それを取り込んだ自社開発ソフトウェアでの対策等でカバーしていること。	○ 対策が必要な脅威が存在する場合、OSS を開発しているコミュニティと連携して対策を入れるか、それを取り込んだ自社開発ソフトウェアでの対策等でカバーしていること。
TEC-S03	○	○ ソフトウェア自身に実行コード改ざん防止機能がなければ、それを取り込んだ自社ソフトウェアや実行環境等でカバーしていること。	○ ソフトウェア自身に実行コード改ざん防止機能がなければ、それを取り込んだ自社ソフトウェアや実行環境等でカバーしていること。
TEC-S04	○	○	× ソースが公開されているので難読化に効果はない。
TEC-S05	○	○	○
TEC-S06	○	△ サードパーティにセキュアコーディングを実施するよう要求していることが望ましい。	○
TEC-S07	○	△ ソースコードがないため、５のソフトウェアコンポジションアナリシスのみ実施する。	△ １以外を実施。ソースコードのフォーマットまでは変更しない。
TEC-S08	○	×	△ OSS 部分のソースコードのレビューも行うことが望ましい。

#	自社開発ソフトウェア	商用ソフトウェア	OSS
TEC-S09	○	△ 実施することが望ましいが、リバースエンジニアリングを伴うコードチェックは、ソフトウェアのライセンスに留意すること。	○
TEC-S10	○	○	○
TEC-S11	○	○	○
TEC-S12	○	○	○
TEC-S13	○	○	○
TEC-S14	○	○	○
TEC-S15	○	○ ソフトウェア自身にソフトウェアアップデートの検証機能がなければ、それを取り込んだ自社開発ソフトウェアや実行環境等でカバーしていること。	○ ソフトウェア自身にソフトウェアアップデートの検証機能がなければ、それを取り込んだ自社開発ソフトウェアや実行環境等でカバーしていること。

【凡例】○：実施する、△：一部実施する、×：実施しない

参考）検証ツールの例

この検証で使用するツールの例を次の表 2-20 に示す。対象がソフトウェア単体でなく、ソフトウェアが動作する実行環境であるものも含む。

表 2-20 検証で使用するツール例[19]

検証手法	ツール	関連項目番号
ソースコード解析	・CodeSonar ・Coverity ・Fortify Static Code Analyzer ・Veracode ・VDOO Vision ・BlackDuck	#TEC-S07
バイナリ解析	・angr ・Ghidra ・IDA Pro ・Binwalk Enterprise ・HERCULES SecSAM ・BlackDuck	#TEC-S09
ファジング	・American Fuzzy Lop ・beStorm ・Defensics ・Peach Fuzzer ・Raven	#TEC-S11
脆弱性スキャン	・Nessus ・Vuls ・Hydra ・GVM	#TEC-S12
侵入テスト	・Metasploit ・angr ・Ghidra ・IDA Pro	#TEC-S13

[19] 参考）株式会社インプレス
「攻撃手法を学んで防御せよ！押さえておくべき IoT ハッキング」2022 年 6 月 11 日初版発行

1.2.3.5.2　組織マネジメントにおける対策

1.2.3.3.2 項の表 2-13～表 2-16 にまとめた組織運用に関するセキュリティプラクティスのうち、検証対象がソフトウェア（自社開発ソフトウェア、サードパーティソフトウェア、商用ソフトウェア、OSS を含む）であるセキュリティプラクティス、表 2-2 の「組織運用（O-1～O-3)」に該当するセキュリティプラクティス、およびサプライチェーンに関わるセキュリティプラクティスを抜粋したものを表 2-21 に示す。

表 2-21 組織マネジメントにおけるセキュリティプラクティス

ID	大項目	中項目	小項目	表 3.2-4～3.2-7 で対応する項目の番号
ORG-S01	セキュリティ要件定義	製品/ソフトウェアのセキュリティ要件を定義し、文書化する。 製品/ソフトウェアのセキュアな設定/構成を決定し文書化する。 設定した定量的なセキュリティ基準に基づいて測定する。	以下について定義し、文書化すること。 ・開発プロセスの中で行われた仮定や製品に関係する想定など。 ⇒想定ユースケースや設置環境、ネットワークアクセス要件、想定される入出力データ、想定されるセキュリティ要件、関連法規、想定される寿命、など ・製品の設計関連 ⇒ハードウェアおよびソフトウェア（OSS、サードパーティソフト、内作)、セキュリティ要素（セキュアブートなど)、セキュアなソフトウェア開発プラクティス、セキュリティ認証結果、など ・セキュアなメンテナンスのための要件 ・リリース時、使用中、サポート終了後などのライフサイクルにわたって考慮されたセキュリティ ・脆弱性管理ポリシー	ORG-05

ID	大項目	中項目	小項目	表 3.2-4～3.2-7 で対応する項目の番号
ORG-S02			・ソフトウェア コンポーネントの安全な構成を決定し、開発者が構成を簡単に使用できるように、これらを使用可能にする （例えば、コードとしての構成として）。 ・ソフトウェアに対して承認された構成が正しく行われていることを確認する。 ・各設定の目的、オプション、デフォルト値、セキュリティの関連性、潜在的な操作上の影響、およびその他の設定との関係を文書化する。 ・プログラムによる技術的メカニズムを使用して、ソフトウェア管理者が各設定を実装および評価する方法を記録する。 ・デフォルトの構成を使用できる形式で保存し、変更管理の手法に従って変更する（例えば、コードとしての構成）。	ORG-06
ORG-S03			・ツールチェーンを使用して、セキュリティの意思決定を知らせる情報を自動的に収集する。 ・条件をサポートする情報の生成と収集をサポートするために必要な場合は、追加のツールを展開する。 ・条件を使用して意思決定プロセスを自動化し、これらのプロセスを定期的に見直す。 ・権限を持つ担当者のみが収集した情報にアクセスできるようにし、情報の変更や削除を防止する。	ORG-07

ID	大項目	中項目	小項目	表 3.2-4～3.2-7 で対応する項目の番号
ORG-S04		構成管理を行う。	・ソフトウェア構成管理のためのリポジトリを保守する。 ・出自データを組織のポリシーに従ってソフトウェアの取得者が利用できるようにする。 ・出自データを組織の運用チームおよび対応チームが利用できるようにして、ソフトウェアの脆弱性を軽減する支援を行う。 ・ソフトウェアのコンポーネントのいずれかが更新されるたびに、出自データを更新する。 ・フィールドに展開済みの古いバージョンのソフトウェアについて、新しいバージョンへの移行が正常に完了するまで、古いバージョンのソフトウェアを保持する。	ORG-13
ORG-S05			・出自データの整合性を保護し、受信者が出自データの整合性を検証する方法を提供する。	ORG-14
ORG-S06			・出自データは標準形式を使用する。	ORG-15
ORG-S07		構成管理により、ソースコードの改ざん防止策を講じる。	・すべてのソースコードとコードの構成をコードリポジトリに格納し、コードの性質に基づいてアクセスを制限する。 ・リポジトリのバージョン管理機能を使用して、個々のアカウントに対する説明責任を持ってコードに加えられたすべての変更を追跡する。 ・コード所有者に、他のユーザがコードに加えたすべての変更を確認し、承認する。 ・リリース・ファイル、関連イメージなどを、組織の確立されたポリシーに従ってリポジトリに保管する。必要な担当者が読み取り専用でアクセスできるようにし、他のユーザによるアクセスを許可しない。	ORG-16

ID	大項目	中項目	小項目	表 3.2-4～3.2-7 で対応する項目の番号
ORG-S08			・暗号化 （暗号化のハッシュなど） を使用して、ファイルの整合性を保護する。 ・リリース整合性検証情報を、リリース ファイルとは別の場所に保管したり、データに署名したりして保管し、保護する。	ORG-17
ORG-S09	サプライチェーンのセキュリティモデルの構築	サプライチェーンを考慮した標準、モデルを開発する。	IoT サプライチェーンのための標準を作る、もしくは既存の標準を IoT サプライチェーン適用する。	SPC-01
ORG-S10		サプライチェーンの脅威/信頼モデルを開発する。	サプライチェーンにおいて、自組織が担う役割を特定し関係者と共有する。	SPC-02
ORG-S11			あらかじめ定められた自組織の優先事業、優先業務と整合したセキュリティポリシー・対策基準を明確化し、自組織の取引に関係する者（サプライヤー、第三者プロバイダ等を含む）に共有する。	SPC-03
ORG-S12			リスクアセスメント結果及びサプライチェーンにおける自組織の役割から自組織におけるリスク許容度を決定する。	SPC-04
ORG-S13	契約	セキュリティに取り組んでいるサプライヤを採用する。	何らかの包括的なセキュリティ対策を実装しているサプライヤを採用する。	SPC-05
ORG-S14			一般的なセキュリティレコメンデーション（例：NISTIR 8259.10）を実装している、あるいはセキュリティ標準（例：ISO27036、ISO28000）に準拠しているサプライヤを採用する。	SPC-06
ORG-S15		セキュリティに関する契約を結ぶ。	セキュリティの監査を契約に含める。	SPC-07
ORG-S16			ソフトウェアに対して、セキュアブートやファームウェア署名を実装させるような契約を結ぶ。	SPC-08

ID	大項目	中項目	小項目	表 3.2-4～3.2-7 で対応する項目の番号
ORG-S17	サードパーティソフトウェア	組み込むにあたって評価を行う。	・サードパーティのソフトウェアを、想定されるユースケースでレビュー、評価する。（ユースケースが変わったらそのときに再びレビュー、評価を行う） ・ソフトウェアのセキュアな構成を決定する。	SPC-09
ORG-S18			・自社ソフト同様のコードレビュー、分析、テストを行う。	SPC-10
ORG-S19			・デジタル署名またはその他のメカニズムを使用して、ソフトウェアコンポーネントの整合性を確認する。	SPC-11
ORG-S20		構成管理を行う。	・各ソフトウェアコンポーネントの出自情報（SBOM、ソース構成分析、バイナリソフトウェア構成分析など）を分析して、脅威や OSS のライセンスなどのリスクを評価する。 ・認可した OSS をホストするソフトウェアリポジトリを構築する。 ・認可した商用ソフトウェアとバージョンのリストを、出自データと共に管理する。 ・フィールドに展開済みの古いバージョンのソフトウェアについて、新しいバージョンへの移行が正常に完了するまで、古いバージョンのソフトウェアを保持する。 ・バイナリで取得したソフトウェアの完全性または出自を確認する。確認できない場合は、ソースコードの完全性と出自を確認した後、ソースコードからバイナリをビルドする。	SPC-12

ID	大項目	中項目	小項目	表 3.2-4～3.2-7 で対応する項目の番号
ORG-S21		定期的に脆弱性などをチェックし、必要なアクションを取る。	・ソフトウェア/サービスに、ベンダがまだ修正していない既知の脆弱性があるかどうかを定期的に確認する。 ・ソフトウェアコンポーネントの既知の脆弱性を自動検出する機能をツールチェーンに組み込む。 ・各ソフトウェアコンポーネントが積極的に保守されており、サポートが切れていないことを確認する。 ・保守が終了しているか、近い将来メンテナンスが終了するソフトウェアコンポーネントに対して、アクションプランを決定する。	SPC-13

1.2.4 ＜調査項目３＞検証のためのルール作りや認証等を行う検証機関のあり方調査

1.2.4.1　調査の背景、目的

本調査項目（調査項目３）では、検証のためのルール作りや認証等を行う検証機関（検証センター）のあり方を国内外の実態調査を踏まえて、国内で実現するために必要な要件を調査する。また、日本国内で実施可能かつ実効的な検証機関の制度・ルール・体制・手順などをまとめた指針書を作成する。

1.2.4.2　検証機関の認定基準の調査

本項では、国内で既に制度運用されている認証機関への要求事項及び、検証機関の認定に関する要求事項を、ISO 規格を中心に調査を実施した。また調査結果をもとに、国内で実現可能かつ、実効的な認証機関の要件及び、検証機関の認定基準に対する要件の整理を行う。

表 3-1　調査対象文書一覧

規格・文書名	発行元	発行年
ISO/IEC 17065 : 2012　適合性評価－製品，プロセス及びサービスの認証を行う機関に対する要求事項	ISO/IEC 規格文書	2012 年 12 月制定
ISO/IEC 17025 : 2017　試験所及び校正機関の能力に関する一般要求事項	ISO/IEC 規格文書	2017 年 11 月改正
CCM-03　IT セキュリティ評価機関承認等に関する要求事項（JISEC[20]の評価機関認定基準）	IPA	2021 年 10 月改訂
※参考情報 システム・ソフトウェア品質標準 SQuaRE シリーズ	ISO/IEC 規格文書	※規格文書によって異なる

[20] IT セキュリティ評価及び認証制度（JISEC：Japan Information Technology Security Evaluation and Certification Scheme）」IPA
https：//www.ipa.go.jp/security/jisec/scheme/index.html

1.2.4.2.1　ISO/IEC17065：2012 に定義された認証を行う機関への要求事項

ISO/IEC 17065 （JIS Q 17065）は、認証機関が製品、プロセス、サービスの認証機関に対する能力、一貫性のある運営や公平性に関する要求事項に規定した国際標準規格文書となる。2012 年 12 月に制定された。

ISO/IEC 17065 は認証機関に求められる要求事項について、大きく下記の 5 つの項目に分類され、定義されている。

（1）一般的な要求事項：公平性の担保や、機密保持を含む法的責任に対する要求事項
（2）組織構造や意思決定に対する要求事項：組織の構造やマネジメントに対する要求事項
（3）資源に対する要求事項：資源（要員や施設、設備等）に対する要求事項
（4）認証プロセスに対する要求事項：認証の申請や、評価、決定など一連の認証プロセスや情報管理等に対する要求事項
（5）マネジメントシステムへの要求事項：文書や記録の管理、内部監査や予防、是処置等に対する要求事項

1.2.4.2.2　ISO/IEC17025：2017 に定義された試験所・校正機関に関する要求事項

ISO/IEC 17025 （JIS Q 17025）は ISO 9001：1994 をベースに、試験所・校正機関に対する固有の要求事項を付加した国際標準規格となる。

試験所・校正機関の能力を、認定機関が認定する際の基準として利用される。2005 年に制定され、2017 年 11 月に規格の改正版が制定された。 ISO/IEC 17025 の認定を受けた試験所・校正機関が発行する証明書類には、認定マークを記載可能であり、国際的に通用する証明書としての信頼性を高めることができる。本書においては、検証機関に求める要求事項の参考として調査を実施した。

ISO/IEC17025 は試験所・校正機関に求められる要求事項について、大きく下記の 5 つの項目に分類され、定義されている。

（1）一般的な要求事項：公平性の担保や、機密保持に関する要求事項
（2）組織の構成に対する要求事項：試験所・校正機関の組織構成や管理構造、マネジメントシステム等への要求事項
（3）資源に対する要求事項：要員や施設及び環境、設備、製品・サービスの外部提供に関する要求事項
（4）試験のプロセスに対する要求事項：契約レビューや試験の方法、校正や記録等の一連の試験プロセスに関する要求事項
（5）マネジメントシステムへの要求事項：文書や記録の管理、内部監査や予防、是処置等に対する要求事項

1.2.4.2.3　IT セキュリティ評価及び認証制度における要求事項

「IT セキュリティ評価及び認証制度（JISEC：Japan Information Technology Security Evaluation and Certification Scheme）」は国際標準規格「ISO/IEC15408（Common Criteria）」に基づく適合性を審査する認証制度となる。

本制度では、IPA が認証機関として JISEC を運営しており、認証機関としての基本方針、評価機関が承認されるための要求事項を定めている。

認証機関としては、ISO/IEC 17065 に準拠した組織、運営方針、組織運営に関する基本方針を「CCM-01　IT セキュリティ認証機関の組織及び業務運営に関する規程」として定め、公開している。

評価機関に対する要求事項としては、評価機関が認証機関による承認を得るために必要な事項、承認を得た評価機関がその承認を維持するために必要な事項を「CCM-03　IT セキュリティ評価機関承認等に関する要求事項」として公開している。同文書は ISO/IEC 17025 に準拠しつつ、CC 認証を対象としたより詳細な条件を定義している。参考情報として、表 3-2 に JISEC における評価機関の承認プロセスの概要を示す。

本制度で認証機関、評価機関に求められる要求事項は、いずれも前項で記載した ISO/IEC 規格に準拠しているため、内容については 1.2.4.2.1 項、1.2.4.2.2 項を参照とする。

表 3-2　参考情報）JISEC における評価機関の承認プロセス概要

出典）IPA「IT セキュリティ評価及び認証制度に関する説明会」P.56～P.59 より抜粋[21]

認証機関（JISEC）による承認審査		
運営審議委員会	認定審査に先立ち、認証機関に評価機関として参入希望する旨を通知	□事前確認 ・認定審査に先立ち、認証機関に評価機関として参入希望する旨を通知 ・参入目的、製品分野、事業情報を提供
		□運営審議委員会での審議 ・参入の妥当性や必要性を官・学の有識者により事前に判断
教育訓練プログラム	評価機関として適切な訓練プログラムを有し定期的に実施されることを確認	□確認事項 ・評価技術に関する教育プログラム ・最新のセキュリティ技術情報の共有手段 ・脆弱性分析、侵入テスト等の訓練状況 ・対象分野の PP 等の教育状況
評価者資格	評価者は認証機関との評価業務に係	□評価者要件

[21] IPA「IT セキュリティ評価及び認証制度に関する説明会」
https：//www.ipa.go.jp/security/jisec/seminar/documents/cc_semi_20170317.pdf

	る窓口 評価機関には 1 名以上の評価者資格保持者を要する	・情報技術処理の専門知識、脆弱性や侵入テストの経験 ・制度規程を理解 ・試行評価において評価ならびに報告書作成が適切に実施できる ・公平性・コミュニケーション能力

1.2.4.2.4　参考情報）システム・ソフトウェア品質標準 SQuaRE シリーズ

本項では参考情報として、セキュリティを含むソフトウェア品質に関する要求事項の調査結果を示す。

ISO/IEC JTC 1 SC 7/WG 6（Software Product and System Quality）では ISO/IEC 25000 SQuaRE（Systems and software Quality Requirements and Evaluation）シリーズとして、日本主導によるシステムおよびソフトウェアの品質要求事項および評価の国際標準化を推進している。ISO/IEC 25000 SQuaRE シリーズは、審議中の内容も含めて、規格が幅広く、品質管理のサイクルの定義や、要求事項の構造モデル、要求事項の詳細定義や評価プロセス、評価結果の定量的な測定方法の定義など、全 6 部門に及ぶ検討が行われている（規格の策定状況と概要を表 3-3 に示す）。またセキュリティについては、品質への要求事項の構造モデルを示す ISO/IEC 25010 において品質特性の一部として定義されている（図 3-1）。

表 3-3 システム・ソフトウェア品質標準 SquaRE シリーズの文書一覧

出典）IPA「つながる世界のソフトウェア品質ガイド」P.26　表 2.2-1 より

部門	規格番号	内容
品質管理部門	ISO/IEC 25000	システム及びソフトウェアの品質要求定義と評価に関する基本概念と用語を定義
	ISO/IEC 25001	SQuaRE の考え方に沿って品質要求定義や品質評価を実施する組織が、それらに関する技術やノウハウを蓄積し、標準化し、プロジェクトでの適用を支援し、改善するサイクルを規定
品質モデル部門	ISO/IEC 25010	システム及びソフトウェアの品質の構造をモデル化し、それを構成する品質の観点（品質特性及び品質副特性）を、利用時の品質モデル及び製品品質モデルとして定義
	ISO/IEC 25011 （審議中）	システム及びソフトウェアを用いたサービスの品質の構造をモデル化し、それを構成する品質の観点（品質特性及び品質副特性）をサービス利用時の品質モデル及びサービス製品品質モデルとして定義

	ISO/IEC 25012	システム及びソフトウェアが処理するデータの品質の構造をモデル化し、それを構成する品質の観点（品質特性）をデータ品質モデルとして定義
品質測定部門	ISO/IEC 25020	品質測定の基本概念と作業要件を規定
	ISO/IEC 25021	品質測定量の算出によく用いられる要素（例：規模、欠陥数など）を品質測定量要素（QME： Quality Measure Elements）として定義
	ISO/IEC 25022 （審議中）	ISO/IEC 25010 の利用時の品質モデルの品質特性/品質副特性を定量化するのに用いる品質測定量を定義
	ISO/IEC 25023 （審議中）	ISO/IEC 25010 の製品品質モデルの品質特性/品質副特性を定量化するのに用いる品質測定量を定義
	ISO/IEC 25024 （審議中）	ISO/IEC 25012 のデータ品質モデルの品質特性を定量化するのに用いる品質測定量を定義
品質要求部門	ISO/IEC 25030	品質要求定義の基本概念と作業要件を規定
品質評価部門	ISO/IEC 25040	品質評価の基本概念と作業要件を規定
	ISO/IEC 25041	開発者、取得者、独立評価者のそれぞれの立場に応じた品質評価の進め方を規定
	ISO/IEC 25045	回復性（Recoverability）の実用的な品質測定量を規定
拡張部門	ISO/IEC 25051	品質保証部門や独立評価機関が実施する既製ソフトウェア製品（RUSP： Ready to Use Software Product）の第三者評価を規定
	ISO/IEC 25060 ～ 66 （一部審議中）	ユーザビリティの要求定義と評価に用いる様式を規定

（注） カッコ内は 2015 年 3 月時点

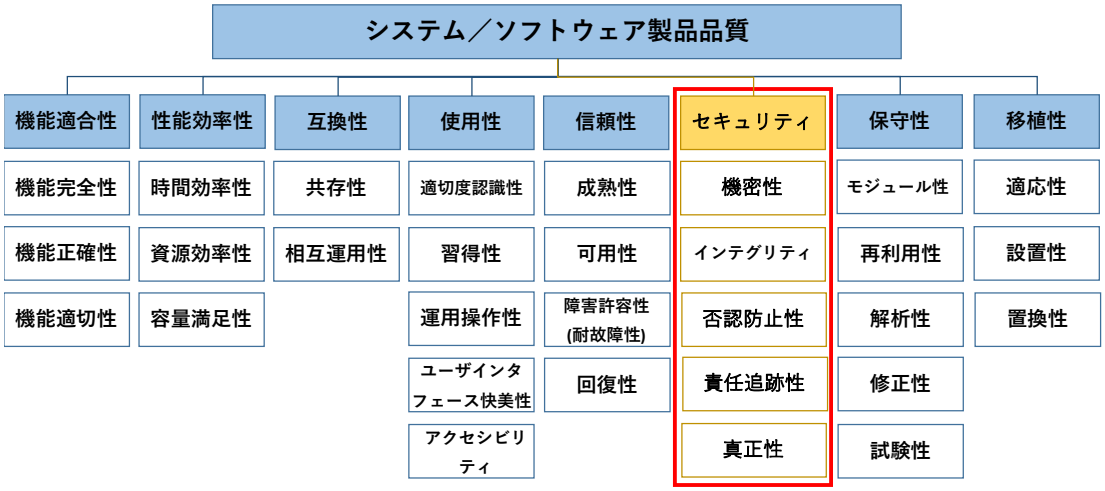


図 3-1 ISO/IEC 25010：2011 における「セキュリティ」の位置づけ

1.2.4.2.5　ソフトウェア（OSS）に対する認証機関、検証機関への要求事項の考察

ソフトウェア（OSS）に対する認証機関、検証機関への要求事項の考察を行うにあたり、まずは前項まで調査を行った ISO 規格で求められる認証機関、検証機関への要求事項を整理する。

■ISO/IEC 17065 で定義されている認証機関への要求事項の整理

①公平性（認証対象となる製品やサービスの事業活動と営利的な取引関係がない）
　の担保、機密保持の徹底
②事業運営が可能な安定した財務基盤や債務状況
③資源（知識や力量を持つ要員、施設や環境、設備等）の確保
④一連の認証プロセスの管理及び、記録の継続的な保持
⑤ISO9001（もしくは ISO/IEC 17065）に基づくマネジメントシステムの保持

■ISO/IEC 17025 で定義されている検証機関への要求事項の整理

①公平性の担保、機密保持の徹底
②組織構成や管理構造の明確化、マネジメントシステムの実施や維持、改善
③資源（要員、検証の施設、環境、設備）の確保、外部提供製品、サービスの管理
④一連の検証プロセスの管理及び、記録の継続的な保持（レビュー、検査、設備校正等）
⑤ISO9001（もしくは ISO/IEC 17025）に基づくマネジメントシステムの保持

次に ISO 規格で求められる認証機関、検証機関への要求事項について、ソフトウェア（OSS）を認証の対象とした場合に、ヒアリング結果（表 1-8）を前提とした条件緩和事項に対する考察を示す。

■認証機関に対する要求事項（ISO/IEC 17065）の考察

①公平性の担保、機密保持の徹底：

・緩和すべき事項の有無：なし

・緩和すべき理由：ー

・参考の緩和案：ー

②事業運営が可能な安定した財務基盤や債務状況：

・緩和すべき事項の有無：なし

・緩和すべき理由：ー

・参考の緩和案：ー

③資源（知識や力量を持つ要員、施設や環境、設備等）の確保

・緩和すべき事項の有無：なし

・緩和すべき理由：ー

・参考の緩和案：ー

④一連の認証プロセスの管理及び、記録の継続的な保持

・緩和すべき事項の有無：なし

・緩和すべき理由：ー

・参考の緩和案：ー

⑤ISO9001（もしくは ISO/IEC 17065）に基づくマネジメントシステムの保持

・緩和すべき事項の有無：あり

・緩和すべき理由：ヒアリングの結果、認証制度については、対応負荷やコスト負担をなるべく低減したい旨の要望が提示されていた。ISO9001 もしくは、ISO17065 に準拠したマネジメントシステムの維持は、認証機関にとっても対応の負荷が高く、認証コストの増加につながる。

・参考の緩和案：認証制度のスキームの中で、最低限管理が必要すべき事項を明確化することで条件を緩和する。

■検証機関に対する要求事項（ISO/IEC 17025）の考察

①公平性の担保、機密保持の徹底：

・緩和すべき事項の有無：あり
・緩和すべき理由：ヒアリングの結果、自社での検証と第三者での検証を柔軟に選択可能な制度が要望されていた。ISO の要望事項では、公平性の担保は求められているが、自社での検証（自己適合性評価）の実施可否については明確に定義されていない。
・参考の緩和案：開発に関わらない要員による検証実施を義務付け、公平性を担保し、自己適合性評価の実施を可能とする。

②組織構成や管理構造の明確化、マネジメントシステムの実施や維持、改善：
・緩和すべき事項の有無：なし
・緩和すべき理由：－
・参考の緩和案：－

③資源（要員、検証の施設、環境、設備）の確保、外部提供製品、サービスの管理
・緩和すべき事項の有無：あり
・緩和すべき理由：ヒアリングの結果、検証機関の認定基準は、第三者機関への委託もしくは自社での検証のどちらも可能となるような適度な基準が要望されている。要員の力量に関する要求事項への対応（必要スキル定義、教育訓練の実施記録の保持、監視、監督等）は対応のハードルが高く、自己適合性評価を希望する企業の認定取得を困難にする恐れがある。
・参考の緩和案：対象となる認証の資格、講習制度に対応し、資格取得者の記録を管理することで代替可能とする。

④一連の検証プロセスの管理及び、記録の継続的な保持（レビュー、検査、設備校正等）
・緩和すべき事項の有無：なし
・緩和すべき理由：－
・参考の緩和案：－

⑤ISO9001（もしくは ISO/IEC 17025）に基づくマネジメントシステムの保持
・緩和すべき事項の有無：あり
・緩和すべき理由：ヒアリングの結果、認証や検証の費用は製品価格に対して現実的な費用であることが要望として提示されていた。ISO9001 もしくは、ISO17025 に準拠したマネジメントシステムの維持は、検証機関にとって対応の負荷が高く、検証コストの増加につながる。

・参考の緩和案：認証制度のスキームの中で、最低限管理が必要すべき事項を明確化することで条件を緩和する。

1.2.4.3　認証制度の調査結果

本項では、実現可能かつ実効的な OSS の認証制度を提案するにあたり、既に運用中のセキュリティ関連の国内外の認証制度について調査した。

1.2.4.3.1　既存の認証制度の調査結果

調査候補とした認証制度を、国際規格による認証制度と独自標準の認証制度に分けてそれぞれ、表 3-4、表 3-5 に示す。

表 3-4 国際規格の認証制度

対象認証制度	規格	概要
IT セキュリティ評価及び認証制度（JISEC）	・JIS X 5070 ・ISO/IEC 15408（Common Criteria）	ISO/IEC によって標準化されている、IT 製品が備えるべきセキュリティ機能が適切に開発され実装されているかを評価する認証制度。主に政府調達で利用されている。
EDSA	IEC 62443-4	ISA（国際計測制御学会）によって開発され、IEC によって標準化されている、組込機器、制御機器のセキュリティ保証に関する認証制度。IEC 62443-4-2：2019 の発行に伴い、CSA（Component Security Assurance）認証に刷新された。

表 3-5 独自標準の認証制度

対象認証制度	規格	概要
PCI[22] Secure Software Program	PCI Secure Software Standard（PCI 独自規格）	PCI Security Standards Council によって開発、実施されているクレジットカード決済関連ソフトウェアを対象とするセキュリティ認証プログラム。
ioXt Certification	ioXt Alliance 独自規格	ioXt Alliance によって開発、実施されている IoT 製品を対象とするセキュリティ認証プログラム。PSA Certified 認証を取得済みなら ioXt 要件の一部を満たしているとみなす、認証プ

[22] Payment Card Industry の略。国際クレジットカードブランド会社によって設立された協議会：PCI SSC (Payment Card Industry Security Standards Council)が策定しているセキュリティ規格群の総称。

		ログラム間の連携あり。
PSA Certified	PSA Certified 独自規格	チップ、デバイス、システムソフトウェアを対象とするセキュリティ認証プログラム。

各認証制度について調査した結果の概要をまとめたものを一覧にし、表 3-8 に示す。各制度の調査を進めた結果、ioXt Certification はアライアンスメンバにのみ詳細情報が公開されており、公開されている情報の範囲で記載を行った。また、PSA Certified はチップに装備した Root of Trust を根拠とするセキュリティ方式の認証であり、本書の表 2-10 の FNC-24「ハードウェアレベルの認証メカニズムを実装する」に挙げた項目に分類される方式であるが、ハードウェアをセキュリティの根拠とする方式は本書では検討対象外となるため、参考情報として表 3-6 に記載した。

表 3-6 調査対象とした認証制度まとめ

項目名	JISEC	EDSA	PCI Secure Software Program	ioXt Certification
概要	ISO/IEC によって標準化されている、IT 製品が備えるべきセキュリティ機能が適切に開発され実装されているかを評価する認証制度。主に政府調達で利用されている。	ISA（国際計測制御学会）によって開発され、IEC によって標準化されている、組込機器、制御機器のセキュリティ保証に関する認証制度。IEC 62443-4-2：2019 の発行に伴い、CSA（Component Security Assurance）認証に刷新された。	PCI Security Standards Council によって開発、実施されているクレジットカード決済関連ソフトウェアを対象とするセキュリティ認証プログラム。	ioXt Alliance によって開発、実施されている IoT 製品を対象とするセキュリティ認証プログラム。PSA Certified 認証を取得済みなら ioXt 要件の一部を満たしているとみなす、認証プログラム間の連携あり。
認証スキームタイプ	国際規格に準拠した認証スキーム	国際規格に準拠した認証スキーム	独自プログラムによる認証スキーム	独自プログラムによる認証スキーム
認証対象	情報技術関連製品およびシステム	産業用制御機器	クレジットカード決済関連ソフトウェア	IoT 製品 ※ioXt Base Profile／ioXt Android Profile／ioXt Smart Speaker Profile／ioXt Mobile Application Profile／ioXt Residential Camera Profile の 5 プロファイルあり、製品のカテゴリに合わせて適用
認証機関	IPA （独立社団法人　情報処理推進機構）	CSSC-CL（CSSC 認証ラボラトリ）	PCI SSC	ioXt Alliance

項目名	JISEC	EDSA	PCI Secure Software Program	ioXt Certification	
認定機関	NITE 認定センター（IAjapan）	JAB（公益財団法人　日本適合性認定協会）	PCI SSC	ioXt Alliance	
検証機関	NITE 認定センターが認定した事業者[23]（３社）	CSSC-CL（CSSC 認証ラボラトリ）[24]	PCI SSC が認定した事業者[25]（４１社）	■認定ラボテスト ioXt Alliance が認定した事業者[26]（７社）	■自己テスト 申請者

[23] https：//www.ipa.go.jp/security/jisec/eval-list.html
[24] http：//www.cssc-cl.org/jp/aboutus/index.html
[25] https：//listings.pcisecuritystandards.org/assessors_and_solutions/software_security_framework_assessors
[26] https：//ja.ioxtalliance.org/authorized-labs

項目名	JISEC	EDSA	PCI Secure Software Program	ioXt Certification
取得コスト	・EAL1：539,000 円 ・EAL2：704,000 円 ・EAL3：825,000 円 ・EAL4：1,045,000 円 ・EAL5：1,133,000 円 ※別途、評価費用が検証機関に対して必要。製品、EAL によって異なる。	・申請料：35 万 ・FSA＋SDSA 審査料：500 万 ・CRT 審査料：300 万 ・認証料：110 万 ・審査付帯費用：25 万 ・認証書等発行費用：3 万 ・ISCI の Web サイトへの情報掲載料： 会員 82.5 万/非会員 137.5 万 ※追加審査料含め検証機関よりプライスリストを提出	・申請料（Web サイト登録込み）：3,000 USD ※別途、評価費用が検証機関に対して必要。ソフトウェアによって異なる。	非公開
審査対象	・開発プロセス ・実機テスト	・開発プロセス ・実機テスト	・開発プロセス ・実機テスト	※公開情報から推定 ・実機テスト
認証の段階（レベル）	EAL1～EAL7 の 7 段階のセキュリティレベル	レベル 1～3 の 3 段階のセキュリティレベル	階層なし（合否のみ）	階層なし（合否のみ） ※要件にレベル 1 から最大 5 までの 5 段階のセキュリティレベルあり。プロファイルによって要求されるレベルが決まっており、合格にはそのレベルと同等あるいは超える要件を満たす必要がある。

項目名	JISEC	EDSA	PCI Secure Software Program	ioXt Certification	
認証申請方法	第三者認証（第三者検証）	第三者認証（第三者検証）	第三者認証（第三者検証）	第三者認証（第三者検証）	第三者認証（自己評価）
テスト/評価方法の規定	CEM（Common Criteria for Information Technology Security Evaluation）として共通の評価手法が定義	・EDSA 認証は下記の３つの要素で構成され、それぞれレベル別に評価項目が定義されている。 １）ソフトウェア開発セキュリティ評価（SDSA）：130～169 項目 ２）情報セキュリティ評価（FSA）：19 項目～82 項目 ３）通信ロバストネス試験（CRT）：69 項目（どのレベルも一定）	・PCI SSS 規格（PCI Secure Software Standard）に、要件かつテスト要件が規定されている。	・プロファイルごとにテストケースが規定されている。	

認証申請に必要な文書群	■申請者 → 認証機関 A） 認証申請書（TOE） B） 法人格を証明できる書類 C） 誓約書 D） 評価用提供物件のリスト、提供スケジュール E） TOE の理解に役立つ資料 F） ST G） 適合する PP ※PP 適合の場合 H） 評価作業実施計画書 I） 評価の公平性及び独立性チェックリスト（検証機関用、評価者用） J） 秘密保持契約書（申請者用、開示者用） K） 認証中の申請案件掲載依頼書 ※希望時 ■申請者 → 検証機関 A） 評価用提供物件	■申請者 → 検証兼認証機関 A） 認証申請書 B） 別紙　申請対象製品情報シート	■申請者 → 認証機関 A） 同意書（Vendor Release Agreement） ■申請者 → 検証機関 A） 申請書兼誓約書（Attestation of Validation） B） 実装ガイド（Implementation Guidance） C） 開発証拠資料 ■検証機関 → 認証機関 A） 検証レポート（Reports on Validation）	※公開情報から推定 ■申請者 → 認証機関 A） 製品情報 B） テスト結果レポート（必要に応じて開発証拠資料含む） ■申請者 → 検証機関 A） 開発証拠資料 ■検証機関 → 認証機関 なし	・同左 申請者→検証機関はなし

項目名	JISEC	EDSA	PCI Secure Software Program	ioXt Certification	
	■検証機関 → 認証機関 A） 評価報告書				
認証審査の方法	・申請書面による審査 ・立ち合いによる審査 ・開発エビデンスや評価結果に基づく審査	・申請書面による審査 ・立ち合いによる審査 ・開発エビデンスや評価結果に基づく審査	・申請書面による審査 ・実機テストによる審査（リモート審査もあり） ・関係者へのインタビューによる審査 ・開発エビデンスや評価結果に基づく審査	※公開情報から推定 ・申請書面による審査 ・開発エビデンスや評価結果に基づく審査	・同左
認証書	認証報告書、認証書の発行	認証報告書、認証書の発行	PCI SSC によって連署された申請書兼誓約書（Attestation of Validation）。専用の認証書はなし。	・なし	
Web での公開	あり：IPA の Web ページにてリスト公開 ※要望に応じて英文 ST 含め、CCRA のポータルサイトに掲載	あり：CSSC-CL、ISASecure の Web ページにてリスト公開	あり：PCI SSC の Web ページにてリスト公開	あり：ioXt Alliance の Web ページにてリスト公開（ラボ認証なのか自己認証なのかの区別あり）	
認証マーク	あり：JISEC の認証マーク、CCRA 認証マーク	あり：ISASecure の認証マーク発行	なし	あり：ioXt の認証マーク（ioXt SmartCert）発行	

項目名	JISEC	EDSA	PCI Secure Software Program	ioXt Certification	
認証取得の実績	740 件 2002～2022 年 ※日本での実績	５件 2014～2022 年 ※日本での実績	14 件 2020 年～2022 年 ※グローバルでの実績	39 件（black box：37 件、white box：2 件） 2014～2022 年 ※グローバルでの実績	196 件 2014～2022 年 ※グローバルでの実績

※認証取得の実績等の数字は 2022 年 7 月 12 日時点

次に、調査した認証制度ごとに制度の詳細を説明する。本項では JISEC、EDSA、PCI Secure Software Program があるべき OSS のセキュリティ認証制度の参考となると考え、詳細を以下に示す。

1.2.4.3.1.1 JISEC 認証制度

■対象

デジタル複合機やバイオメトリクス照合製品、データベース管理システムといった、情報技術関連製品およびシステムを対象としている。

■特徴

認定機関である IAJapan[27]が JIS Q 17025 または ISO/IEC 17025 に基づいて検証機関を認定する。その検証機関が申請された製品やシステムの適合性評価を行い、第三者認証機関である IPA が認証を行うスキームである。IPA は認証機関として、JIS Q 17065 または CCRA[28]で規定された要件を満たすように体制を整備し、運営を行っている。

認証対象の製品カテゴリに適した定義済みセキュリティ要件（PP：Protection Profile）に対し、セキュリティ設計仕様（ST：Security Target）と実装が適切かを確認するものである。

政府調達において、情報システムの構成上、攻撃にさらされやすいもの、あるいは情報システムの基盤となるもの、あるいは攻撃事例の報告が多いものといった観点から、デジタル複合機やファイアウォールなどに対して調達要件になっている[29]。また、セキュリティ評価基準の国際標準である ISO/IEC 15408 に基づき、CCRA 加盟国間で認証製品を相互承認する制度が確立されている。

JISEC 認証制度では、認証機関が評価開始前のキックオフミーティングに参加し、検証機関による開発現場の評価であるサイト訪問に同行するなど、認証機関が評価開始時から関わる。

■認証要件

TOE に適用するセキュリティ機能要件として、大きく 11 の機能分類が定義されており、そ

---

27 独立行政法人 製品評価技術基盤機構（NITE）の適合性認定分野を担当している認定センターの呼称、International Accreditation Japan の略。

28 CC 承認アレンジメント。Common Criteria Recognition Arrangement の略。国内に認証制度をもつ認証国は 17 カ国、認証制度は持たないが認証された製品を受け入れる受入国は 14 カ国（2022 年 6 月時点）

29 IPA, セキュアな製品調達のために, https：//www.ipa.go.jp/security/jisec/choutatsu/index.html
経済産業省, IT 製品の調達におけるセキュリティ要件リスト, https：
//www.meti.go.jp/policy/netsecurity/cclistmetisec2018.pdf

の下位分類として機能ファミリ、さらなる下位分類としてコンポーネントが定義されている。

機能分類：

- セキュリティ監査
- 通信
- 暗号サポート
- 利用者データ保護
- 識別と認証
- セキュリティ管理
- プライバシー
- TSF の保護
- 資源利用
- TOE アクセス
- 高信頼パス/チャネル

さらに、設計から製品化に至る過程でセキュリティ機能が確実に実現されていることを保証するための要件として、セキュリティ保証要件も定義されている。機能分類と同様、下位の分類として保証ファミリ、保証コンポーネントが定義されている。

セキュリティ保証要件：

- PP 評価
- ST 評価
- 開発
- ガイダンス文書
- ライフサイクルサポート
- テスト
- 脆弱性評定
- 統合

このセキュリティ保証要件にどの程度対応しているかによって EAL（評価保証レベル）が決定される。EAL は EAL1 から EAL7 までの 7 段階あり、それぞれの段階で要求されるセキュリティ保証要件が保証コンポーネント単位で定義されている。

■認証タイプ
第三者検証による第三者認証

■認証申請の関係機関

・認証機関： 独立社団法人　情報処理推進機構（IPA）

・認定機関： 独立行政法人　製品評価技術基盤機構（NITE）認定センター（IAJapan）

・検証機関： IAJapan が認定した事業者

■認証申請費用

EAL1： 539,000 円

EAL2： 704,000 円

EAL3： 825,000 円

EAL4： 1,045,000 円

EAL5： 1,133,000 円

※検証費用については別途検証機関の見積もりによる

■認証申請に必要な文書

申請者→認証機関：

A） 認証申請書（TOE）

B） 法人格を証明できる書類

C） 誓約書

D） 評価用提供物件のリスト、提供スケジュール

E） TOE の理解に役立つ資料

F） ST

G） 適合する PP ※PP 適合の場合

H） 評価作業実施計画書

I） 評価の公平性及び独立性チェックリスト（評価機関用、評価者用）

J） 秘密保持契約書（申請者用、開示者用）

K） 認証中の申請案件掲載依頼書 ※希望時

申請者→検証機関：

A） 評価用提供物件

検証機関→認証機関：

A） 評価報告書

■JISEC 認証スキーム（図 3-2）

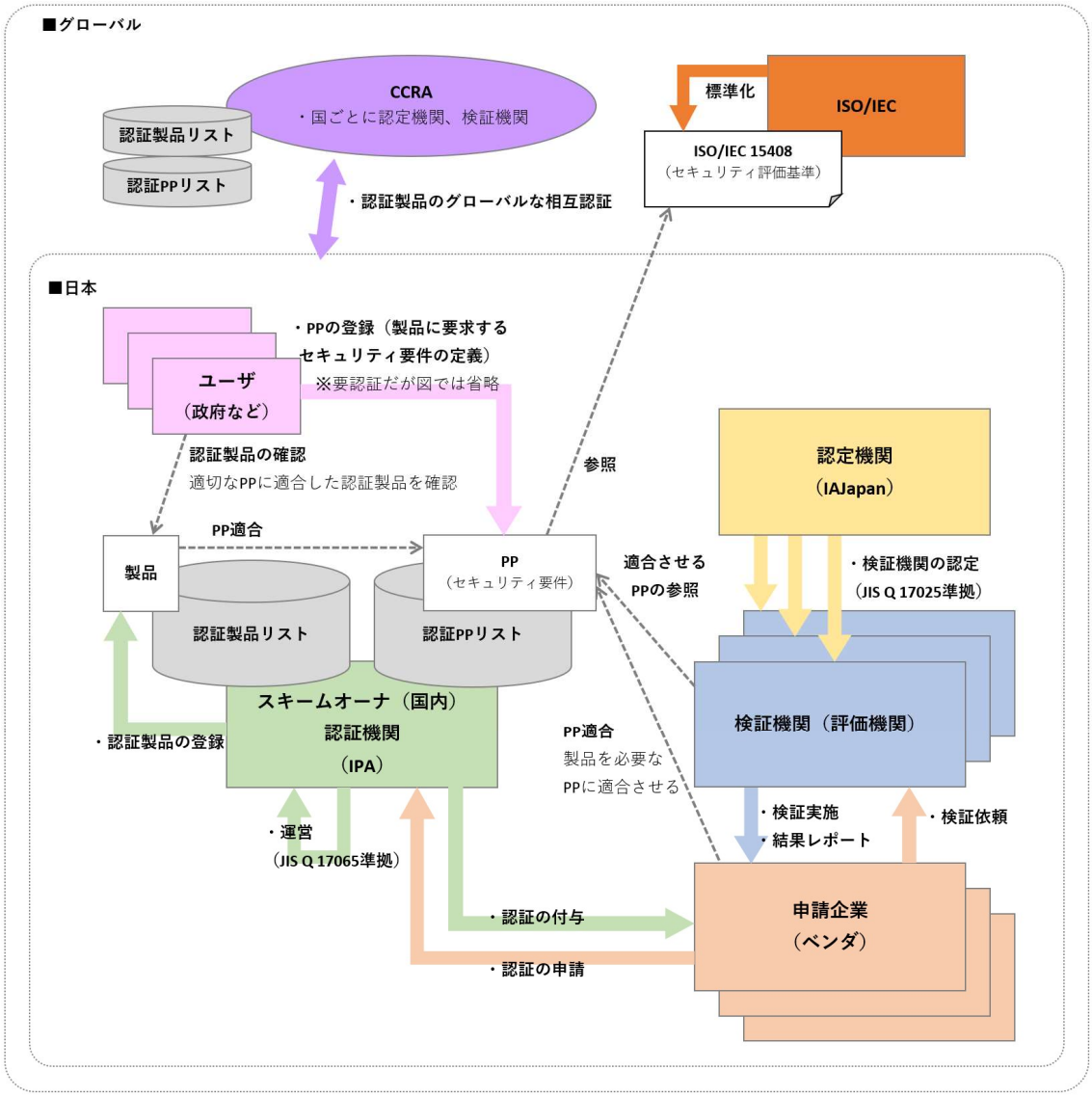
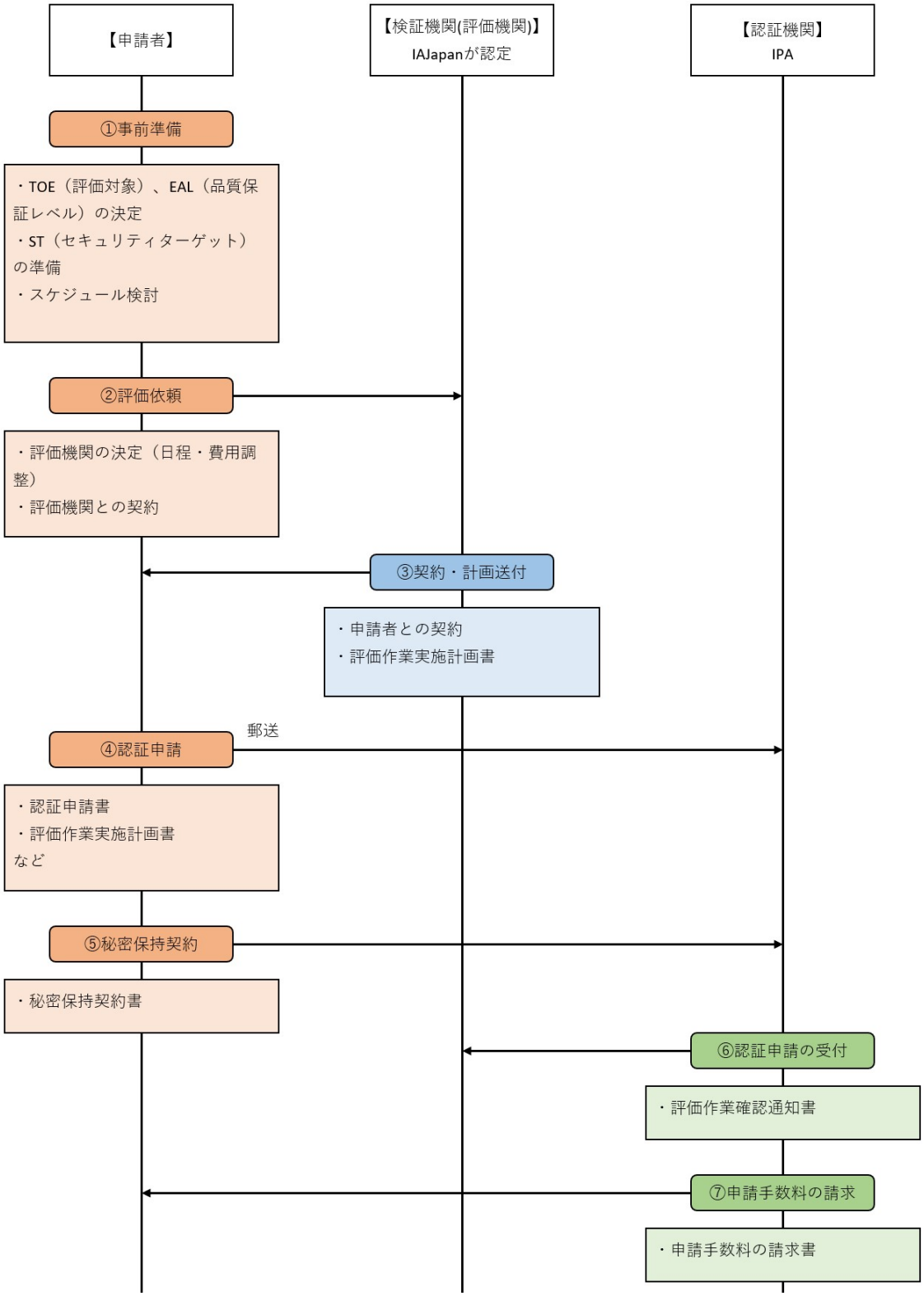


図 3-2　JISEC における認証スキーム

■JISEC 認証申請プロセス（図 3-3）



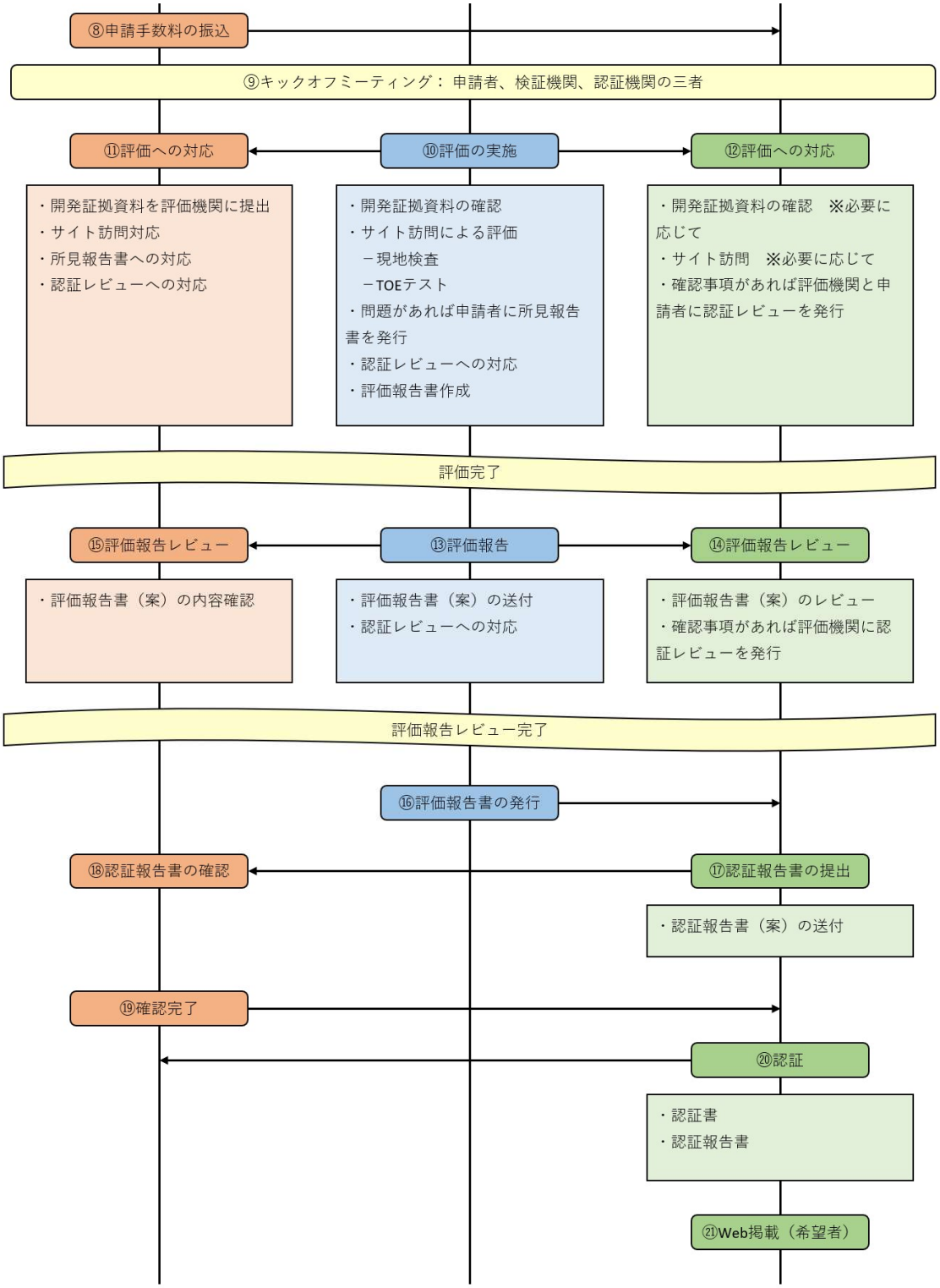


図 3-3　JISEC における認証フロー図

■考察

CCRA 加盟国間で認証が共有されるため、OSS を含むソフトウェアのセキュリティ認証要件がグローバルで統一された場合、同一製品をグローバル展開するベンダにとっては、認証処理を一度行えばよいので負担の削減が期待できる。一方、セキュリティ要件や適合基準が日本国内のみに適用するものである場合や、グローバルに製品を展開していないベンダにとっては、これはメリットとなりづらい。

設計書レベルでの整合性を、第三者がトレーサビリティをもって確認するので申請者の負担は大きいと考えられる。

調査結果 1 の要望（表 1-8）と JISEC 認証制度と比較考察した結果を表 3-7 に示す。なお、あくまで本調査における要望との比較であって当認証制度自体の良し悪しの判断ではないことに留意。

表 3-7 要望と JISEC 認証制度との比較

ID	要望事項	要望との合致	比較内容
R1	第三者認証機関が検証を行う場合は、比較的対応負荷や費用（コスト負担）が大きい、自社での検証は対応負荷や費用は小さいが信頼度に課題。	×	認定された検証機関が適合性評価を行い、第三者認証機関が認証を行う方式である。さらに認証機関も検証段階から関わり、検証に立ち会う。
R2	第三者認証機関が検証を行う方式及び、自社で認証が完結する方式よりも、自社や委託先で検証を行ったものを第三者が認証する方式が望まれている。		
R3	検証機関の認定基準は、第三者機関への委託もしくは自社での検証実施が可能となるような、適度なハードルである事が望まれる。	×	検証は、JIS Q 17025 又は ISO/IEC 17025 に基づいて認定された検証機関しか実施できない。

ID	要望事項	要望との合致	比較内容
R4	一定の信頼性を確保するために、当該認証制度に合わせた資格制度は要望が高い。	×	検証は認定された検証機関しか実施できない。検証実施者は基本的な教育経験や実務経験の他、検証機関が認めた教育・訓練プログラムを履修することが求められる。認証制度専用の資格制度があるわけではない。
R5	ソフトウェアコンポーネント一覧やソースコードは機密性が高く、認証機関であっても情報提出のハードルは高い。	○	これらは基本的に検証機関のみに開示され、認証機関は検証機関が作成する検証レポートにより認証を行う。
R6	認証の申請、認証付与のプロセスが明確であり、申請のオンライン化や申請の進捗が把握できるようにすることが望まれている。	△	申請は、申請書や秘密保持契約書などを郵送することで行い、オンライン化には対応していない。またオンラインで進捗が把握できるようなシステムは用意されていない。
R7	取得費用が安くその金額が明確なことや、対応すべきセキュリティ基準が明確で信頼できることが望まれている。	△	認証費用はレベルごとに明確に決まっている。検証機関に支払う検証にかかる費用が別途必要で、その費用は製品によって異なるため検証機関の見積もりによる。また、必要に応じて認証機関のサイト訪問にかかる費用が別途必要。 必要な要件が PP で明確に定義される。PP の信頼度は PP の提供者に依存する。
R8	国や業界団体によって取得が推奨されているなど、認証取得や検証のコストを顧客や消費者に説明しやすく、理解を得やすい制度運用が望まれる。	○	政府調達では、デジタル複合機やファイアウォールなどの製品分野で調達要件になっており取得の必要性が明確である。

【凡例】○：合致する、△：一部合致する、×：合致しない

1.2.4.3.1.2　EDSA 認証制度

■対象

PLC（Programmable Logic Controller）や DCS（Distributed Control System）コントローラといった、産業用制御機器を対象としている。海外では、2019 年に CSA 認証として刷新され、ソフトウェアアプリケーションや、ホストデバイス、ネットワークデバイスも認証対象となった。（ただし、本書では日本における EDSA 認証プロセスについて説明する）

■特徴

ISA 配下の ISCI[30]がスキームオーナとして提供している産業向けの機器、システム、開発プロセスのセキュリティを認証するプログラム：ISASecure 認証制度の中で、機器を対象とするもの。

認定機関である JAB[31]が JIS Q 17065 または ISO/IEC 17065 に基づいて CSSC 認証ラボラトリを認証機関に認定している。さらにスキームによって、認証機関は JIS Q 17025 または ISO/IEC 17025 に基づいて検証機関としての認定を取得することも要求されており、CSSC 認証ラボラトリが検証機関と認証機関を兼ねている。

制御機器のセキュリティにフォーカスしていて、機器のセキュリティ機能及び通信機能の堅牢性を検証する一方で、その機器のソフトウェア開発プロセスも併せて検証される。

■認証要件

３つの要素と３つのレベルで構成される。

通信ロバストネステスト（CRT）　… 全レベルにおいて必須（ファジング重視）：

- テスト対象：IEEE 802.3　（Ethernet）、ARP、IPv4、ICMPv4、UDP、TCP

機能セキュリティアセスメント（FSA）　… セキュリティ機能の実装評価：

- アクセス制御、使用コントロール（デバイス認証、監査証跡）、データ完全性、データ機密性、データフロー制御、イベントへのタイムリーレスポンス、ネットワークリソースの可用性、を評価

ソフトウェア開発セキュリティアセスメント（SDSA）　… 開発プロセスの評価：

- 開発ドキュメント（計画/成果物）とレビュー記録、PDCA プロセスの妥当性と記録確

[30] ISA Security Compliance Institute
[31] 公益財団法人 日本適合性認定協会（Japan Accreditation Board）

認、を評価

※レベル 2 やレベル 3 は FSA と SDSA の評価要求事項が増加する

■認証タイプ

第三者検証による第三者認証

■認証申請の関係機関

・認証機関： CSSC 認証ラボラトリ （CSSC-CL）

・認定機関： 公益財団法人 日本適合性認定協会 （JAB）

・検証機関： CSSC 認証ラボラトリ （CSSC-CL）

■認証申請費用（検証費用含む）

申請料：	350,000 円
FSA/SDSA 審査料：	5,000,000 円
CRT 審査料：	3,000,000 円
認証料：	1,100,000 円
追加審査料：	個別見積もり
審査付帯費用：	250,000 円
認証書等発行料：	30,000 円

■認証申請に必要な文書

申請者→検証兼認証機関：

A）認証・試験申請書

・申請者情報

・申請区分（新規、流用）

・試験区分（CRT 試験、EDSA 認証：レベル１～３）

・希望する CRT 試験デバイス（Achilles、Synopsys Defensics）

・申請対象の製品タイプ（Embedded Device、その他）

・製品情報

・製品の製造業者情報

・確認事項（情報開示の承諾等）

B）別紙 申請対象製品情報シート（申請時に別途提出が必要な内容）

・製品名

・製品型式

・製品バージョン

・評価対象範囲 ※システム構成図で図示
・ハードウェアブロック図
・ソフトウェアブロック図
・アクセス可能なインタフェース
・データフロー図
・脅威分析に基づく保護資産
・脅威分析に基づく脅威シナリオ
・想定するセキュリティ環境（前提環境条件）

■EDSA 認証スキーム（図 3-4）

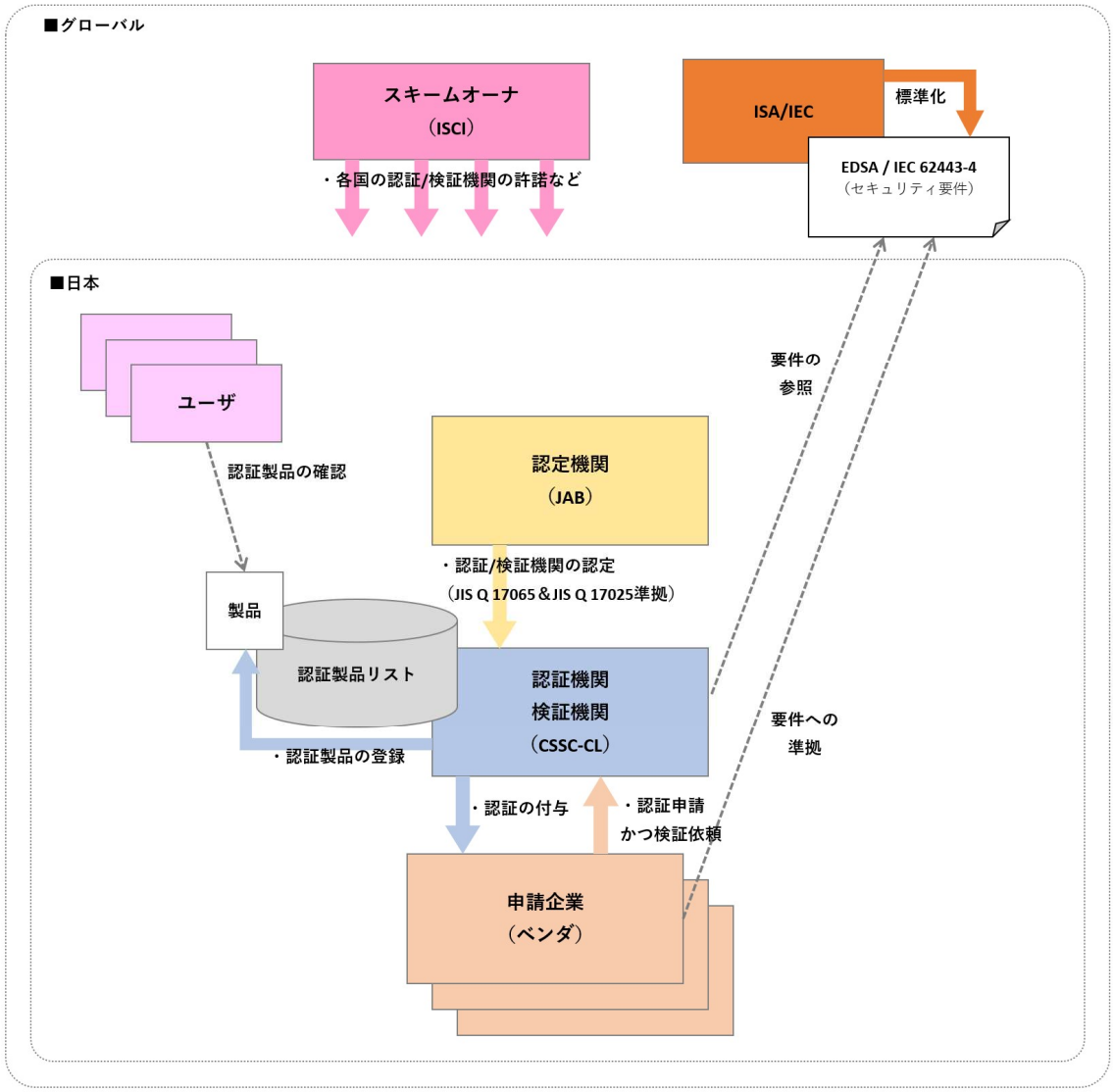
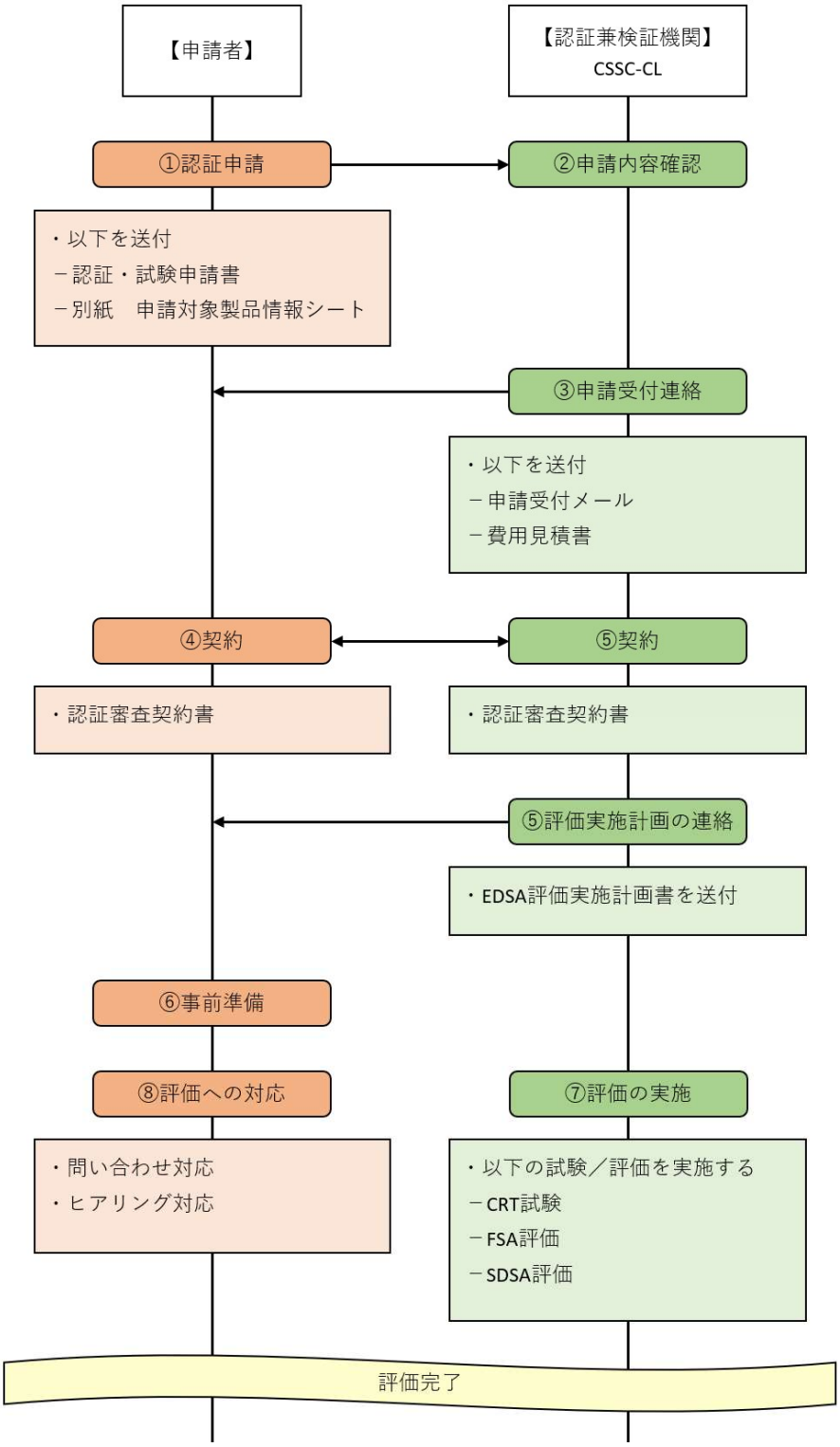


図 3-4　EDSA における認証スキーム

■EDSA 認証申請プロセス（図 3-5）



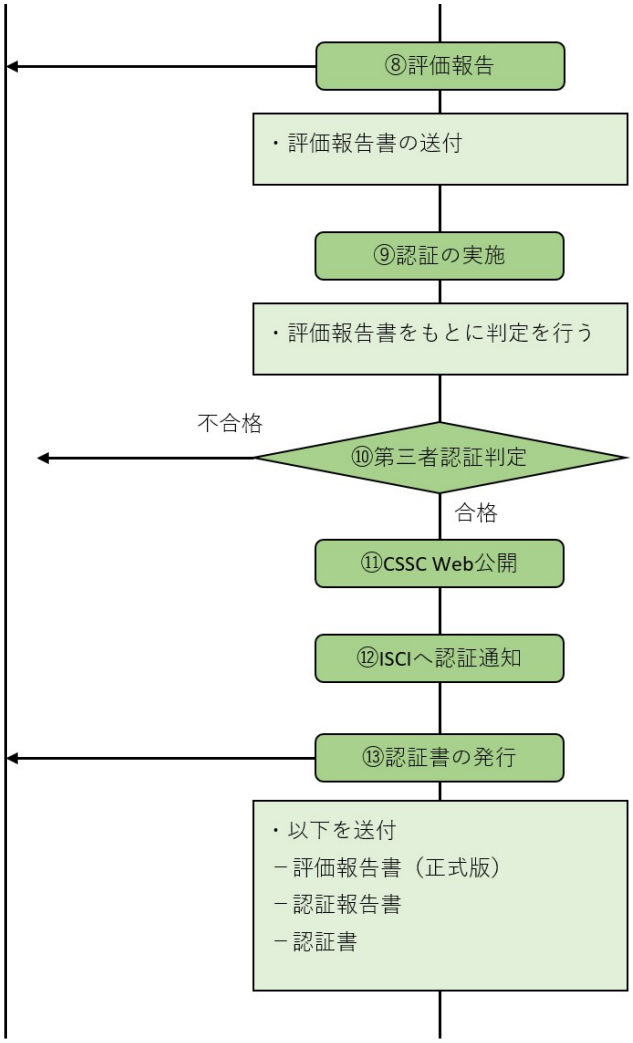


図 3-5　EDSA における認証フロー図

【考察】

産業制御機器向けの制度ということもあり、制度開始の 2014 年からの認証済み製品が最も要求事項が少ないレベル 1 のみで 5 つだけである。当認証のハードルが高いことがうかがえる。

調査結果 1 の要望（表 1-8）と EDSA 認証制度と比較考察した結果を表 3-8 に示す。なお、あくまで本調査における要望との比較であって当認証制度自体の良し悪しの判断ではないことに留意。

表 3-8 要望と EDSA 認証制度との比較

ID	要望事項	要望との合致	比較内容
R1	第三者認証機関が検証を行う場合は、比較的対応負荷や費用（コスト負担）が大きい、自社での検証は対応負荷や費用は小さいが信頼度に課題。	×	EDSA では、認証機関には JIS Q 17065 の認定の他、検証機関に要求される ISO/IEC 17025 に基づく認定の取得も併せて求められている。 検証機関を兼ねる認証機関自身が適合性評価を行う方式であり、自社や委託先で検証を行うことは認められていない。
R2	第三者認証機関が検証を行う方式及び、自社で認証が完結する方式よりも、自社や委託先で検証を行ったものを第三者が認証する方式が望まれている。		
R3	検証機関の認定基準は、第三者機関への委託もしくは自社での検証実施が可能となるような、適度なハードルである事が望まれる。	×	上述の通り、検証機関は同時に認証機関であることも求められる。JIS Q 17065 によって、認証機関は申請者の開発などに関わることができないため、ベンダが自社で検証を実施することは制度上不可能である。
R4	一定の信頼性を確保するために、当該認証制度に合わせた資格制度は要望が高い。	×	委託もしくは自社での検証実施をした場合の要望であるが、EDSA では検証は認定された検証機関しか実施できない
R5	ソフトウェアコンポーネント一覧やソースコードは機密性が高く、認証機関であっても情報提出のハードルは高い。	×	認証機関が検証を行うため、認証機関にソフトウェアコンポーネント一覧やソースコードなどの機密性が高い情報を提供する必要がある。 要望は認証制度の対象に民生品を含んでおり、産業用制御機器の認証制度である EDSA との違いが出ていると考えられる。

ID	要望事項	要望との合致	比較内容
R6	認証の申請、認証付与のプロセスが明確であり、申請のオンライン化や申請の進捗が把握できるようにすることが望まれている。	△	認証の申請については認証機関の Web サイトで提示されている認証手続き規定に触れられているが、具体的な申請方法などについては、別途 Web サイトの「お問い合わせ」にあるメールアドレス宛に連絡して個別に進めることになっている。 また、特にオンラインシステムは用意されていない。
R7	取得費用が安くその金額が明確なことや、対応すべきセキュリティ基準が明確で信頼できることが望まれている。	×	検証費用を一部含んでいるものの、認証申請費用が最低 1,000 万円程度と比較的高額である。 対応すべきセキュリティ基準自体は IEC 規格にもなっていて明確であるものの、当規格に精通していない申請者に対して、JIS Q 17065 の規定によって、認証機関は申請者に対するコンサルテーションを行うことができない。おそらく認証を希望する申請者は別途 EDSA のコンサルテーションを提供している民間の企業にまず相談することになるが、認証機関の Web サイトにはそういった情報がないため、申請者が自身で探すことになると想定される。
R8	国や業界団体によって取得が推奨されているなど、認証取得や検証のコストを顧客や消費者に説明しやすく、理解を得やすい制度運用が望まれる。	△	対象となる制御機器は国内では EDSA 準拠が推奨されている。ただし、国際スキームでは CSA に移行している一方で、日本ではまだ EDSA のままであり立ち遅れている。

【凡例】○：合致する、△：一部合致する、×：合致しない

1.2.4.3.1.3　PCI Secure Software 認証制度

■対象

クレジットカード決済に関連するソフトウェアを対象としている。

■特徴

JISEC や EDSA と異なり、業界団体による独自の認証スキームである。PCI SSC が国際規格の認証制度における認証機関及び認定機関の役割を果たしており、検証機関に当たる企業と人員に対してトレーニングの提供を含めた認定を行っている[32]。

クレジット決済アプリケーションのセキュリティを認証する規格：PA-DSS（Payment Application Data Security Standard）の後継として 2020 年にスタートした。

製品ではなくベンダを対象とした PCI Secure Software Life Cycle Standard program という、セキュアにソフトウェアを開発する体制・プロセスがあるかを保証する認証プログラムも存在する。この認証を取得したベンダに対して、一度 PCI Secure Software 認証を取った製品を機能追加などにより再認証する際の検証を自社で行うことを認めている。

申請者は、検証機関による検証が完了してから認証申請を行う。

PCI PO（Participating Organizations）会員になれば、セキュリティ基準が改訂されるときに正式版がリリースされる前のドラフトを見ることができる。さらにフィードバックを出すことで規格改定に参加することも可能となっている。

■認証要件

前述の PA-DSS と大きく変わった点の一つとして、要件に Objective-Based Approach の考え方が導入された点が挙げられる。具体的な値ではなく達成すべきセキュリティ目標が示されており、基本的にベンダ自身がソフトウェアの性質を考慮して満たすべき具体的な値を決定するようになった。

すべてのタイプのクレジットカード決済関連ソフトウェアを対象とするコア要件のほかに、ソフトウェアの性質に合わせた要件モジュールに追加で対応する。

コア要件：

[32] 検証機関に対しては、コードレビューの経験、暗号技術に関する知識の保有、ペネトレーションテストの経験などが求められる。人員に対しては、業界に認知されている資格の保有に加えて毎年 PCI SSC が提供するトレーニングを受けて試験に合格することなどが求められる。

- アタックサーフェスの最小化
  - 重要資産の特定
  - セキュアなデフォルト設定
  - 機密データの保持
- ソフトウェアの保護機能
  - 重要資産の保護
  - 認証とアクセス制御
  - 機密データの保護
  - 暗号の使用
- セキュアなソフトウェア運用
  - 行動追跡
  - 攻撃検知
- セキュアなソフトウェアライフサイクル管理
  - 脅威及び脆弱性の管理
  - セキュアなソフトウェア更新
  - ソフトウェアのベンダ実装ガイド

モジュール A 要件（アカウントデータを保存、処理、送信するソフトウェアが対象）:

- アカウントデータの保護
  - 機密認証データ
  - カード保有者データの保護

モジュール B 要件（PCI 承認済みデバイス上で実行することが前提のソフトウェアが対象）:

- 端末ソフトウェアのセキュリティ
  - 端末ソフトウェアの文書化
  - 端末ソフトウェアの設計
  - 端末ソフトウェアの攻撃の軽減
  - 端末ソフトウェアのセキュリティテスト
  - 端末ソフトウェアの実装ガイド

■認証タイプ
第三者検証による第三者認証

※ただし特徴に記載した通り、PCI Secure Software Lifecycle Standard に準拠したベンダは、認証済みソフトウェアを更新する際の再検証を自社で行うことが可能。

■認証申請の関係機関（独自スキームのため、ISO 等における定義とは異なる）

・認証機関： PCI SSC（Payment Card Industry Security Standards Council）

・認定機関： PCI SSC（Payment Card Industry Security Standards Council）

・検証機関： Software Security Framework Assessor（PCI SSC が認定した事業者）

■認証申請費用

新規登録： 3,000 US ドル

高影響変更： 1,500 US ドル

低影響変更： 500 US ドル

登録情報変更： 275 US ドル

年次更新： 300 US ドル

※検証費用については別途検証機関の見積もりによる

■認証申請に必要な文書

申請者→認証機関：

A） 同意書（Vendor Release Agreement）

・機密保持や脆弱性対応の義務などベンダが認証プログラムに参加するにあたって同意すべき事項が書かれている

※検証機関経由で認証機関に提出される

申請者→検証機関：

A） 申請書兼誓約書（Attestation of Validation）

・申請者情報

・申請者が Secure SLC プログラムの認証を受けているかどうか

・申請区分（完全評価、年次更新、登録情報変更、低影響変更、高影響変更）

・製品情報（ソフトウェア名、ソフトウェアバージョン、ソフトウェアのタイプ）

・申請対象の製品タイプ（Embedded、その他）

・誓約事項へのチェック

・署名

B） 実装ガイド（Implementation Guidance）

・ソフトウェアをセキュアにインストール、設定するためのユーザマニュアル

C） 開発証拠資料（以下一例）

・ソフトウェア設計書（以下のような、要件に書かれている機能が実現されていることが

分かるもの）

－扱うセンシティブデータおよびその保護方法一覧

－提供するセンシティブ機能およびその保護方法一覧

－データフロー図

－デフォルトで提供する API やインタフェースの一覧とその正当性を示すもの

－ソフトウェアのセキュリティ機能がデフォルトで有効になっていること

－デフォルトの資格情報、暗号鍵、電子証明書などの一覧

－ソフトウェアが要求する権限が必要最小限であること

－組み込みアカウントの権限が制限されていること

－認証方法

－暗号機能および鍵管理

－トレース機能

－攻撃検知機能

など

・脅威分析に基づく脅威シナリオおよび緩和策

・脆弱性対応について

・ソフトウェア更新について

検証機関→認証機関：

A） 検証レポート（Reports on Validation）

・各要件に対して検証機関が判断した内容、理由などが書かれる

■PCI Secure Software 認証スキーム（図 3-6）

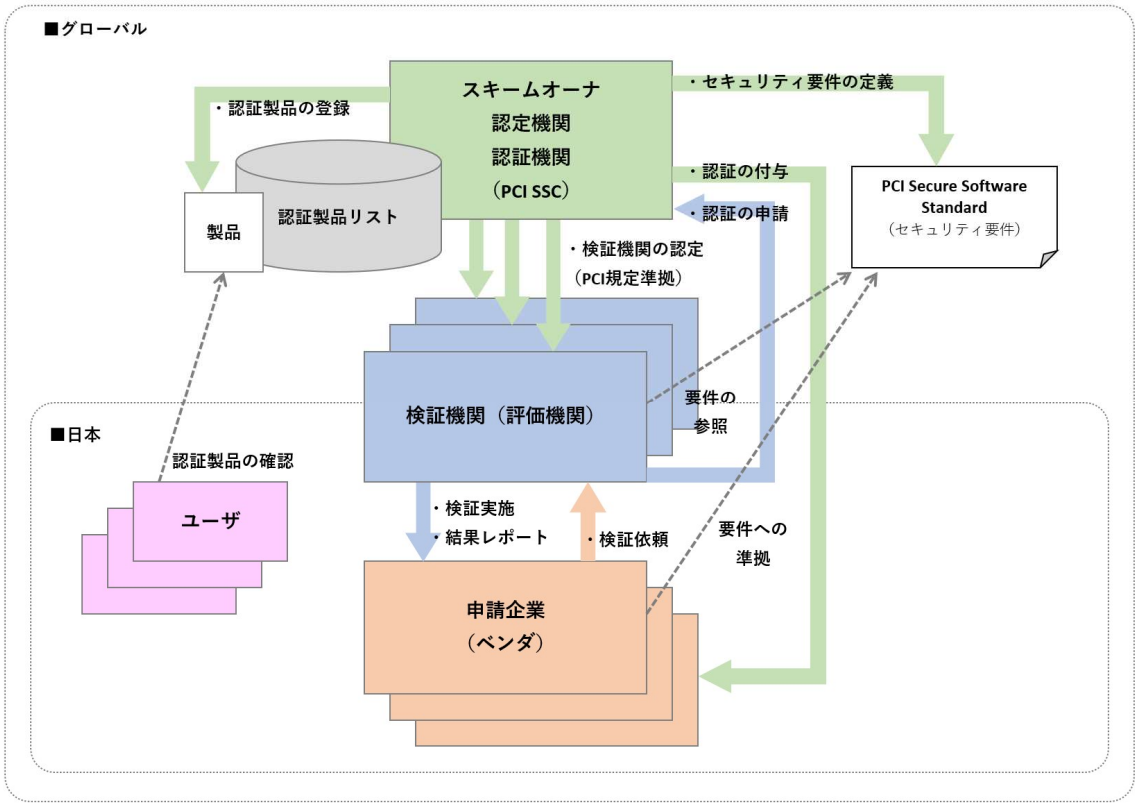
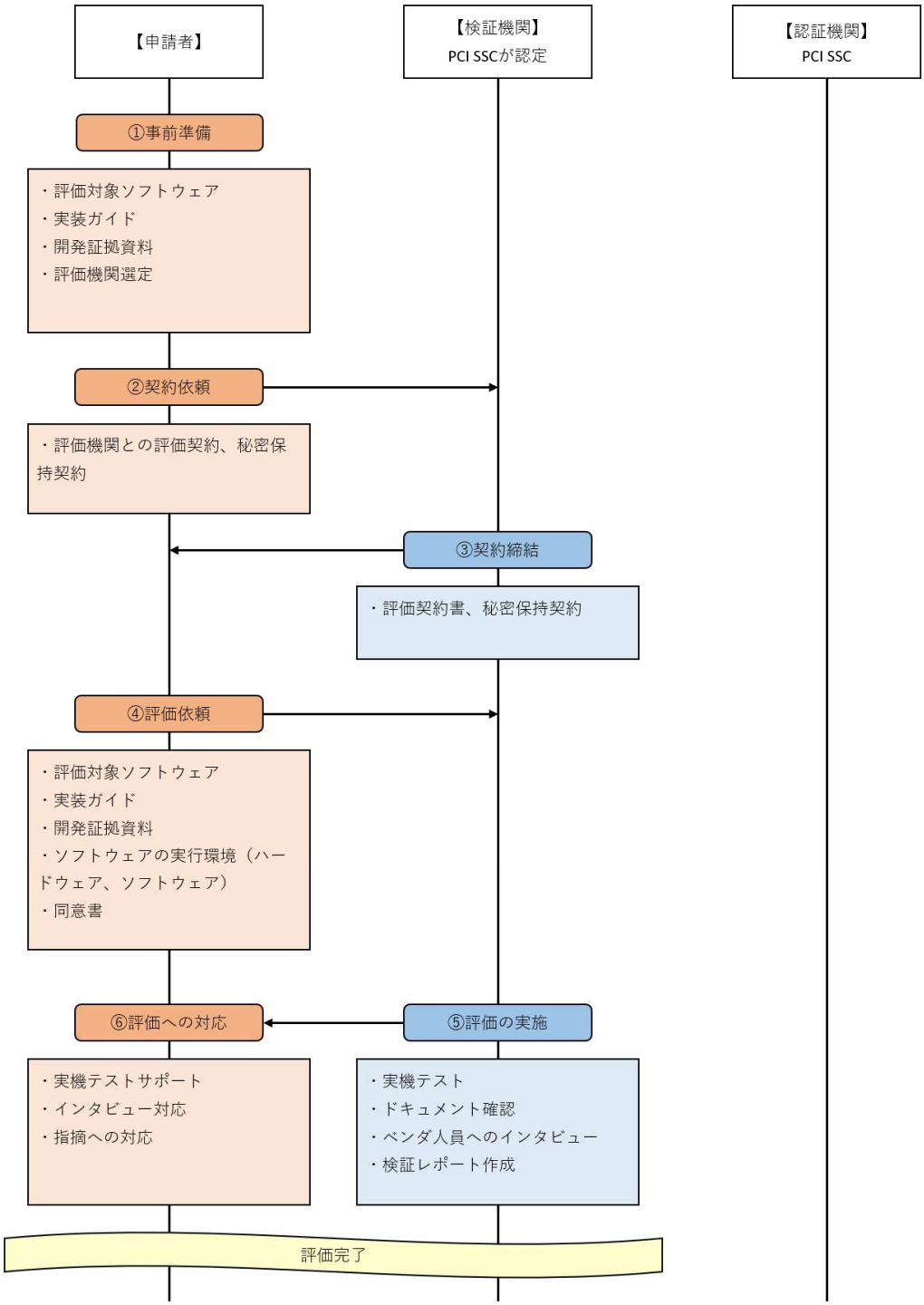


図 3-6　PCI Secure Software における認証スキーム

■PCI Secure Software 認証申請プロセス（図 3-7）



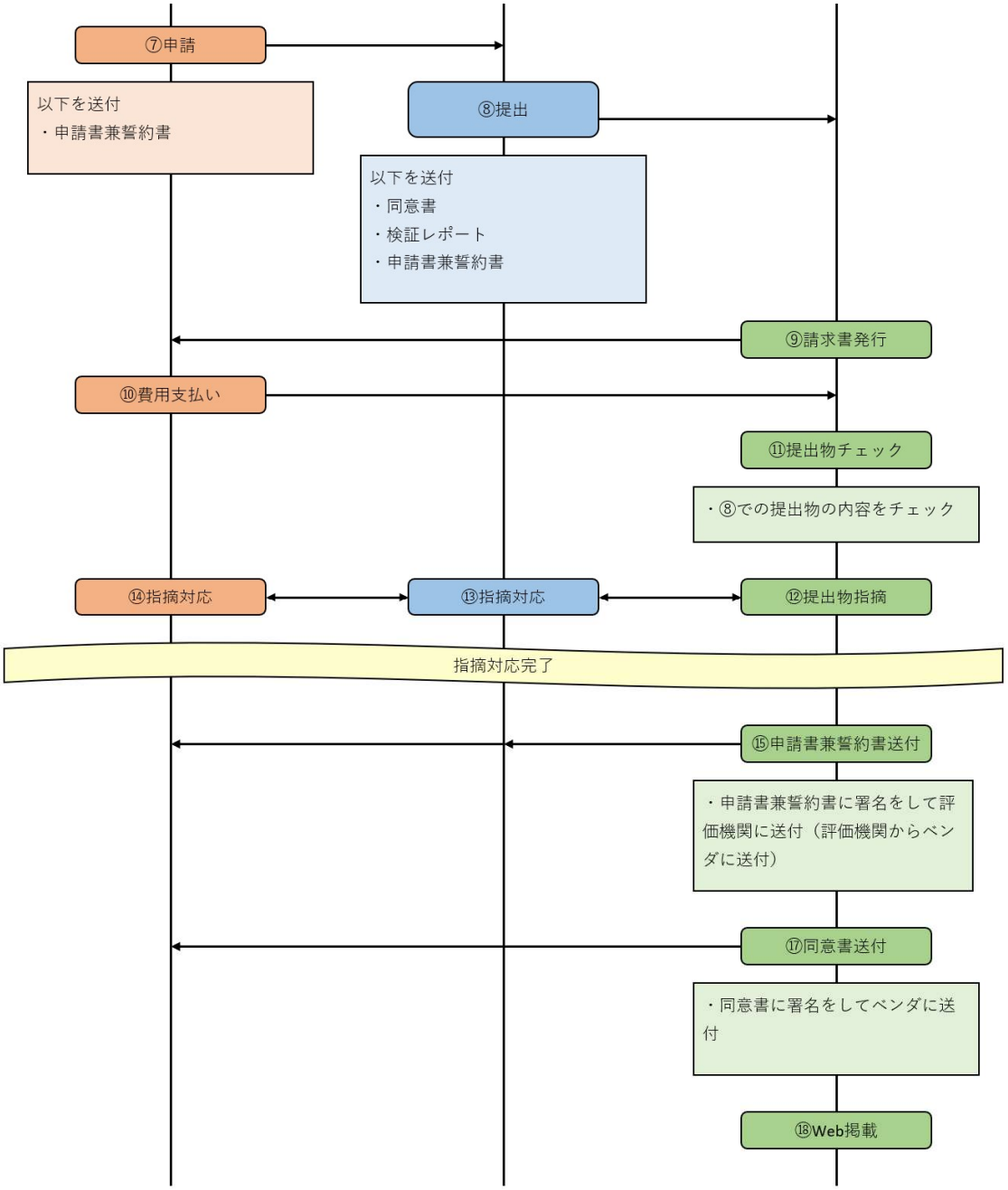


図 3-7　PCI Secure Software における認証フロー図

■考察

2022 年 6 月の時点で認定製品は 12 件だが、2020 年から始まった規格であること、及び、前身の規格：PA-DSS による認証を 2021 年まで受け付けていたことが要因。PA-DSS での認定製品は現在認証が有効なもの（過去に認証されて現在失効しているものを除く）だけで 751 件あることから、事実上の認定実績は多い。

調査結果 1 の要望（表 1-8）と PCI Secure Software 認証制度と比較考察した結果を表 3-9 に示す。なお、あくまで本調査における要望との比較であって当認証制度自体の良し悪しの判断ではないことに留意。

表 3-9 要望と PCI Secure Software 制度との比較

ID	要望事項	要望との合致	比較内容
R1	第三者認証機関が検証を行う場合は、比較的対応負荷や費用（コスト負担）が大きい、自社での検証は対応負荷や費用は小さいが信頼度に課題。	△	認証機関を兼ねている PCI SSC が認定した検証機関が検証を行う。ベンダ認定プログラムを併せて提供し、ベンダ自身でも一部検証（認証済み製品に対して、機能追加などの更新の際の再検証）を行うことを認めている。
R2	第三者認証機関が検証を行う方式及び、自社で認証が完結する方式よりも、自社や委託先で検証を行ったものを第三者が認証する方式が望まれている。		
R3	検証機関の認定基準は、第三者機関への委託もしくは自社での検証実施が可能となるような、適度なハードルである事が望まれる。	△	検証機関に対しては、ISO などの国際規格への準拠は求められないが、セキュリティ関連として、コードレビューの経験や暗号技術に関する知識の保有、ペネトレーションテストの経験などが求められる。 また、検証を担当する人員に対しては、業界に認知されている資格の保有に加えて、毎年 PCI SSC が提供するトレーニングを受けて試験に合格することなどが求められる。 ベンダ認定プログラムではベンダが規格に準拠していることを認証してもらう必要がある。ベンダの要員に対しては特に資格を要求していない。 ※セキュリティを主とする企業に属する、セキュリティを主たる業務とする要員が検証を実施することを前提としているレベルと思われる。

ID	要望事項	要望との合致	比較内容
R4	一定の信頼性を確保するために、当該認証制度に合わせた資格制度は要望が高い。	△	PCI SSC が検証機関の人員向けにトレーニングと試験を提供している。ただし、業界のセキュリティ資格の保有も必要。 ベンダ認定プログラムではベンダが規格に準拠していることを認証してもらう必要がある。ベンダの要員に対しては特に資格を要求していない。
R5	ソフトウェアコンポーネント一覧やソースコードは機密性が高く、認証機関であっても情報提出のハードルは高い。	○	これらは検証機関のみに開示され、認証機関は検証機関が作成する検証レポートと顧客に提示するマニュアル類により認証を行う。
R6	認証の申請、認証付与のプロセスが明確であり、申請のオンライン化や申請の進捗が把握できるようにすることが望まれている。	△	申請者は、まず認証機関ではなく検証機関にアクセスする。認証機関とのやり取り（申請を含む）は、認証費用の支払いなどを除き、基本的に検証機関経由で行われる。 検証機関は PCI SSC とポータルサイトでやり取りする。申請者が申請状況をオンラインでの把握できるようにはなっていない。

ID	要望事項	要望との合致	比較内容
R7	取得費用が安くその金額が明確なことや、対応すべきセキュリティ基準が明確で信頼できることが望まれている。	△	認証費用は明確に決まっており、初回申請が 3,000 US ドルと比較的安価。検証機関に支払う検証にかかる費用が別途必要で、その費用はソフトウェアによって異なるため検証機関の見積もりによる。 対応すべきセキュリティ基準自体は認証機関の Web サイトに公開されていて明確であるが、当規格に精通していない申請者は検証機関にコンサルテーションを依頼することも可能（条件はあるものの、検証依頼しようとしている検証機関も可能）。認証機関の Web サイトに検証機関とその窓口の一覧が掲載されている。
R8	国や業界団体によって取得が推奨されているなど、認証取得や検証のコストを顧客や消費者に説明しやすく、理解を得やすい制度運用が望まれる。	○	PCI Secure Software 認証の上位スキームである PCI DSS 認証について、日本では割賦販売法[33]で、クレジットカードを扱う小売店やカード情報を保持する事業者に対して準拠が求められている。その PCI DSS 対応において、PCI Secure Software 認証取得済みのソフトウェアを適切に導入している場合、その部分についてのセキュアなソフトウェア開発に関する要件は充足しているとみなされる。このため、顧客へのコストの説明だけではなく、顧客の PCI DSS 対応コストの軽減に寄与できる面もある制度となっている。

【凡例】○：合致する、△：一部合致する、×：合致しない

[33] ２か月を超えて支払う信用取引（クレジットカードの分割払いなど）などについて規定している法律。クレジットカード取引に関わるセキュリティに関しても規定が追加され 2018 年に施行された。
経済産業省，割賦販売法の一部を改正する法律について, https：//www.meti.go.jp/policy/economy/consumer/credit/kappuhannbaihounoichibuwokaiseisuruhouritsu.pdf

1.2.4.3.2　認証制度の調査結果の整理

本項におけるまとめとして、ソフトウェアを対象とした認証制度の実現において、ヒアリング結果による要望事項と各認証制度の状況について、比較結果の一覧を示す。

表 3-10 要望と各制度との比較一覧

ID	要望事項	要望との合致			要望事項との比較考察
		JISEC	EDSA	PCI	
R1	第三者認証機関が検証を行う場合は、比較的対応負荷や費用（コスト負担）が大きい、自社での検証は対応負荷や費用は小さいが信頼度に課題。	×	×	△	いずれの制度でも自社や委託先で検証が完結する方式は認めていない。 PCI Secure Software のみベンダ認証プログラムを別途設けている。このプログラムでは、個々の人員には特定の資格の取得・維持を求めていない一方、経営層を含めた組織的な対応が求められ、これが適度なハードルと感じるかどうかは会社の規模や体制に依存すると思われる。
R2	第三者認証機関が検証を行う方式及び、自社で認証が完結する方式よりも、自社や委託先で検証を行ったものを第三者が認証する方式が望まれている。				
R3	検証機関の認定基準は、第三者機関への委託もしくは自社での検証実施が可能となるような、適度なハードルである事が望まれる。	×	×	△	
R4	一定の信頼性を確保するために、当該認証制度に合わせた資格制度は要望が高い。	×	×	△	PCI Secure Software のみ、検証機関の検証要員向けではあるが専用の資格制度を設けている。JISEC と EDSA では専用の資格制度は特にない。
R5	ソフトウェアコンポーネント一覧やソースコードは機密性が高く、認証機関であっても情報提出のハードルは高い。	○	×	○	EDSA では制度上、認証機関が検証機関を兼ねるため、認証機関に対して機密性の高い情報を開示する必要がある。

ID	要望事項	要望との合致			要望事項との比較考察
		JISEC	EDSA	PCI	
R6	認証の申請、認証付与のプロセスが明確であり、申請のオンライン化や申請の進捗が把握できるようにすることが望まれている。	△	△	△	いずれの制度でも、申請のオンライン化や、オンラインで申請の進捗状況を申請者に示すようなシステムは用意されていない。
R7	取得費用が安くその金額が明確なことや、対応すべきセキュリティ基準が明確で信頼できることが望まれている。	△	×	△	取得費用に関して、EDSA では検証費用を一部含んでいるものの、認証費用が最低 1,000 万円程度と比較的高額である。その他の認証制度でも検証費用は別途見積もりによるため、取得費用について安いとまでは言い切れない。 セキュリティ基準に関して、JISEC では認証対象の製品のセキュリティ機能が十分かつ信頼できるかは、適合させる PP に依存するため、PP の作成者に依存する。
R8	国や業界団体によって取得が推奨されているなど、認証取得や検証のコストを顧客や消費者に説明しやすく、理解を得やすい制度運用が望まれる。	○	△	○	いずれの制度においても国や業界団体によって取得が推奨あるいは必須になっており、認知度も高い。 EDSA については、グローバルでは CSA 認証に移行しており、制度が立ち遅れている。

【凡例】○：合致する、△：一部合致する、×：合致しない

1.2.4.4　実現可能かつ実効的な認証制度や検証機関のプロセス・ルール提言

前項までの調査結果を踏まえ、本項では実現可能かつ実行的な認証機関及び検証機関のプロセスやルールについての提言を示す。

1.2.4.4.1　認証制度の開始までに必要とされる段階

認証制度の運用開始までには、対象のセキュリティ要件の定義に加え、認証の適合判定基準の定義や、認定機関（第三者認証を採用する規格において審査機関を認定する組織）の決定や、認証機関の決定、認証スキームの定義（及び検証機関の要求事項定義）、検証機関の決定といった一連の段階が必要となる。（注：上記は、国際標準規格に基づく第三者認証を実施するケースであり、民間による認証制度では独自に規定されたスキームやプロセス、要求事項によって運営されるケースもある）。参考情報として、図 3-8 に IoT 製品・サービスを対象とした認証制度の整備の現状を示す。なお、図 3-8 には民間主導による認証制度の例として、重要生活機器連携セキュリティ協議会（CCDS）[34]、及び ioXt Alliance[35]が推進する制度を記載している。

[34] CCDS：一般社団法人 重要生活機器連携セキュリティ協議会
https：//www.ccds.or.jp/
[35] ioXt Alliance
https：//ja.ioxtalliance.org/

認証国	規格・制度・法令	認証基準の策定				認証スキーム整備								制度開始
		セキュリティ要件の定義	⇒	認証の適合判定基準の定義	⇒	認定機関の決定	⇒	認証機関の決定	⇒	認証スキームの定義 ※検証機関選定の要求事項定義	⇒	検証機関の決定	⇒	
国際標準規格	ISO/IEC15408（CC）	Protection Profile（PP） ※対象機器ごとに定義	⇒	CEM（Common Criteria for Information Technology Security Evaluation）	⇒	各認証国で定義 ※CC相互承認アレンジメント（CCRA）にて、相互承認	⇒		⇒	ISO/IEC 17065：2012	⇒	各認証国で決定 ※ISO/IEC 17025：2017に準拠	⇒	○
	ISO/IEC 27400 （ISO/IEC 27402）	ISO/IEC 27400 DIS ISO/IEC 27402 CD ※SC27でCommittee Draft段階	⇒	未対応	⇒	未対応	⇒	未対応	⇒	未対応	⇒	未対応	⇒	×
米国	IoT Cybersecurity Improvement Act of 2020	NIST "Recommended Criteria for Cybersecurity Labeling for Consumer Internet of Things (IoT) Products"	⇒	未対応 ※通常であれば米国規格協会（ANSI）にて定義	⇒	未対応	⇒	未対応	⇒	未対応	⇒	未対応	⇒	×
欧州	Cybersecurity Act	ETSI EN 303 645	⇒	TS 103 701	⇒	未対応	⇒	未対応	⇒	未対応	⇒	未対応 ※要件の一部は欧州の「RED委任規制2022/30」として先行義務化	⇒	×
米国 （デファクト基準）	ioXt Certification	「ioXt Base Profile」他、製品種別ごとに異なる計5つのProfileを定義。	⇒	「ioXt Base Profile」他、製品種別ごとに異なる計5つのProfileを定義。実施すべきテストケースもProfileに記載されている。	⇒	ioXt Alliance	⇒	ioXt Alliance	⇒	※不明（詳細情報は登録メンバーに情報開示）	⇒	ioXt Allianceが認定した事業者	⇒	○
日本 （デファクト基準）	CCDSサーティフィケーションプログラム	「IoT機器セキュリティ要件ガイドライン2021年版」	⇒	「CCDS_IoT機器セキュリティ要件_検査ガイドライン2021年版」	⇒	なし	⇒	CCDS	⇒	「CCDSサーティフィケーションプログラム規程」	⇒	CCDS指定検査資格者が所属する法人	⇒	○

図 3-8　IoT 製品・サービスを対象とした認証制度の現状

1.2.4.4.2　認証制度や検証機関のプロセス・ルール提言

次に本書の調査項目１によるヒアリング結果（要望事項）及び、調査項目３の認証制度の調査結果を踏まえ、ソフトウェア（OSS）を対象とした認証制度について提言を示す。

提言の前提として、調査結果１の要望を踏まえた認証制度のポイントを以下に示す。

■要望を踏まえた認証制度のポイント（表 1-8 より抜粋）

R1～R2：自社又は委託先で適合性評価を行い、第三者認証機関が認証する方式。

R3：検証機関の認定を資格者の在籍等、一定の信頼性を得られる制度とする。

R4：認証制度に対応する資格制度の検討。

R5：認証機関への情報提出は、認証に関連する設計文書やセキュリティ検証結果やエビデンスまでが容認可能。

R6：認証の申請、認証付与のプロセスの分かりやすさ及び、情報開示が必要。またオンライン手続き等を活用した進捗状況の透明性の確保。

R7：認証を取得するための費用が製品単価に照らして現実的であり、費用が明確に提示されている。対応すべきセキュリティの基準が明確であり、信頼できる。

R8：資格取得が国や業界団体に推奨され、顧客や消費者の理解を得られやすい制度の検討

1.2.4.4.2.1　認証スキーム

・認証スキームは「A）認定された検証機関が検証を行う制度（図 3-9)」と「B）自社での検証を可能とする制度（図 3-10)」の 2 パターンを提言する。

・A、B どちらのパターンも、認証機関がスキームオーナとなり、認証スキーム（要求事項、適合基準を含む）を定義する。認証機関は要求事項への準拠が可能な組織であれば、複数の組織による認証業務の運営も可能とする。

・検証機関は、A と B の認証スキームで、それぞれ認定方法や検証機関が異なる。

パターン A では、ISO 規格に準拠した検証機関を認定機関が認定し、検証は認定された検証機関のみが実施可能となる。

パターン B では、認証に対応する資格制度を認証機関が整備し、必要資格を備えた要員を有する組織であれば、自社検証、第三者検証のいずれも実施可能とする。

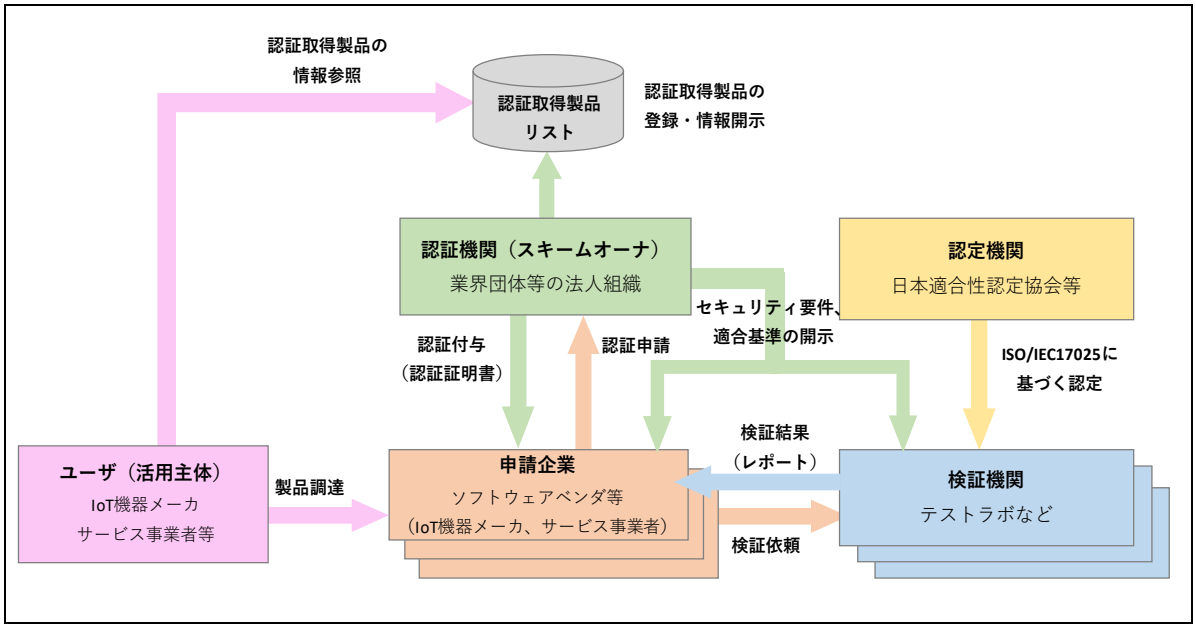


図 3-9　認定制度のスキーム：A.認定された検証機関が検証を行う制度

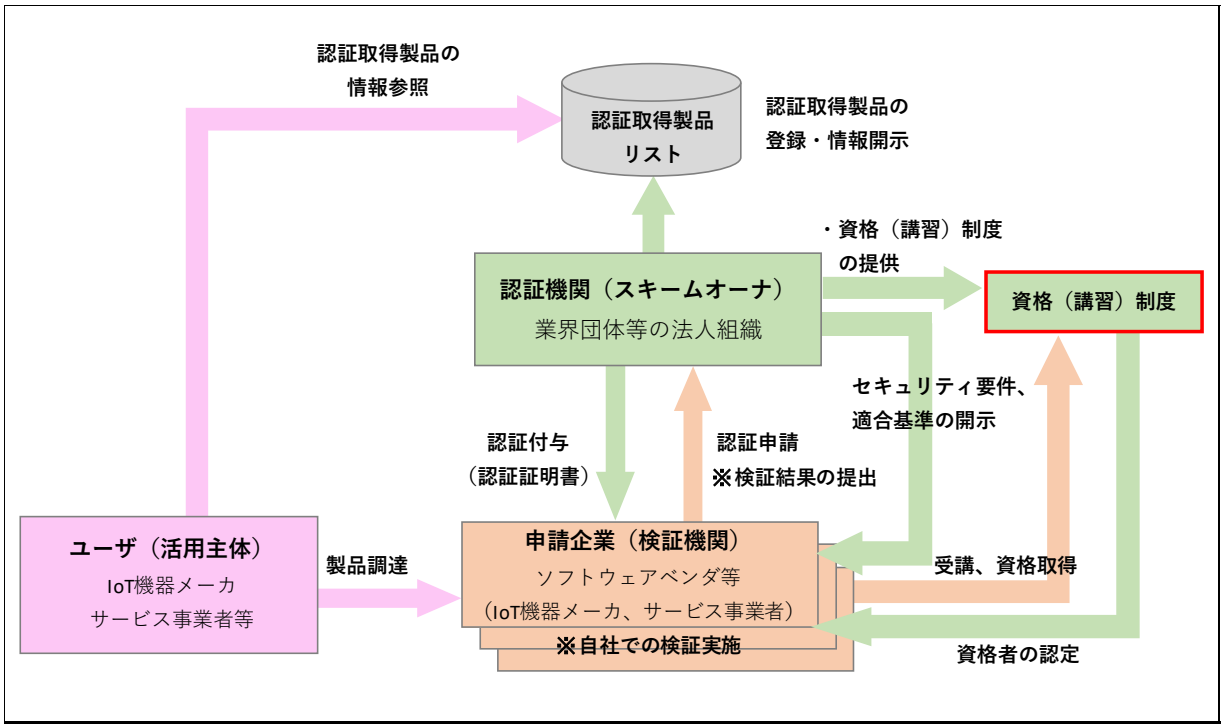


図 3-10　認定制度のスキーム図：B.自社での検証を可能とする制度

1.2.4.4.2.2　関係機関への要求事項

１）認証機関に求められる要求事項

①公平性の担保、機密保持の徹底

・認証機関は公平性を担保し、認証の対象製品やサービスの事業に直接的な関与を行わないこと。

・対象となる製品、サービスを明確にし、対象範囲であれば認証申請者が公平に制度を利用可能であること。

・認証の申請者とは機密保持契約を締結し、取得した情報は機密事項として取り扱うこと。

・認証業務の一部を外部委託する場合は、認証機関と外部委託先で機密保持契約を締結すること。

②事業運営が可能な安定した財務基盤や債務状況

・認証機関は、運営に必要な経営資源をもち、生じる債務への備えをもつこと。

③資源（知識や力量を持つ要員、施設や環境、設備等）の確保

・認証制度を運用する上で、必要な人数、知識、力量（必要資格）を備えた要員を有すること。

・認証業務の一部を外部委託する場合、外部委託先の要員が、必要な知識、力量を備えていること。

④一連の認証プロセスの管理及び、記録の継続的な保持

・認証機関は、認証のプロセスやスキームを規程文書として整備し、規程に準拠した運営を行うこと。

・製品を検証するための要求事項や適合基準を文書として定義すること。

⑤マネジメントシステムの保持

・ISO9001 か ISO/IEC17065、もしくは認証スキームに定義された事項に基づきマネジメントシステムを保持すること。

２）検証機関に求められる要求事項

①公平性の担保、機密保持の徹底

・検証機関は公平性に責任を持ち、利害関係者による影響を受けない運営を行うこと。

・検証機関は、検証の過程で取得した情報の管理について、顧客（申請企業）との機密保持契約を締結すること。

②組織構成や管理構造の明確化、マネジメントシステムの実施や維持、改善

・検証機関の組織や管理構造及び、検証業務に関する要員の権限、責任範囲を明確化すること。

・検証業務に伴うマネジメントシステムの実施、維持、改善のプロセスを有すること。

③資源（要員、検証の施設、環境、設備）の確保、外部提供製品、サービスの管理

・検証の実施おいて要求される知識、力量（必要資格）を備えた要員を有すること。

・要員の教育や管理の仕組みを整備し、記録を保持すること。（要員に求められる資格取得に向けた教育の仕組み、取得実績の管理を含む）
・検証機関は、検証の過程で取得した情報の管理について、顧客（申請企業）との機密保持契約を締結すること。
・検証機関は、適切な検証を実施するための設備が利用可能であること。
・検証機関が検証業務の外部委託を行う場合は、検証の実施おいて要求される知識、力量（必要資格）を備えた要員を有する委託先を選定すること。

④一連の検証プロセスの管理及び、記録の継続的な保持（レビュー、検査、設備校正等）

・検証結果は、顧客（申請企業）へ報告する前に、レビューされ、承認されるプロセスを有すること。
・検証に利用するツールの最適化（バージョン、脆弱性データの更新）及び、機器の校正を行うプロセスが整備され、記録の管理を行うこと。
・検証機関は、検証結果や関連する証票の情報管理を行い、利用可能とすること。
・認証機関より、検証結果の妥当性に関する調査要請があった場合には、対応する仕組みを有すること。

⑤マネジメントシステムの保持

・ISO9001 か ISO/IEC17025、もしくは認証スキームに定義された事項に基づきマネジメントシステムを保持すること。

1.2.4.4.2.3　セキュリティ要件（要求事項）・適合基準の定義

・認証の適合に求められるセキュリティ要件は、認証機関が整備し、認証対象の製品やサービスの事業に直接関与をしない委員を含む第三者委員会による、レビューと承認を行う。
・承認されたセキュリティ要件については、認証制度の情報とあわせて情報開示を行う。
・認証機関は認証の適合基準として、セキュリティ要件に対応したセキュリティ検証の方法や手順、判定基準を整備し、検証機関や認証申請者に情報開示を行う。
・セキュリティ要件及び適合基準については、国内外のセキュリティや脅威の情勢を調査し、適宜更新が可能な仕組みとする。

1.2.4.4.2.4　検証機関の認定

・検証機関の認定については、前述した 2 パターンの認証スキーム（「A.認定された検証機関が検証を行う制度（図 3-9）」と「B.自社での検証を可能とする制度（図 3-10）」）に応じて、それぞれ方法が異なる。A は ISO 規格に準拠し、組織として検証機関を認定する方法であり、B はヒアリングによる要望事項を踏まえ、認証に対応した資格制度（講習制度）により、資格者を認定する方法となる。それぞれのケースについて、以下に詳細を示す。

A.認定された検証機関が検証を行う制度：検証機関を組織として認定

・上記「検証機関に求められる要求事項」を満たし、ISO/IEC17025 に準拠した検証事業者を認定機関が認定する。

表 3-11　A）検証機関を組織として認定する場合のメリット、デメリット

メリット	デメリット
企業の認定による、検証業務、検証結果の信頼性、客観性が向上する。	組織として要求事項及び ISO 規格に準じた対応が必要となり、検証機関としては対応コストが増加する。
－	検証機関の認定基準のハードルがあがり、申請企業による自社での検証（自己適合性検証）が実現しにくい。

B.自社での検証を可能とする制度：資格制度により要員を認定

・認証機関が対応する資格制度（講習制度）を整備し、認証に必要な力量（知識や検証能力）を備えた資格者を認定する。（ポイント：R4）

・認定を受けた資格者が所属する法人であれば、自社製品（自己適合性検証）、他社製品（第三者検証）のいずれも対応可能とする。（ポイント：R1～R3）

※資格制度の運営、教育

・資格制度のスキーム構築や運営、資格者の認定は認証機関が担い、認定した資格者の情報を管理する。

・また資格制度に対応した検証の方法や適合基準、申請方法などの講習制度についても認証機関（あるいは認証機関が認定する事業者）が運営を行う。

表 3-12　資格制度により、要員を認定する場合のメリット、デメリット

メリット	デメリット
検証機関の組織としての対応コストが低減できる。また自社で検証を行うことで、開発部門に対する課題のフィードバックが効率的に実施できる。	検証機関を認定する基準がないため、検証機関に対する信頼性や客観性を証明できない。
検証機関の認定基準のハードルが下がり、多くの企業が事業に参入可能となる。また、申請企業による自社での検証（自己適合性検証）が実現しやすい。※要望を踏まえた認証制度のポイント「R1～R3」に対応。	－

A～B のメリット、デメリットを踏まえた提言

・本調査の IoT 機器メーカを対象としたヒアリング結果では、コスト負担における課題から、自社による検証実施の仕組みや、資格制度が望まれている。また多くの IoT 機器メーカからは、ソフトウェアの調達基準として活用可能な認証制度を期待されている。この場合、認証取得の主体はソフトウェアベンダとなるが、ソフトウェアベンダにとっても、同様のコスト負担への課題から、自社での検証の仕組みに対するニーズが高いことが想定される。

1.2.4.5　認証及び検証のプロセス・ルール

１）認証プロセス・ルール

・認証の申請から付与までのプロセスは、複雑化せず、明確なフローを検討する。
（要望を踏まえた認証制度のポイント：R6）

・提言する認証制度のフロー案を図 3-11 に示す。

[申請]

・認証の概要やプロセス、申請手順について、認証機関がスキームを定義し、認証を取得するために必要な情報を公開する。

・認証機関は、認証申請に必要な文書（申請文書）について、機密情報や申請者による提供可能範囲を配慮し、定義を行う。（要望を踏まえた認証制度のポイント：R5）。また、申請文書について認証機関の審査業務、検証機関の検証業務に分類し、それぞれの対応に必要な文書を定義する。

・認証に掛かる費用については、認証機関が製品やサービスの販売単価に応じて、適切な金額を算定し、情報を公開する。（要望を踏まえた認証制度のポイント：R7）

[認証付与の決定]

・認証機関は、検証結果と検証レビュー結果に基づき、認証決定を行う。

・認証の決定は、検証業務に従事していない要員が最終判断を行う。

・認証付与が不可である場合、その理由を含めて認証申請者に通知を行う。

[認証結果文書]

・認証機関は、以下の事項を含む文書を認証申請者に提出し、認証付与の履歴を管理する。
　－認証機関、認証申請者の名称や住所、認証授与の日付、認証の範囲（レベル）、
　　有効期限

[結果通知]

・認証の結果については、認証機関から申請者へ結果を通知する。また、結果が不適合と

なった場合には、不適合箇所と理由を明確化し、申請者へ通知を行う。

[製品・サービスの登録簿]

・認証機関は、認証が付与された製品やサービスについて、以下の情報を管理する。
　－認証された製品やサービスの情報、基準となる規格や規準文書、依頼者の情報
・認証取得製品やサービスの情報については、認証事業者が管理を行い、申請企業と合意した範囲で情報の開示を行う。

[認証プロセス・スキームの変更]

・認証プロセスやスキームに変更があった場合、認証機関は変更に関係する認証申請者に変更連絡を行う。
・変更を加えた規程等の文書について、継続的な管理を行う。

[認証の終了や一時停止、取消し]

・追跡調査により、要求事項への不適合が確認された場合、認証機関は規程に定められた対処を行う。
・認証の終了、一時停止、取消しの場合、認証結果文書や、公開情報、適合マークの付与権限など関連する情報の修正を行う（認証の復帰が可能となった場合にも、同様の修正を行う）

[記録の情報管理と保持]

・認証機関（または外部委託先）は認証の付与した製品やサービスに関する文書（検証やレビュー結果、認証結果など）を機密事項として管理し、規程に定められた期間、保持する。

[苦情及び異議申立て]

・認証機関は、苦情及び異議申立てについて、受領や妥当性の調査、対応決定のプロセスを文書として整備する。

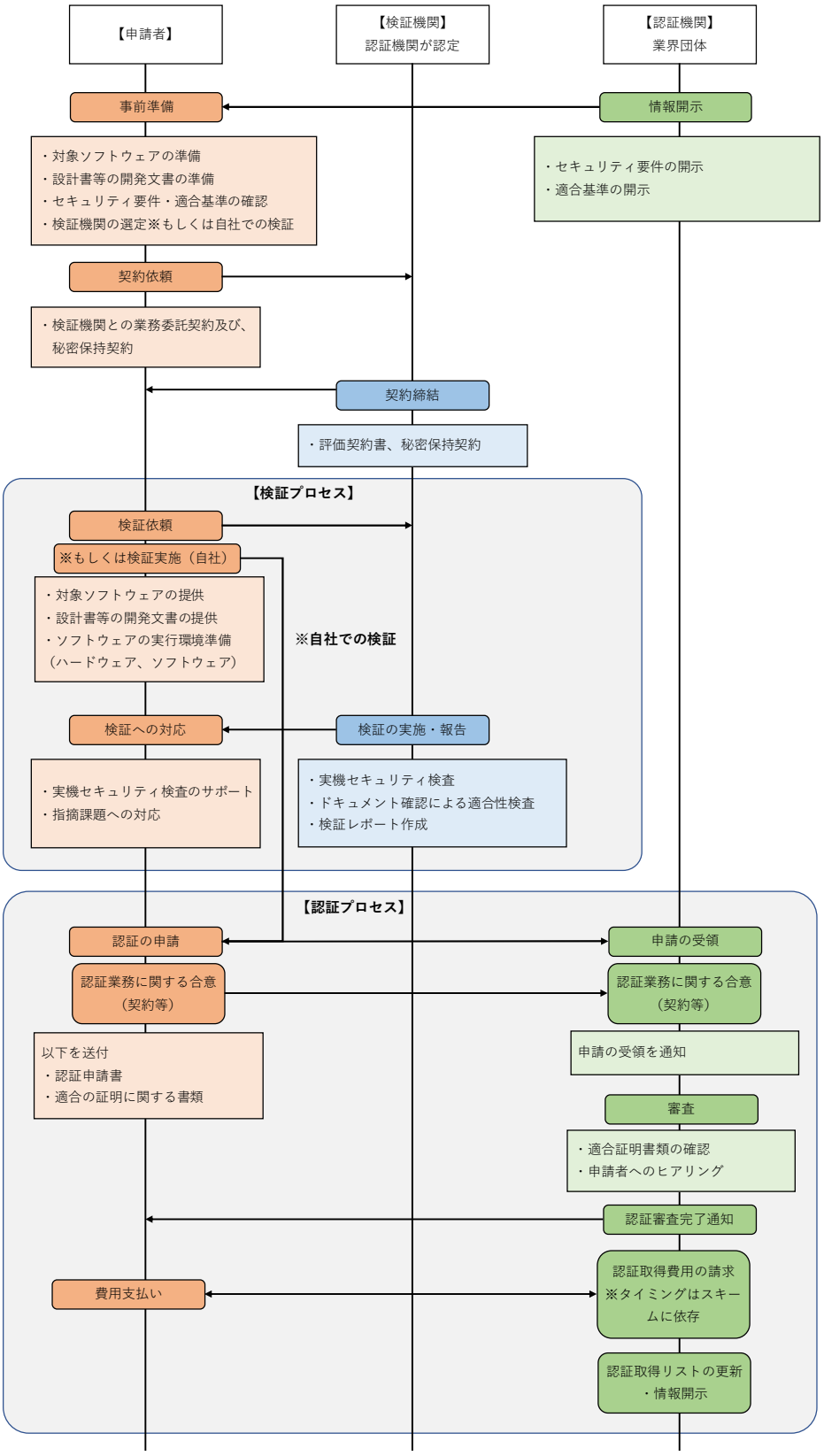


図 3-11　認証制度のフロー案（申請～付与）

2）検証プロセス・ルール

[検証業務に関する契約のレビュー]

・検証機関は、顧客の依頼、見積及び契約条件についてレビューの仕組みを有する。
・顧客の要求や契約内容が、認証の対象や要求事項と齟齬がないことを確認する。

[検証方法の選定、妥当性確認]

・認証の適合性に関するセキュリティ検証の方法、手順、判定基準については、認証機関がセキュリティ要件と共に定義し、検証機関へ情報提供を行う。
・検証の方法、手順、判定基準に関する文書は、最新の状態を維持し、要員がいつでも利用可能とする。

[検証ツール・校正対象機器の取扱い]

・セキュリティ検査に利用可能な検証ツールについては、認証機関が適合基準と共に定義する。また、検証ツールの利用条件（ソフトウェアのバージョン、脆弱性データの更新など）についても、定義を行う。
・検査に利用されるツール（ツール）は、文書で定められた更新手順に準拠し、記録を保持すること。（ソフトウェアバージョン、脆弱性データの更新日など）
・校正の対象となる機器（通信モニター機器）を使用する場合は、文書に校正手順を定め、校正の記録を保持する。

[検証結果の記録]

・検証機関が実施した検証結果は、曖昧性を極力排除し、実行時に近い条件で実行可能な形式で記録を行う。
・また、検証結果の妥当性を追跡するために必要な情報（検証時のログ等のエビデンス）を記録として保持する。

[検証結果の報告]

・検証結果は、顧客へ報告する前に、レビューされ、承認されるプロセスを有する。
・顧客、製品が識別可能であり、検証実施者（責任者）の氏名や実施内容及び、明確な結果判定（判定結果に考察が入る場合はその根拠）を報告書に含める。

[苦情の受領と対応]

・検証機関は苦情を受領し、その内容の妥当性評価及び、対応の決定を行うためのプロセスを有する。

[不適合の場合の対応]

・検証機関は、検証結果が、要求事項や基準に適合しない場合の対応手順を有する。
・不適合の場合の対応結果を、記録として保持する。

[検証結果の情報管理と保持]

・検証機関は、検証結果や関連する証票の情報を機密事項として管理し、利用可能とする。また管理対象となる情報は、認証期間により指定された期間、保持する。

[認証機関からの調査要請対応]

・認証機関より、検証結果の妥当性に関する調査要請があった場合には、対応する仕組みを有する。

1.2.4.6　認証制度の普及啓発

・認証機関はセキュリティ対策の必要性や認証制度の利点について、製品やサービスの利用者に対する普及啓発活動を行う。※行政機関に対するアウトリーチ支援活動を含む。
（要望を踏まえた認証制度のポイント：R8）。

1.2.4.7　参考情報）民間主導による認証制度の対応状況

参考情報として、民間主導による認証制度（CCDS サーティフィケーション、ioXt Certification）について、ヒアリングにより提示された要望事項（1.2.2.3.4 項）及び、認証機関への要求事項（1.2.4.4.2.2 項）に対する対応状況を表 3-13 として示す。

表 3-13　民間主導による認証制度の要望・要求事項対応状況

<table>
<tr><th rowspan="2">種別</th><th rowspan="2">No.</th><th rowspan="2">ヒアリング要望事項・
要求事項</th><th colspan="2">認証制度の対応状況</th></tr>
<tr><th>CCDS
サーティフィケーション</th><th>ioXt Certification</th></tr>
<tr><td rowspan="4">ヒアリング要望事項</td><td>R1</td><td>第三者認証機関が検証を行う場合は、比較的対応負荷や費用（コスト負担）が大きい、自社での検証は対応負荷や費用は小さいが信頼度に課題</td><td rowspan="2">○

検証は自社でも可能とし、認証は第三者機関である CCDS が書面審査を行い、認証する。</td><td rowspan="2">○

ioXt Alliance が認定するラボでの検証及び、自社での検証のどちらも対応が可能。認証は ioXt Alliance が検証結果を審査し、認証する。</td></tr>
<tr><td>R2</td><td>第三者認証機関が検証を行う方式及び、自社で認証が完結する方式よりも、自社や委託先で検証を行ったものを第三者が認証する方式が望まれている。</td></tr>
<tr><td>R3</td><td>検証機関の認定基準は、第三者機関への委託もしくは自社での検証実施が可能となるような、適度なハードルである事が望まれる。</td><td>○

認証制度に対応した資格制度、講習制度により、資格者を認定する。</td><td>×

ioXt Alliance による認定ラボは 7 社（アジア圏では 1 社）のみであり、比較的認定のハードルは高い。</td></tr>
<tr><td>R4</td><td>一定の信頼性を確保するために、当該認証制度に合わせた資格制度は要望が高い</td><td>○

認証制度に対応した資格制度、講習制度を整備。</td><td>×

認定のための講習や資格制度ははない。（ラボを認定する制度となる）</td></tr>
</table>

	R5	ソフトウェアコンポーネント一覧やソースコードは機密性が高く、認証機関であっても情報提出のハードルは高い。	○ 検証結果や検証エビデンスを中心に審査を行う。	○ テスト結果を中心とした審査が行われる。
	R6	認証の申請、認証付与のプロセスが明確であり、申請のオンライン化や申請の進捗が把握できるようにすることが望まれている。	△ 認証申請から付与までのプロセスは、ウェブにより情報開示。オンラインによる申請は現状未対応。	△ 認証申請から付与までのプロセスは、アライアンス会員限定で情報を開示。オンライン申請については不明。
	R7	取得費用が安くその金額が明確なことや、対応すべきセキュリティ基準が明確で信頼できることが望まれている。	○ 取得費用は製品・サービスの販売価格に合わせて変動するが、比較的安価。セキュリティ要件は諮問委員会による審議を経ており、信頼性は高い。また承認されたセキュリティ要件はウェブにより情報公開されている。	△ 登録費用は比較的安価ではあるが、費用の情報はアライアンス登録メンバーにのみ開示される。セキュリティ基準は独自の Profile を定義しており、情報はウェブにより公開されている。
	R8	国や業界団体によって取得が推奨されているなど、認証取得や検証のコストを顧客や消費者に説明しやすく、理解を得やすい制度運用が望まれる。	○ 国や業界団体のセキュリティガイドラインと整合性を持った要件を定義している。また認証取得企業への保険制度により、コストの説明はしやすい制度となっている。	× 国内においては制度の認知度がまだ低く、国や業界団体による推奨もされていない。
認証機関への要求	①	公平性の担保、機密保持の徹底	○ 認証対象の事業とは中立の団体による制度運営。認証	— ※情報非開示につき詳細不明。

			申請書において、申請者と機密保持を合意。	
	②	事業運営が可能な安定した財務基盤や債務状況	○ 多数の幹事、正会員企業を有する。	― ※情報非開示につき詳細不明。
	③	資源（知識や力量を持つ要員、施設や環境、設備等）の確保	○ 組織内でセキュリティ要件や適合基準の定義が可能な要員を有する。	― ※情報非開示につき詳細不明。
	④	一連の認証プロセスの管理及び、記録の継続的な保持	○ CCDS 及び、検証機関、申請企業に一定期間、記録の保持を求めている。	― ※情報非開示につき詳細不明。
	⑤	マネジメントシステムの保持	△ 組織内で独自のマネジメントシステムを整備しているが、ISO9001 もしくは ISO/IEC 17065 には準拠していない。	― ※情報非開示につき詳細不明。

２．研究発表・講演、文献、特許等の状況

（１）研究発表・講演

なし

（２）論文

なし

（３）特許等（知財）

なし

（４）受賞実績

なし

（５）成果普及の努力（プレス発表等）

なし

契約管理番号：	21501786-0