経済産業省　御中

令和４年度 サイバー・フィジカル・セキュリティ対策促進事業
SBOM を導入・活用するサプライチェーン
モデルの構築に向けた調査・実証事業
報告書

2023 年 3 月

MRI 三菱総合研究所

目 次

1. 背景と目的 1

2. 産業活動においてソフトウェアを安全に利活用するための枠組みに関する取組動向等の調査 3

2.1. 米国政府機関における取組動向等の調査 3

2.1.1. CISA：Vulnerability Exploitability eXchange(VEX) – Use Cases 4

2.1.2. NIST：Software Security in Supply Chains / NIST SP 800-161 Rev.1 9

2.1.3. DHS：Silicon Valley Innovation Program 10

2.1.4. CISA：Vulnerability Exploitability eXchange (VEX) - Status Justifications 12

2.1.5. CISA：SBOM Listening Sessions 14

2.1.6. OMB：Enhancing the Security of the Software Supply Chain through Secure Software Development Practices 17

2.1.7. S.4913 - Securing Open Source Software Act of 2022 19

2.1.8. CISA/NSA/ODNI：Securing Software Supply Chain Series - Recommended Practices for Developers 20

2.1.9. CISA/NSA/ODNI：Securing Software Supply Chain Series - Recommended Practices for Suppliers 21

2.1.10. CISA/NSA/ODNI：Securing Software Supply Chain Series - Recommended Practices for Customers 22

2.1.11. White House：Natinonal Cybersecurity Strategy 24

2.1.12. Food and Drug Administration（FDA）における動向調査 25

2.2. 欧州政府機関における取組動向等の調査 27

2.2.1. EC：Proposal for Cyber Resilience Act 28

2.2.2. 英国 NCSC：Mapping your supply chain 35

2.3. 企業、業界団体等におけるソフトウェア管理の取組動向等の調査 35

2.4. OSS を含むソフトウェアの安全な利活用に関する調査 39

2.4.1. ソフトウェアに関連するインシデント・脆弱性事例の調査 39

2.4.2. 脆弱性管理手法ツールに関する調査 45

2.4.2.1. Security Content Automation Protocol（SCAP） 45

2.4.2.2. SCAP の取組例 46

2.4.2.2.1. National Vulnerability Database（NVD） 46

2.4.2.2.2. JVN iPedia 46

2.4.2.2.3. OpenSCAP 46

2.4.2.3. SBOM・VEX・SCAP において検討される関係性 48

2.5. CISA 会合の要点整理 49

3. OSS の利活用等に関する事例集の英訳 53

4. SBOM を導入や活用するサプライチェーンモデルの構築に向けた調査・実証 54

4.1. 目的 54

4.2. 基本方針 54

4.3. SBOM 実証全体の計画 55
4.3.1. SBOM 実証に関する要件 56
4.3.2. 実証対象分野と実証事業者の選定 57
4.3.3. 分野ごとの前提条件・制約条件等 59
4.3.3.1. 法制度及び推奨事項 59
4.3.3.1.1. 医療機器分野 59
4.3.3.1.2. 自動車分野 61
4.3.3.1.3. ソフトウェア分野 61
4.3.3.2. 業界の取引形態・開発形態 62
4.3.4. SBOM 実証の設計 66
4.3.5. SBOM 実証の実施スケジュール 71
4.4. SBOM 対応モデル 72
4.4.1. SBOM 対応モデルの必要性 ～自律的な市場メカニズムの形成～ 72
4.4.2. SBOM 対応モデルのフレームワーク 75
4.4.2.1. SBOM 対応モデルの基本的な考え方と活用方法 75
4.4.2.2. SBOM 対応項目の区分と選択肢 77
4.4.2.3. SBOM 対応モデルの記述方法 79
4.5. 医療機器分野 81
4.5.1. 実証の狙い・重点ポイント 81
4.5.2. 実証の手順 81
4.5.3. 実証の対象システムの概要と構成 81
4.5.4. 対象分野の前提条件・取引慣行等 83
4.4.5. 実証項目の特定 87
4.5.5. 実証結果 88
4.5.5.1. SBOM 初期導入 88
4.5.5.1.1. SBOM 管理ツールの選定基準 88
4.5.5.1.2. 実施内容とコスト評価 90
4.5.5.1.3. 課題、解決ノウハウ、留意点 93
4.5.5.2. SBOM の生成・共有 96
4.5.5.2.1. 実施内容とコスト評価 96
4.5.5.2.2. 開発委託先との SBOM 共有方法 103
4.5.5.2.3. 医療機関への SBOM の提供方法 104
4.5.5.2.4. 課題、解決ノウハウ、留意点 105
4.5.5.3. SBOM の活用・管理 108
4.5.5.3.1. 実施内容とコスト評価 108
4.5.5.3.2. 脆弱性マネジメントフロー 116
4.5.5.3.3. 脆弱性マネジメント工数の想定 117
4.5.5.3.4. 脆弱性マネジメント工数の結果 122
4.5.5.3.5. 脆弱性マネジメントに関する分析 130
4.5.5.3.6. 医療機器セキュリティポリシーにおけるサポート計画 132
4.5.5.3.7. サポート計画の分析 136
4.5.5.3.8. 医療機関へのヒアリング 137
4.5.5.3.9. 課題、解決ノウハウ、留意点 139

4.5.5.4. 効果評価 145
4.5.5.5. SBOM 適用範囲（SBOM 活用モデル）の改訂案 147
4.5.5.6. 取引契約における要求条項等 156
4.6. 自動車分野 157
4.6.1. 実証の狙い・重点ポイント 157
4.6.2. 実証の手順 157
4.6.3. 実証の対象システムの概要と構成 158
4.6.3.1. 対象システム 1：オープン系非リアルタイム OS ベースシステム 158
4.6.3.2. 対象システム 2：制御系システム（一般的なヒータコントローラの例） 159
4.6.3.3. システム構成 1（オープン系非リアルタイム OS ベースシステム）とシステム構成 2（制御系システム）のソフト開発の違いについて 159
4.6.3.4. 実証の進め方について 160
4.6.4. 対象分野の前提条件・取引慣行等 160
4.6.5. 実証項目の特定 161
4.6.6. 実証結果 162
4.6.6.1. SBOM 初期導入 162
4.6.6.1.1. 実施内容とコスト評価 162
4.6.6.1.2. 課題、解決ノウハウ、留意点 169
4.6.6.2. SBOM の生成・共有 169
4.6.6.2.1. 実施内容とコスト評価 169
4.6.6.2.2. SBOM 作成工数 174
4.6.6.2.3. 精査工数 175
4.6.6.2.4. バイナリスキャンによる検出 176
4.6.6.2.5. SPDX ファイル作成工数 176
4.6.6.2.6. 米大統領令最小要素と SPDX,及び SPDX-Lite ファイルの比較 176
4.6.6.2.7. SBOM 共有工数 177
4.6.6.2.8. 課題、解決ノウハウ、留意点 178
4.6.6.3. SBOM の活用・管理 181
4.6.6.3.1. 実施内容とコスト評価 181
4.6.6.3.2. 事前に全て精査済みの SBOM を元に深刻度評価を行う場合と、脆弱性検知後に SBOM を精査し深刻度評価をする場合の比較 188
4.6.6.3.3. 脆弱性監視工数の比較 190
4.6.6.3.4. 週 1 回などの定期監視と常時監視の対応期間の比較 192
4.6.6.3.5. 課題、解決ノウハウ、留意点 192
4.6.6.4. 効果評価 192
4.6.6.4.1. 工数低減効果 195
4.6.6.4.2. SBOM の精査 198
4.6.6.4.3. 部品の包含関係を含む全ての粒度の SBOM を保持することが有効か 198
4.6.6.4.4. 脆弱性対応期間の短縮効果について 198
4.6.6.4.5. SBOM 活用の嬉しさについて 199
4.6.6.5. SBOM 適用範囲（SBOM 活用モデル）の改訂案 199
4.6.6.5.1. SBOM 適用範囲の変更箇所 206
4.6.6.5.2. 変更理由 206
4.6.6.6. 取引契約における要求条項等 206

4.6.6.6.1. 開発中の作業について 206
4.6.6.6.2. 開発終了後の作業について 207
4.6.6.6.3. SBOM データの共有について 207
4.6.6.6.4. 今後の課題について 207
4.7. ソフトウェア分野（トレンドマイクロ） 208
4.7.1. 実証の狙い・重点ポイント 208
4.7.2. 実証の手順 208
4.7.3. 実証の対象システムの概要と構成 209
4.7.4. 対象分野の前提条件・取引慣行等 211
4.7.5. 実証項目の特定 213
4.7.5.1. SBOM 初期導入 213
4.7.5.2. SBOM 生成 213
4.7.5.3. SBOM のユーザへの共有 214
4.7.5.4. SBOM 活用・管理 215
4.7.6. 実証結果 217
4.7.6.1. SBOM 初期導入 217
4.7.6.1.1. 実施内容とコスト評価 217
4.7.6.1.2. 課題、解決ノウハウ、留意点 220
4.7.6.2. SBOM の生成・共有 221
4.7.6.2.1. 実施内容とコスト評価 221
4.7.6.2.1. 課題、解決ノウハウ、留意点 232
4.7.6.3. SBOM の活用・管理 239
4.7.6.3.1. 実施内容とコスト評価 239
4.7.6.3.1. 課題、解決ノウハウ、留意点 243
4.7.6.4. 効果評価 246
4.7.6.5. SBOM 適用範囲（SBOM 活用モデル）の改訂案 250
4.7.6.6. 取引契約における要求条項等 252
4.8. ソフトウェア分野（コラボスタイル） 257
4.8.1. 実証の狙い・重点ポイント 257
4.8.2. 実証の手順 257
4.8.3. 実証の対象システムの概要と構成 258
4.8.4. 対象分野の前提条件・取引慣行等 260
4.8.5. 実証項目の特定 261
4.8.6. 実証結果 264
4.8.6.1. SBOM 初期導入 264
4.8.6.1.1. 実施内容とコスト評価 264
4.8.6.1.2. 課題、解決ノウハウ、留意点 272
4.8.6.2. SBOM の生成・共有 274
4.8.6.2.1. 実施内容とコスト評価 274
4.8.6.2.2. 課題、解決ノウハウ、留意点 283
4.8.6.3. SBOM の活用・管理 286
4.8.6.3.1. 実施内容とコスト評価 286
4.8.6.3.2. 課題、解決ノウハウ、留意点 292

4.8.6.4. 効果評価 292
4.8.6.5. SBOM 適用範囲（SBOM 活用モデル）の改訂案 298
4.8.6.6. 取引契約における要求条項等 308
4.8.7. 実証成果と課題の整理 309
4.8.7.1. 検討すべき課題 309
4.8.7.2. 今後の取組の方向性整理 311
4.9. ソフトウェア分野（さくらインターネット） 313
4.9.1. 実証の狙い・重点ポイント 313
4.9.2. 実証の手順 313
4.9.3. 実証の対象システムの概要と構成 314
4.9.4. 対象分野の前提条件・取引慣行等 315
4.9.5. 実証項目の特定 316
4.9.6. 実証結果 317
4.9.6.1. SBOM 初期導入 317
4.9.6.1.1. 実施内容とコスト評価 320
4.9.6.1.2. 課題、解決ノウハウ、留意点 320
4.9.6.2. SBOM の生成・共有 321
4.9.6.2.1. 実施内容とコスト評価 321
4.9.6.2.2. 課題、解決ノウハウ、留意点 325
4.9.6.3. SBOM の活用・管理 327
4.9.6.3.1. 実施内容とコスト評価 327
4.9.6.3.2. 課題、解決ノウハウ、留意点 330
4.9.6.4. 効果評価 332
4.9.6.5. SBOM 適用範囲（SBOM 活用モデル）の改訂案 335
4.9.6.5.1. 大口顧客とのコミュニケーションコスト削減 340
4.9.6.6. 取引契約における要求条項等 340
4.9.6.6.1. SBOM の悪用に関する懸念と期待 340
4.9.6.6.2. フォーマットに関する期待 340
4.9.6.6.3. メーカーの対応が消極的だったことについて 341
4.10. 実証成果と課題の整理 342
4.10.1. 検討すべき課題 343
4.10.2. SBOM 対応モデル（案）の比較 344
4.10.3. SBOM 取引モデル 347
4.10.3.1. SBOM 取引モデルの意義と位置付け（全体像） 347
4.10.3.2. SBOM 取引モデル ～取引契約に規定すべき事項（案）～ 347
5. ソフトウェアの利活用に係るセキュリティリスク、課題及び対応策の検討 349
5.1. ソフトウェア利活用に係るセキュリティリスク・課題 349
5.1.1. ソフトウェア利活用にかかる法的リスク 349
5.1.1.1. ライセンス訴訟事例 351
5.1.1.2. ソフトウェア利活用におけるライセンスリスクの課題 353
6. ソフトウェアタスクフォースの運営 355
6.1. タスクフォースの実施概要 355

6.1.1. 第 7 回ソフトウェアタスクフォース 355
6.1.2. 第 8 回ソフトウェアタスクフォース 355
6.1.3. 第 9 回ソフトウェアタスクフォース 355
6.2. 実施スケジュール 355
6.3. タスクフォースの実施結果の要点整理 357
6.3.1. 第 7 回ソフトウェアタスクフォースの意見概要 357
6.3.2. 第 8 回ソフトウェアタスクフォースの意見概要 357
6.3.3. 第 9 回ソフトウェアタスクフォースの意見概要 357
7. 今後の課題と取組案のまとめ 358

図 目 次

図 2-1　VEX を導入することのメリット例 5
図 2-2　VEX ドキュメントに含めるべき最小要素 6
図 2-3　CSAF フォーマットの 3 つのプロパティ 7
図 2-4　VEX ドキュメントの構造（3 つのプロパティの概要とプロパティの関係）..... 8
図 2-5　CSAF 形式による VEX ドキュメントの例 9
図 2-6　大統領令に対する NIST のこれまでの取組と C-SCRM との対応関係 10
図 2-7　ソフトウェアサプライチェーンのセキュリティ確保のための新たな概念 10
図 2-8　SVIP で募集対象となったツールに求められる要件 11
図 2-9　VEX で「脆弱性の影響を受けない」ステータスを正当化するため 5 つの主張 13
図 2-10　VEX で「脆弱性の影響を受けない」ステータスを正当化するための主張を判断するためのフローチャート 14
図 2-11　MyJVN バージョンチェッカのイメージ図 46
図 2-12　OpenSCAP の概要 47
図 2-13　OpenSCAP に用いられている XCCDF の記述例 47
図 2-14　SBOM・VEX・SCAP において検討される関係性のイメージ図 48
図 4-1　SBOM 実証全体の計画 55
図 4-2　SBOM に関する課題と取組策（2021 年度結果） 56
図 4-3　医療機器の市販前のセキュリティに関わる規制の概要 60
図 4-4　医療機器の市販後のセキュリティに関わる規制の概要 60
図 4-5　自動車分野の SBOM に係る法制度の関係整理 61
図 4-6　ソフトウェア分野の SBOM に係る法制度の関係整理 62
図 4-7　実証の実施スケジュール 71
図 4-8　SBOM 対応範囲の可視化を通じた市場メカニズムの形成 73
図 4-9　SBOM 対応モデルと関連する要素の関係 75
図 4-10　SBOM 対応範囲の可視化の考え方 76
図 4-11　システム構成図 82
図 4-12　脆弱性マネジメントフロー 116
図 4-13　製品ライフサイクルにおけるレガシー医療機器の概念フレームワーク 133
図 4-14　システム構成 1：オープン系非リアルタイム OS ベースシステム 158
図 4-15　システム構成 2：制御系システム 159
図 4-16　パッケージマネージャによる検出部品とツールによる検出部品の関係 178
図 4-17　SBOM 精査まとめ 180
図 4-18　TMSL の基本機能 210
図 4-19　TMSL の構成と SBOM 作成範囲 210
図 4-20　チームによる脆弱性調査フローの違い 215
図 4-21　SBOM 活用に関するヒアリング結果 229
図 4-22　SBOM ファイルの提供形式 229
図 4-23　angularjs のコンポーネント構成と検出結果 233
図 4-24　SCA ツール上のプロジェクト階層構造 235

図 4-25　プロジェクト構成変更後の SBOM（抜粋）.....236
図 4-26　SBOM に出力される SCA ツール上の設定情報例.....238
図 4-27　脆弱性調査手法と作業内容.....241
図 4-28　対象システムの構成と SBOM 生成範囲.....259
図 4-29　SBOM 実証 対象システム構成(さくらインターネット株式会社).....315
図 4-30　サプライチェーンにおける委託契約に基づく SBOM の普及促進.....347
図 7-1　実証・調査を通じて抽出された主な課題と対応の方向性（再掲）.....358
図 7-2　SBOM 管理・脆弱性管理に関する課題の分類.....359
図 7-3　脆弱性の共通管理方式、脆弱性マッチング、脅威情報活用策のイメージ....359

表 目 次

表 2-1　米国政府機関における取組動向等の調査対象.....3
表 2-2　SVIP のツール開発において各フェーズに設定された成果物・活動目標の概要 11
表 2-3　CISA SBOM Listening Sessions における主な議論内容・意見等.....14
表 2-4　CISA SBOM Workstream における主な焦点・活動状況.....15
表 2-5　OMB 覚書発行日以降の各機関・OMB・CISA に対する要求事項 18
表 2-6　「Securing Open Source Software Act of 2022（法案）」の概要 19
表 2-7　ソフトウェア開発者に対する推奨事項の概要.....20
表 2-8　ソフトウェアサプライヤーに対する推奨事項の概要.....22
表 2-9　ソフトウェア消費者に対する推奨事項の概要.....23
表 2-10　国家サイバーセキュリティ戦略における 5 つの柱と戦略目標の概要.....24
表 2-11　FDA に承認申請を行う医療機器メーカーに対する推奨事項.....26
表 2-12　欧州政府機関における取組動向等の調査対象.....28
表 2-13　EU サイバーレジリエンス法（草案）における要求事項一覧 29
表 2-14　企業等におけるソフトウェア管理の取組動向等の調査結果.....36
表 2-15　ソフトウェアに関連するインシデント・脆弱性事例.....39
表 2-16　SCAP における 6 つの標準仕様 45
表 2-17　OpenSCAP の具体的なツール.....47
表 4-1　実証対象分野・対象事業者等の概要 57
表 4-2　分野ごとの取引形態等 63
表 4-3　分野ごとの開発形態等 64
表 4-4　実証対象分野における SBOM の作成・活用の制約条件 65
表 4-5　SBOM 実証におけるプロセス区分と実証要求事項（案）の整理 67
表 4-6　SBOM 対応項目選択肢（最終案） 78
表 4-7　各フェーズの実証内容 81
表 4-8　医療機器の一般的な取引慣行 83
表 4-9　今回の実証形態での取引慣行 84
表 4-10　医療機器分野の規制要求事項 84
表 4-11　SBOM 実証における実施項目 87
表 4-12　有償ツール 89
表 4-13　日立 SOL での MEND SCA のサポート内容及び価格.....89
表 4-14　無償ツール 89
表 4-15　MEND SCA 初期導入・設定・習熟工数計測.....90
表 4-16　Syft&Grype 初期導入・設定・習熟工数計測.....91
表 4-17　Daggerboard 初期導入・設定・習熟工数計測.....91
表 4-18　各ツールの初期導入・設定・習熟コスト比較.....92
表 4-19　MEND SCA の初期導入・設定・習熟の課題.....93
表 4-20　Syft&Grype の初期導入・設定・習熟の課題.....94
表 4-21　Daggerboard の初期導入・設定・習熟の課題.....94
表 4-22　各ツールの SBOM 生成に要した工数とその評価 96

表 4-23　各ツールの IMDRF 追補ガイダンス案 8 要素の確認工数.....97
表 4-24　サードパーティ製 SBOM の IMDRF 追補ガイダンス案 8 要素の確認工数.....98
表 4-25　手動と MEND SCA での検出結果.....100
表 4-26　MEND SCA と Syft&Grype での検出結果.....103
表 4-27　開発委託先との SBOM 共有方法検討.....103
表 4-28　医療機関への SBOM 提供方法検討結果.....104
表 4-29　SBOM 生成・共有の課題.....105
表 4-30　ツール別の脆弱性の特定及び深刻度評価工数.....111
表 4-31　ツール別の脆弱性の特定及び深刻度評価及び対処方法特定比較.....113
表 4-32　MEND SCA による対象医療機器の脆弱性の特定及び深刻度評価.....115
表 4-33　サンプルコードにおける脆弱性マネジメントの想定シナリオ.....117
表 4-34　本実証における文書作成・更新、レビュー、社内会議等共通項目の工数..117
表 4-35　脆弱性マネジメントにおける標準設定工数(サンプルコード).....118
表 4-36　脆弱性マネジメントにおける標準設定工数(対象医療機器のソフトウェア) 119
表 4-37　サンプルコードでの脆弱性マネジメント工数の算出結果.....122
表 4-38　対象医療機器での脆弱性マネジメント工数の算出結果.....127
表 4-39　各マイルストーンにおける医療機関に提供する情報.....134
表 4-40　各ライフサイクルにおける医療機関に提供する主なサポート内容.....134
表 4-41　OS の EOL 及び EOS の確認結果.....135
表 4-42　Windows10 LTSC 2019 を搭載した KR-X SCAN のサポート計画.....135
表 4-43　医療機関ヒアリングにおける要点.....137
表 4-44　SBOM 活用・管理の課題.....139
表 4-45　製品ポリシーにおけるサポート計画等の課題.....141
表 4-46　医療機関ヒアリングを通じての課題抽出（2022 年 11 月 30 日時点）.....143
表 4-47　SBOM 作成、共有方法、活用・管理まとめ.....145
表 4-48　本実証での実証範囲.....148
表 4-49　実証項目の改定案.....150
表 4-50　開発委託先及び直接契約における契約に含める条項.....156
表 4-51　実証手順.....157
表 4-52　オープン系非リアルタイム OS ベースシステムの開発言語.....159
表 4-53　制御系システムの開発言語.....159
表 4-54　開発環境等(注).....161
表 4-55　実施項目.....161
表 4-56　MEND を使用時の SBOM 初期導入コスト小計.....163
表 4-57　MEND 使用時の SBOM 初期導入コスト内訳.....163
表 4-58　Black Duck 使用時の SBOM 初期導入コスト小計.....164
表 4-59　Black Duck を使用時の SBOM 初期導入コスト内訳.....165
表 4-60　SBOM 手動生成時の SBOM 初期導入コスト小計.....166
表 4-61　SBOM 手動生成時の SBOM 初期導入コスト内訳.....167
表 4-62　SBOM 導入工数.....168
表 4-63　MEND を使用時の SBOM 生成・共有コスト小計.....170
表 4-64　MEND を使用時の SBOM 生成・共有コスト内訳.....170

表 4-65　Black Duck を使用時の SBOM 生成・共有コスト小計 171
表 4-66　Black Duck を使用時の生成・共有コスト内訳.....171
表 4-67　SBOM 手動生成時の SBOM 生成・共有コスト小計.....172
表 4-68　SBOM 手動生成時の SBOM 生成・共有コスト内訳.....172
表 4-69　SBOM 作成工数（オープン系 OS 採用システム） 174
表 4-70　SBOM 作成工数（制御系システム） 174
表 4-71　精査工数 175
表 4-72　SPDX 作成工数.....176
表 4-73　米大統領令最小要素と SPDX 及び SPDX-Lite の比較 176
表 4-74　MEND を使用時の SBOM 活用・管理コスト小計 182
表 4-75　MEND を使用時の SBOM 活用・管理コスト内訳 182
表 4-76　Black Duck を使用時の SBOM 活用・管理コスト小計 184
表 4-77　Black Duck を使用時の SBOM 活用・管理コスト内訳 184
表 4-78　SBOM 手動生成時の SBOM 活用・管理コスト小計.....186
表 4-79　SBOM 手動生成時の SBOM 活用・管理コスト内訳.....186
表 4-80　深刻度評価工数の SBOM 精査タイミングによる比較 189
表 4-81　脆弱性監視工数 191
表 4-82　効果測定結果 193
表 4-83　効果の算出用情報一覧 194
表 4-84　初期工数比較 195
表 4-85　SPDX ファイル出力工数比較 195
表 4-86　ライセンス情報のみの SBOM 作成工数比較 196
表 4-87　深刻度評価工数比較 197
表 4-88　ツール導入工数に対する採算性 197
表 4-89　SBOM 精査工数比較.....198
表 4-90　SBOM 適用範囲（SBOM 活用モデル）の改訂案.....201
表 4-91　人的コストの算出法 209
表 4-92　TMSL の開発言語.....210
表 4-93　契約形態および取引慣行について 211
表 4-94　開発言語、環境およびツールに関する情報 212
表 4-95　生成された SBOM の正確性に関する検討項目 213
表 4-96　顧客への SBOM 提供に関する検討事項.....214
表 4-97　比較検証した脆弱性の調査手法 215
表 4-98　社内 SBOM 管理システムの機能.....216
表 4-99　評価に用いた脆弱性と TMSL における実際の影響度.....216
表 4-100　SCA ツールを使用時の SBOM 初期導入コスト小計.....218
表 4-101　SCA ツールを使用時の SBOM 初期導入コスト内訳.....219
表 4-102　SBOM 作成ツール検討時の確認項目例 220
表 4-103　SCA ツールを使用時の SBOM 生成・共有コスト小計.....221
表 4-104　SCA ツールを使用時の SBOM 生成・共有コスト内訳.....221
表 4-105　SBOM 作成ツール導入コスト 222
表 4-106　SCA ツールでの OSS の検出結果と SBOM への出力 225

表 4-107　SCA ツールの正確性.....226
表 4-108　SBOM 作成および正確性の確認に要した時間.....227
表 4-109　ユーザ側での SBOM 活用に関する期待と課題.....230
表 4-110　SBOM 提供に関する現時点での方針.....231
表 4-111　サブコンポーネントの検出率.....233
表 4-112　SCA ツールを使用時の SBOM 活用・管理コスト小計.....240
表 4-113　SCA ツールを使用時の SBOM 活用・管理コスト内訳.....240
表 4-114　単一製品での影響度調査における作業時間.....242
表 4-115　ユーザとしての脆弱性の調査手法.....245
表 4-116　効果測定結果.....247
表 4-117　効果の算出用情報一覧.....248
表 4-118　作成主体が自社の場合の SBOM 適用モデル案検討結果.....250
表 4-119　SBOM 提供に関する現時点での方針.....254
表 4-120　サプライヤーとの契約において期待される契約事項.....254
表 4-121　実証に参加したメンバーの経験、スキルレベル.....257
表 4-122　コラボフロークラウドの開発言語.....258
表 4-123　対象分野の前提条件・取引慣行等.....260
表 4-124　開発環境等.....261
表 4-125　SBOM 実証フェーズ毎の実証項目.....261
表 4-126　Mend を使用時の SBOM 初期導入コスト小計.....264
表 4-127　Mend を使用時の SBOM 初期導入コスト内訳.....265
表 4-128　Yamory を使用時の SBOM 初期導入コスト小計.....266
表 4-129　Yamory を使用時の SBOM 初期導入コスト内訳.....267
表 4-130　SBOM 手動生成時の SBOM 初期導入コスト小計.....268
表 4-131　SBOM 手動生成時の SBOM 初期導入コスト内訳.....268
表 4-132　SBOM 実証対象ツール.....270
表 4-133　SBOM 初期導入コスト.....270
表 4-134　SBOM 定期生成、評価、運用コスト.....271
表 4-135　SBOM を生成するために必要なツール.....272
表 4-136　Mend を使用時の SBOM 生成・共有コスト小計.....274
表 4-137　Mend を使用時の SBOM 生成・共有コスト内訳.....275
表 4-138　Yamory を使用時の SBOM 生成・共有コスト小計.....276
表 4-139　Yamory を使用時の生成・共有コスト内訳.....277
表 4-140　SBOM 手動生成時の SBOM 生成・共有コスト小計.....278
表 4-141　SBOM 手動生成時の SBOM 生成・共有コスト内訳.....278
表 4-142　SBOM 手動生成時の SBOM 生成・共有コスト内訳.....280
表 4-143　対象のコンポーネント数.....282
表 4-144　SBOM の運用コスト.....282
表 4-145　検証に利用したツールと SBOM 出力可否.....283
表 4-146　OSS ライブラリ毎の間接利用部品の検出工数.....283
表 4-147　Mend を使用時の SBOM 活用・管理コスト小計.....286
表 4-148　Mend を使用時の SBOM 活用・管理コスト内訳.....287

表 4-149　Yamory を使用時の SBOM 活用・管理コスト小計.....288
表 4-150　Yamory を使用時の SBOM 活用・管理コスト内訳.....289
表 4-151　SBOM 手動生成時の SBOM 活用・管理コスト小計.....290
表 4-152　SBOM 手動生成時の SBOM 活用・管理コスト内訳.....290
表 4-153　効果測定結果.....292
表 4-154　効果の算出用情報一覧.....294
表 4-155　脆弱性特定作業 1 回毎のコスト.....295
表 4-156　特定された脆弱性の深刻度、及び悪用可能性を評価作業コスト.....296
表 4-157　利用部品のライセンス準拠確認作業 1 回毎のコスト.....297
表 4-158　SBOM 適用範囲(ソフトウェア).....298
表 4-159　(a1)自社 ＞(b1)直接利用部品.....304
表 4-160　(a1)自社 ＞(b2)間接利用部品.....304
表 4-161　(a2)サプライヤー（開発委託先）取引契約あり ＞(b1)直接利用部品.....305
表 4-162　(a2)サプライヤー（開発委託先）取引契約あり ＞(b2)間接利用部品.....305
表 4-163　(a3)サプライヤー（サードパーティ）取引契約なし ＞(b1)開発者自身.....306
表 4-164　(a3)サプライヤー（サードパーティ）取引契約なし ＞(b2)開発者以外（調達者、利用者）.....306
表 4-165　SBOM に関する認知調査①.....311
表 4-166　SBOM に関する認知調査②.....311
表 4-167　当社における前提条件及び取引慣行等.....315
表 4-168　本実証の実証項目.....316
表 4-169　yamory を使用時の SBOM 初期導入コスト小計.....318
表 4-170　yamory を使用時の SBOM 初期導入コスト内訳.....318
表 4-171　導入タスクの内訳と所要工数.....319
表 4-172　実証におけるタスクの内訳と所要工数.....320
表 4-173　yamory を使用時の SBOM 生成・共有コスト小計.....322
表 4-174　yamory を使用時の SBOM 生成・共有コスト内訳.....322
表 4-175　yamory を使用時の SBOM 活用・管理コスト小計.....328
表 4-176　yamory を使用時の SBOM 活用・管理コスト内訳.....328
表 4-177　効果測定結果.....332
表 4-178　効果の算出用情報一覧.....333
表 4-179　本実証を踏まえた SBOM 適用範囲(SBOM 活用モデル)の改定案.....335
表 4-181　実証で抽出された課題に対する解決ノウハウ・今後の取組施策.....343
表 4-182　取引契約で規定すべき事項.....348
表 5-1 主な OSS ライセンスの種類.....349
表 5-2　日本におけるライセンス違反事例.....352

用語集・略語集

本報告書では、以下のとおり用語を定義する。

用語	概要
CISA	Cybersecurity and Infrastructure Security Agency の略で、米国国土安全保障省に属するサイバーセキュリティ・インフラストラクチャ・セキュリティ庁のこと。
CPE	Common Platform Enumeration の略で、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のこと。MITRE が中心となり仕様が策定された。
CSIRT	Computer Security Incident Response Team の略で、セキュリティインシデントに関する報告を受け、調査や対応活動を行う組織体のこと。
CVE	Common Vulnerabilities and Exposures の略。情報セキュリティにおける脆弱性やインシデントに関する識別子で、脆弱性に一つの CVE ID が付与される。米国非営利団体 MITRE が採番。
CVSS	Common Vulnerability Scoring System の略。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評価方法であり、0～10.0 の間でスコアが定まる。FIRST (Forum of Incident Response and Security Teams) が管理。
CycloneDX	SBOM の規格の一つで、OWASP（Open Web Application Security Project）が中心となって開発された。
DINUM	Direction Interministérielle du Numérique の略で、フランスのデジタル省庁間総局のこと。
DHS	Department of Homeland Security の略で、米国国土安全保障省のこと。
DISA	Defense Information System Agency の略で、米国国防総省の内局である米国国防情報システム局のこと。
DoC	Department of Commerce の略で、米国商務省のこと。
ENISA	European Network and Information Security Agency の略で、欧州連合の機関である欧州ネットワーク情報セキュリティ機関のこと。
FAR	Federal Acquisition Regulations の略で、米国の政府調達に関する一般的な調達原則を定めた連邦調達規則のこと。
FDA	Food and Drug Administration の略で、米国食品医薬品局のこと。
GSA	General Services Administration の略で、米国調達庁のこと。
IMDRF	International Medical Device Regulators Forum の略で、国際的な医療機器規制の整合化と収束を促進する世界各国の医療機器規制当局による任意の活動のこと。日本語では国際医療機器規制当局フォーラムと呼ばれる。
INL	Idaho National Laboratory の略で、米国エネルギー省の国立研究所の一つであるアイダホ国立研究所のこと。

用語	概要
JVN	Japan Vulnerability Notes の略で、日本で流通・利用されているソフトウェアの脆弱性に関連する情報を収集・公開するデータベースのこと。JPCERT/CC と IPA により共同で運営されている。
MITRE ATT&CK	戦術や手法にまとめられ、少数のマトリックスだけでなく STIX/TAXII でも表現された既知の攻撃者の行動の構造化されたリストのこと。
NAS	Network Attached Storage の略で、ネットワーク上に接続可能な記憶装置のこと。
NIST	National Institute of Standards and Technology の略で、米国国立標準技術研究所のこと。
NIST NVD	NIST が管理している脆弱性情報データベース（NVD：National Vulnerability Database）のこと。
NTIA	National Telecommunications and Information Administration の略で、米国商務省電気通信情報局のこと。
NSA	National Security Agency の略で、米国国家安全保障局のこと。
ODNI	Office of the Director of National Intelligence の略で、国家情報長官室のこと。
OMB	Office of Management and Budget の略で、大統領行政府の機関の一つである米国行政管理予算局のこと。
PSIRT	Product Security Incident Response Team の略で、自社で製造・開発する製品やサービスを対象に、セキュリティレベルの向上やインシデント発生時の対応を行う組織のこと。
SAN	Storage Area Network の略で、複数のコンピュータと記憶装置を結ぶネットワークのこと。
SAST	Static Application Security Testing の略で、アプリケーションのソースコードやバイナリコード等を静的に解析し、脆弱性を特定する解析手法のこと。
SCA	Software Composition Analysis の略で、ソフトウェアが使用している部品（特に OSS 部品）の情報を明らかにする分析手法のこと。
SIEM	Security Information and Event Management の略で、ネットワーク製品やセキュリティ製品を含む機器のログを一元管理・解析し、インシデントにつながる脅威を検知するセキュリティ製品のこと。
SPDX	Software Package Data Exchange の略で、SBOM の規格の一つ。Linux Foundation が中心となって開発され、ISO/IEC 5962:2021 として国際標準化されている。
SWID	Software ID の略で、SBOM の規格の一つ。ISO/IEC 19770-2 として国際標準化されているほか、NISTIR 8060 として、NIST より SWID 作成のガイドラインが発表されている。

1. 背景と目的

産業活動のサービス化に伴い、産業に占めるソフトウェアの重要性は高まる傾向にある。特に、近年は、産業機械や自動車等の制御にもソフトウェアの導入が進んでおり、IoT 機器・サービスや今後実用化される 5G 技術においても、汎用的な機器でハードウェア・システムを構築した上で、ソフトウェアにより多様な機能を持たせることで、様々な付加価値を創出していくことが期待される。

ソフトウェアを利活用した製品・サービスの安全・安心を担保するには、利活用するソフトウェアの脆弱性の管理が求められる。セキュリティ・バイ・デザインの考え方に基づいて、企画・設計段階で脆弱性を含めないようソフトウェアが構成されていたとしても、リリース後に脆弱性が発見されることが多く、その場合、ソフトウェアを利活用する側でのソフトウェア更新等の対応が求められる。また、自社の製品・サービスで利活用しているソフトウェアの保守・サポートが終了する場合には、それ以降に発見された脆弱性の管理について代替ソフトウェアへの変更を含めた検討が求められる。しかしながら、オープンソースソフトウェア（以下、「OSS」という。）の利用が一般化する中で、自社製品において利用するソフトウェアであっても、コンポーネントとしてどのようなソフトウェアが含まれているのかを把握することが困難な状況が生じてしまっており、そのような状況において脆弱性の管理を事業者単独で実施することは費用対効果の面からも困難である。

このような課題を踏まえ、米国商務省の電気通信情報局（以下、「NTIA」という。）では、平成 30 年 7 月から Software Component Transparency という官民合同の検討体制を構築し、ソフトウェアの成分構成を表す Software Bill of Materials（以下、「SBOM」という。）をサプライチェーンで共有すること等を通じたソフトウェアの透明性向上に向けた実証実験を、ヘルスケア産業をはじめ複数の産業分野において行っている。

また、米国では、令和 3 年 5 月に署名された大統領令に基づき、政府調達のソフトウェアサプライチェーンセキュリティの対策強化に向け、SBOM の提供を含む基準作りが進められているほか、令和 3 年 12 月には国土安全保障省サイバーセキュリティ・インフラストラクチャセキュリティ庁（以下、「CISA」という。）が SBOM に関するイベントを開催し、今後の課題検討を推進していくことを発表する等、SBOM に係る取組が加速化している。

経済産業省では、「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（以下、「ソフトウェアタスクフォース」という。）」を令和元年 9 月に設置し、SBOM も含めたソフトウェア管理手法等に関して幅広い議論を行っているが、ソフトウェア分野における業界構造や商習慣が異なる我が国においては、総論として SBOM の有用性は理解されているものの、実際の活用に向けては様々なハードルがあると認識している。

本事業は、ソフトウェアを安全に利活用するための脆弱性の管理の在り方や、安全な OSS の選定及び活用するための枠組みを検討するために、OSS の利活用に起因するセキュリティリスクを洗い出すとともに、国内外におけるソフトウェアの脆弱性の管理の取組動向や、OSS を含むソフトウェアを企業等で自社製品・サービスとして活用する際のセキュリティリスクに対応するための取組について調査を行い、多角的な観点から OSS を含むソフトウェアを安全に利活用できる基盤の構築に向けた検討の基礎資料を作成し、我が国のサイバーセキュリティ政策立案に資することを目的とする。併せて、我が国の企業やサプライチェーンにおける SBOM の導入・活用に関する調査及び実証事業を通じ、我が国における SBOM

の普及への課題や効果等について整理を行うものである。

2. 産業活動においてソフトウェアを安全に利活用するための枠組みに関する取組動向等の調査

本項目では、企業等において自社製品・サービスの一部として利活用されるソフトウェアのセキュリティ確保を実現するためのソフトウェア管理に関連する国内外の取組について調査した。特に、以下の項目について調査等を実施した。

1. 米国政府機関における取組動向等の調査
2. 欧州政府機関における取組動向等の調査
3. 企業、業界団体、OSS コミュニティ等におけるソフトウェア管理の取組動向等の調査
4. OSS を含むソフトウェアの安全な利活用に関する調査
5. 米国 CISA との勉強会の開催

2.1. 米国政府機関における取組動向等の調査

本事業で調査対象とした米国政府機関における近年の取組動向等を表 2-1 に示す。以降では各取組の概要等について示す。

表 2-1 米国政府機関における取組動向等の調査対象

#	取組名・文書名	取組日・発行日	取組主体
1-1	Vulnerability Exploitability eXchange(VEX) - Use Cases[1]	2022 年 4 月	CISA
1-2	Software Security in Supply Chains / NIST SP 800-161 Rev.1[2]	2022 年 5 月	NIST
1-3	Silicon Valley Innovation Program[3]	2022 年 5 月	DHS
1-4	Vulnerability Exploitability eXchange (VEX) - Status Justifications[4]	2022 年 6 月	CISA
1-5	SBOM Listening Sessions[5]	2022 年 7 月	CISA
1-6	Enhancing the Security of the Software Supply Chain through Secure Software Development Practices[6]	2022 年 9 月	OMB

[1] https://www.cisa.gov/sites/default/files/publications/VEX_Use_Cases_Document_508c.pdf
[2] https://www.nist.gov/itl/executive-order-14028-improving-nations-cybersecurity/software-security-supply-chains, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-161r1.pdf
[3] https://www.dhs.gov/science-and-technology/svip
[4] https://www.cisa.gov/sites/default/files/publications/VEX_Status_Justification_Jun22.pdf
[5] https://www.federalregister.gov/documents/2022/06/01/2022-11733/public-listening-sessions-on-advancing-sbom-technology-processes-and-practices
[6] https://www.whitehouse.gov/wp-content/uploads/2022/09/M-22-18.pdf

#	取組名・文書名	取組日・発行日	取組主体
1-7	S.4913 - Securing Open Source Software Act of 2022[7]	2022 年 9 月	米国上院
1-8	Securing Software Supply Chain Series - Recommended Practices for Developers[8]	2022 年 9 月	CISA/NSA/ODNI
1-9	Securing Software Supply Chain Series - Recommended Practices for Suppliers[9]	2022 年 10 月	CISA/NSA/ODNI
1-10	Securing Software Supply Chain Series - Recommended Practices for Customers[10]	2022 年 11 月	CISA/NSA/ODNI
1-11	Natinonal Cybersecurity Strategy[11]	2023 年 3 月	White House

2.1.1. CISA：Vulnerability Exploitability eXchange(VEX) – Use Cases

Vulnerability Exploitability eXchange（VEX）とは、ある製品が既知の脆弱性の影響を受けるかどうかを示す機械判読可能なセキュリティ勧告の一つであり、米国 NTIA を中心として開発された。VEX の主な目的は、製品利用者に対して、製品が特定の脆弱性の影響を受けるかどうか、そして影響を受ける場合に是正するために推奨されるアクションがあるかどうかの追加情報を提供し、製品サプライヤー及び利用者の双方のコスト・労力を軽減することにある。VEX を導入することのメリット例を図 2-1 に示す。VEX は、SBOM を活用した脆弱性管理をより効率化するために開発されたが、必ずしも SBOM に VEX を含める必要性はなく、また、VEX を活用する際にも SBOM と併用する必要はないとしている。

[7] https://www.congress.gov/bill/117th-congress/senate-bill/4913/text

[8] https://www.cisa.gov/sites/default/files/publications/ESF_SECURING_THE_SOFTWARE_SUPPLY_CHAIN_DEVELOPERS.PDF

[9] https://www.cisa.gov/resources-tools/resources/securing-software-supply-chain-recommended-practices-guide-suppliers-and

[10] https://www.cisa.gov/news-events/news/cisa-nsa-and-odni-release-guidance-customers-securing-software-supply-chain

[11] https://www.whitehouse.gov/wp-content/uploads/2023/03/National-Cybersecurity-Strategy-2023.pdf

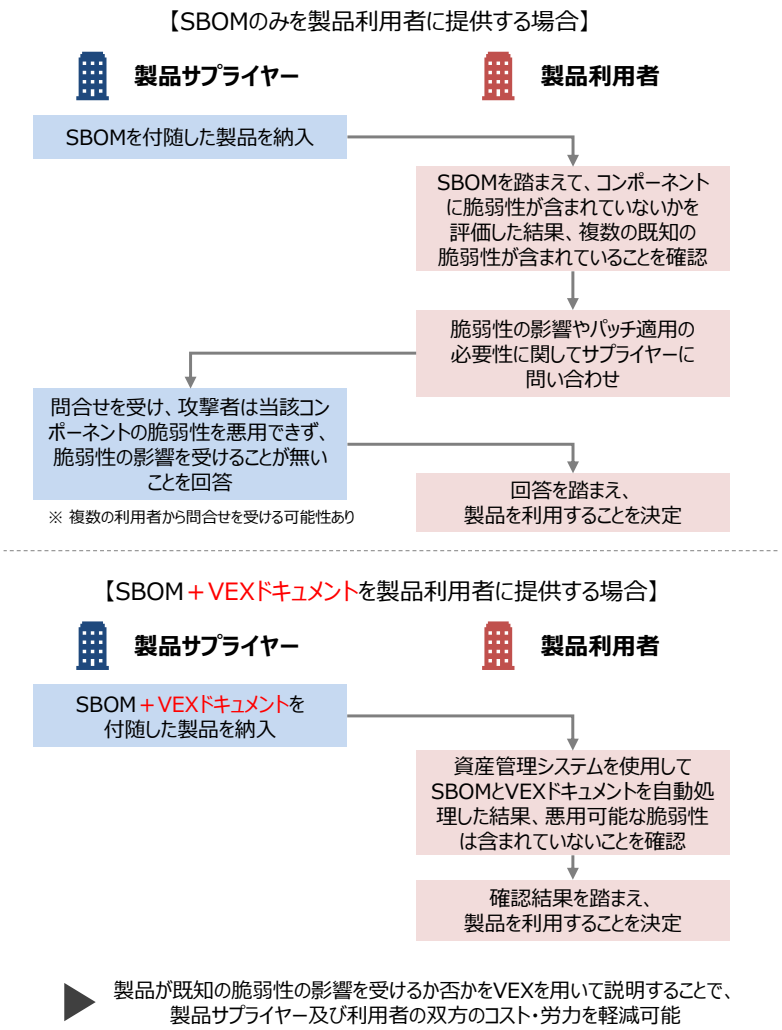


図 2-1　VEX を導入することのメリット例

2022 年 4 月に公開された「Vulnerability Exploitability eXchange(VEX) – Use Cases」では、VEX ドキュメントに含めるべき最小要素及びユースケースが示されている。CISA は、VEX ドキュメントの最小要素として、図 2-2 に示すとおり、メタデータ、製品の詳細、脆弱性の詳細、脆弱性のステータスを含めるべきとしている。脆弱性のステータスについて、「影響を受けない（Not Affected）」、「影響を受ける（Affected）」、「調査中（Under Investigation）」、「修正済み（Fixed）」の 4 つのステータスを定義している。

最小要素の項目	最小要素の具体的な定義
VEXドキュメントのメタデータ	VEXドキュメントのメタデータについて、以下の情報を含めること。 • VEXフォーマットの識別子 • VEX文書の作成者 • VEX文書のタイムスタンプ • VEX文書の識別文字列 • VEX文書の作成者の役割
製品の詳細	VEXドキュメントの対象製品について、以下のいずれかの情報を含めること。 • 製品の識別子 • 製品群の識別子（例：一意の識別子、サプライヤー名・製品名・バージョン文字列の組み合わせ）
脆弱性の詳細	対象製品に存在する脆弱性について、以下の情報を含めること。 • 脆弱性の識別子（CVEまたは他の識別子） • 脆弱性の説明（例：CVEの説明）
脆弱性のステータス	対象製品に存在する脆弱性のステータスについて、以下のいずれかの情報を含めること。 • 影響を受けない（NOT AFFECTED）：脆弱性について、対策は必要ない状態 • 影響を受ける（AFFECTED）：脆弱性を対策または対処が推奨される状態で、脆弱性に対する対処策も情報として含める • 修正済み（FIXED）：脆弱性に対する修正がされている状態 • 調査中（UNDER INVESTIGATION）：脆弱性の影響を受けるかどうか不明な状態

図 2-2　VEX ドキュメントに含めるべき最小要素

同文書では、VEX ドキュメントを作成する方法を理解するために、対象とする製品数、製品のバージョン数、脆弱性数、脆弱性に関する製品ステータスに関する複数のシナリオを想定したユースケースが示されている。各ユースケースでは、脆弱性情報が公開された際に、企業が保有する製品の状況に応じた VEX ドキュメントを作成するに当たっての対応事例（具体的な VEX ドキュメント例を含む）に加えて、VEX ドキュメントに関する補足）が紹介されている。例えば、ある企業の製品の特定のバージョンにおいて、Log4Shell（CVE-2021-44228）の脆弱性の影響を受ける（Affected）場合に作成される VEX ドキュメントの例[12]が示されているほか、ある企業の製品の特定のバージョンにおいて Ripple20 の複数の脆弱性が存在するが、その影響は受けない（Not Affected）場合に作成される VEX ドキュメントの例[13]も示されている。さらに、複数の製品に Log4Shell（CVE-2021-44228）の脆弱性が存在したが、既に修正済み（Fixed）である場合に作成される VEX ドキュメント例[14]や、複数の製品において複数の脆弱性が存在する場合の VEX ドキュメント例[15]も示されている。

なお、VEX は、機械判読可能なセキュリティアドバイザリー標準の一つである CSAF（Common Security Advisory Framework）のプロファイルとして実装されている。CSAF とは、標準化団体 OASIS Open が開発した JSON ベースの機械判読可能なセキュリティアドバイザリー標準のことで、動的に変動する製品のセキュリティアドバイザリー情報を効率的に自動処理する目的で開発された。CSAF では、図 2-3 に示すとおり、メタデータに関するプロパティ（document）、アドバイザリーの対象となる製品に関するプロパティ（product_tree）、脆弱性に関するプロパティ（vulnerabilities）の 3 つのプロパティが記載され、Schneider Electric 社など一部の海外企業では、既に CSAF 形式でのアドバイザリー情報提供を始めている[16]。

[12] https://github.com/oasis-tcs/csaf/blob/master/csaf_2.0/examples/csaf/csaf_vex/2022-evd-uc-01-a-001.json
[13] https://github.com/oasis-tcs/csaf/blob/master/csaf_2.0/examples/csaf/csaf_vex/2022-evd-uc-02-na-001.json
[14] https://github.com/oasis-tcs/csaf/blob/master/csaf_2.0/examples/csaf/csaf_vex/2022-evd-uc-07-001.json
[15] https://github.com/oasis-tcs/csaf/blob/master/csaf_2.0/examples/csaf/csaf_vex/2022-evd-uc-08-001.json
[16] Schneider Electric, Cybersecurity Support Portal https://www.se.com/ww/en/work/support/cybersecurity/security-notifications.jsp

document

セキュリティアドバイザリーに関するメタデータが記述される（色付き下線は必須オブジェクト）：

- 謝辞（acknowledgments）
- 統合的な深刻度（aggregate_severity）
- カテゴリ（category）
- CSAFバージョン（csaf_version）
- 配布範囲（distribution）
- 言語（language）
- ノート（notes）
- 発行者（publisher）
- 参考情報（references）
- ソース言語（source_lang）
- タイトル（title）
- 来歴（tracking）

product_tree

セキュリティアドバイザリーの対象製品のベンダー、バージョン、名称、ID等を、ブランチ（branches）の項目に基づき、木構造で記述される。

記載例

```
"product_tree": {
  "branches": [
   {
    "name": "Cisco",
    "category": "vendor",
    "branches": [
     {
      "name": "IOS",
      "category": "product_name",
      "branches": [
       {
        "name": "12.2SE",
        "category": "product_version",
        "branches": [
    ....
```

vulnerabilities

1つ以上の脆弱性に関するオブジェクトが記述される：

- 謝辞（acknowledgments）
- CVE番号（cve）
- CWE情報（cwe）
- 検出日（discovery_date）
- フラグ（flags）
- 脆弱性ID（ids）
- 関与（involvements）
- ノート（notes）
- 製品ステータス（product_status）
- 参考情報（references）
- 発表日（release_date）
- 改善策（remediations）
- スコア（scores）
- 脅威（threats）
- タイトル（titles）

図 2-3　CSAF フォーマットの 3 つのプロパティ

VEX ドキュメントは CSAF の 3 つのプロパティに基づき記述される。CSAF の 3 つのプロパティの概要と関係性を図 2-5 に示す。メタデータに関するプロパティ（document）では、発行者、タイトル、ID、初回発行日、改訂履歴、VEX ドキュメントのステータス等、当該 VEX ドキュメントに関する情報が記述される。製品に関するプロパティ（product_tree）では、ベンダー、製品名、バージョンの関係性をツリー構造が記述される。脆弱性に関するプロパティ（vulnerabilities）では、各製品・各バージョンに存在する脆弱性について、4 つの脆弱性ステータスのどれに該当するかが記述される。

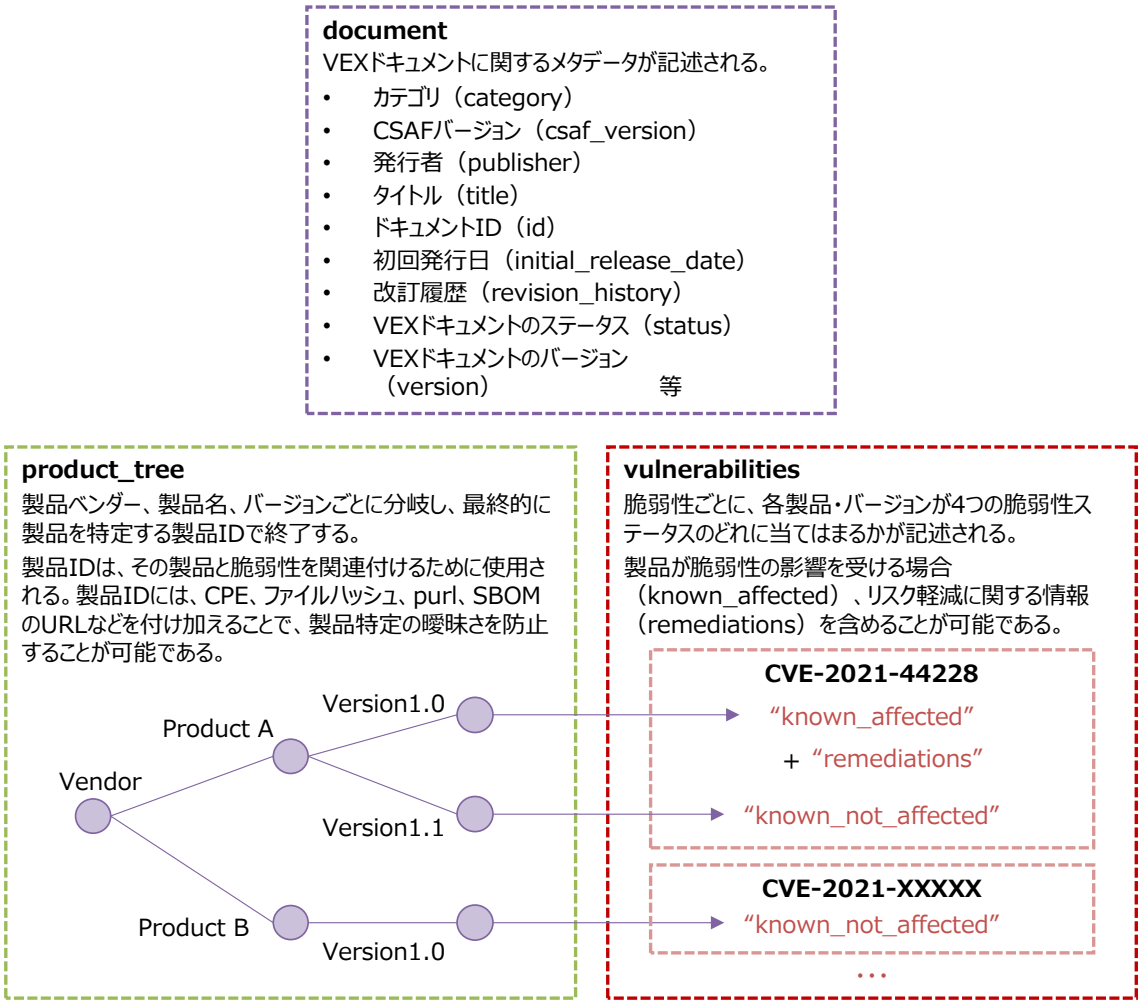


図 2-4　VEX ドキュメントの構造（3 つのプロパティの概要とプロパティの関係）

CSAF で記述される VEX ドキュメントのイメージを図 2-5 に示す。上述のとおり、CSAF のプロパティに基づき、メタデータに関するプロパティ、製品に関するプロパティ、脆弱性に関するプロパティの 3 つに分かれ、ベンダー情報、改善策、脆弱性スコア等の多様な情報を記載できる。なお、CSAF 形式の VEX ドキュメントを作成・編集するツールとしては、ドイツ BSI が開発した Secvisogram[17]が存在する。

VEX と SBOM の関係性について、標準フォーマットの CSAF 内に SBOM 情報を直接記述することはできないが、SBOM 標準フォーマットの SPDX、CyclonDX、SWID タグで記述された SBOM 情報を外部参照することが可能である。SPDX では、VEX 情報を直接記述することができないが、外部参照として CSAF 形式の VEX ドキュメントを参照することができる。一方、CycloneDX では、SPDX と同様に CSAF 形式の VEX ドキュメントを外部参照することができるほか、SBOM 情報内に VEX 情報を直接記述することができる。ただし、CycloneDX を開発した OWASP は、脆弱性情報は動的に変化するため、VEX 情報を SBOM 情報に埋め込むのではなく、外部参照することを推奨している。

[17] https://github.com/secvisogram/secvisogram

```
1.  {
2.    "document": {
3.      "category": "csaf_vex",
4.      "csaf_version": "2.0",
5.      "notes": [
6.        {
7.          "category": "summary",
8.          "text": "Example Company VEX document. Unofficial content for demonstration purposes only.",
9.          "title": "Author comment"
10.       }
11.     ],
12.     "publisher": {
13.       "category": "vendor",
14.       "name": "Example Company ProductCERT",
15.       "namespace": "https://psirt.example.com"
16.     },
17.     "title": "Example VEX Document Use Case 1 - Not Affected",
18.     "tracking": {
19.       "current_release_date": "2022-03-03T11:00:00.000Z",
20.       "generator": {
21.         "date": "2022-03-03T11:00:00.000Z",
22.         "engine": {
23.           "name": "Secvisogram",
24.           "version": "1.11.0"
25.         }
26.       },
27.       "id": "2022-EVD-UC-01-NA-001",
28.       "initial_release_date": "2022-03-03T11:00:00.000Z",
29.       "revision_history": [
30.         {
31.           "date": "2022-03-03T11:00:00.000Z",
32.           "number": "1",
33.           "summary": "Initial version."
34.         }
35.       ],
36.       "status": "final",
37.       "version": "1"
38.     }
39.   }
40.   "product_tree": {
41.     "branches": [
42.       {
43.         "branches": [
44.           {
45.             "branches": [
46.               {
47.                 "category": "product_version",
48.                 "name": "4.2",
49.                 "product": {
50.                   "name": "Example Company ABC 4.2",
51.                   "product_id": "CSAFPID-0001"
52.                 }
53.               }
54.             ],
55.             "category": "product_name",
56.             "name": "ABC"
57.           }
58.         ],
59.         "category": "vendor",
60.         "name": "Example Company"
61.       }
62.     ]
63.   },
64.   "vulnerabilities": [
65.     {
66.       "cve": "CVE-2021-44228",
67.       "notes": [
68.         {
69.           "category": "description",
70.           "text": "Apache Log4j2 2.0-beta9 through 2.15.0 (excluding security releases 2.12.2, 2.12.3, and 2.3.1) JNDI features
      used in configuration, log messages, and parameters do not protect against attacker controlled LDAP and other JNDI related
      endpoints. An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP
      servers when message lookup substitution is enabled. From log4j 2.15.0, this behavior has been disabled by default. From version
      2.16.0 (along with 2.12.2, 2.12.3, and 2.3.1), this functionality has been completely removed. Note that this vulnerability is
      specific to log4j-core and does not affect log4net, log4cxx, or other Apache Logging Services projects.",
71.           "title": "CVE description"
72.         }
73.       ],
74.       "product_status": {
75.         "known_not_affected": [
76.           "CSAFPID-0001"
77.         ]
78.       },
79.       "threats": [
80.         {
81.           "category": "impact",
82.           "details": "Class with vulnerable code was removed before shipping.",
83.           "product_ids": [
84.             "CSAFPID-0001"
85.           ]
86.         }
87.       ]
88.     }
89.   ]
90. }
```

VEXドキュメントのメタデータ

脆弱性の影響を受ける製品の情報

脆弱性の情報

図 2-5　CSAF 形式による VEX ドキュメントの例

2.1.2. NIST：Software Security in Supply Chains / NIST SP 800-161 Rev.1

2022 年 5 月、NIST は、2021 年 5 月大統領令で指示されたソフトウェアサプライチェーンのセキュリティ強化に向けたガイダンスを、SP 800-161（Cybersecurity Supply Chain Risk Management (C-SCRM) Practices for Systems and Organizations）の Rev. 1 の付録及び専用の Web サイトにおいて公開した。NIST は、関連する大統領令文書との相互な関わりを促進するとともに、SP 800-161 に影響を与えることなく本ガイダンスの更新を可能とするために Web サイトでの公開を行ったとしている。本ガイダンスでは、大統領令に対する NIST の取組と SP 800-161 Rev.1 で規定された C-SCRM の管理策との対応関係（図 2-6 参照）を示すほか、

ソフトウェアサプライチェーンのセキュリティ確保のための新たな 4 つの概念（図 2-7 参照）を概説している。

「重要なソフトウェア」の定義と求められる対策	• 各政府機関が優先的にセキュリティ対応を行う対象となる「重要なソフトウェア」の定義及び「重要なソフトウェア」のセキュリティ対策に関する取組。 • **C-SCRMの管理策や補足ガイダンスは、「重要なソフトウェア」に求められるセキュリティ対策を実施するための有効な手段**である。
開発者と消費者のためのソフトウェアサイバーセキュリティ	• ソフトウェア開発を安全に実施するため、開発者が準拠すべき対策（SSDF）と、消費者が開発者へ要求すべき最低限の推奨事項に関する取組。 • **C-SCRMの管理策や補足ガイダンスを用いて、開発者へ要求すべき最低限の推奨事項を確認**できる。
ソフトウェア検証の最低基準	• サプライヤー（ベンダーや開発者など）によるソフトウェア検証の際に推奨される最低基準に関する取組。 • **C-SCRMの管理策や補足ガイダンスを用いて、最低基準を満たしたソフトウェア検証が実施されていることを確認**できる。

図 2-6　大統領令に対する NIST のこれまでの取組と C-SCRM との対応関係

SBOM	• 各省庁は、ソフトウェア製品やサービスのサプライヤーが、**大統領令及びNTIAが公表しているガイドラインに準拠したSBOMを作成していることを確認すべきである**。
サプライヤーにおけるリスク評価の強化	• サプライヤーが作成したソフトウェアだけではなく、**サプライチェーンにおいてソフトウェアに関与する事業体に対してもリスク評価を強化すべき**である。
OSSの管理	• 各省庁は、**サプライヤーのソフトウェアにおけるOSSの使用状況をより深く理解するよう努めるべき**である。
脆弱性管理	• ゼロトラストアーキテクチャの議論を踏まえ、**各省庁の戦略は、検出された脆弱性をいかに効率的かつ包括的に管理するかについて焦点を当てるべき**である。

図 2-7　ソフトウェアサプライチェーンのセキュリティ確保のための新たな概念

NIST は、連邦政府機関に対して、本ガイダンスを活用して C-SCRM の管理策を調達先に対して適用することのほか、可能な場合に新たな 4 つの概念を採用することを求めている。

2.1.3. DHS：Silicon Valley Innovation Program

2021 年 5 月より、DHS は 5 年間・$75M の予算を投じる研究開発プログラムである Silicon Valley Innovation Program（SVIP）を開始した。そして、SVIP プログラムの一つとして、ソフトウェアサプライチェーンの可視化を目的としたツール開発を行う事業者の募集を開始した。募集期間は 2022 年 10 月 3 日まで設定された。図 2-8 に示すとおり、本研究開発プログラムで開発するツールについて、ソフトウェアサプライチェーンの関係者に対して SBOM に関する機能を展開することを目的に、SBOM 自動生成機能や脆弱性情報の可視化機能等を有した OSS ライブラリであることを必須要件としつつ、SBOM に関連する追加機能やプラグインを一つ以上有することが追加要件として設定された。

要件	詳細
OSSライブラリであること	以下の機能を有したOSSライブラリであること。 • SBOM標準フォーマット（SPDX、SWID、CycloneDX）のインポート、相互変換、エクスポートに関する機能 • CPE、SWIDタグ、purl、SWHIDsのソフトウェア識別子の間でコンポーネントの対応関係を取る機能
SBOMの自動生成・更新機能を有していること	SDLCへ統合し、SBOMの自動生成・自動更新ができること。
脆弱性の可視化機能を有していること	様々なデータ形式のSBOMを読み取り、SBOM情報を信頼できる脆弱性情報とリンクさせ、利用可能なパッチや緩和策に関する情報を提供することができること。
SBOM対応のIDEプラグインを有していること	ソフトウェア開発者に対して、以下の情報の読み取り・可視化の機能を提供するSBOM対応IDEプラグインであること。 • 開発ソフトウェアに対応するSBOM • CVEや開発しているソフトウェアの関数・ライブラリ・関連コードに関する脆弱性情報 • 脆弱性の重要度（例：CVSS、SSVC）、脆弱性の状況 • 利用可能なパッチや緩和策
SBOM対応のSIEMプラグインを有していること	既存SIEMツールに統合し、稼働中ソフトウェアに関する以下の情報を提供するSBOM対応プラグインであること。 • 脆弱性の影響を受けるシステム • ソフトウェアに関連するCVE • 脆弱性の重要度（例：CVSS、SSVC） • 利用可能なパッチや緩和策

図 2-8　SVIP で募集対象となったツールに求められる要件

なお、ツールの開発に当たって表 2-2 に示す 5 つのフェーズが設定されており、フェーズごとに設定された成果物や活動に関する条件を満たすことで、継続的な資金提供（各フェーズ$50,000～$500,000）を受けることができる。

表 2-2　SVIP のツール開発において各フェーズに設定された成果物・活動目標の概要

フェーズ	成果物・活動目標概要	成果物・活動に関する条件 （次期フェーズに進むための条件）	資金	期間
1	PoC を実施するに当たって必要な最低限の機能を有する製品を開発すること。	PoC を実施するための最低限の機能を有する製品を開発するとともに、検証可能なテストのエビデンスや設計書、ソフトウェアデモ等を提示すること。なお、本フェーズ終了時点で以下のいずれかを完了すること。 ・ 本募集に適した新技術での PoC を実施すること ・ 既存技術を活用した場合、本募集に適するように修正を加えた上で、PoC を実施すること ・ 開発した製品の市場参入のための戦略文書を作成すること	$50,000～$200,000	採択後、3～9 ヶ月

フェーズ	成果物・活動目標概要	成果物・活動に関する条件（次期フェーズに進むための条件）	資金	期間
2	応募時の全ての機能を有するプロトタイプを開発すること。	フェーズ 1 の結果を踏まえてプロトタイプを開発し、プロトタイプが実行可能であることを実証すること。なお、本フェーズの終了時点で、プロトタイプは以下の条件を満たすこと。 ・ エンド・ツー・エンドでの運用可能性に関して実証すること ・ プロトタイプに対する独立したレビューと評価のための準備を行うこと ・ 潜在的な顧客やパートナーと共に、市場参入のための戦略を検証すること	$50,000 ～ $500,000	フェーズ 1 終了後、3～9ヶ月
3	プロトタイプを現実的なテスト・評価環境に展開し、応募要件に対する適合性の評価・検証を実施すること。	製品の生産準備がなされたプロトタイプに対し、現実的なテスト・評価環境におけるテスト・評価プロセスを経て、製品の運用性を確保すること。なお、本フェーズの終了時には、各テストが DHS の各部門や運用関係者と十分に周知され、全テスト結果を踏まえたフィードバックが製品へ組み込まれる必要がある。	$50,000 ～ $500,000	フェーズ 2 終了後、3～9ヶ月
4	テストを完了させ、応募時の全ての機能を有する製品を開発すること。	製品設計が完了し、応募時に提案された全機能を有する製品を提示する必要がある。なお、本フェーズでのテストや製品デモは、DHS の各部門及び運用関係者と調整が完了し、複数のユーザーシナリオや条件でのテストが実施される必要がある。	$50,000 ～ $500,000	フェーズ 3 終了後、3～9ヶ月
5	本フェーズは、政府機関のニーズや追加テストの要件が発生した場合にのみ、フェーズ 4 終了後に開始される。提供資金について、フェーズ 1～4 の資金枠を超える可能性がある。			

2.1.4. CISA：Vulnerability Exploitability eXchange (VEX) - Status Justifications

2022 年 6 月、CISA は、VEX 文書の最小要素における「脆弱性のステータス」の内、脆弱性の「影響を受けない（NOT AFFECTED）」ステータスを正当化するための具体的な 5 つの主張を定義し公開した。具体的な 5 つの主張は図 2-9 に示すとおりである。

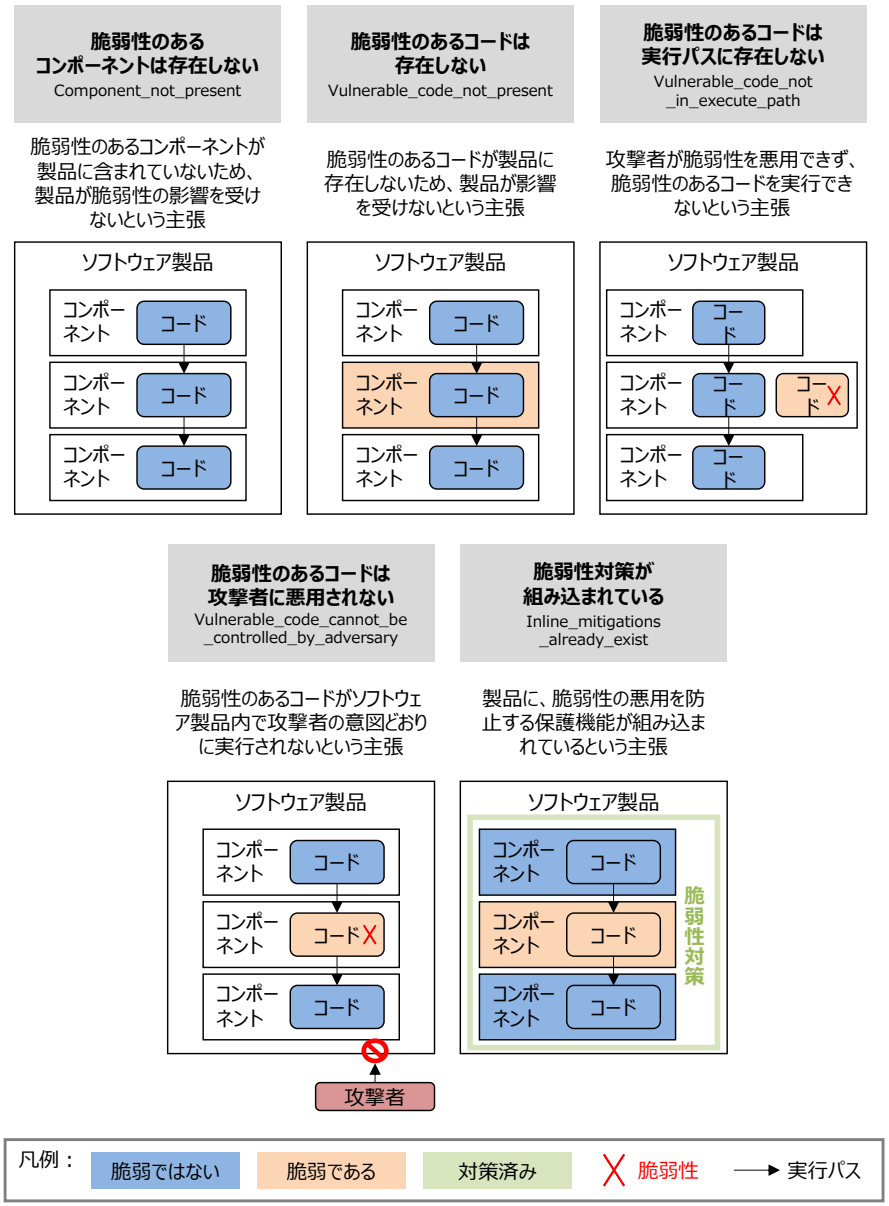


図 2-9　VEX で「脆弱性の影響を受けない」ステータスを正当化するため 5 つの主張

また、文書では、各主張の概要について説明するとともに、どのような場合にどの主張を実施すべきかを判断するためのフローチャートが示されている（図 2-10 参照）。なお、CISA は今後の検討によって、新たな主張を追加する可能性があるとしている。

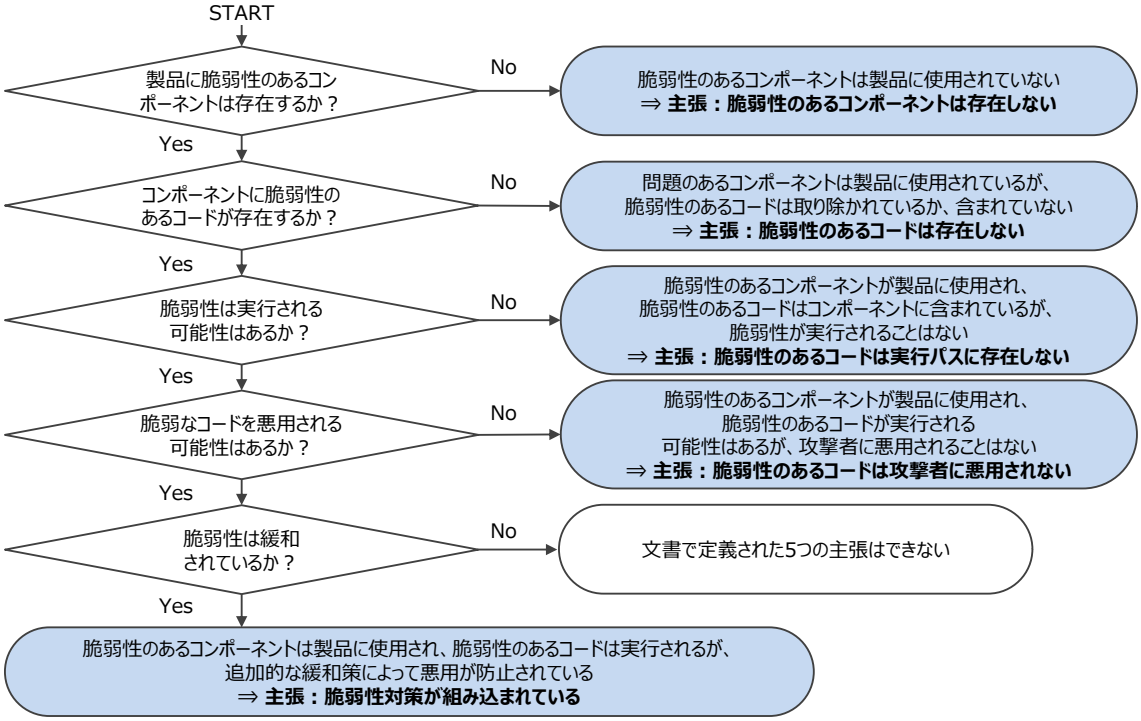


図 2-10　VEX で「脆弱性の影響を受けない」ステータスを正当化するための主張を判断するためのフローチャート

2.1.5. CISA：SBOM Listening Sessions

2022 年 7 月 12 日から 21 日にかけて、CISA は、SBOM の特定のテーマ（クラウドとオンラインアプリケーション、導入と採用、共有と交換、ツールと実装）について、コミュニティにおける取組を開始するため、公開のリスニング・セッションを実施した。各テーマにおける主な議論内容と、議論において挙げられた主な意見等は表 2-3 に示すとおりである。リスニング・セッションにおいては、CISA はファシリテーターとなりセッション参加者からの意見を聴取し、聴取した意見を踏まえ今後の各コミュニティの方向性を検討するとした。

表 2-3　CISA SBOM Listening Sessions における主な議論内容・意見等

テーマ	議論内容	主な意見等
Cloud & Online-Applications	・ オンラインアプリケーション、最新のインフラストラクチャと SBOM の関係 ・ クラウドと SBOM の関係	・ クラウドの SBOM を検討する場合、副次的な影響についても念頭に置く必要。 ・ 「クラウド」は非常に広いトピックであり、議論を深掘りするためには、どのようなタイプのクラウドサービスをまず対象とするか絞り込む必要。 ・ アウトカムとして、クラウドやオンラインアプリケーションに対する SBOM 生成のベスト・プラクティスが分かるとよい。
On-Ramps &	・ SBOM の容易・	・ 複数の組織の成果や情報をうまく統合し、

テーマ	議論内容	主な意見等
Adoption	廉価な導入方法	情報をファシリテートする取組が求められる。SBOM に関する教育を支援する取組も必要。情報が不足している SBOM に対する情報集約方法、SBOM データの信頼性に関するリスクも考慮する必要。 ・ 開発者においてもメリットが享受できるよう、SBOM 生成は簡単かつ廉価である必要。
Sharing & Exchanging	・ SBOM の情報共有方法	・ SBOM 情報の共有に当たっては、対象ソフトウェアとその SBOM に対する名前付けとバージョン付けが重要。SBOM に対する署名やアシュアランスも必要。 ・ SBOM 共有による脆弱性管理やインシデント対応に対するユースケースを考える必要。本取組の潜在的なアウトカムとして、ガイダンスが望まれる。
Tooling & Implementation	・ ツールのカテゴライズ化（生成・利活用、分野におけるツールの特徴等）、導入における障壁、互換性（フォーマット間、ツール間等）など	・ ツールによってカバーされる Third-party OSS やツールごとの機能をわかりやすく示す必要。Plugfests（相互接続性を確認する取組）を実施して、ツールの違いを認識するのは重要。SBOM の保守、更新とバージョンの管理をサポートするツールも必要。 ・ SBOM 共有による脆弱性管理やインシデント対応に対するユースケースを考える必要。

リスニング・セッション後、表 2-3 に示す 4 つのテーマに加え、VEX に関するテーマの計 5 つのテーマについてワークストリームを構築し、当該コミュニティでの打合せを週次で実施している。2023 年 3 月 14 日に CISA から共有された情報によれば、各ワークストリームの焦点と現在の活動状況はそれぞれ以下のとおりである。

表 2-4　CISA SBOM Workstream における主な焦点・活動状況

ワーク ストリーム	**主な焦点・活動状況（2023 年 3 月 14 日時点）**
VEX	このワークストリームでは、製品がある脆弱性の影響を受けるか受けないかを証明できる VEX（Vulnerability Exploitability eXchange）モデルの定義と改良を行い、VEX のユースケースと運用を特徴付けている。現在の活動状況は以下のとおり。 ・ 「VEX の最小要件」文書がワーキンググループにより完成し、CISA SBOM ページで共有される予定である。この文書は、拡張性

ワーク ストリーム	主な焦点・活動状況（2023 年 3 月 14 日時点）
	のある実装を支援し、期待値の調和を図る役割を果たす。 ・ 次のステップ：VEX ガイダンスの作業を継続し、VEX を発行する時期に関するいくつかの選択肢を共有することを含む。
Cloud & Online-Applications	このワークストリームでは、SBOM に関する現在の理解を、オンラインアプリケーションや分散型オンデマンドインフラストラクチャに統合することに重点を置いている。SBOM の使用例に関する既存の議論のほとんどは、オンプレミスのソフトウェアに焦点を当てているが、オンラインやクラウドベースのアプリケーションは、ソフトウェアエコシステムの中で大きく成長しているセグメントである。SBOM に関する現在の理解をクラウドネイティブテクノロジーの新たな進歩と統合し、クラウドにおける SBOM の使用例についてよりよいストーリーを構築し、組織の境界を越え、どのように扱われるかを理解することが重要である。現在の活動状況は以下のとおり。 ・ SaaS プロバイダーと顧客向けの SBOM に関するガイダンスを構築する。 ・ クラウドスタックにおける透明性の定義を探究する。 ・ オンラインアプリケーションのサードパーティサービス利用の推移グラフを追跡するための、サービスの透明性モデルを定義する。
On-Ramps & Adoption	このワークストリームでは、SBOM に関する教育と認識を促進し、SBOM 導入のコストと複雑さを軽減することに重点を置いている。特に、新たな組織や成熟度の低い組織が、組織のリスクを確保しつつ、SBOM を提供、要求、使用できるために、障壁を取り除き、SBOM 導入を加速させることを目標としている。このワークストリームでは、SBOM のさらなるユースケースを定義する予定である。加えて、全ての SBOM 関連ワークストリームの取組を調整し、コミュニケーションを支援するとともに、実質的な重複を回避することも目的としている。現在の活動状況は以下のとおり。 ・ 購入・調達のユースケースにおける SBOM の使用に関する明確なガイダンスを構築する。 ・ SBOM を求める組織をどのように可能にし、支援するかを議論する。 ・ SBOM の FAQ の更新と拡張に向けた議論を行う。
Sharing & Exchanging	このワークストリームでは、ソフトウェアのサプライチェーン全体で、SBOM や関連するメタデータを共有するというトピックに焦点を当てている。ソリューションの相互運用性の重要性を強調しつつ、SBOM の検出や SBOM へのアクセスを可能にする方法について議論している。現在の活動状況は以下のとおり。 ・ 共有要件をよりよく理解するために、特定の共有ユースケースを調査する。

ワーク ストリーム	主な焦点・活動状況（2023 年 3 月 14 日時点）
	・ このようなユースケースの内、単純なユースケースとしては、ソフトウェアを配布する場合や、顧客が SBOM を要求する場合などがある。 ・ より複雑なユースケースとしては、(1)様々なアクセス管理を伴う多部門間でのサプライチェーン、(2)資産管理や脆弱性管理を伴うネットワークに SBOM データを統合する運用のアプローチ等がある。
Tooling & Implementation	このワークストリームでは、SBOM エコシステムの自動化に向けた機会と課題に焦点を当てている。このエコシステムは、オープンソース及びプロプライエタリの両方において、アクセス可能で建設的なツール及び実現アプリケーションの範囲によって推進される。この作業により、既存の SBOM データにさらなる実装を与え、相互運用性を促進し、ツール市場の進歩及び効率化を促進する可能性がある。現在の活動状況は以下のとおり。 ・ 「SBOM ドキュメントの種類」に関する 2 ページの概要が完成し、CISA SBOM ページで共有される予定である。この文書は、SBOM ツールが生成できる様々な種類の SBOM（例えば、ソースからの SBOM、ビルド時に作成する SBOM など）の違いや価値を明確にするものである。 ・ SBOM の相対的な品質をどのように測定し、伝達するかについて議論し、ガイダンスを示す。 ・ SBOM ツールの相互運用性をテストし、向上させるための公開「プラグフェスト（plugfests）」の計画を開始する。

2.1.6. OMB : Enhancing the Security of the Software Supply Chain through Secure Software Development Practices

2022 年 9 月 14 日、OMB は、安全なソフトウェア開発手法の実装を通じたソフトウェアサプライチェーンの確保に関する覚書を発行した。覚書では、機関[18]は、安全なソフトウェア開発手法（SSDF）の実装を証明できるソフトウェアベンダーが提供するソフトウェアのみを使用すべきとしており、そのために、各機関の最高情報責任者（CIO）は、OMB 及び最高調達責任者（CAO）と連携し、ソフトウェアベンダーによる SSDF の実装、実装の適合性を確保しなければならないとしている。そして、このために、機関は以下を実施する必要があるとしている。

- 機関は、ソフトウェア使用前に、SSDF の実装の適合性を証明する自己適合証明書をソフトウェアベンダーから取得することを要求する。
- 機関は、必要に応じて、自己適合証明書に付随する成果物（SBOM 等）をソフトウェ

[18] 44 U.S.C. § 3502(1)で定義される「機関（Agency）」を意味しており、連邦政府、軍関係省、行政法人、行政管理法人、政府行政部門のその他施設（大統領府を含む）、独立規制機関が対象。

アベンダーから入手する。

覚書の対象となるソフトウェアには、ファームウェア、OS、アプリケーション、アプリケーションサービス（クラウドベースのソフトウェア）、ソフトウェアに使用される OSS 及びソフトウェアを使用する製品が含まれるが、機関によって開発されたソフトウェアや直接的に入手した OSS は対象外となる。また、要件の適用範囲に関して、覚書発行日以降に開発されたソフトウェア（既存ソフトウェアのメジャーバージョンアップ含む）を機関が使用する場合に適用される。本覚書では、一定期間内に機関が使用するソフトウェアの目録作成や、NIST のガイダンスに基づく自己適合証明書をソフトウェアベンダーに要求することなどを求めている。関連主体に対する要求事項具体的な対応スケジュールは表 2-5 に示すとおりである。

表 2-5　OMB 覚書発行日以降の各機関・OMB・CISA に対する要求事項

スケジュール	各機関	OMB	CISA
覚書発行後 90 日以内	対象となる全てのソフトウェアの目録を作成すること[19]	各機関が、本覚書の適用を延長・免除するための申請方法を専用 WEB サイトへ掲載すること	—
覚書発行後 120 日以内	—	—	OMB と協議し、文書業務削減法（PRA）に準拠した、自己適合証明書の共通様式を制定すること
覚書発行後 180 日以内	—	CISA、GSA と協議し、機関間で保護及び共有する仕組みを備えた、自己適合証明書や付随する成果物を管理するためのリポジトリー要件を確立すること	—
覚書発行後 270 日以内	「重要なソフトウェア」[20]に関するベンダーの自己適合証明書を取得すること	—	—

[19] 「重要なソフトウェア」に対するソフトウェア目録の作成は、2021 年 8 月 10 日に発行された OMB 覚書「Protecting Critical Software Through Enhanced Security Measures」において、当該覚書発行後 60 日以内に対応することが既に求められている。

[20] 2021 年 5 月の大統領令に基づき定義された「重要なソフトウェア（Critical Software）」を意味する。具体的な定義は NIST 文書「Definition of Critical Software Under Executive Order (EO) 14028」にて規定されている。

スケジュール	各機関	OMB	CISA
覚書発行後 365 日以内	全てのソフトウェアに関するベンダーの自己適合証明書を取得すること	—	—
OMB によるリポジトリー要件確立から 365 日以内	—	—	GSA、OMB と協議し、自己適合証明書や付随する成果物を管理するリポジトリーのためのプログラム計画を策定こと

2.1.7. S.4913 - Securing Open Source Software Act of 2022

2022 年 9 月、連邦政府の OSS 利用におけるセキュリティ確保に向けた法案が米国上院へ提出された。本法案の要求事項の概要を表 2-6 に示す。本法案では、OSS 利用におけるセキュリティ確保には特徴的な課題が存在するとし、国土安全保障法（合衆国法典 Title 6）に対して、OSS のセキュリティに関する内容の加筆を要求している。加えて、CISA に対して、OSS 利用におけるセキュリティ確保における責任の明確化、OSS のリスクを評価するためのフレームワークの公開、各省庁に向けた OSS に関するガイダンスの発行を要求している。また、CISA に選定された機関に対しては、機関内にオープンソースプログラムオフィス（OSPO）の設立が要求されている。なお、本法案はあくまで法案であり、法制化していないことに留意が必要である。

表 2-6 「Securing Open Source Software Act of 2022（法案）」の概要

区分	要求事項
オープンソースソフトウェアのセキュリティ義務 **（Section 3）**	国土安全保障法第 22 編（合衆国法典 Title. 6 §651）の副題 A を以下のとおりに改正すること。 ● 第 2201 条（合衆国法典 Title. 6 §651 ）において、OSS という用語の定義を追加する。 ● 第 2202 条(c) (合衆国法典 Title. 6 §652(c))において、CISA の責任として「各省庁におけるソフトウェア開発ライフサイクルにおける、OSS を含むソフトウェアの安全な使用を支援」を追加する。 ● 末尾に、「第 2220 条 E. OSS セキュリティの責任」を追加する。具体的には、CISA は主に以下の取組に責任を与える。 ➢ OSS に関する各省庁の取組の支援、OSS に関する公的な窓口、OSS の脆弱性情報の開示を支援することによるサプライチェーンのセキュリティ確保の支援を実行する。

区分	要求事項
	➢ 本条の制定日から 1 年以内に、OSS のリスクを評価するためのフレームワーク（OSS 内のコードのセキュリティ情報、OSS にパッチが適用されていない脆弱性の数と重大度等の評価観点や基準）を公開し、適宜更新する。 ➢ 上記のフレームワーク公開後、1 年以内に、各省庁が使用している OSS を対象にリスクを評価する。
ソフトウェア・セキュリティアドバイザリー委員会 （**Section 4**）	国土安全保障法第 2219 条(d)(1)（合衆国法典 Title. 6 §665 e(d)(1)）を以下のとおり改正すること。 ● サイバーセキュリティ問題に対処するための小委員会の議題として、「OSS セキュリティを含むソフトウェアセキュリティ」を追加する。
オープンソースソフトウェアガイダンス （**Section 5**）	● CISA は、本法案制定日から 1 年以内に、各省庁に向けた OSS に関するガイダンス（OSS の安全な利用や OSS への貢献をどのように実現するか等）を発行すること。 ● CISA によって選定された機関は、本法案制定日から 1 年以内に、OSS の安全な使用やリスク管理等を実行する試験的なオープンソースプログラムオフィス（OSPO）を設立すること。

2.1.8. CISA/NSA/ODNI : Securing Software Supply Chain Series - Recommended Practices for Developers

2022 年 9 月、CISA、NSA、ODNI は共同で、ソフトウェアサプライチェーンのセキュリティ保護に関するソフトウェア開発者向けのガイダンスを発表した。本ガイダンスは、ソフトウェア開発者、ソフトウェアサプライヤー（2.1.9 参照）、消費者（2.1.10 参照）の 3 つの役割ごとに焦点を当てた 3 部のガイダンスシリーズの内、第 1 部に該当する。本ガイダンスでは、安全なソフトウェアサプライチェーンを確保するためのソフトウェア開発者に対する推奨事項が整理されており、表 2-7 に示すとおり、サードパーティコンポーネントを含むソフトウェアの SBOM の作成、脆弱性の評価等が推奨されている。

表 2-7　ソフトウェア開発者に対する推奨事項の概要

区分	推奨事項概要
安全なソフトウェアの基準と管理	セキュリティに焦点を当てた原則とガイドラインを確保するためのポリシー（例：脅威モデル、セキュリティ検証計画、脆弱性対応方針・手順）を策定する。
セキュアコードの開発	攻撃者による悪意あるコードの組込み、開発者による脆弱なコードの組込みを防止するため、以下の事項を実施する。 ・ ソースコードの改変・窃取のリスクを軽減する ・ 使用する OSS を管理する

区分	推奨事項概要
	・ 安全な開発手法を実践する ・ 全ての依存関係の統合・可視化する ・ 脆弱性情報を取得し、製品との関連性を評価する ・ 安全な開発手法に基づき製品を拡張する（SBOM による拡張箇所の特定を含む）
サードパーティコンポーネントの検証	脆弱なサードパーティのソースコード・バイナリのソフトウェアへの組込みを防止するため、以下の事項を実施する。 ・ サードパーティコンポーネントへの構成解析を実施する ・ サードパーティコンポーネントのリスク評価を実施する ・ 信頼できるサプライヤーからコンポーネントを取得する ・ コンポーネントをメンテナンスする（例:脆弱性管理、脆弱性への対処） ・ SBOM を作成し、更新する
ビルド環境のハード化	ビルド環境における脆弱性の組込みや悪意のあるソフトウェアの組込みを防止するため、以下の事項を実施する。 ・ ビルド環境のセキュリティ対策を実施する（例:外部ネットワークからの保護、データ漏洩監視） ・ ソースコード署名によるセキュリティ対策を実施する
安全なソフトウェアの提供	配信システム（リポジトリー）におけるソフトウェア改変や悪意のあるソフトウェアの組込みを防止するため、以下の事項を実施する。 ・ 消費者への納品前に最終パッケージを検証する ・ リポジトリーへのアップロード前にパッケージを暗号化する ・ リポジトリーに対するセキュリティ対策を実施する

2.1.9. CISA/NSA/ODNI : Securing Software Supply Chain Series - Recommended Practices for Suppliers

2022 年 10 月、CISA、NSA、ODNI は共同で、ソフトウェアサプライチェーンのセキュリティ保護に関するソフトウェアサプライヤー向けのガイダンスを発表した。前述のとおり、本ガイダンスは、3 部のガイダンスシリーズの内、第 2 部に該当する。本ガイダンスでは、安全なソフトウェアサプライチェーンを確保するためのソフトウェアサプライヤーに対する推奨事項が整理されており、表 2-8 に示すとおり、サプライヤーは開発者と消費者の間の仲介役として、ソフトウェアの保護や脆弱性に関する対応・通知等が推奨されている。なお、本ガイダンスは、ソフトウェア開発者も想定読者として位置づけられており、ソフトウェアの保護や脆弱性への対応等が推奨されている。

表 2-8　ソフトウェアサプライヤーに対する推奨事項の概要

区分	推奨事項概要
安全なソフトウェアを提供するための組織の構築	安全なソフトウェアを提供するため、必要な基準やプロセスを確立し、実施する。また、基準やプロセスは、ソフトウェアサプライチェーンにおいて役割を担う全ての人が参照でき、認知している必要がある。なお、プロセスには、脆弱性対応やサポート期限に関する情報を消費者へ通知する内容を含める。
ソフトウェアの保護	不正アクセスや改ざんからソフトウェアを保護するため、以下の事項を実施する。 ・ 最小権限や認証によるアクセス制御によって、ソースコードへの不正アクセスを防止する ・ デジタル署名によって、ソフトウェアの完全性を担保する ・ ソフトウェアをバージョンごとにアーカイブする（災害復旧・フォレンジック調査にも有効）
安全性の高いソフトウェアの提供	自組織で開発又はサードパーティから取得したソフトウェアを安全性の高い状態で提供するため、以下の事項を実施する。 ・ セキュリティ要件を満たすソフトウェアを設計、開発する ・ サードパーティソフトウェアがセキュリティ要件に準拠していることを確認する ・ ソフトウェアのコンパイルとビルドのプロセスを適切に設定して、実行可能ファイルを保護する ・ 脆弱性の特定、セキュリティ要件への準拠を確認するため、ソースコードを解析する ・ 脆弱性の特定、セキュリティ要件への準拠を検証するため、実行可能ファイルをテストする ・ 標準でセキュアに設定する（アクセス制御等）
脆弱性への対応	公開されている既知の脆弱性が消費者へ提供されるソフトウェアに存在しないことを保証するため、ソフトウェアの脆弱性テストを実施し、必要に応じて脆弱性を取り除く。また、継続的に、脆弱性を特定、分析、除外した上で、セキュリティに関する情報を消費者へ通知する。

2.1.10. CISA/NSA/ODNI : Securing Software Supply Chain Series - Recommended Practices for Customers

2022 年 11 月、CISA、NSA、ODNI は共同で、ソフトウェアサプライチェーンのセキュリティ保護に関する消費者向けのガイダンスを発表した。前述のとおり、本ガイダンスは、3 部のガイダンスシリーズの内、第 3 部に該当する。本ガイダンスでは、安全なソフトウェアサプライチェーンを確保するための消費者（ソフトウェアを調達した組織）に対する推奨事項が整理されており、表 2-9 に示すとおり、ソフトウェア調達時のセキュリティに関する要求・評価、脆弱性の管理・報告等が推奨されている。

表 2-9　ソフトウェア消費者に対する推奨事項の概要

区分	推奨事項概要
ソフトウェアの調達	ソフトウェアの調達段階で脆弱性を軽減し、契約段階でサプライチェーンリスクを軽減するため、以下の事項を実施する。 ・ ソフトウェアセキュリティ、サプライチェーンリスク管理（SCRM）に関する活動をサプライヤーへ要求する ・ SBOM を基に脆弱性を評価し、RFI や RFP を基にソフトウェアセキュリティ、SCRM の要件を満たすか評価する ・ ソフトウェアサプライチェーンのセキュリティ管理・改善を可能にする契約上の合意を得て、サプライヤーによる安全性の自己適合証明（提出物には SBOM を含める）を要求する
ソフトウェアのデプロイメント	ソフトウェアのデプロイメント段階で、ソフトウェアのセキュリティリスクを軽減するため、以下の事項を実施する。 ・ ソフトウェアが調達前の評価と同一であるかを確認する（例：SBOM と取得したソフトウェアを照合する） ・ 機能テストを行い、構築したテスト環境は保存し、将来の検証に活用する ・ リスク評価結果に応じた対処を可能にするため、リスク管理プログラムを確立する ・ 構成（又は、変更）管理組織によって、ソフトウェアの構成・変更の管理を行う ・ テスト環境の多層防御、ログの管理等により、ソフトウェアの統合プロセスにおける脅威を軽減する ・ ソフトウェアの整合性を確保するための仕組み（設定ファイルのハッシュ等）をサプライヤー等から確認し、ソフトウェアの展開プロセスにおける脅威を軽減する ・ ソフトウェアのアップグレード計画を立てる（セキュリティ、運用テストに関する内容を含める） ・ EOL 等の製品利用終了計画を立てる（撤去中のセキュリティ管理に関する内容を含める） ・ トレーニングプログラム（主要なセキュリティ機能やリスクの内容を含める）を実施する
ソフトウェアの運用	ソフトウェアサプライチェーン上におけるセキュリティリスクを管理、特定、軽減するため、以下の事項を実施する。 ・ バグや異常を報告（サプライヤーへの通知を含む）する体制や仕組み（例：ヘルプデスク、SOC）を構築する ・ アップデート時に発生するセキュリティリスクに対処する（例：SBOM の自動更新） ・ ソフトウェアの潜在的なリスクを評価し、サプライヤーによるセキュリティに関する活動を可視化する等、SCRM を実施する

2.1.11. White House：Natinonal Cybersecurity Strategy

2023 年 3 月、White House は、全ての米国民において安全でセキュアなデジタルエコシステムによる利益を十分に確保することに向けて、新たな「国家サイバーセキュリティ戦略（National Cybersecurity Strategy）」を発表した。米国は複雑なサイバーリスクに直面しているとしており、サイバーリスクに対処し、本戦略を実行することで、米国におけるインフラの再建、クリーンエネルギー分野の開発、技術・製造基盤の再構築への投資が担保されると示している。また、同盟国や友好国とともに、デジタルエコシステムを構築することを目指している。なお、米国政府は、「国家安全保障戦略」、「サイバーセキュリティ強化のための大統領令（EO 14028）」等にて、サイバー空間における取組を進めている。本戦略はこれらの取組を発展させ、サイバー空間自体が存在することを目的とせず、米国が目指すデジタルエコシステムを実現するためのツールとしてサイバー空間が存在するものであると示している。

本戦略では、表 2-10 に示すとおり、5 つの柱を軸として、それぞれの柱に対して戦略目標が掲げられており、ソフトウェアセキュリティや SBOM に関しても言及されている。具体的には、連邦政府機関全体におけるサイバー攻撃からの防衛力強化に向けて、NIST と連携し、ソフトウェアサプライチェーンリスク軽減目標を構築する予定としている。なお、本目標は、SBOM に関する取組、NIST の「ソフトウェアの脆弱性を軽減するためのソフトウェア開発者向けの手法をまとめたフレームワーク（Secure Software Development Framework）」、OSS セキュリティ向上に関連する取組等を含み、EO 14028 に基づいて目標達成を目指すとしている。また、デジタルエコシステムのセキュリティとレジリエンスを構築するために、ソフトウェアの脆弱性によって生じるリスクに対する責任を規定する法律を再構築しなければならないとしている。加えて、安全なソフトウェア開発手法の採用を促進するため、連邦政府における協調的な脆弱性情報の開示や SBOM の導入を促進し、ソフトウェアが持つリスクを特定・軽減するためのプロセスを開発する予定としている。

本戦略を実現するため、Office of the National Cyber Director（ONCD）は各省庁と連携し、本戦略の実施計画を策定し、公表する予定としている。また、本戦略について、実施に向けた投資、進捗状況、最終成果を評価し、大統領や連邦議会等へ毎年報告するものとしている。

表 2-10　国家サイバーセキュリティ戦略における 5 つの柱と戦略目標の概要

柱	戦略目標概要
1. 重要インフラの防衛	重要インフラとインフラが提供するサービスの可用性とレジリエンスについて米国民に信頼感を与えるため以下の事項を実施する。 1. 国家安全保障と公共安全を支援するためのサイバーセキュリティ要件の確立 2. 官民協力の拡大 3. 連邦政府のサイバーセキュリティセンターの統合 4. 連邦政府のインシデント対応計画やプロセスの明確化 5. 連邦政府におけるシステムやサイバー攻撃に対する防衛の近代化
2. サイバー攻撃者の排除と解体	サイバー攻撃者が米国の国家安全保障や公共の安全を脅かすことができないようにするため、以下の事項を実施する。 1. サイバー攻撃者を排除するための活動の統合化

柱	戦略目標概要
	2. サイバー攻撃者をかく乱するための官民との協力強化 3. 情報共有と被害者通知の速度と規模の拡大 4. 米国拠点のインフラ悪用の防止 5. サイバー犯罪への対抗とランサムウェアの撲滅
3. セキュリティとレジリエンスを推進するための市場を形成	デジタルエコシステムをより信頼できるものにするに当たって、リスクを低減し、サイバーセキュリティの不備を起因とした不利益を回避するため、以下の事項を実施する。 1. データ管理者へ説明責任の付与 2. 安全な IoT 機器開発の促進 3. 安全でないソフトウェアやサービスに対する責任の再構築 4. 連邦政府の補助金やその他のインセンティブを利用したセキュリティの構築 5. 連邦政府の調達におけるサイバーセキュリティに関する説明責任の向上 6. 連邦政府によるサイバー保険市場の支援検討
4. レジリエンスな未来への投資	戦略的な投資と協調的な行動を通じ、安全でレジリエンスに優れた次世代技術やインフラの革新において、世界をリードし続けるため、以下の事項を実施する。 1. インターネットの技術的基盤の確保 2. サイバーセキュリティのための連邦政府研究開発の活性化 3. ポスト量子暗号の未来への備え 4. クリーンエネルギーの確保 5. デジタル ID を活用したソリューションやインフラによって構築されるデジタルエコシステムの開発支援 6. サイバー人材強化のための国家戦略策定
5. 国際的パートナーシップの構築	サイバー空間における国家の責任ある行動が期待され、無責任な行動にコストがかかるような世界を構築することに向けて、以下の事項を実施する。 1. デジタルエコシステムへの脅威に対抗するための連合体の構築 2. 国際的なパートナーの能力の強化 3. 同盟国やパートナーを支援するための能力向上 4. 責任ある国家行動の世界的規範を強化するための連合体構築 5. 情報・通信・運用技術に関する製品やサービスのための安全なグローバルサプライチェーンの確保

2.1.12. Food and Drug Administration（FDA）における動向調査

本項では、米国医療機器分野における SBOM に関する動向として、米国 Food and Drug Administration（以下「FDA」という。）における動向を示す。2022 年 4 月、FDA は、FDA の承認申請において医療機器メーカーが含めるべきサイバーセキュリティ対策に関する推奨事項を示した FDA-2021-D-1158 Cybersecurity in Medical Devices: Quality System Considerations

and Content of Premarket SubmissionsDraft（以下、「市販前ガイダンス案」という。）を公開し、パブリックコメントを 2022 年 7 月 7 日締め切りで実施した。市販前ガイダンス案では、医療機器に含まれる既製ソフトウェアコンポーネントに対して機械判読可能な SBOM を作成し、FDA に提出すること、既知の脆弱性を有するサードパーティのソフトウェアコンポーネントに対して、当該脆弱性に対するリスク評価結果とセキュリティ管理策を FDA に提出することを推奨しているほか、医療機器の品質システム規制（QSR[21]）において SBOM を活用可能であるとしている。

なお、FDA のガイダンス文書は、発行時点での FDA の考えや推奨事項を記述するものであるため、法的強制力はないが、規制当局から発表されるガイダンスであるため一定の影響力を有している。

FDA に承認申請を行う医療機器メーカーに対するソフトウェアコンポーネントに関連する推奨事項を以下表に示す。

表 2-11　FDA に承認申請を行う医療機器メーカーに対する推奨事項

区分	推奨事項概要
既製ソフトウェアコンポーネントに関する機械判読可能な SBOM の生成・提出	医療機器に含まれる既製ソフトウェアコンポーネントについて、以下の要素を SBOM に含めること。 ・当該ソフトウェアコンポーネントを含んでいる資産名称 ・当該ソフトウェアコンポーネントの名称 ・当該ソフトウェアコンポーネントのバージョン ・当該ソフトウェアコンポーネントの開発者 ・開発者により提供される保守・監視による当該ソフトウェアコンポーネントのサポートレベル ・当該ソフトウェアコンポーネントのサポート終了日 ・脆弱性データベースで公開されている当該ソフトウェアコンポーネントにおける既知の脆弱性 ・SBOM は、機械判読可能な形式とし、業界標準のフォーマットを利用すること。 FDA の審査・承認を支援するため、FDA に対して SBOM を提出すること。なお、上記の要素が SBOM に含まれていない場合、FDA による審査を効率化するために、その旨を付加的に FDA に提供すること。
サードパーティのソフトウェアコンポーネントに関するリスク評価結果・セキュリティ管理策の提出	・既知の脆弱性を有する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、各脆弱性の安全性及びセキュリティに関するリスク評価を行い、その情報を FDA に提出すること。 ・各脆弱性に対処するために適用されるセキュリティ管理策を詳細化し、その情報を FDA に提出すること。

また、FDA は 2022 年 11 月 15 日 に The Mitre Corporation[22]（以下、「MITRE」という。）

[21] https://www.ecfr.gov/current/title-21/chapter-I/subchapter-H/part-820
[22] https://www.mitre.org/

と共同で、医療機関がサイバーセキュリティインシデントに備えるためのリソースである「The Medical Device Cybersecurity Regional Incident Preparedness and Response Playbook[23]」を更新し、FDA サイトに公開した。この Playbook の主な更新内容は、以下の 3 点である。

- 臨床医、医療技術管理専門家、IT、緊急対応、リスク管理、施設スタッフ等、多様な職種によるチームがサイバーセキュリティへの備えと対応の演習に参加する必要性を強調。
- 地域対応におけるモデルやパートナーの活用が有効なサイバーセキュリティインシデントにおける広範囲な影響や長時間のダウンタイムに対する配慮を強調。
- 医療機器のサイバーセキュリティインシデント（ランサムウェア対策を含む）への準備と対応を支援するツール、参考資料、その他のリソースを容易に発見することができるリソースの付録を追記。

具体的には、MITRE サイトに米国 FDA との契約に基づき、MITRE が作成した以下の 2 文書を公開している[24]。なお、このプレイブックに含まれる見解、意見、発見は、機関のポリシー、ガイドライン、推奨事項、または法的強制力のある要件を構成するものではありません。

- Medical Device Cybersecurity Regional Incident Preparedness and Response Playbook[25]
- Medical Device Cybersecurity Regional Incident Preparedness and Response Playbook Quick Start Companion Guide[26]

2022 年 11 月に新たに改訂された Playbook は、医療機器のサイバーセキュリティインシデントに対処するための実践的な検討事項を提供している。そのためのツール、技術、リソースを特徴としており、医療機関等が医療機器を含むサイバーセキュリティインシデントに備え、対応することを支援することを目的としている。医療機関が SBOM を要求することにより、脆弱なデバイスコンポーネントを識別して対処できるようになる。また、組織内のデバイスインベントリ全体のトリアージと優先順位付けを可能にし、インシデントが発生した際により迅速に対応することを容易にし、インシデントレスポンス計画の開発に役立つとされている。

2.2. 欧州政府機関における取組動向等の調査

本事業で調査対象とした欧州政府機関における近年の取組動向等を表 2-12 に示す。以降では各取組の概要等について示す。

---

23 https://www.fda.gov/medical-devices/digital-health-center-excellence/cybersecurity
24 https://www.mitre.org/news-insights/publication/medical-device-cybersecurity-regional-incident-preparedness-and-response
25 https://www.mitre.org/sites/default/files/2022-11/pr-2022-3034-medical-device-cybersecurity-regional-preparedness-response-playbook.pdf
26 https://www.mitre.org/sites/default/files/2022-11/pr-2022-3616-medical-device-cybersecurity-regional-preparedness-response-companion-guide.pdf

表 2-12　欧州政府機関における取組動向等の調査対象

#	取組名・文書名	取組日・発行日	取組主体
2-1	Proposal for Cyber Resilience Act[27][28]	2022 年 9 月	EC
2-2	Mapping your supply chain[29]	2023 年 2 月	英国 NCSC

2.2.1. EC：Proposal for Cyber Resilience Act

欧州委員会は 2022 年 9 月、EU 市場に投入されるデジタル製品のセキュリティ対応を義務付ける「EU サイバーレジリエンス法（Cyber Resilience Act）」の草案を発表した。本法案では、デジタル製品を上市する際のルール、製品におけるサイバーセキュリティに関する要求事項、製造業者に課される脆弱性対応の要求事項、当該要求事項への順守を担保するための市場監督者へのルールを規定している。具体的な要求事項は表 2-13 に示すとおりであり、対象となるデジタル製品に対する要求事項として、リスクに応じた適切なレベルのサイバーセキュリティを確保するように設計・開発・生産することや、悪用可能な既知の脆弱性がない状態で提供することを求めている。また、製造業者に対する要求事項として、デジタル製品の脆弱性ハンドリングに関する要求事項、製品のセキュリティ評価に関する要求事項、利用者への情報提示に関する要求事項、製品の適合性評価に関する要求事項、脆弱性の報告義務に関する要求事項が規定されているほか、SBOM や OSS 管理に係る要求事項も規定されている。具体的には、製品のトップレベルの依存関係を網羅した機械判読可能な SBOM の作成を求めているほか、外部コンポーネントの脆弱性に関する情報共有手段の確立や使用しているコンポーネントに脆弱性が見つかった場合の報告についても要求している。なお、SBOM の様式や要素は、欧州委員会が実装法令によって規定できるとしている。

本法案の対象製品について、ソフトウェアやハードウェアを含む、他の製品やネットワークへの直接的・間接的な接続が存在するあらゆるデジタル製品が対象となるが、既存の EU 法により要求事項が課されている医療機器、自動車関連製品等は除外される。また、対象製品の上市に当たって、当該製品に対するセキュリティ要件への適合性証明（自己適合宣言若しくは第三者認証）が求められる。製造業者が本法案のセキュリティに関する要件を遵守にしない場合、最大 1,500 万ユーロ若しくは前会計年度の世界全体の総売上高の最大 2.5%のいずれか高い方が罰金として科される。

欧州委員会は、本法案の 2023 年後半の発効、2025 年後半（発効から 2 年後）の適用を目指しているが、製造業者における脆弱性の報告義務は 2024 年後半（発効から 1 年後）からの適用を位置づけている。

[27] https://www.fda.gov/medical-devices/digital-health-center-excellence/cybersecurity
[28] https://digital-strategy.ec.europa.eu/en/library/cyber-resilience-act
[29] https://www.ncsc.gov.uk/guidance/mapping-your-supply-chain

表 2-13　EU サイバーレジリエンス法（草案）における要求事項一覧

要求対象	区分	具体的な要求事項
デジタル製品に対する要求事項	デジタル製品に対するセキュリティ要求事項	(1) デジタル製品は、リスクに応じた適切なレベルのサイバーセキュリティを確保するように設計、開発、生産すること。 (2) デジタル製品は、悪用可能な既知の脆弱性がない状態で提供すること。 (3) 第 10 条（2）に言及されたリスクアセスメントに基づき、デジタル製品は、以下のようにしなければならない。 (a) デフォルトでセキュアな設定で提供され、製品を元の状態に戻すことが可能であること。 (b) 認証、ID、アクセス管理システムを含むがこれに限定されない適切な管理機構により、不正アクセスから確実に保護すること。 (c) 保存、送信、又はその他の方法で処理された個人又はその他のデータの機密性を、最新のメカニズムによって関連データを暗号化するなどして保護すること。 (d) 保存、送信、又はその他の方法で処理されたデータ、個人又はその他のデータ、コマンド、プログラム、設定の整合性を、ユーザーによって許可されていない操作又は修正から保護し、また破損について報告すること。 (e) 個人又はその他のデータを、適切かつ関連性のある、製品の使用目的に関連する必要なものに限定して処理すること（「データの最小化」）。 (f) サービス妨害（DoS）攻撃に対する回復力及び軽減を含む、重要な機能の可用性を保護すること。 (g) 他の機器やネットワークが提供するサービスの可用性に及ぼす自らの悪影響を最小限に抑えること。 (h) 外部インターフェースを含む攻撃面を制限するように設計、開発、製造すること。 (i) 適切な悪用防止メカニズムや技術を用いて、インシデントの影響を軽減するように設計、開発、製造すること。 (j) データ、サービス、機能へのアクセスや変更を含む、関連する内部活動を記録や監視することにより、セキュリティ関連情報を提供すること。 (k) 脆弱性について、セキュリティ更新（該当する場合、自動更新及び利用可能な更新のユーザーへの

要求対象	区分	具体的な要求事項
		通知を含む）を通じて対処されることを保証すること。
製造業者に対する要求事項	脆弱性ハンドリングに関する要求事項	デジタル製品の製造者は、以下のことを行わなければならない。 (1) 製品に含まれる脆弱性とコンポーネントを特定し、文書化すること。これには、少なくとも製品のトップレベルの依存関係を網羅する、一般的に使用され機械判読可能な形式のソフトウェア部品表を作成することが含まれる。 (2) デジタル製品にもたらされるリスクに関連して、セキュリティアップデートの提供を含め、脆弱性に遅滞なく対処し、改善すること。 (3) デジタル製品のセキュリティについて、効果的かつ定期的なテストとレビューを適用すること。 (4) セキュリティアップデートが提供された後、修正された脆弱性についての情報（脆弱性の説明、影響を受けるデジタル製品を特定できる情報、脆弱性の影響、深刻度、脆弱性を修正するための情報など）を一般に公開すること。 (5) 脆弱性の協調的な開示に関するポリシーを導入し、実施すること。 (6) デジタル製品及びその製品に含まれる第三者のコンポーネントの潜在的な脆弱性に関する情報の共有を促進するための手段を講じること。（デジタル製品で発見された脆弱性を報告するための連絡先を提供することを含む） (7) デジタル製品のアップデートを安全に配布し、悪用可能な脆弱性が適時に修正又は軽減される仕組みを提供すること。 (8) 特定されたセキュリティ問題に対処するためのセキュリティパッチやアップデートが利用可能な場合、それらが遅滞なく、かつ無料で配布されることを保証すること。
	製品に対するセキュリティ評価に関する要求事項	デジタル製品を市場に出す場合、製造業者は、その製品が附属書 I の第 1 節に定める要件（デジタル製品の特性に関するセキュリティ要求事項）に従って設計、開発及び製造されていることを確認しなければならない。 製造業者は、デジタル製品に関連するサイバーセキュリティリスクの評価を実施し、サイバーセキュリティリスクの最小化、セキュリティ事故の防止及び利用者の健康及び安全との関連性を含む当該事

要求対象	区分	具体的な要求事項
		故の影響の最小化を目的として、デジタル製品の計画、設計、開発、生産、引渡し及び保守の段階において当該評価の結果を考慮に入れなければならない。 デジタル製品を市場に出す場合、製造者は、第 23 条及び附属書 V に定める技術文書にサイバーセキュリティリスク評価を含めなければならない。特定の必須要件がデジタル製品に適用されない場合、製造者はその文書に明確な正当性を含めなければならない。 製造業者は、第三者から調達した構成部品をデジタル製品に統合する際、十分な注意を払わなければならない。製造業者は、そのようなコンポーネントがデジタル製品のセキュリティを損なわないことを保証しなければならない。 製造業者は、製品の性質及びサイバーセキュリティリスクに見合った方法で、デジタル製品に関する関連するサイバーセキュリティの側面（自らが認識する脆弱性及び第三者から提供される関連情報を含む）を体系的に文書化し、該当する場合には、製品のリスク評価を更新しなければならない。 デジタル製品を市場に出す場合、予想される製品寿命又は製品を市場に出してから 5 年間のいずれか短い期間、製造業者は、当該製品の脆弱性が必須要件（脆弱性ハンドリングに関する要求事項）に従って効果的に処理されることを保証しなければならない。 製造業者は、内部又は外部の情報源から報告されたデジタル製品の潜在的な脆弱性を処理し、是正するために、附属書 I の第 2 節（5）で言及されている協調的脆弱性開示方針を含む適切な方針及び手順を有していなければならない。
	利用者への情報提示に関する要求事項	・ 製造業者は、デジタル製品に、附属書Ⅱに規定する情報及び指示を電子的又は物理的形態で添付することを確実にしなければならない。当該情報及び指示は、利用者が容易に理解できる言語でなければならない。それらは、明確で、理解しやすく、分かりやすく、読みやすいものでなければならない。また、デジタル技術を用いた製品の安全な設置、操作、使用を可能にしなければならない。 ・ 製造業者は、EU 適合性宣言をデジタル製品に添付するか、EU 適合性宣言にアクセスできるインターネットアドレスを付属書 II に定める指示と情報に含めなければならない。
	セキュリティ要件の適	・ デジタル製品の上市から予想される製品寿命又は上市後 5 年間のいずれか短い期間、デジタル製品

要求対象	区分	具体的な要求事項
	合性の維持に関する要求事項	又は製造者が実施したプロセスが付属書 I（サイバーセキュリティの必須要件）に定める必須要件に適合していないことを知っているか、適合していないと理由できる製造業者は、デジタル技術を用いた製品又は製造者のプロセスを適合させるために必要な修正措置を直ちに講じ、必要に応じて製品を撤回するか回収しなければならない。 ・ 製造業者は、市場監視当局からの理由ある要請があれば、同当局が容易に理解できる言語で、デジタル製品及び製造事業者が実施するプロセスが附属書 I に定める必須要件に適合していることを示すために必要な全ての情報及び文書を紙又は電子形式で提供しなければならず、その要請に応じて、市場に投入したデジタル製品がもたらすサイバーセキュリティ上のリスクを排除するために講じた措置について同局と協力しなければならない。 ・ 事業を停止し、その結果、本規則に定める義務を遵守することができなくなった製造業者は、事業の停止が効力を発する前に、関連する市場監視当局に状況を通知するとともに、利用できるあらゆる手段で、可能な限り、市場に出されたデジタル製品の使用者にもこの状況を通知しなければならない。
	脆弱性の報告義務	・ 製造業者は、過度の遅滞なく、いかなる場合においても、認識してから 24 時間以内に、デジタル製品に含まれる積極的に悪用される脆弱性を ENISA に通知しなければならない。通知には、当該脆弱性に関する詳細、及び、該当する場合には講じられた是正又は軽減措置が含まれるものとする。 ・ 製造業者は、デジタル要素を有する製品のセキュリティに影響を及ぼす事象について、過度の遅滞なく、いかなる場合であってもその認識から 24 時間以内に ENISA に通知しなければならない。 ・ 製造業者は、過度の遅滞なく、かつ、認識した後に、デジタル製品の利用者に対して、事故について、また必要に応じてユーザーが事故の影響を軽減するために展開できる是正措置について通知しなければならない。 ・ 製造業者は、オープンソースコンポーネントを含む、デジタル製品に組み込まれたコンポーネントの脆弱性を特定した場合、そのコンポーネントを保守する個人又は団体に報告しなければならない。
	罰則	・ 附属書 I に規定されたサイバーセキュリティの必須要件及び第 10 条と第 11 条に規定された義務（セ

要求対象	区分	具体的な要求事項
		キュリティ要件の対応やセキュリティに関する報告等）を遵守しない場合、最高 1,500 万ユーロ、又は違反者が事業者の場合、前会計年度の全世界の年間総売上高の 2.5%のいずれか高い方の行政罰の対象となる。 ・ 本規則に基づくその他の義務に違反した場合、最高 1,000 万ユーロ、又は違反者が事業者の場合は、前会計年度の全世界の年間総売上高の 2%のいずれか高い方の行政罰が課される。 ・ 要求に対する回答として、不正確、不完全又は誤解を招く情報を通知機関及び市場監視当局に提供した場合、5 百万ユーロ又は違反者が事業者の場合、前会計年度の全世界の年間総売上高の 1%以下のいずれか高い方の行政罰の対象となる。
EU 加盟国関係機関に対する要求事項	EU 加盟国に対する要求事項	・ 加盟国は、この規則に従って適合性評価を実施する権限を有する適合性評価機関を欧州委員会及び他の加盟国に通知しなければならない。 ・ 加盟国は、適合性評価機関の評価及び届出並びに届出機関の監視（第 31 条の遵守を含む）のために必要な手続を設定し実施する責任を負う届出機関を指定しなければならない。 ・ 加盟国は、適合性評価機関の審査及び届出、届出機関の監視に関する自国の手続き並びにその変更について欧州委員会に報告しなければならない。
	欧州委員会に対する要求事項	・ 欧州委員会は、附属書 I の第 2 節（1）に定めるソフトウェア部品表の様式及び要素を実装法令によって規定することができる。これらの実装法令は、第 51 条（2）にいう審査手続に従って採択されなければならない。 ・ 欧州委員会は、実装法令によって、11 条 1 及び 2 に従って提出される届出の情報の種類、様式及び手続をさらに定めることができる。これらの実装法令は、第 51 条（2）にいう審査手続に従って採択されるものとする。
	ENISA に対する要求事項	・ ENISA は、サイバーセキュリティリスクに関連する正当な理由がない限り、過度の遅延なく、製造者から脆弱性通知を受領次第、当該加盟国の指令（NIS2 指令）に従い、協調的脆弱性開示を目的として指定された CSIRT に転送し、市場監視当局に通知した脆弱性について通知するものとする。 ・ ENISA は、サイバーセキュリティリスクに関連する正当な理由がない限り、製造者からデジタル要

要求対象	区分	具体的な要求事項
		素を有する製品のセキュリティに影響を及ぼす事象を受領次第、過度の遅滞なく、NIS2 指令に従って指定された単一の連絡先に通知を転送するものとする。 ・ ENISA は、通知されたインシデントについて市場監視当局に通知する。インシデントの通知にはインシデントの重大性と影響に関する情報を含め、該当する場合、製造者がインシデントを不法行為又は悪意ある行為によるものと疑っているか、又は国境を越えた影響があるとみなしているかを示すものとする。 ・ ENISA は、NIS2 指令によって設立された欧州サイバー危機連絡組織ネットワーク（EU-CyCLONe）に、11 条 1 及び 2 に従って通知された情報が、大規模サイバーセキュリティ事件及び危機の運用レベルでの協調管理に関連している場合、提出しなければならない。 ・ ENISA は、11 条 1 及び 2 に従って受領した通知に基づいて、デジタル要素を有する製品におけるサイバーセキュリティリスクに関する新たな傾向について 2 年ごとの技術報告書を作成し、NIS2 指令で言及されている協力グループに提出するものとする。最初の当該報告書は、11 条 1 及び 2 に規定する義務の適用開始後 24 か月以内に提出するものとする。

2.2.2. 英国 NCSC：Mapping your supply chain

2023 年 2 月、英国 NCSC は、「サプライチェーンマッピングのガイダンス（Mapping your supply chain）」を発表した。サプライチェーンマッピング（SCM：Supply Chain Mapping）とは、ソフトウェアサプライチェーンに関わるサプライヤーから収集した情報を記録、保存、活用するプロセスのことである。本ガイダンスの対象は、サプライヤーから供給されたソフトウェアの脆弱性が対処されていることを確認・保証する必要がある、中規模から大規模の組織としている。SCM を導入することでサプライヤーが誰で、何を、どのように供給しているかを理解し、発生し得るサイバーセキュリティのリスクをマネジメントするのに役立つとしている。また、サイバーリスクに関する情報を含む、より多くの情報に基づいたビジネス上の意思決定（例：サイバーインシデント時の対応、法律や規制への準拠）を行うことが可能になる。

SCM の実施に当たっては、サプライヤーや供給されたソフトウェアに関する情報を一貫した方法で収集し、アクセス管理された一元的なリポジトリーに保存することで、分析と管理が容易になる。既存のサプライヤーに関する情報は、既存の調達システムに存在している場合があるほか、各サプライヤーの情報を取得できた場合でも、サプライヤーに関するエントリーポイントが複数で、関連情報を集約する必要がある場合がある。これらのことを踏まえて、既存のサプライヤーに関する情報の照合やサプライヤーに関する情報を最新に保つ等の機能を有する商用ツールを検討することが有益であるとしている。また、サプライヤーの下請け業者も把握すべきか、何を重要として把握すべきか等のサプライチェーンにおける把握すべき範囲という観点においては、情報取得の必要性とコストを踏まえて、事前に検討する必要があるとしており、サプライヤーとの契約においては、サイバー攻撃に対するインシデント対応能力や対応におけるサポート時間、サプライヤーの監査等に関する条件を課すことを考慮する必要があるとしている。

SCM を導入する際の手段について、組織における調達とリスクマネジメントのプロセス、利用可能なツールに依存することになる。ただし、初めて導入する組織は、調達システムなどの既存システムを利用した既知のサプライヤーリストの作成、組織にとって重要なサプライヤー、システム、製品、サービスの優先順位付け等を最優先事項として取り組むべきであるとしている。

2.3. 企業、業界団体等におけるソフトウェア管理の取組動向等の調査

本調査項目では、企業等において自社製品・サービスの一部として利活用するソフトウェアに既存の脆弱性が存在せず、安全なソフトウェアであることを確認するための取組や、当該目的を達成するための製品・サービスの最近の動向について調査した。調査結果を表 2-14 に示す。

表 2-14　企業等におけるソフトウェア管理の取組動向等の調査結果

#	取組タイトル	取組主体	公表時期	概要
1	「情報システムにおけるセキュリティコントロールガイドライン Ver.1.0」の公表[30]	Software-ISAC	2022 年 5 月	・ システム運用のライフサイクルを通じて必要なセキュリティの管理策をまとめたガイドライン。 ・ SBOM に関して、アプリケーションの SBOM を作成し、定期的に更新することの重要性が明記されているほか、サプライチェーン全体での SBOM 管理の重要性も触れられている。
2	業界と政府のリーダーを集めた OSS に関するサミット（Open Source Security Summit Ⅱ）の開催[31]	Linux Foundation、OpenSSF	2022 年 5 月 12 日	・ 2022 年 1 月に開催された米国国家安全保障会議（NSC）が主導した第 1 回の Summit に続くもので、OSS とソフトウェアサプライチェーンに関するセキュリティを向上するための計画に関して議論・合意を行った。 ・ 37 社の企業、CISA、NIST、OMB 等の政府関係機関が参加。 ・ 本 Summit の参加企業の一部は、合意された計画の実施に向けた資金提供を誓約した。具体的には、Amazon、Ericsson、Google、Intel、Microsoft、VMWare の企業が、合計 3,000 万ドル以上の資金提供を誓約した。

[30] Software-ISAC、情報システムにおけるセキュリティコントロールガイドライン Ver.1.0 http://www.softwareisac.jp/HP/data/SecurityCtrlGuidelineV1.pdf

[31] Linux Foundation, Linux Foundation と Open Source Security Foundation (OpenSSF)、業界と政府のリーダーを集めたオープンソースソフトウェアセキュリティサミット II を開催 https://www.linuxfoundation.jp/press-release/2022/05/linux-foundation-openssf-gather-industry-government-leaders-open-source-software-security-summit/

#	取組タイトル	取組主体	公表時期	概要
3	OSS のセキュリティ強化に向けた計画（Open Source Software Security Mobilization Plan）の公開[32]	Linux Foundation、OpenSSF	2022 年 5 月 13 日	・ Open Source Security Summit II で合意された文書で、OSS のソフトウェアサプライチェーンのセキュリティ向上に向けた計画書。 ・ 「セキュアな OSS の作成」、「脆弱性の発見と修正方法の強化」、「ソフトウェアエコシステム全体へのパッチ適用のための時間短縮」の目標を実現するための計画が打ち出されている。 ・ 計画では SBOM に関して言及されており、SBOM ツールの導入、SBOM に関する教育・啓発、SBOM の利用を促進するための計画が示されている。
4	SBOM 作成ツール（sbom-tool）をオープンソースとして公開[33]	Microsoft	2022 年 7 月 12 日	・ OSS として公開された SBOM を作成するツール。 ・ Windows、Linux、macOS 等のプラットフォームで動作する。また、様々なパッケージ管理システムと統合することで、コンポーネントの自動検出を可能としている。 ・ 作成された SBOM は、SPDX 形式に基づく 4 つの主要なセクション（SBOM ドキュメント情報、ファイル情報、パッケージ情報、リレーションシップ情報）を含んでいる。また、完全な依存関係を把握するために、他の SBOM を参照することが可能となっている。

[32] Linux Foundation, The Open Source Software Security Mobilization Plan https://openssf.org/oss-security-mobilization-plan/
（日本語翻訳版）https://www.linuxfoundation.jp/publications/2022/08/oss_security_mobilization_plan/

[33] Microsoft, Microsoft open sources its software bill of materials (SBOM) generation tool https://devblogs.microsoft.com/engineering-at-microsoft/microsoft-open-sources-software-bill-of-materials-sbom-generation-tool/

#	取組タイトル	取組主体	公表時期	概要
5	業界と政府のリーダーを集めた OSS に関する日本向けサミット（Open Source Security Summit Japan）の開催[34]	Linux Foundation、Open Source Software Security Foundation（OpenSSF）	2022 年 8 月 23 日	・ 2022 年 5 月にホワイトハウスで開催された Summit に続くもので、OSS のセキュリティ課題とこれを向上する方法に関して議論を行った。 ・ 国内からは、日立製作所、富士通、トヨタ等の企業のほか、経済産業省、IPA、JPCERT/CC の政府関係機関も参加。
6	OSS に対象を絞ったバグ報奨金プログラム（Open Source Software Vulnerability Rewards Program）の構築[35]	Google	2022 年 8 月 30 日	・ OSS のソフトウェアサプライチェーンにおけるセキュリティを確保するため、OSS に存在する脆弱性やバグを発見した者に対して報奨金が支払われるプログラムを開始した。 ・ 脆弱性の重要度や OSS プロジェクトの重要性に応じて、報奨金は 100 ドル～31,337 ドルの範囲とされている。
7	オープンソースプロジェクト向け脆弱性スキャナ（OSV-Scanner）の公開[36]	Google	2022 年 12 月 13 日	・ OSS 開発者が、開発した OSS に関連する脆弱性情報を取得するための無償ツール。 ・ OSS に含まれるコードや依存関係を既知脆弱性と照合し、パッチやバージョンの更新が必要かどうかを開発者へ通知する機能を持つ。

[34] Linux Foundation, Linux Foundation と OpenSSF、Open Source Security Summit Japan を開催 https://www.linuxfoundation.jp/press-release/2022/08/linux-foundation-openssf-gather-industry-government-leaders-open-source-software-security-summit-japan/

[35] Google, Announcing Google's Open Source Software Vulnerability Rewards Program https://security.googleblog.com/2023/08/Announcing-Googles-Open-Source-Software-Vulnerability-Rewards-Program%20.html

[36] Google, Announcing OSV-Scanner: Vulnerability Scanner for Open Source https://security.googleblog.com/2022/12/announcing-osv-scanner-vulnerability.html

2.4. OSS を含むソフトウェアの安全な利活用に関する調査

2.4.1. ソフトウェアに関連するインシデント・脆弱性事例の調査

本調査項目では、OSS を含むソフトウェアの安全な利活用の必要性について理解を助けるための基礎資料として、ソフトウェアの安全を脅かす要因やその対処方法、OSS 関連の最近の動向等、特にソフトウェアの脆弱性事例やインシデント事例について、公開情報を中心に収集した。調査結果を表 2-15 に示す。そして、調査結果について、ソフトウェアタスクフォースで紹介することを前提として取りまとめた。本事業では、表 2-15 に示す事例の内、No. 6 の事例についてソフトウェアタスクフォースの資料として取りまとめた。

表 2-15　ソフトウェアに関連するインシデント・脆弱性事例

#	事例タイトル	事例 報告日	概要
1	VMware 製品における脆弱性：CVE-2022-22954、CVE-2022-22960	2022 年 4 月	・ 2022 年 4 月に VMware の 5 つのサービスにおける脆弱性が発表された。悪用された場合、遠隔からのコード実行や root への特権昇格が行われる可能性がある。悪用可能性の高い脆弱性であり、高い CVSS がつけられている。 ・ 本脆弱性が公表された 48 時間以内に攻撃コードが作成され、積極的な攻撃が行われていることが CISA より発表されている。そのため、VMware や各国当局においても繰り返し注意とアップデートの実施を呼びかけている。APT などの犯罪グループにおける利用が判明され、対象製品を利用している全ての連邦政府機関に対して、脆弱性に対処するよう緊急指令を CISA より発出した。

#	事例タイトル	事例 報告日	概要
2	Linux に権限昇格につながる脆弱性群：Nimbuspwn[37]	2022 年 4 月 26 日	・ 本脆弱性群は、Linux のデスクトップ環境において使用されるコンポーネントである networkd-dispatcher におけるディレクトリトラバーサル、シンボリックリンクの競合、Time of check to time of use の競合等の複数の脆弱性である。悪用された場合、Linux デスクトップのエンドポイントにおいて root 権限へ昇格され、マルウェアやランサムウェア感染等を実行される可能性がある。 ・ 本脆弱性は Microsoft により発見された。Microsoft は、当該脆弱性群を発見後、networkd-dispatcher のメンテナンスを行う開発者へ開示した。その後、開発者は当該脆弱性群を修正したバージョンを公開し、networkd-dispatcher の利用者に対して、最新版へアップデートすることを促した。
3	Microsoft Support Diagnostic Tool に関するゼロデイ脆弱性：Follina（CVE-2022-30190）[38]	2022 年 5 月 30 日	・ 本脆弱性は、Word 等の Microsoft Office 文書のアプリケーションから Microsoft Support Diagnostic Tool を呼び出す場合にリモートで任意コードが実行される脆弱性である。悪用された場合、呼び出し元のアプリケーションの権限で任意コードが実行されるほか、プログラムのインストール、データの表示、変更、削除、アカウント作成が実行される可能性がある。 ・ Microsoft は、当該脆弱性を修正したバージョンを 2022 年 6 月 15 日に公開し、利用者に対して、脆弱性対応のための迅速なアップデートを促した。

[37] Microsoft, Microsoft finds new elevation of privilege Linux vulnerability, Nimbuspwn https://www.microsoft.com/en-us/security/blog/2022/04/26/microsoft-finds-new-elevation-of-privilege-linux-vulnerability-nimbuspwn/

[38] Microsoft,Guidance for CVE-2022-30190 Microsoft Support Diagnostic Tool Vulnerability https://msrc.microsoft.com/blog/2022/05/guidance-for-cve-2022-30190-microsoft-support-diagnostic-tool-vulnerability/

#	事例タイトル	事例 報告日	概要
4	Atlassian Confluence Server に RCE の脆弱性：CVE-2022-26134	2022 年 6 月 3 日	・ Atlassian Confluence Server と Atlassian Confluence Data Center にリモートで任意のコードが実行される脆弱性が発見された。脆弱性の悪用により、資格情報やデータの窃取につながるおそれがある。 ・ 本脆弱性は、CISA の Known Exploited Vulnerability Catalog に登録されており、既に攻撃に利用されていることが判明している。また、米国の CISA より、政府機関に対して Atlassian Confluence Server と Atlassian Confluence Data Center へのネットワークを遮断することを求めている。
5	Microsoft Exchange Server のゼロデイのリモートコード実行の脆弱性：ProxyNotShell [39]	2022 年 9 月	・ 本脆弱性は、Microsoft Exchange Server における脆弱性であり、悪用された場合、サーバーサイドリクエストフォージェリ攻撃を仕掛けられることで、リモートから任意コードを実行される可能性がある。ただし、悪用するには、攻撃者は標的対象の Microsoft Exchange Server へのアクセス権が必要となる。 ・ CISA は当該脆弱性が悪用されたことを確認したとして、「既知の悪用された脆弱性カタログ（Known Exploited Vulnerablities Catalog）」に追加し、注意を促した。 ・ Microsoft は、当該脆弱性を修正したセキュリティ更新プログラムを 2022 年 11 月 8 日に公開し、利用者に対して、脆弱性対応のための迅速な更新プログラムの適用を促した。

[39] Microsoft,Customer Guidance for Reported Zero-day Vulnerabilities in Microsoft Exchange Server https://msrc.microsoft.com/blog/2022/09/customer-guidance-for-reported-zero-day-vulnerabilities-in-microsoft-exchange-server/

#	事例タイトル	事例 報告日	概要
6	OpenSSL の脆弱性：CVE-2022-3602、CVE-2022-3786[40]	2022 年 10 月 25 日	・ 本脆弱性は、OpenSSL の X.509 証明書の検証処理に起因する脆弱性であり、悪用された場合、システムがクラッシュする可能性があるほか、環境によってはリモートから任意コードを実行される可能性がある。ただし、これらの問題は証明書チェーンの署名検証の後で発生するため、悪意のある証明書に対して認証局（CA）が署名し、その後の処理が継続してしまう場合においてのみ悪用可能であるため、悪用可能性は限定的である。 ・ OpenSSL Project は、当該脆弱性を修正した「OpenSSL 3.0.7」を 2022 年 11 月 1 日に公開し、利用者に対して、脆弱性対応のための迅速なアップデートを促した。
7	Microsoft Exchange Server のゼロデイのリモートコード実行の脆弱性：CVE-2022-41080[41]	2022 年 11 月	・ 本脆弱性は、Microsoft Exchange Server における権限昇格の脆弱性であり、ProxyNotShell の脆弱性の軽減策（URL 書き換えモジュールの設定）を回避する。悪用されることで、Outlook Web Access（OWA）を介してサーバーサイドリクエストフォージェリ攻撃が仕掛けられ、リモートから任意コードを実行される可能性がある。 ・ CISA は当該脆弱性が悪用されたことを確認したとして、「既知の悪用された脆弱性カタログ（Known Exploited Vulnerablities Catalog）」に追加し、注意を促した。 ・ 当該脆弱性を発見した CrowdStrike は、Microsoft が 2022 年 11 月 8 日に公開したセキュリティ更新プログラムを適用した場合には当該脆弱性が悪用されることはないことを確認しており、セキュリティ更新プログラムの適用を促した。また、未適用の場合は、OWA を無効化することを促した。

[40] JPCERT/CC, OpenSSL の脆弱性（CVE-2022-3602、CVE-2022-3786）に関する注意喚起
https://www.jpcert.or.jp/at/2022/at220030.html

[41] NIST, NATIONAL VULNERABILITY DATABASE CVE-2022-41080 Detail
https://nvd.nist.gov/vuln/detail/CVE-2022-41080,
CrowdStrike, OWASSRF: CrowdStrike Identifies New Exploit Method for Exchange Bypassing ProxyNotShell Mitigations
https://www.crowdstrike.com/blog/owassrf-exploit-analysis-and-recommendations/

#	事例タイトル	事例 報告日	概要
8	Google Chrome におけるシンボリックリンクフォローの脆弱性：CVE-2022-3656[42]	2022 年 11 月 1 日	・ 本脆弱性は、Google Chrome や Chromium ベースの Web ブラウザにおけるファイルシステムの検証処理に起因する脆弱性で、悪用された場合、リモートの攻撃者によって作成された偽の Web サイトを介し、暗号通貨ウォレットやクラウドサービスの認証情報等が含まれる機密ファイルが窃取される可能性がある。当該脆弱性を発見した Imperva は、PoC を作成し、機密ファイルを窃取することが可能であることを確認した。 ・ Imperva は、当該脆弱性について Google へ情報を開示した。Google は、当該脆弱性を修正したバージョンを公開し、利用者に対して、最新版へアップデートすることを促した。
9	FortiOS におけるゼロデイ脆弱性：CVE-2022-42475[43]	2022 年 12 月 12 日	・ 本脆弱性は、FortiOS SSL-VPN におけるヒープベースのバッファーオーバーフローの脆弱性であり、悪用された場合、認証されていない第三者によって、悪意のあるリクエストが送信され、リモートから任意コードが実行される可能性がある。Fortinet は、本脆弱性を悪用する攻撃を確認したことを発表した。 ・ Fortinet は、当該脆弱性を修正したバージョンを公開し、利用者に対して、脆弱性対応のための迅速なバージョンアップを促した。

[42] NIST, NATIONAL VULNERABILITY DATABASE CVE-2022-3656 Detail
https://nvd.nist.gov/vuln/detail/CVE-2022-3656,
Imperva, Google Chrome “SymStealer” Vulnerability: How to Protect Your Files from Being Stolen
https://www.imperva.com/blog/google-chrome-symstealer-vulnerability/

[43] JPCERT/CC, FortiOS のヒープベースのバッファーオーバーフローの脆弱性（CVE-2022-42475）に関する注意喚起
https://www.jpcert.or.jp/at/2022/at220032.html

#	事例タイトル	事例 報告日	概要
10	Citrix ADC 及び Citrix Gateway における脆弱性：CVE-2022-27518[44]	2022 年 12 月 13 日	・ 本脆弱性は、Citrix ADC 及び Citrix Gateway における脆弱性であり、悪用された場合、認証されていない第三者によって、リモートから任意コードが実行される可能性がある。NSA は、本脆弱性に関するガイダンスを公開し、本脆弱性を悪用する攻撃が確認されていること、攻撃に対する被害状況の調査方法や被害を最小化するための対処方法を示した。 ・ Citrix は、当該脆弱性を修正したバージョンを公開し、利用者に対して、脆弱性対応のための迅速なバージョンアップを促した。
11	機械学習フレームワーク PyTorch に不正プログラム混入のおそれ[45]	2022 年 12 月 31 日	・ 本脆弱性は、機械学習フレームワークである PyTorch における脆弱性で、2022 年 12 月 25 日から 12 月 30 日の期間に、PyTorch の nightly 版をパッケージマネージャの pip 経由でインストールした場合、悪意あるプログラムがインストールされるおそれがあった。 ・ PyTorch の開発チームによると、悪意あるプログラムは、依存関係にあるパッケージの torchtriton と同名で、ソースコードリポジトリーの PyPI 上に公開されていた。依存関係にあるパッケージは、PyPI 上のプログラムが優先される仕様であるため、悪意あるプログラムが優先的にインストールされるおそれがあった。 ・ PyTorch の開発チームは、脆弱性が存在した期間中に PyTorch の nightly 版をインストールした利用者に対して、PyTorch 及び torchtriton をアンインストールし、2022 年 12 月 30 以降に公開した最新版を利用するよう促した。

[44] JPCERT/CC, Citrix ADC 及び Citrix Gateway の脆弱性（CVE-2022-27518）に関する注意喚起 https://www.jpcert.or.jp/at/2022/at220033.html

[45] PyTorch,Compromised PyTorch-nightly dependency chain between December 25th and December 30th, 2022 https://pytorch.org/blog/compromised-nightly-dependency/#how-to-check-if-your-python-environment-is-affected

2.4.2. 脆弱性管理手法ツールに関する調査

SBOM の利活用に向けて、ソフトウェア、ソフトウェア部品の脆弱性管理に関する調査を行った。SBOM は、ソフトウェアの構成要素のみを示しているものであり、開発支援・透明性の確保・効率的な脆弱性管理など様々な目的で SBOM を利活用する必要がある。また、SBOM は機械判読な形式であるため、利活用の自動化も注目されている。本章では、脆弱性管理における SBOM の活用や SBOM による自動化に向けて、脆弱性対応やポリシー確認の自動化に既に利用されている SCAP と SCAP の取組事例を紹介すると共に、SBOM・VEX との関係性を示す。

2.4.2.1. Security Content Automation Protocol（SCAP）

SCAP とは、情報セキュリティ対策の自動化と標準化を目的として、NIST において開発された技術仕様である。SCAP は、6 つの標準仕様から構築され、現在は脆弱性管理や脆弱性スキャンツールなどで幅広く活用されている。SCAP を利用した代表的なスキャンツールとして、OpenSCAP が挙げられる。また、世界的な脆弱性データベースである National Vulnerability Database（NVD）においても利用されている。脆弱性の識別子である CVE を付けられる組織は、世界各国で 220 程度の組織が存在する。

米国政府では SCAP が特に幅広く利用されている。連邦政府の IT 製品のベース基準である United States Government Configuration Baseline（USGCB）や DoD/DISA のサイバーセキュリティ基準である Security Technical Implementation Guides（STIGs）などでは、OS やソフトウェアの設定の基準への準拠確認を自動化するために SCAP が用いられている。(2022 年 12 月時点で、602 件の SCAP ファイルが NIST の National Checklist Program で公開されている。)

表 2-16　SCAP における 6 つの標準仕様

標準仕様名	概要
共通脆弱性識別子（CVE）	個別製品中の脆弱性に一意の識別番号
共通セキュリティ設定一覧（CCE）	コンピュータのセキュリティ設定項目に対する一意の識別番号（例：パスワードの長さ等）
共通プラットフォーム一覧（CPE）	ハードウェア、ソフトウェアに対する名称基準（例：Microsoft Office 等）
共通脆弱性評価システム（CVSS）	脆弱性の深刻度を評価するためのシステム
セキュリティ設定チェックリスト記述形式（XCCDF）	セキュリティチェックリストやベンチマークなどを記述するためのフォーマット
セキュリティ検査言語（OVAL）	コンピュータのセキュリティ設定状況を検査するための仕様（XML 言語）

2.4.2.2. SCAP の取組例

2.4.2.2.1. National Vulnerability Database（NVD）

National Vulnerability Database とは、SCAP を基に脆弱性を整理しているデータベースである。NVD では、CVE 番号を付随して脆弱性を管理するだけでなく、脆弱性毎に CVSS を評価している。NVD は、CVE 番号を付けられている脆弱性をデータベース化しており、API 等の自動化の手助けとなるシステムも提供している。NVD では、全世界の脆弱性が集められ、ほぼ全ての脆弱性情報にアクセスすることができる。

2.4.2.2.2. JVN iPedia

JVN iPedia とは、日本で使用されているソフトウェアなどの脆弱性対策情報や国外で公開される脆弱性対策情報をまとめたデータベースである。JVN iPedia は、JPCERT/CC と IPA が運用している。JVN iPedia においても、NVD と同じく SCAP に則り脆弱性がまとめられている。JVN iPedia では、CVE での管理だけでなく、JVNDB という独自の ID も利用して管理されている。また、JVN iPedia を利用した「MyJVN バージョンチェッカ」というサービスでは、PC 内のソフトウェアのバージョンを XCCDF や OVAL 等の SCAP ファイルを利用して自動で確認している。

図 2-11　MyJVN バージョンチェッカのイメージ図[46]

2.4.2.2.3. OpenSCAP

OpenSCAP とは、SCAP の標準仕様に従い脆弱性の診断を自動で行うオープンソースのソ

[46] MyJVN バージョンチェッカ for.NET、https://jvndb.jvn.jp/apis/myjvn/vccheckdotnet.html

フトウェアである。OpenSCAP は LinuxOS（Red Hat, Fedora, CentOS 等）において広く活用され、LinuxOS の標準ライブラリとして OpenSCAP が既に組み込まれている。OpenSCAP は、セキュリティポリシー遵守の自動確認・脆弱性アセスメントの自動化の 2 点を主な利用用途としている。OpenSCAP の自動化は、SCAP における OVAL や XCCDF という標準仕様を用いて実現している。OpenSCAP は、具体的に表 2-17 に示した 5 つのツールで構成されている。STIGs や USGCB への準拠を確認するツールとしても利用されている。

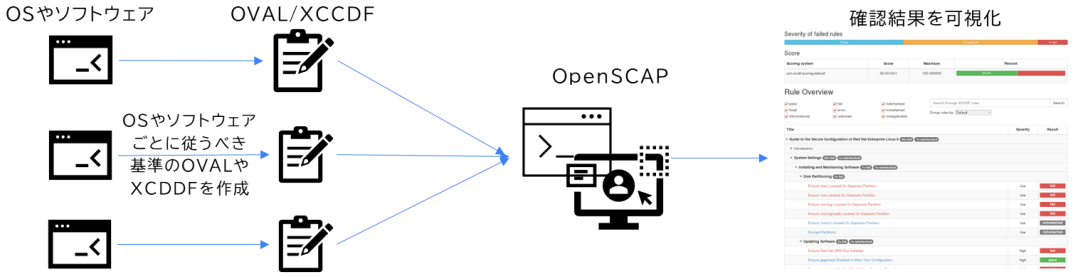


図 2-12　OpenSCAP の概要

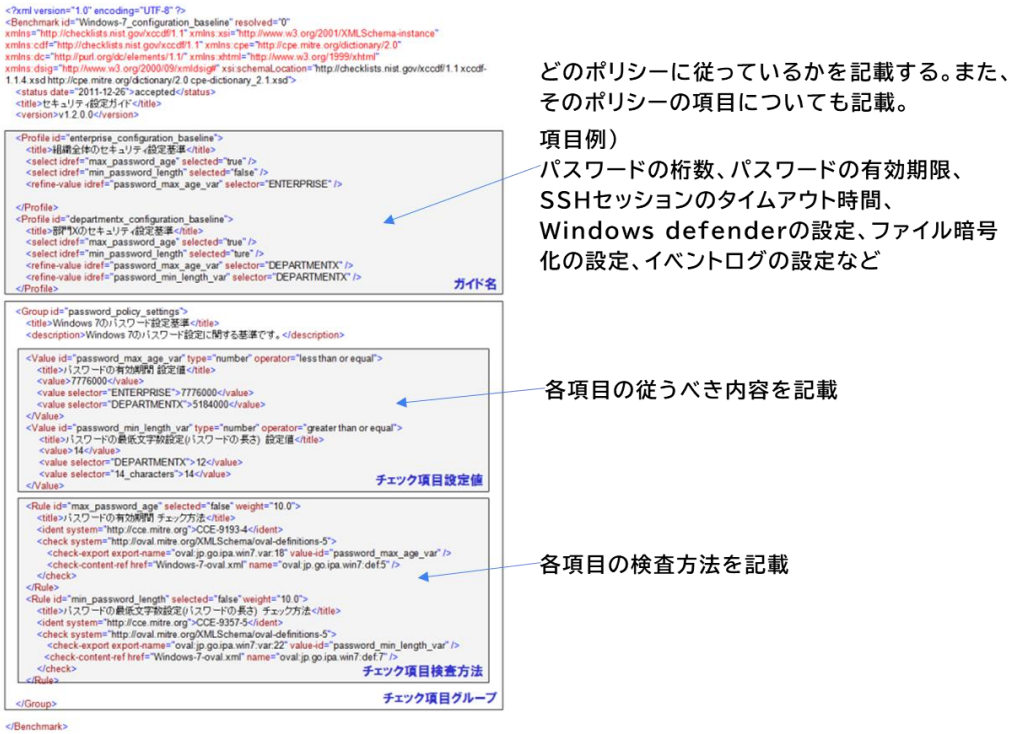


図 2-13　OpenSCAP に用いられている XCCDF の記述例[47]

表 2-17　OpenSCAP の具体的なツール

ツール名	概要
OpenSCAP Base	CLI 上で実施可能な OpenSCAP
SCAP Workbench	GUI 上で実施可能な OpenSCAP
OpenSCAP Daemon	OpenSCAP を自動で動かし、経年変化を可視化する
SCAPTimony	複数の機器で動いている OpenSCAP の結果

[47] セキュリティ設定チェックリスト記述形式 XCCDF 概説、https://www.ipa.go.jp/security/vuln/XCCDF.html

	をまとめる
OSCAP Anaconda Add-on	OS のインストール時に、同時に OpenSCAP を実施することができる

2.4.2.3. SBOM・VEX・SCAP において検討される関係性

今後 SBOM の脆弱性管理を自動化・効率化する上で、2.1.1 で確認した VEX や SCAP を用いて図 2-14 に示すようなイメージが検討される。SBOM は SCAP における CPE などの製品識別子と対応させることで、NVD 等の脆弱性データベースとの対応関係の確認が容易となり、脆弱性の特定の効率化につながると考えられる。類似の取組として、各企業では既に CPE と CVE の対応付けがされている NVD や JVN を用いて脆弱性管理の効率化が図っている。また、VEX においても CVE などの識別子を利用することが検討されているため、SCAP 標準で作成されている脆弱性データベースと VEX を連携することで、VEX が示す脆弱性が明確化され、脆弱性管理の効率化につながると考えられる。

更なる自動化に向けて、SBOM の構成要素に関する設定チェックファイル（OVAL・XCCDF など）を構築することで、特定のソフトウェアに脆弱性が残存していないかを自動で確認することが可能となる。類似の取組として、OpenSCAP 等でソフトウェアが規制・基準に準拠しているかを自動で確　　認している事例が挙げられる。また、VEX と設定チェックリストを連携することで、確認すべき脆弱性が限定され、一部の設定項目のみを抽出して精査することが可能となり、脆弱性管理の効率化につながると考えられる。

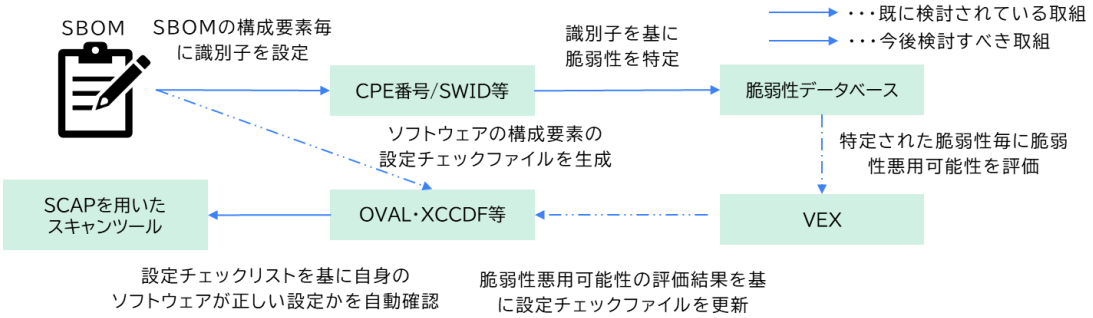


図 2-14　SBOM・VEX・SCAP において検討される関係性のイメージ図

2.5. CISA 会合の要点整理

CISA 会合における主な要点を以下にまとめる。会合は、最新の回から順に記載し、先行する回は情報が古くなっていることから絞り込みを行っている。

(1) CISA12 月会合

区分	Friedman 氏発言ポイント	アクション等
ヘルスケア PoC	ヘルスケア PoC は、政府資金を受けず、医療機器メーカー、医療機関の活動として発足。2022 年に開始された PoC 3.0 では、商用グレード SBOM を、契約に基づき規制当局や他社に共有する場合を想定した SBOM を対象とする。商用グレードの SBOM は、現時点の最先端のレベルを目指すもの。	SBOM のレベル差の認識は重要であり、SBOM 対応モデルの考え方で対応できる。
	PoC3.0 では、VEX と SBOM の並行した運用、SBOM の契約体系、NDA について検討を予定	VEX 活用の成果については METI 来年度取組の参考となる。
	病院が自主的に SBOM の利用を推進する組織は、2 つに分けられる。一つは、潤沢なリソースがあり、セキュリティの技術力が高い組織、もう一つは、セキュリティの関心が高く、将来の重要課題と捉えている組織。	自主的な取組を促進する上で、リソース、技術力、問題認識（関心）が挙げられる。関心は、SBOM 対応モデルの論点である。
自動車 PoC	本 PoC は、Auto-ISAC が中心の活動であり、政府による取組ではない。自動車分野は最も成熟したサプライチェーン管理モデルを有している。米国自動車業界は、訴訟分野が非常に強く、訴訟の懸念が背景にある。	米国ヘルスケア分野、自動車分野共に、業界の自主性が強い。責任と法制度に係る SBOM 取引モデルでの対応が期待される。
	SBOM WG では、ECU と OSS についての事項に注力して取り組む予定。SBOM WG は、基本的には自主自立で推進したい意向と見受けられる。	OSS の課題は大きい。外部からの規制を嫌うため、SBOM 対応モデルによる可視化を通じた自主性の促進が重要。
CISA 取組	“VEX”グループでは、VEX の最小要素の定義を進めており、現在ドラフトがほぼ完成に近づいている。1 月中旬完成の見通し。日本語訳を公開したい場合、提供する。	経産省の今後の取組の参考となる。

	“Sharing and Exchanging SBOMs”グループでは、SBOM 共有のユースケースとして DBOM のソリューションについての情報収集中。	SBOM 共有の効率化に有効。
	“On-ramps and Adoption”グループでは、SBOM の購買・調達におけるユースケースについての議論が行われている	SBOM 取引モデルの対価負担の参考となる。
その他	CISA 局長にブリーフィングする予定だが、その際、特に経済産業省が世界中の政府の中でもリーダーであることを強調したい。	日本の SBOM 実証が評価されており、米国との連携が進めやすくなる。
	RSA カンファレンスに何人か派遣するのであれば、スタッフの技術レベルではなく、ハイレベルの専門家を招聘することも可能かもしれない。アメリカの技術専門家と実務レベルの会議を直接開催したい。	RSA のオフサイト会合は、ステークホルダー間の交渉の場として有益。

(2) CISA10 月会合

区分	Friedman 氏発言ポイント	アクション等
SBOM 活用レベル	日本の SBOM 活用レベルの取組は、他に類を見ない分析で、日本は SBOM 推進をリードする国と認識。米国でも苦戦している論点に関するものであり、日本の取組は非常に実践的、実用的である。	日本の SBOM 実証が評価されており、米国との連携は進めやすくなる。
ソフトウェア ID	製品 ID の集中管理は困難である。CISA は製品 ID 解決のアプローチについてこの冬に報告書を発行予定。それは異なる ID (CPE, PURL 等）を許容しつつ、曖昧さを解消するアプローチである。 CPE の利用は限定的で、OSS では使われていないなど課題がある。	来年度の取組の共通 ID 方式の参考とする。
	CVE numbering authority(CNA)で実現されていることと対比して、CPE の分散管理(decentralized)でできることで、できないことを検討している。中小企業の部品 ID 定義などに課題あり。	来年度の取組の共通 ID 方式の参考とする。
VEX 課題	VEX は誰でも作成できるという前提で設計され、それ自体では本質的な信頼はないが、信頼できるデータ（会社）もある。主な作成者は製品ベンダーの PSIRT を想定。電力会社から、脆弱性に対して、自社が使用している製品への影響有無や	米国の VEX 取組成果を注視する。

	対応要否の判断に寄与する VEX のニーズがあがっている（自動的に SBOM を消費する方法）。	
	現状では、VEX はサードパーティが作成しており、信頼性の課題がある。独 CSAF, CSAT はアドバイザリーの機械可読の方法であり、VEX に利用できる。VEX 作成は分析とデータ生成があり、分析は、成熟した PSIRT が必要、後者はツールの開発が必要。	VEX の現実的な対応について米国の取組を参考とする。
	VEX ワーキンググループの ML に参加、フィードバックも可能。	今後の連携による情報共有を期待。
各国動向	欧州と米国で SBOM に差異がないようにするため、政府の実務者レベルでの国際 SBOM サミットを実施したい。	政府間での整合化は重要。
	サイバーレジリエンス法は SBOM データの共有を義務づけておらず、「持っていること」「要求があれば出せること」が条件だが、要求できるのは政府だけである。	政府間での整合化は重要。
	米国政府最小要素では、“top level dependencies”という言葉を使用したが、log4j の脆弱性から“top level dependencies”だけでは不十分であることが明確になった。(“Deps Matters”)	SBOM 対応モデルで明確に可視化できる。
	連邦政府内の様々な機関で SBOM を共有できるようなシステムを構築するように指示されている。	日米整合の観点で米国政府の動向の注視が必要。
	ヘルスケア分野が最も成熟した取組を進めており、今後 VEX も積極的に導入し医療機関に VEX ドキュメントを使用としている。	VEX の現実的な対応について米国の取組を参考とする。

(3) CISA7 月会合

区分	Friedman 氏発言ポイント	アクション等
SBOM 対応モデル・信頼性	SBOM をどこまで生成するかの違いの影響は大きい。（Deps Matters: Log4j） 日本の SBOM 適用範囲のレベル分けの取組は新しく、興味深い。	SBOM 対応モデルが認識され、今後の連携が期待される。
	SBOM の誤検知は最大の懸念事項である。VEX を活用して対処することが考えら	SBOM 対応モデルと VEX の組合せた取組の参

	れる。	考とする。
OSS 対応	OSS のメンテナンスに SBOM は有効である。OSS の利用促進に向け OpenSSF がセキュアな OSS に計画を検討。	今後の動向を注視する。
政府調達要件	大統領令に基づく政府調達要件化はまだ調整中である。	ソフトウェア分野の政府調達要件の整合化を検討。

3. OSS の利活用等に関する事例集の英訳

令和 2 年度事業で作成し令和 3 年度に拡充した「OSS の利活用及びそのセキュリティ確保に向けた管理手法に関する事例集」について、英訳を行った。具体的には、令和 3 年度事業で拡充した以下の 5 社の事例について、英訳を行った。

- SCSK 株式会社：
  意図しない OSS が混入していないかを検査する OSS 混入検査システム、安全性を確認した OSS を登録し、OSS 調達時に利用できる選定調達システムを構築。また、良質な OSS の選定のため、独自の OSS 評価結果をレーダーチャートで可視化するシステム（OSS Radar Scope として公開）を開発し、OSS 利活用に係る課題やリスクに対応している事例。
- OSSTech 株式会社：
  中小企業のシステム開発でも構成管理が可能な例として、ビルドシステム等の OSS に備わっているパッケージ管理システム等の機能を活用しながら、ソフトウェアの依存関係を開発者自らが管理し、省力化及び効率化した構成管理を実施している事例。
- 三菱電機インフォメーションシステムズ株式会社：
  通信キャリア向けに OSS を含んだソリューションを提供する際、OSS コミュニティによるサポート終了のリスクや、OSS を長期間利用する上での脆弱性管理やアップデート対応に係るコストの考え方等について、顧客と事前に合意することの重要性を示した事例。
- 日本電気株式会社：
  PSIRT を設立し、CVE Numbering Authority（CNA）としても活動を開始。脆弱性情報の収集、対応方針の整理、対応の社内調整を行っている。また、開発するシステムの構成情報を登録し、構成情報とともに脆弱性対策の有無及び報告を管理するシステムを構築・運用することで、構成管理と脆弱性対策の効率化、対策漏れ防止を実施している事例。
- 株式会社ラキール：
  ある OSS での商用利用を制限するライセンス変更をきっかけに、利用している OSS 全てをチェックするために、ツールを利用。従来の Excel 管理による管理漏れを防止し、ツールによって早い段階で危険な OSS を把握できるようになった事例。

4. SBOM を導入や活用するサプライチェーンモデルの構築に向けた調査・実証

4.1. 目的

産業分野ごとのリスク、法制度に応じて、SBOM を用いたソフトウェア部品のリスク管理を効果的に行うための方法についてコスト・効果の比較評価を行い、現実的な適用範囲と課題について整理する。実証を通じて、初級者向け SBOM 導入ガイダンス、SBOM の適用範囲を例示する SBOM 対応モデル、取引契約の例示により SBOM 導入を促進する SBOM 取引モデルの主な契約事項を整理することを目的とする。

4.2. 基本方針

分野ごとの法制度、取引形態、リスクに応じて、SBOM によるソフトウェア管理のコストと効果を考慮して、効果的、現実的な SBOM の適用範囲とレベルを示すことを基本方針とする。そのために、以下のような考え方に基づき実際の SBOM 適用に基づく成果物を取りまとめる。

(1) 分野ごとの法制度などの前提条件の整理
分野ごとに準拠すべき法制度は最低限満たすべき前提条件として整理する。

(2) 分野ごとの取引形態、開発形態などの条件の把握
産業分野ごとに取引形態、開発形態は多様であり、現実的に SBOM を適用する場合の制約条件として把握することが必要である。これらの制約条件は、将来的には改善・変更により、より効率化を図ることができる。

(3) SBOM の適用範囲とレベルを可視化するフレームワークの整理
SBOM 適用の選択肢は多様であり、どこまで適用するかにより効果・コストは大きく異なる。SBOM 適用の選択肢を洗い出し整理し、効果・コストに与える影響に基づき、どのような選択肢の組合せが考えられるかを整理し可視化することで、分野ごとに自律的に、必要な SBOM の適用範囲とレベルを考える手段を提供することができる。これにより受発注におけるソフトウェアの透明性、信頼性を確保し、市場メカニズムを機能させることができるようになるため重要である。

(4) SBOM 導入・活用のプロセス整理
SBOM 導入・活用のライフサイクル全体を俯瞰して漏れなく体系的にプロセスを整理し、コスト・効果の計測・評価の基盤として用いる。

(5) SBOM の実証に基づくコスト・効果の計測・評価
SBOM 導入・活用プロセスに応じて各フェーズについてコスト・効果の計測・評価を行う。

(6) 現実的な SBOM 適用範囲・レベル案の整理
分野ごとの法制度に準拠しつつ、取引形態、開発形態などの制約条件を考慮して、コスト・効果に基づき現実的に期待される SBOM 適用選択肢の組合せ案を整理する。将来的には、これらの案と SBOM 適用範囲の可視化フレームワークを用いて、分野ごとのステークホルダーにより妥当なレベル、基準についてコンセンサスを形成することが期待される。

(7) SBOM 適用における課題・ノウハウの整理

実証を通じて課題・ノウハウを抽出整理し、現場等で活用に寄与する。

分野ごとの法制度における要求、コスト・効果の受容性などに基づき、分野ごとに期待される SBOM 活用モデルを検討する。実証等で得られた知見を分野間で共有し、SBOM 導入促進のためのガイダンスに反映する。実証分野以外については、実証結果をもとに最小限求められる SBOM 活用モデルを検討する。

対象分野は、SBOM に関して「規制や推奨化が見込まれる分野」や「効果が大きいと思われる分野」を候補に、本年度の実証では、医療機器分野、自動車分野、ソフトウェア分野での SBOM 実証を実施する。

4.3. SBOM 実証全体の計画

実証の基本方針に基づく以下のような手順で実証を行った。

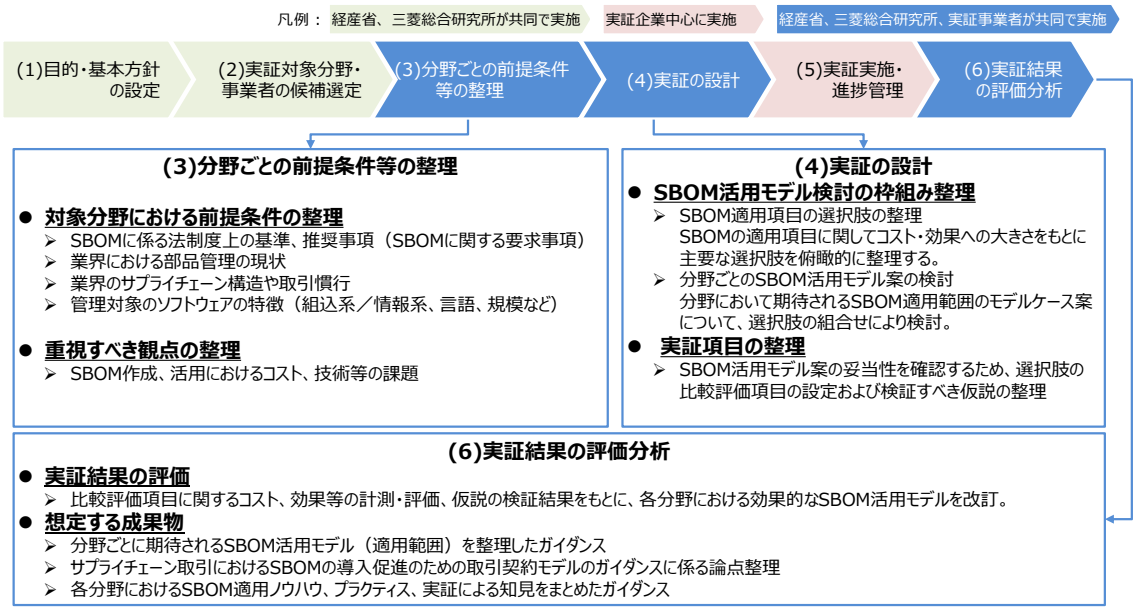


図 4-1　SBOM 実証全体の計画

(1) 目的・基本方針の設定
実証の目的と基本方針は 4.1, 4.2 章に示すとおり設定した。

(2) 実証対象分野・事業者の候補選定
SBOM に関して「規制や推奨化が見込まれる分野」や「効果が大きいと思われる分野」を候補に選定し、分野ごとに業界団体、ソフトウェアタスクフォース委員からの意見などをもとに選定、依頼する。

(3) 実証の設計
各分野の実証企業ごとに重点ポイントの設定、対象システムの選定、実証項目の設定、実証スケジュール等の設計を行う。

(4) 実証実施・進捗管理
設定した実証項目について、実証スケジュール、WBS に基づき実証作業の実施・管理

を行い、定期的な打合せを通じて、実証項目や方向性の調整を行う。

(5) 実証結果の評価分析

実証企業ごとの成果をもとに、分野横断的なコスト・効果の整理分析、課題・解決ノウハウの整理、分野ごとの SBOM 適用範囲案について整理する。

4.3.1. SBOM 実証に関する要件

2021 年度の結果から SBOM に関する課題と取組策を以下のとおり整理しており、本実証では、今後の課題の内、主に「1. SBOM 活用モデルの最適化」「2. SBOM 共有のための環境整備」「3. ツールの活用促進による効率化」について取り組むものである。

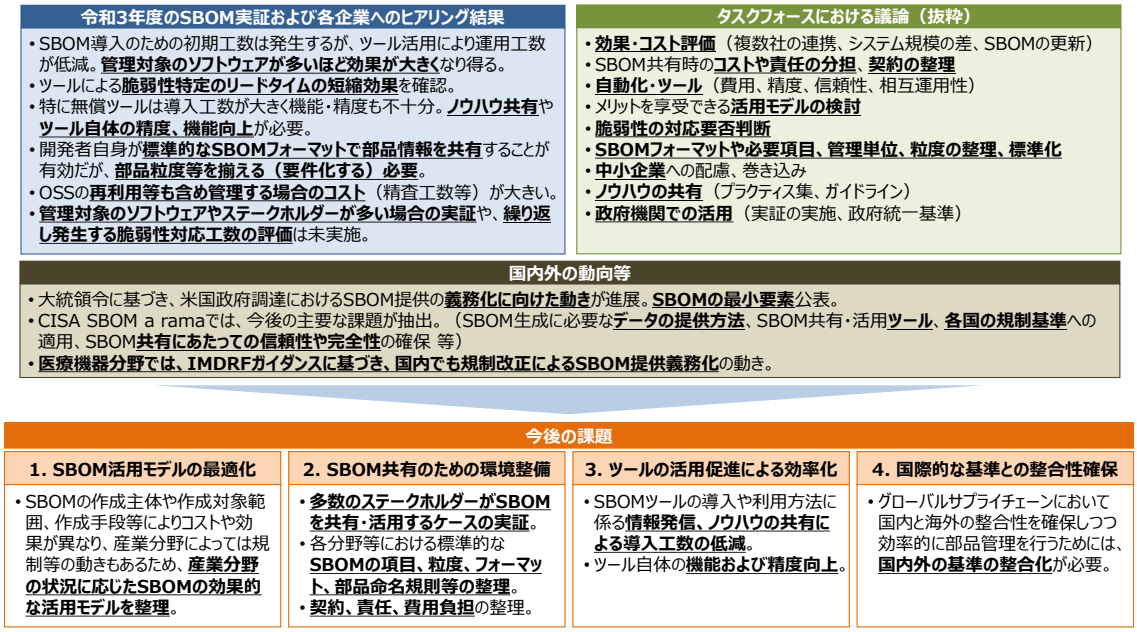


図 4-2　SBOM に関する課題と取組策（2021 年度結果）

4.3.2. 実証対象分野と実証事業者の選定

対象分野は、SBOM に関して「規制や推奨化が見込まれる分野」や「効果が大きいと思われる分野」を候補に、本年度の実証では、医療機器分野、自動車分野、ソフトウェア分野での SBOM 実証を実施する。

表 4-1　実証対象分野・対象事業者等の概要

分野	関連する業界団体	実証実施企業及び関係企業など					関連法制度（前提となる基準等）	対象製品
		ユーザー	最終ベンダー（製品ベンダー、インテグレータ）	ティア 1 サプライヤー	ティア 2 サプライヤー	サードパティサプライヤー		
医療機器	日本医療機器産業連合会	ヒアリング協力： 東京大学医学部附属病院 企画情報運営部 新秀直先生	近畿レントゲン工業 （製品ベンダー）	ライフサイエンスコンピューティング		Microsoft, Google 等	(厚労省) 医療機器　基本要件基準　一部改正案 JIS T 81001-5-1 制定案 医療機器製販業者向けサイバーセキュリティ手引書改訂(案) 医療機関向けサイバーセキュリティ手引書(案) (国際)N60 IMDRF ガイダンス N73 IMDRF 追補ガイダンス案 (米国) FDA 市販前ガイダンス案	歯科用 CT
自動車	（日本自動車工業会）	一般消費者、法人オーナー	(トヨタ自動車助言) （製品ベンダー）	東海理化	サニー技研	BROADCOM ,OSS ベンダー等	(国交省)道路運送車両の保安基準 (国際)UN-R155, 156 (米国)NHTSA ガイダンス	自動車ヒーターコントローラ

分野	関連する業界団体	実証実施企業及び関係企業など					関連法制度（前提となる基準等）	対象製品
		ユーザー	最終ベンダー（製品ベンダー、インテグレータ）	ティア 1 サプライヤー	ティア 2 サプライヤー	サードパティサプライヤー		
ソフトウェア	ソフトウェア協会	法人 (ヒアリング協力)	トレンドマイクロ、さくらインターネット、 コラボスタイル (製品ベンダー、インテグレータ)			Adobe, Amazon, Microsoft 等	(米国)NIST サプライチェーンガイダンス, FedRAMP	ネットワーク脅威検知、データセンター、業務フロー管理 SaaS

4.3.3. 分野ごとの前提条件・制約条件等

4.3.3.1. 法制度及び推奨事項

SBOM に係る前提条件となる法制度について分野ごとに整理する。

4.3.3.1.1. 医療機器分野

医療機器分野においては、医薬品、医療機器等の品質、有効性及び安全性の確保等に関する法律（以下、「薬機法」という。）に基づいて、医療機器の製造、販売、審査・認証、市販後の安全対策にわたって一貫した規制が行われています。薬機法第 41 条第 3 項には、厚生労働大臣は、医療機器、再生医療等製品又は体外診断用医薬品の性状、品質及び性能の適正を図るため、薬事・食品衛生審議会の意見を聴いて、必要な基準を設けることができるとされている[48]。

医薬品、医療機器等の品質、有効性及び安全性の確保等に関する法律第四十一条第三項の規定により厚生労働大臣が定める医療機器の基準（平成 17 年厚生労働省告示第 122 号）[49]（以下、「基本要件基準[50]」という。）では、全ての医療機器又は体外診断用医薬品が具備すべき品質、有効性及び安全性に係る基本的要件を規定している。

基本要件基準第 12 条第 2 項の規定（プログラムを用いた医療機器に対する配慮）では、構成管理の要件が示されており、適合を示すために用いられる規格として JIS T 2304 が示されている。[51]さらに、令和 5 年 3 月 9 日に厚生労働省告示第 67 号「医薬品、医療機器等の品質、有効性及び安全性の確保等に関する法律第四十一条第三項の規定により厚生労働大臣が定める医療機器の基準の一部を改正する件[52]」が官報に掲載され一部改正が公布された。これにより薬機法における基本要件基準第十二条第三項にサイバーセキュリティ要件が明示された。国際整合の一環として日本国内においても International Medical Device Regulators Forum：国際医療機器規制当局フォーラム（以下、「IMDRF」という。）ガイダンスの内容を薬機法規制に取り入れる方向である。国内における動向については、「我が国における医療機器サイバーセキュリティの規制の動向と今後の課題について[53]」を参考とし、最新の動向を調査した。詳細については 4.5 医療機器分野で説明する。

市販前のセキュリティに関わる規制（承認審査を含む）を整理し、 IMDRF ガイダンス国内導入にむけた検討、法改正等の動向について確認した事項を追記し、以下の図に示す。

---

48 薬機法：https://elaws.e-gov.go.jp/document?lawid=335AC0000000145

49 医薬品、医療機器等の品質、有効性及び安全性の確保等に関する法律第四十一条第三項の規定により厚生労働大臣が定める医療機器の基準：https://www.mhlw.go.jp/web/t_doc?dataId=81aa6953&dataType=0&pageNo=1

50 独立行政法人医薬品・医療機器総合機構「基本要件基準とは」https://www.pmda.go.jp/files/000240068.pdf

51 薬生機審発 0517 第 1 号平成 29 年 5 月 17 日 厚生労働省医薬・生活衛生局医療機器審査管理課長通知 https://www.std.pmda.go.jp/stdDB/Data/RefStd/Std_etc/H290517_0517-01_01.pdf

52 令和 5 年 3 月 9 日官報本紙（第 933 号）https://kanpou.npb.go.jp/20230309/20230309h00933/20230309h009330003f.html

53 医療機器学 Vol. 90, No. 6（2020）「我が国における医療機器サイバーセキュリティの規制の動向と今後の課題について」https://www.jstage.jst.go.jp/article/jjmi/90/6/90_534/_article/-char/ja/

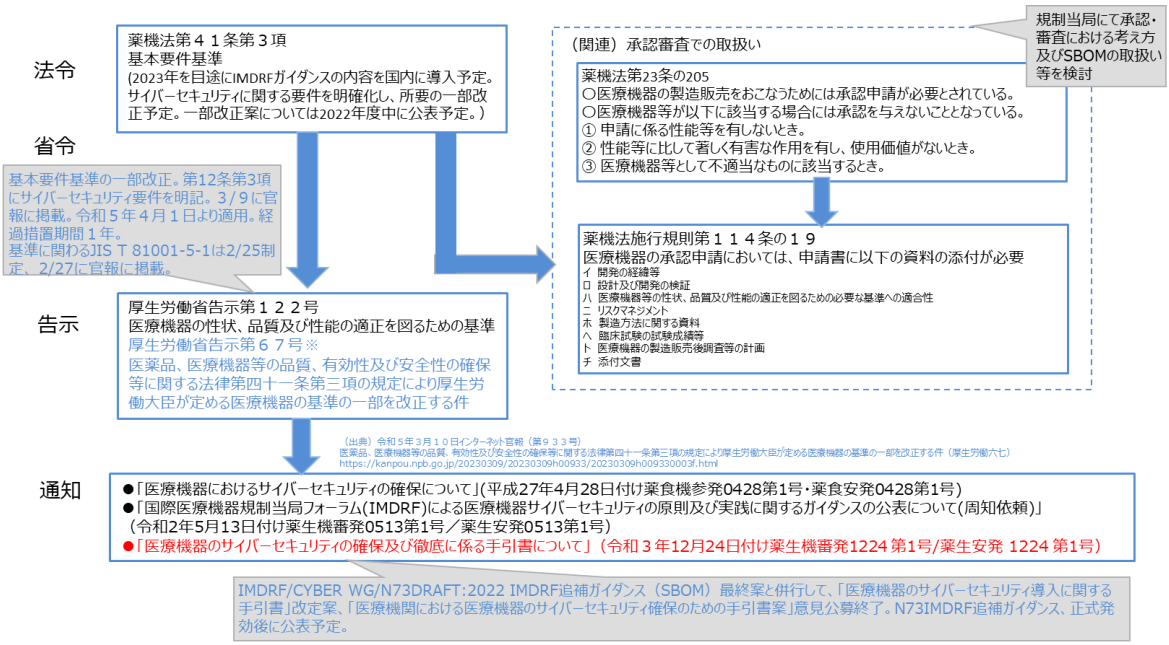


図 4-3　医療機器の市販前のセキュリティに関わる規制の概要

（出所）「我が国における医療機器サイバーセキュリティの規制の動向と今後の課題について」医機学 Vol. 90, No. 6（2020）「図１　医薬品医療機器等法における医療機器のサイバーセキュリティに係る市販前の要求事項」を参照し、株式会社三菱総合研究所にて一部加筆

また、医療機器市販後のセキュリティに関わる規制を整理し、IMDRF ガイダンス国内導入にむけた検討内容について確認した事項を追記した。市販後の安全対策について、以下図に示す。

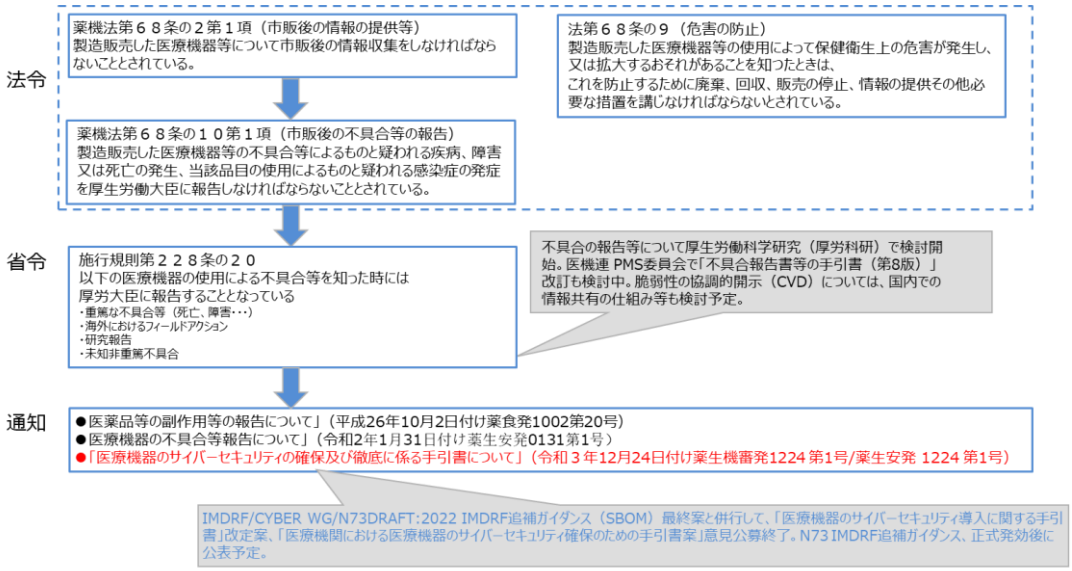


図 4-4　医療機器の市販後のセキュリティに関わる規制の概要

（出所）「我が国における医療機器サイバーセキュリティの規制の動向と今後の課題について」医機学 Vol. 90, No. 6（2020）「図２　医薬品医療機器等法における医療機器のサイバーセキュリティに係る市販後の要求事項」を参照し、株式会社三菱総合研究所にて一部加筆

4.3.3.1.2. 自動車分野

自動車分野では、国連欧州経済委員会の「自動車基準調和世界フォーラム（WP29）」における分科会「自動運転（GRVA）」で取りまとめられた規則（UN-R155、UN-R156）を踏まえ、国内では、令和 4 年 7 月より当該規則の適用が順次求められる。当該規則において SBOM に係る要件は規定されていない。日欧など型式認証で要求される国連協定規則 UN-R155 から参照される ISO/SAE21434 では、構成管理が要求され、例としてソフトウェア部品表の作成が挙げられる。

米国では、NHTSA が 2021 年初にガイダンスのドラフトが公開し、パブコメを実施。OEM に対し、ECU や各車両に使用されるソフトウェアコンポーネントに関する SBOM の作成・維持を求める要件が含まれており、今後 SBOM が推奨化される見通し。

SBOM に係る法制度を含む自動車サイバーセキュリティの法制度の関係を整理すると以下のようになる。

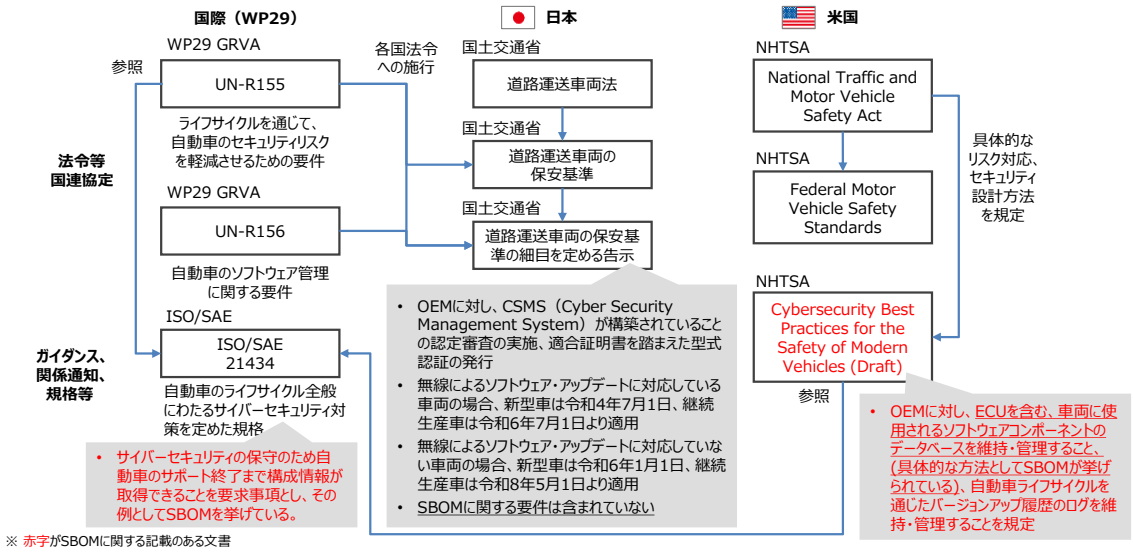


図 4-5　自動車分野の SBOM に係る法制度の関係整理

4.3.3.1.3. ソフトウェア分野

米国大統領令に基づき、連邦政府のソフトウェア調達において SBOM を開示等することが義務化される見通しである(2022 年度内)。米国では、2021 年 5 月の大統領令後、SBOM に関する内容を含んだ複数のガイダンス等の文書が公開された。

大統領令では、2022 年 5 月 12 日までに、これらの内容を踏まえた政府調達に関する勧告を FAR 審議会に対して実施するよう、DHS に指示している。勧告の詳細な内容は不明だが、SBOM に関する要求が含まれている可能性が高い。

今後、当該勧告事項に基づき、政府調達に関する規則（FAR）が改正される予定である。

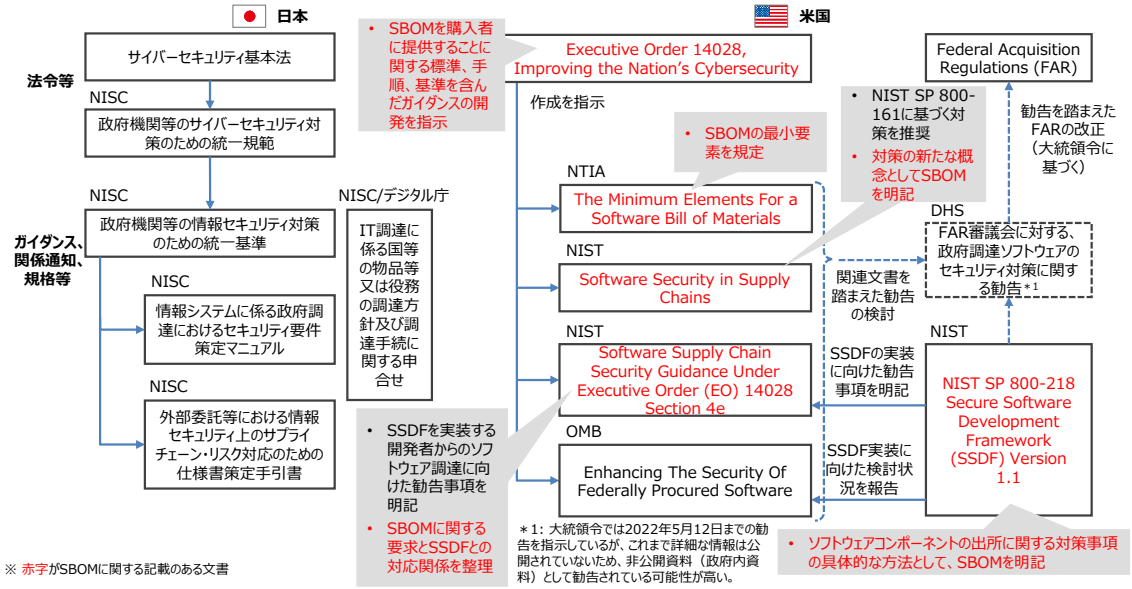


図 4-6　ソフトウェア分野の SBOM に係る法制度の関係整理

4.3.3.2. 業界の取引形態・開発形態

SBOM を作成・活用する上で、業界ごとの取引形態・開発形態等の制約条件を考慮し、現実に対応しつつ、将来に向けて改める点も考慮した対応が求められる。分野ごとに現状の取引形態、開発形態について、本実証の実施企業、関連団体、文献等から主な点を抽出整理したものを以下の表にまとめる。

表 4-2　分野ごとの取引形態等

	ソフトウェア分野（トレンドマイクロ）		ソフトウェア分野（コラボスタイル）		ソフトウェア分野（さくらインターネット）		医療機器分野（近畿レントゲン工業社）	
契約形態・取引慣行等	本実証における形態	業界において想定される主な形態	本実証における形態	業界において想定される主な形態	本実証における形態	業界において想定される主な形態	本実証における形態	業界において想定される主な形態
契約類型	既製品ライセンス契約	既製品ライセンス契約	既製品ライセンス契約（委託契約は行っていないため、ユーザ企業との契約のみ）	既製品ライセンス契約。（請負/準委任での開発委託契約を行っているケースも多いと思われる）	既製品ライセンス契約など	個別のケースで異なる	販売業者を介した製品売買 修理業者を介した製品保守	販売業者を介した製品売買 修理業者を介した製品保守
部品情報提供の要件	サポート期間中の製品について、契約ユーザ（場合によっては購入検討中の見込み顧客）に無償提供（予定）	契約ユーザに対して無償提供（ファイルまたはユーザが自ら取得できるコマンド等の提供）	利用者への構成・部品情報の提供義務無し。今後開示を検討するが、SBOMが脆弱性の洗い出し材料として利用されないように対策は必要	弊社と同様のケースが多いと想定。	サプライヤとの間：規定なし 顧客との間：規定なし	同左	なし	オープンソース及び市販のソフトウェア部品の透明性を確保するためのSBOMの提供
第三者部品の申告	OSSおよび有償ライブラリについては製品同梱のテキストファイルにてライセンス情報を提供	OSSおよび有償ライブラリのライセンス形態に合わせた情報提供（ウェブページからの参照、電子ファイルにて提供など）	規約上、利用者への告知義務なし。ただしOSSのライセンス条項に基づき、第三者部品は、ライセンス情報ファイルに利用しているライブラリを記載。	弊社と同様のケースが多いと想定。	サプライヤとの間：規定なし 顧客との間：規定なし	同左	なし	オープンソース及び市販のソフトウェア部品の透明性を確保するためのSBOMの提供
脆弱性の通知・修正	・自社が修正すべきと判断した脆弱性に対し、修正プログラムを契約中のユーザに提供（対象はサポート期間中の製品）。 ・SaaS製品以外、ユーザの適用作業が必要。	自社が修正すべきと判断した脆弱性に対して、修正プログラムを提供。原則対象はサポート期間中の製品。影響度が極めて高い場合、サポート対象外でユーザが多い製品・バージョンにも修正が提供される場合がある	通知義務はない。発見された脆弱性は、全ての関連サービスの修正が終わってからリリースノートで公開される。早急なユーザの協力が必要となる場合、の個別連絡や対策を行う。	弊社と同様のケースが多いと想定。	既製品の場合、保守条件に基づく	同左	なし	サイバーセキュリティ脆弱性又はインシデントが検知された場合の対応方法に関する通知及び修正
脆弱性の告知	・ 自社サイトで告知や担当者への連絡。 ・悪用事例があるなど、注意喚起が必要なものは、有償サポート契約ユーザの担当者に電話で連絡。 ・JPCERTと脆弱性公表に関して連携。	サポートウェブページなどで告知	規約上、修正義務は無い。	-	-	-	なし	サイバーセキュリティ脆弱性又はインシデントが検知された場合の対応方法に関する通知及び修正
納品形態	・バイナリパッケージを電子納品、インターネットから自動アップデート機能も使用。SaaS製品はライセンスの提供のみ、製品へのアクセス権を付与。ハードウェアアプライアンスのソフトウェアは機器組み込みで提供。 ・クラウドサービスのMarket Placeなどでは仮想マシンイメージやコンテナで OS 含め提供。	バイナリパッケージのインターネット上での配布。SaaS製品は、ライセンスの提供のみを行い、製品コンソールへのアクセス権を付与。 ハードウェア製品の場合にはソフトウェアは機器組み込み　にて提供	クラウドサービスであるため、メールによるお客様環境開設通知を以って納品とする。	弊社と同様のケースが多いが、追加で紙の納品書を送るケースもあると想定される。	機器組込など	同左	製品一式	製品一式
損害賠償責任	使用許諾契約総覧には以下記載が存在。 ・ 物理的なインストールメディアの欠陥に関しては購入後一定期間内で無償交換 ・上記を除き、ライセンス製品、マニュアルまたはスタンダードサポートに関して一切保証なし ・ライセンス製品、マニュアル、スタンダードサポート、またはサポートが受けられないことに起因して生じた損害に関しては、一切責任を負わない	トレンドマイクロと同等の賠償責任規定を持つ会社が多い印象	クラウド規約に従った本サービスの提供に関して、お客様に発生したいかなる損害について、一切の責任を負わない。	弊社と同様のケースが多いと想定するが、SLAに基づき、一定の損害賠償責任を負う場合もある。	既製品の場合、保守条件に基づく	同左	契約による	契約による
知的財産権の帰属	トレンドマイクロ	自社に帰属	ベンダー保有・使用権のみ提供。	弊社と同様のケースが多いと想定。	既製品の場合、ベンダー保有・使用権のみ提供	個別のケースで異なる	契約による	契約による

表 4-3　分野ごとの開発形態等

	医療機器分野（近畿レントゲン工業社）		自動車分野		ソフトウェア分野（トレンドマイクロ）		ソフトウェア分野（コラボスタイル）		ソフトウェア分野（さくらインターネット）	
項目	本実証	業界において想定される主な形態	本実証	業界において想定される主な例	本実証	業界において想定される主な形態	本実証	業界において想定される主な形態	本実証	業界において想定される主な形態
開発言語	C#,C++,JavaScript	個別のケースで異なる	C++,C,Python	アセンブリ,C,C++等	C,C++,Objective-C,Kotlin,Go,C#,PHP,Java,JavaScript,Python,Luaなど	Python,Java,Go,JavaScript,Rust,Swift,Objective-C,C,C++,C#など	ColdFusion(CFML),JavaScript	・サーバサイド nodejs(express),PHP,GO,Ruby等 ・クライアントサイド JavaScriptベースのフレームワーク（React, Vue.js等）	Java, Python, Goなど	個別のケースで異なる
部品形態	アプリケーション	個別のケースで異なる	ミドルウェア,アプリケーション	ミドルウェア,アプリケーション	OS,OS ライブラリ,Webアプリケーションフレームワーク,データベースサービスなど SaaSの場合はクラウドサービス提供のサービスなどを含む	同左	OS,システムライブラリ,コンパイラライブラリ,ミドルウェア,アプリケーション,データベースサービス,メールサービス,Webサービス	OS,システムライブラリ,コンパイラライブラリ,ミドルウェア,アプリケーション,データベースサービス,メールサービス,Webサービス等。基本的には自社と大きく変わらない想定。	ミドルウェア,アプリケーション	個別のケースで異なる
開発環境ツール	主にMicrosoft Windowsでの統合環境 VisualStudio VisualStudioCode	個別のケースで異なる	Qt Creator IDE	ティア1指示または関係者で選定 (適用マイコン用の統合開発環境,エディタ等)	Visual Studioシリーズ ,Eclipse, Android Studio, XCodeなど	同左	VisualStudioCode	VisualStudioCode	特になし	個別のケースで異なる
ビルドツール	MSBuild HAXE(JavaScript用)	個別のケースで異なる	GCC	ティア1指示または関係者で選定 (適用マイコン用のコンパイラやアセンブラ等)	CI/CD Pipeline : Jenkins,Circle CI, Github Actions	CI/CD Pipeline : Jenkins, Circle CI, Gitlab, Bamboo	webpack	クライアントサイドではwebpackやbabelが多いと想定。	Java:Ant Go: Mage	個別のケースで異なる
構成管理ツール	GitHub Bazaar(インストーラモジュール管理)	個別のケースで異なる	Git,Subversion	ティア1指示または関係者で選定（Subversion,Git等)	Github Enterprise Edition	Github（Github.com / Github EE）, Gitlab, Team Foundation Server, subversion	Bitbucket(git)	Gitによる管理が多いと想定。	Ansibleなど	個別のケースで異なる

以上の整理結果をもとに、SBOM の作成・活用において制約条件として特に考慮すべき点を整理すると以下のようになる。

表 4-4　実証対象分野における SBOM の作成・活用の制約条件

前提等	医療機器分野	自動車分野	ソフトウェア分野（情報系）
サプライチェーンの構造	・ソフトウェアの委託開発は一段階層程度が多い ・サードパーティ部品（既製品）の構成管理が義務化されており、OSS は対応の困難さから避ける傾向がある。（※ただし、AI/ML を搭載したプログラム医療機器等では OSS は少なからず利用あり。）	• サプライチェーンは多段階層も多く、中小、海外も含めて裾野が広い。 • サプライチェーンを通じた CSMS が要件化されており、サプライヤーに対してもリスク管理のエビデンスが求められる（ISO/SAE21434)。	• 商用、OSS 等のサードパーティ部品を使うケースも多い。 • パッケージソフトのサプライチェーンの階層は浅い • 情報システム構築では、多段階層のサプライチェーンも多い。
開発対象・開発環境等	・C/C++が多く、アセンブラ、Java、Python も利用実績あり。 ・AI/ML を搭載したプログラム医療機器では OSS 利用もあり。 ・Nessus（脆弱性チェックツール）が FDA 対応でよく用いられる。	• 制御系で C/C++/アセンブラが多い。 • 情報系でOSSの利用増加が見込まれる • ECU の場合、納品はバイナリ/ソースコードの両ケースがある。 • 構成管理ツールとして、Git, Subversion など利用するケースがある。	• Java, C/C++, Python、JavaScript などを含め多様な開発言語 • SaaS 等のクラウドやオンプレミスなど • ビルド・構成管理ツールは Jenkins,GitHub など、開発環境は, VisualStudio, Eclipse などが多い
取引慣行・契約	・開発委託、保守委託が多い。 ・市販後は修理業が対応する場合もある。	• 請負契約、準委任契約など各種存在。 • OSS 使用については報告を要件化する場合あり • 脆弱性の監視と連絡を要件化する場合あり	• ユーザー向けライセンス契約あり • 将来の部品情報、SBOM の提供は要件化の有無は両ケースあり

4.3.4. SBOM 実証の設計

SBOM 実証を推進するために、SBOM の実施プロセスを体系化し、実証分野ごとに実証項目に関する要件を整理、実証事業者との協議を通じて実証において実施すべき事項、計測・評価すべき事項、検討分析すべき事項について整理する。

表 4-5　SBOM 実証におけるプロセス区分と実証要求事項（案）の整理

フェーズ	項目	医療機器分野	自動車分野	ソフトウェア分野
基本方針（SBOM 活用モデルを実施する際のコスト計測、実現性、課題を確認する。）		・精度の高い SBOM を作成するための方法・選択肢（適用範囲）について活用モデル案の妥当性を確認する。 ・中小企業を含めた契約関係にある全ての開発者について、SBOM 作成する場合のフィージビリティや課題を検討する。 ・医療機関の SBOM 活用方法（脆弱性検出時の機器利用中断など）を確認する。 ・実証における製品開発に関連する全ての文書化は、QMS に従って行う。	・自動車のコスト制約と安全性の要求を考慮し、サプライチェーン上の契約関係にある各開発者が自身の直接利用部品(※1)の SBOM を生成し、それ以外の間接利用部品（※2)は、ティア 1 などがツールを用いて一括生成することで全体のコストを最小化することを想定する。 ・ツールの誤検知による SBOM 誤差に対する精査は、コードレビューなどのコスト評価に基づき実施の可否を考察する。 ・部品の粒度は、脆弱性 DB を用いた脆弱性特定などの目的を考慮して、部品の包含関係を含む全ての粒度の SBOM を保持することが有効か考察する。	・サプライチェーンにおける最終開発者（Sier 又はは機器ベンダー）を SBOM 作成の主体とし、委託契約のある開発者の SBOM 生成は、実証のコスト評価に基づく実運用での作成可否を考察する。 ・SBOM を生成できない委託先開発者から再帰的に利用される部品（間接利用部品※）について、最終開発者等がツールにより SBOM を生成し、その精査の可否、コストについて評価する。 ・SBOM の先行企業が参加できる場合、SBOM の信頼性を高めるための精査方法やコストについて評価する。
実証主体		医療機関（ヒアリング協力）、医療機器製造販売業者（以下略称：機器製販業者）、ソフトウェアサプライヤー	OEM（確認協力）、ティア 1、ティア 2	SIer、サプライヤー(ソフトウェア受託開発者又はパッケージソフトベンダー)
対象ソフト		歯科用 CT 装置	ヒータコントローラ	パッケージソフト、情報システム等
初期導入		・ツールの選定、インストール、利用方法習得の工数、コストの評価、課題の特定	・SBOM ツールの選定、インストール、利用方法習得の工数・コストの評価、課題の特定	・ツールの選定、インストール、利用方法習得の工数などのコストの評価、課題の特定
作成・共有	**SBOM 作成主体（ユーザー／ベンダー／サプライヤー／サー**	・サプライヤーが対象部品の範囲を特定し、SBOM を作成する。作成の工数等のコスト評価、課題の確認を行う。その結果から委託先中小企業を含むサプライヤーが SBOM 作成する上で、コスト、技術の困難がないか考察する。 ・サードパーティ(MS 等）による既製品	・ティア 1、ティア 2：各開発者がソフトウェアの実証範囲を特定し、各社の直接利用部品の SBOM を生成し、工数などのコスト評価、課題の確認を行う。実証結果をもとに、中小サプライヤー等がコスト的、技術的に SBOM 作成を現実的に行えるか考察する。	・サプライヤーが、自社開発部分の SBOM を生成することのフィージビリティを確認するため、生成コストの計測と課題を確認する。 ・CSP、サプライヤーの両者による計測を想定するが、サプライヤーの実証が難しい場合、CSP などが代替して計測する。

フェーズ	項目	医療機器分野	自動車分野	ソフトウェア分野
	ドパーティ）	の SBOM 提供の予定・見通しを確認する。		
	対象範囲（直接利用する部品のみ／再利用等も含む） 作成手段（無償／有償の別、ツールの種類、機能など）	・直接利用部品（※1）は、各開発者自身が特定し、ツールで SBOM 変換するコスト・課題を評価。 ・間接利用部品（※2）は、有償/無償ツールで SBOM 生成し、精査の方法確認（コードレビュー、ヘッダーレビュー等）、コスト比較、フィージビリティを評価する。コード解析ができない場合、OTS 提供者（サードパーティ）から情報提供の可否やコストを確認する。	・直接利用部品（※1）は、各開発者自身が特定し、ツールで SBOM 変換するなどし、そのコスト・課題を評価。 ・間接利用部品（※2）は、ティア 1 が SBOM 有償ツールにより一括生成する。生成された SBOM について、ソフトウェアなどの精査を行う。既製品が商用・バイナリの場合、既製品ベンダーから部品情報を取得できるか確認し、SBOM ツールの出力と比較などにより精査する。これらにより SBOM の精査の工数・コスト評価を行い、実運用で精査の現実性について考察する。	・CSP、サプライヤーが、自身の開発部分の直接利用部品の SBOM を生成し、コストを評価する。 ・SBOM を作成できないサプライヤーや既製品から再帰的に利用される部品（間接利用部品）は CSP が有償ツール等を用いて一括生成し、コストを評価する。 ・直接利用部品については、無償ツールにより SBOM 生成の可否を検討することで、コストの低減が可能か評価する。 ・間接利用部品について、OSS などソースコードレビューによりツールで生成した SBOM の誤検知など精査し、コストを評価する。
	SBOM のデータ形式	・SPDX、SWID、CycloneDX（SPDX, SWID は必須）の標準について、SBOM ツールによる生成の可否確認、コスト比較する（サプライヤー）	SPDX Lite 等	ツール選択肢と入出力を考慮し、妥当な標準形式を選定する。SPDX 等を想定し、CSP,サプライヤーで統一可能か検討する。
	SBOM の項目	・SBOM ツール出力が NTIA　Farming 文書等に対応できるか確認する（サプライヤー）	・米大統領令 SBOM 最小要素に対応できるか確認する（サプライヤー）	ツールにより、大統領令最小要素をカバーできるか確認する。

フェーズ	項目	医療機器分野	自動車分野	ソフトウェア分野
	共有方法（ファイル形式、受入検査等）	・受発注者のツールを共通化可否（ツール費用、入出力機能も考慮）（医療機関、製販業者、サプライヤー） ・受発注者で異なるツールを用いる場合、ツールの相互運用性、データ変換の可否を確認。 ・機器製販業者は、サプライヤーが有償ツールを用いた SBOM 生成を行っていない場合、有償ツールを用いて SBOM を生成し、受入検査（独立検査）について検討する。	・OEM,ティア 1、ティア 2：共通のツールを利用できる場合、SBOM 入出力機能を持つツールによりデータの共有の可否を評価する。 ・OEM,ティア 1、ティア 2：共通のツールを利用できない場合、ツールの入出力データの相互運用性、データ変換の可否を評価する。	・ツールの SBOM 入出力の可否を確認する。 ・サプライチェーンでツールの共通化の可否、課題を検討する。 ・サプライチェーンでデータ形式が統一できない場合、ツールによるデータ変換の検討。 ・SBOM 受入側は、独自に SBOM 生成し比較するなど、SBOM の信頼性検査の要否を検討する。
運用・管理	**活用範囲（調達時のリスク評価、脆弱性対応、資産管理、供給者管理）**	・機器製販業者が SBOM を集約し、脆弱性の特定、深刻度評価（CVSS）、悪用可能性評価及び情報共有フロー構築（VEX）の実施可否、フィージビリティ、コスト計測、課題特定を行う。 ・医療機関については同様の活用策について実施可否の判断を実施。規制当局への不具合報告のレベルの参考情報の整理。	・ティア 1 に SBOM を集約し、脆弱性の特定、深刻度評価（CVSS）、悪用可能性評価（VEX）の実施可否、フィージビリティ、コスト計測、課題特定を行う。 ・SBOM 作成時の部品情報の精査を省略し、脆弱性特定後に精査することで、コストを低減できるか評価する。	・脆弱性特定、脆弱性深刻度評価(CVSS)、悪用可能性評価(VEX)、ライセンス特定を実施し、コスト、課題を評価する。 ・SBOM 誤検知など信頼性の誤差を許容し、脆弱性特定の絞り込み後に SBOM 精査を行うことでコスト低減できるか評価する。
	SBOM の管理・運用の責任主体	・委託開発先のサプライヤーに対する契約上の部品申告の責任、SBOM 生成・脆弱性通知について要件化できるか検討。	・委託開発先に、SBOM の提供、脆弱性の通知を契約で要件化できるか考察する。	・OSS 含めライセンス契約に準拠した部品の申告は必要 ・契約条項として SBOM の要求が可能か考察。 ・最終製品ベンダーの部品情報、SBOM の責任について考察。

フェーズ	項目	医療機器分野	自動車分野	ソフトウェア分野
	管理・運用方法など	・脆弱性が特定された場合の医療機関、製販業者、サプライヤーの対処について確認。 （例：医療機関（使用中断等の事業継続評価、実際に脆弱性を悪用されてサイバー攻撃を受けた場合には所管省庁に報告※3、製販業者に情報共有）、製販業者（医療機関に通知、規制当局への報告、情報共有機関等への情報提供）、サプライヤー（製販業者への通知、脆弱性の修正、パッチ適用）） ・製品リリースに紐づく SBOM の更新、トレーサビリティ、管理の仕組み構築 ・対処の所要期間、頻度、工数コスト等を評価。	・OEM、ティア 1、ティア 2：SBOM を用いて脆弱性が特定された場合の対処法の選択肢（発注者への通知、ソフトの改修・回帰テスト等）を検討し、対処までの所要期間の計測、SBOM 活用による所要期間短縮効果を評価する。	CSP、サプライヤー、ユーザー企業について、脆弱性検出時の対処（通知、利用中断、修正など）を洗い出し、SBOM 有無によりリスク低減効果（対処時間の短縮など）を評価する。

※1：直接利用部品：サプライチェーンにおいて契約関係のある開発者が直接利用する部品

※2：間接利用部品：サプライチェーンにおいて契約関係のないサプライヤー（サードパーティ）が提供する部品から再帰的に利用される部品

実証においては、各分野にこれらの実証要求事項（案）をベースに打合せで、実施項目を具体化し、実施項目を設計した。具体的な実施項目は、4.5 章以降の各分野の実証項目の特定に示す。

4.3.5. SBOM 実証の実施スケジュール

下記のとおり、実証の要件定義・設計の後に、実証事業者との連携により実証を進め、計測・評価・分析を経て、結果の取りまとめを行った。

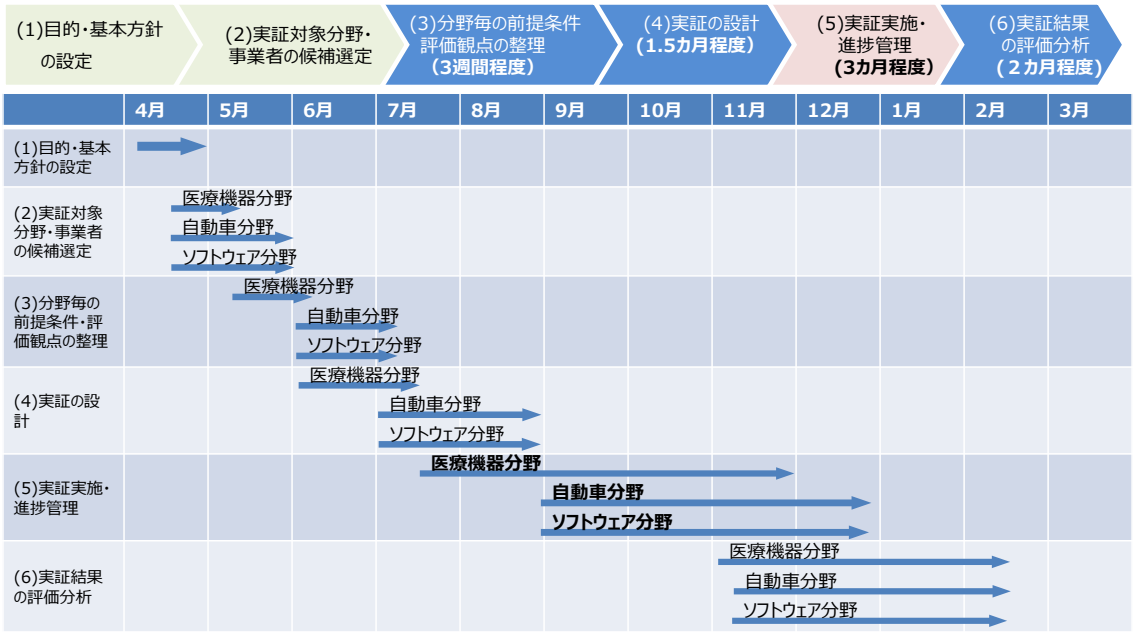


図 4-7　実証の実施スケジュール

4.4. SBOM 対応モデル

4.4.1. SBOM 対応モデルの必要性　～自律的な市場メカニズムの形成～

昨年度からの実証、ソフトウェアタスクフォース議論、CISA 議論の結果、SBOM は対応するかしないかのかの 2 択ではなく、サードパーティ部品の再帰的な利用を含めて、どこまで部品を広く特定し、それによりどこまで広く脆弱性管理を行えるかによって、脆弱性リスクの低減効果やコストに大きな影響を与えることが確認された。このような中で、どこまで SBOM に対応するか可視化しなければ、リスクに対応した適正なコストを負担するインセンティブは働かず、低いレベルの対応にとどまってしまうことが問題となる。このような点について SBOM 可視化の効果と必要性について全体像を図示したものが下図である。

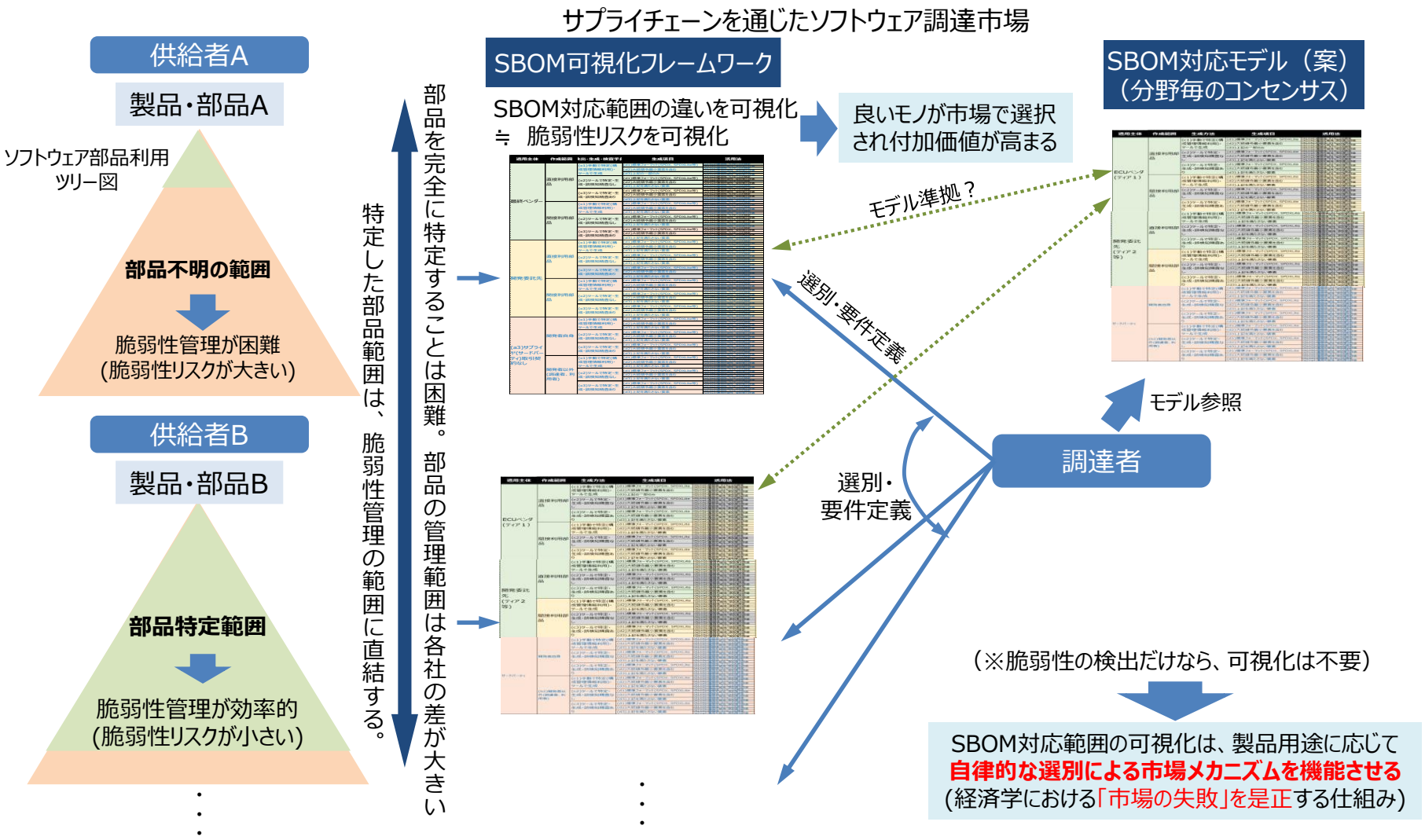


図 4-8　SBOM 対応範囲の可視化を通じた市場メカニズムの形成

サプライチェーンを通じて部品情報を管理することにより、脆弱性情報の特定など脆弱性リスクを低減する効果が得られる。

特定する部品の範囲や脆弱性管理などの活用範囲によりコストと効果は大きく異なる。**現実的には、OSS などサードパーティ部品などの再帰的な利用部品や snippet（改変コード）を含めた部品全体を完全に特定することは技術的にも困難であり、コストと効果の観点から管理する部品の妥当な範囲を判断することが求められことが想定される。**

SBOM において、**どこまで部品を特定し、どこまで活用するかについては、脆弱性管理等のリスク管理の対象範囲に直結**するため、リスクマネジメントの観点で、SBOM の部品範囲・活用範囲（以下、「**SBOM 対応範囲**」という。）の可視化は極めて重要である。

SBOM 対応範囲の可視化がされなければ、ソフトウェアの脆弱性の管理レベルやリスクを把握することができず、脆弱性管理レベルの高いソフトウェアと低いソフトウェアに対する適正な価値評価ができず、**よいソフトウェアが市場で選択されるという市場メカニズムが機能しない**ままとなる（経済学における「情報の非対称性」による「市場の失敗」と呼ばれる）。

このようなことから **SBOM 対応範囲の違いを可視化し、ソフトウェア構成の透明性を確保することは、市場メカニズムを機能させるために不可欠**な条件である。

SBOM の効果は（1）脆弱性の特定やその効率化などの手段としてだけでなく、（2）部品を調達・運用する際に、部品の透明性がどのぐらい確保されて、将来どのぐらいのリスクが想定されるかリスクの把握とコントロールが可能になること。

SBOM の価値は、**(1)脆弱性情報の特定とその効率化の手段、(2)部品を調達・運用する際に、部品の透明性がどのぐらい確保され、リスクがどの程度であるか把握しリスクをコントロールすることが挙げられる。これらの 2 つは、本質的に異なる**。前者(1)は、脆弱性を低減することが目的であり、後者(2)は、**リスクを可視化し、調達部品の選別や脆弱性管理等の要求レベルを判断することによりリスクをコントロールする**ことが目的である。**前者(1)の目的だけであれば、SBOM の対応範囲を可視化する必要性はない。**しかし、サードパーティ部品を含むソフトウェアの**部品を完全に特定することが技術的に困難であり、リスクゼロにはできないため、後者の SBOM 対応範囲の可視化を通じたリスクマネジメントは必須**となる（「**セキュリティ・アシュアランス**」という）

SBOM 対応範囲を可視化する枠組みが確立されれば、ソフトウェアの脆弱性管理の範囲などリスクや信頼性を可視化することができ、コストと効果のトレードオフの関係の中で、**どこまで SBOM に対応するべきかについて外部から強制されるのではなく、産業分野や個社の必要性に応じて調達者と供給者の間で自律的に部品の選別が進み、市場メカニズムのもとで適正レベルに調整が進む仕組み（「市場の失敗」を是正する仕組み）**を形成することができる。

以上のことから、SBOM の対応範囲を可視化する枠組み（「**SBOM 可視化フレームワーク**」と呼ぶ）を検討する。SBOM 可視化フレームワークは、SBOM の部品生成範囲・活用範囲を特定するための選択肢を整理し、それらの組合せとして対応範囲を可視化するものである。実証においては、SBOM 可視化フレームワークを用いて、SBOM 適用選択肢のコスト・効果を評価し、産業分野に応じて適正な SBOM 対応範囲の案を提示する。将来的には、業界

ごとに様々な評価・試行を通じて、適正な SBOM 対応範囲についてコンセンサスを形成することが重要である。そのような検討を通じて形成された分野ごとの SBOM 対応範囲のコンセンサスを「**SBOM 対応モデル**」という。

SBOM 可視化フレームワークは、(1)市場メカニズムに基づき SBOM 対応範囲の適正化を促すためだけでなく、(2)**規制分野における SBOM 対応範囲の基準を示す上でも必要**になる。基準を示すためには、その範囲や要件を、曖昧性なく示す必要があるためである。以上のような内容について、SBOM 対応モデルの必要性について図示したものが下図である。

4.4.2. SBOM 対応モデルのフレームワーク

4.4.2.1. SBOM 対応モデルの基本的な考え方と活用方法

前項において示したとおり、SBOM 対応範囲を可視化するフレームワークとそれに基づき、分野ごとのリスク、要求レベル、取引形態に応じて妥当な SBOM 対応範囲のコンセンサスを形成することが重要である。

コストと効果のトレードオフに基づき妥当なレベルの SBOM の適用範囲を示すことが、SBOM 普及促進させていく上で重要である。そのためのアプローチとして、SBOM 可視化フレームワーク、分野ごとの SBOM 対応モデル（案）、分野ごとに評価・検討した SBOM 対応モデル（業界確定版）が重要になる。それらの関係を図示したものが下図である。

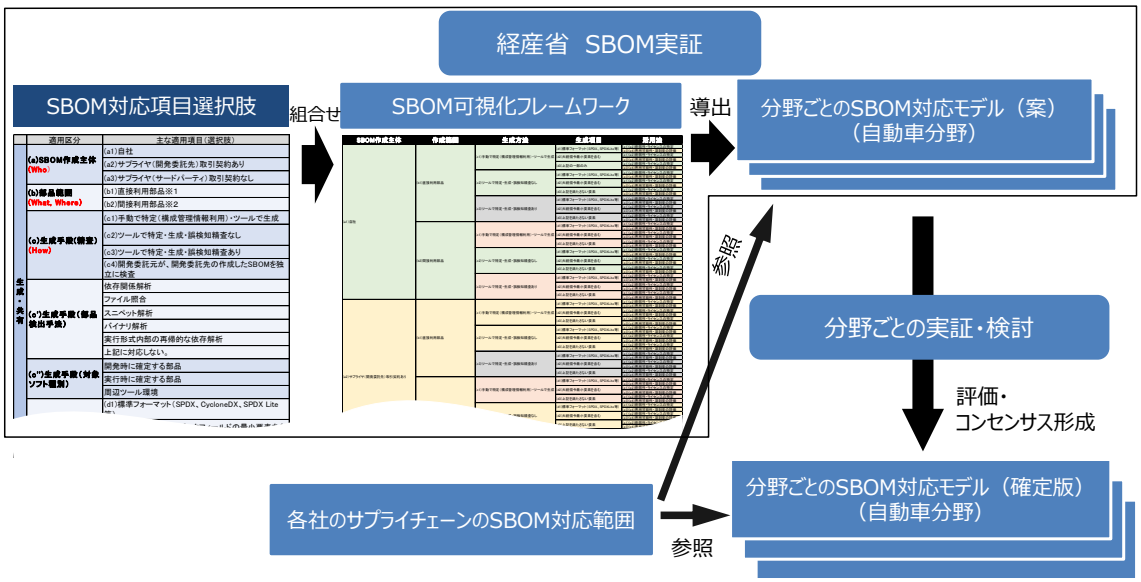


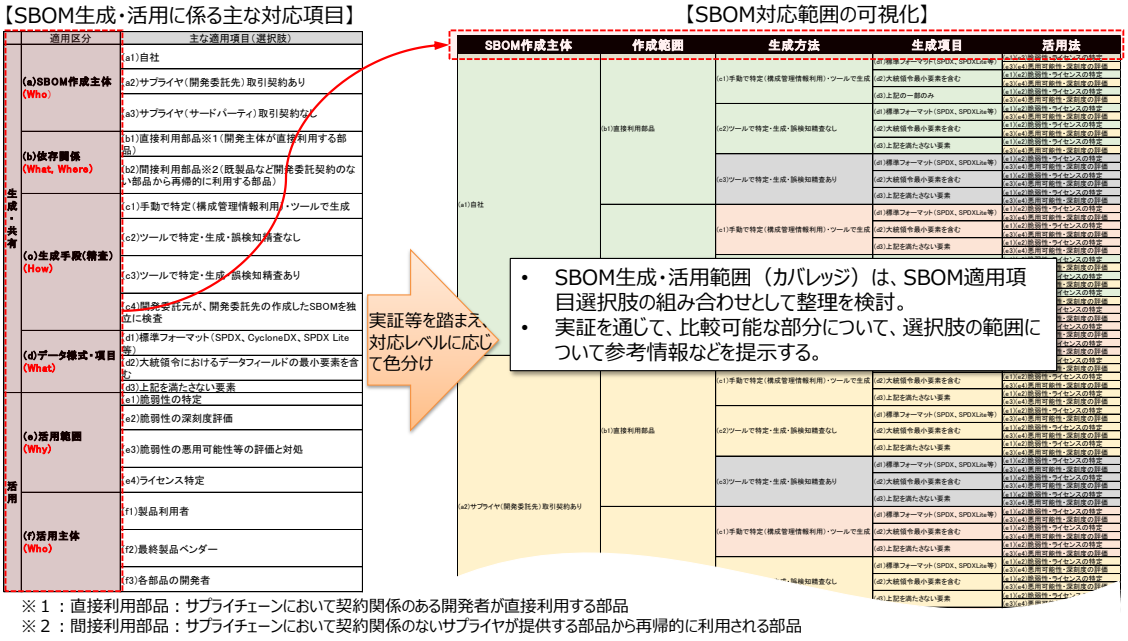
図 4-9　SBOM 対応モデルと関連する要素の関係

SBOM 可視化フレームワークは、SBOM の作成範囲・活用範囲などの対応範囲を可視化する枠組みであり、本枠組みを用いて、SBOM 実証において、分野ごとの法制度、取引形態、開発形態などの前提条件をもとに、コストと効果の計測評価を行い、分野ごとに妥当と考えられる SBOM の適用範囲（案）を示したものが「分野ごとの SBOM 対応モデル（案）」である。各社はサプライチェーンを通じてこのような SBOM 対応モデル（案）を参考に、自社

の SBOM 対応範囲を決定することができる。また、業界によって、主要なステークホルダーを交えて、実証評価や検討を通じて、「分野ごとの SBOM 対応モデル（案）」を精査し、「分野ごとの SBOM 対応モデル（業界確定版）」についてコンセンサス形成することが想定される。いずれの SBOM 対応モデルも、関連分野の法制度・規制に準拠していることが求められる。このような業界確定版が策定された場合には、各社はそちらの SBOM 対応モデルを参照して、必要に応じて、プラスアルファとしての SBOM 対応項目を追加し、ソフトウェアや SBOM の付加価値を高めることができる。

SBOM 可視化フレームワークは、あくまでも、SBOM 対応範囲を可視化する手段を提供するものであり、最終的な SBOM 対応範囲は、業界ごとの検討を通じ精査を行うことが期待される。

SBOM 対応範囲を可視化するためには、生成する SBOM の部品範囲に影響を与える SBOM 作成時の実施項目や SBOM の活用方法の選択肢を洗い出し、それらの組合せによりどこまで部品を生成し、どこまで活用・管理するかを明確に示す必要がある。下図は、その可視化の考え方を示したものである。



※１：直接利用部品：サプライチェーンにおいて契約関係のある開発者が直接利用する部品
※２：間接利用部品：サプライチェーンにおいて契約関係のないサプライヤが提供する部品から再帰的に利用される部品

図 4-10　SBOM 対応範囲の可視化の考え方

SBOM 生成・活用における対応項目の選択肢は、5W1H の形式で、誰が、どの範囲を対象に、どのような方法で生成し、いつ、どこで活用するかなどの観点で洗い出し、コストと効果の観点で影響の大きい主要なものを整理することで整理できる。それらの選択肢について、サプライチェーンを通じて、各主体（ユーザー、機器ベンダー、サプライヤー、サードパーティ等）が、それらの SBOM 適用項目の組合せについて、実施する項目、部分的に実施する項目、実施しない項目等で分類することにより全体の SBOM 対応範囲を可視化することができる。このようにして SBOM 対応範囲を可視化する枠組みを SBOM 可視化フレームワークと呼ぶ。

4.4.2.2. SBOM 対応項目の区分と選択肢

SBOM の作成・活用に関する項目の選択肢は、SBOM 可視化フレームワークを定義する重要な要素である。経産省 SBOM 実証事業においては、以下のような手順により SBOM 対応項目の選択肢を整理した。

(1) SBOM に関する国内外の文献調査
NTIA SOFTWARE BILL OF MATERIALS サイト、CISA Software Bill of Materials (SBOM)サイト、NIST Software Security in Supply Chains: Software Bill of Materials (SBOM)サイト、The Software Package Data Exchange® (SPDX®) Specification Version 2.3 等における文献情報

(2) SBOM に関する国内外の実証成果
NTIA SBOM 実証（PoC）、経済産業省 2021 年度実証結果

(3) 経済産業省ソフトウェアタスクフォースの意見（2021 年度、2022 年度第 7 回まで）
ソフトウェアタスクフォースにおける SBOM に関する意見、SBOM 対応項目選択肢（案）に対する過不足に関する意見

(4) SBOM 対応項目の選択肢（案）の策定
(1), (2)の結果を踏まえ、SBOM の作成・活用に係る主な実施項目を洗い出し、SBOM 対応項目選択肢（案）を整理。

(5) SBOM の分野ごとの実証
経済産業省　2022 年度 SBOM 実証において、医療機器分野、自動車分野、ソフトウェア分野の実証企業、参加業界団体からの SBOM 対応項目選択肢（案）に対する意見をもとに SBOM 対応項目選択肢（案）を改訂。

(6) SBOM 対応項目選択肢（最終案）の整理
(1)～(5)の結果をもとに、2022 年度における最終的な SBOM 対応項目選択肢（最終案）を整理。

以上の検討の結果整理した SBOM 対応項目選択肢（最終案）を以下に示す。

表 4-6　SBOM 対応項目選択肢（最終案）

	適用区分	主な適用項目（選択肢）	コスト	コスト区分の判断理由（主なコスト要素、前提等）
生成・共有	(a)SBOM作成主体 (Who)	(a1)自社	小	自社開発で直接利用する部品を構成ファイルなどから特定し、SBOMを生成する。コード改変部品を含む。
		(a2)サプライヤ（開発委託先）取引契約あり	中	取引契約のある開発委託先のソフトウェアで利用する部品のSBOMを生成する。
		(a3)サプライヤ（サードパーティ）取引契約なし	大	取引契約によるSBOMの要件化できないOSSや既成部品ベンダーがSBOMを作成する。(b2)(c2)
	(b)部品範囲 (What, Where)	(b1)直接利用部品※1	小	開発者が直接利用する部品を構成ファイルなどから特定し、ツールなどでSBOMを生成する。
		(b2)間接利用部品※2	大	サードパーティ部品について、再帰的に利用される部品に対してSBOMを生成する。
	(c)生成手段(精査) (How)	(c1)手動で特定（構成管理情報利用）・ツールで生成	小	直接利用する部品情報を構成ファイルなどを用いて作成する。
		(c2)ツールで特定・生成・誤検知精査なし	中	ツールを用いてSBOMを生成し、精査は省略する。ツールの利用は再帰部品のSBOM生成を主に想定するため商用ツールの利用を想定する。
		(c3)ツールで特定・生成・誤検知精査あり	大	商用ツールを用いてSBOMを生成し、ソースコードレビューを行い、誤検知、検出漏れの精査を行う。（再帰利用部品を含む）
		(c4)開発委託元が、開発委託先の作成したSBOMを独立に検査	大	開発委託元が、開発委託先の作成したSBOMを受け入れる際に、ツールなどで独立してSBOMを作成するなどして信頼性を検査する。
	(c')生成手段（部品検出手法）	依存関係解析	中	パッケージマネージャ等の構成情報を静的に解析する。
		ファイル照合	中	ハッシュ値当を用いてソースコードのファイル単位の一致を検出する。OSSのライブラリの検出なども含む。
		スニペット解析	大	ソースコードの部分的な文字列一致や類似性により検出する。
		バイナリ解析	大	バイナリファイルのビットパターンなどをもと類似性を検出する。
		実行形式内部の再帰的な依存解析	大	実行形式内にリンク済みのライブラリについて、そのライブラリをビルドする際の依存解析を再帰的に行う。
		上記に対応しない。	小	予め認識している部品をSBOMに変換する。
	(c'')生成手段（対象ソフト種別）	開発時に確定する部品	小	スタティックライブラリ、アプリケーション
		実行時に確定する部品	中	ランタイムライブラリ、サービス（ローカル、外部クラウド）、OS、ミドルウェア、実行環境（コンテナ、VM、APサーバ）
		周辺ツール環境	大	開発運用で使用するツール（インストーラ、アップデータ、配布パッケージ、開発環境、ツールチェーン、SBOMツール）
	(d)データ様式・項目 (What)	(d1)標準フォーマット（SPDX、CycloneDX、SPDX Lite等）	中	SPDXなどの標準フォーマットで作成する。
		(d2)大統領令におけるデータフィールドの最小要素を含む	中	大統領令におけるデータフィールドの最小要素を含むSBOMを作成する。
		(d3)上記を満たさない要素	小	独自の最小限の要素を作成する。
活用	(e)活用範囲 (Why)	(e1)脆弱性の特定	小	NVD, JVN等のDBを対象として脆弱性の検索・特定を行う。
		(e2)脆弱性の深刻度評価	中	CVSS値をベースとした深刻度を評価し、脆弱性対応の優先度を設定する。
		(e3)脆弱性の悪用可能性等の評価と対処	中	VEX情報等を用いて悪用可能性、脆弱性対処の必要性を評価する。必要に応じて対処策等のアドバイザリを発行する。
		(e4)ライセンス特定	中	ライセンスの特定と規約の取得を行う。
	(f)活用主体 (Who)	(f1)製品利用者	小	脆弱性が特定された場合、利用を中断し、ベンダーによる修正を待つ。業務中断コストも考慮すれば損害は大きい。
		(f2)最終製品ベンダー	中	利用者に脆弱性を通知するとともに、開発者への修正依頼、修正後のビルド・利用者への提供を行う。必要に応じて当局、ISAC等に報告する。
		(f3)各部品の開発者	大	開発者は、脆弱性の監視と修正を行い、調達者に修正版を提供する。必要に応じて当局、ISAC等に報告する。

※1：直接利用部品（直接部品）：サプライチェーンにおいて契約関係のある開発者が直接利用する部品

※2：間接利用部品（間接部品）：サプライチェーンにおいて契約関係のないサプライヤー（サードパーティ）が提供する部品から再帰的に利用される部品

4.4.2.3. SBOM 対応モデルの記述方法

SBOM 対応モデルは、サプライチェーンを通じた主体（ユーザー、機器ベンダー、サプライヤー、サードパーティ等）が、SBOM 対応項目のいずれを実施するかその組合せを規定することで表現することができる。実施の区分は、全て実施する項目、部分的に実施する項目、実施しない項目、他の項目を実施することで対応済みのため適用不要という 4 区分で全体の SBOM 対応範囲を可視化することができる。

適用主体	作成範囲	生成方法	生成項目	活用法
ティア1 (ECU最終ベンダー)	(b1)直接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記の一部のみ	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
	(b2)間接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
ティア2 (開発委託先)	(b1)直接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
	(b2)間接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
(a3)サプライヤ(サードパーティ)取引契約なし	(b1)開発者自身	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
	(b2)開発者以外(調達者、利用者)	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価

凡例：

適用可能・必須	部分適用（事例依存）	適用困難	不要または対象外

4.5. 医療機器分野

4.5.1. 実証の狙い・重点ポイント

IMDRF が策定した「N60:2020 Principles and Practices for Medical Device Cybersecurity：医療機器サイバーセキュリティの原則及び実践　」（以下、「IMDRF ガイダンス」という。）では医療機器セキュリティ確保において SBOM が推奨されている。2023 年には IMDRF ガイダンスに基づく国内対応が予定されている。

本実証においては、IMDRF ガイダンスに基づく SBOM 対応を検討した。その上で SBOM を作成するための手順や方法、初期導入及び管理、運用、活用における効果等を検証した。それぞれフェーズの工数やコスト等を計測し、各実証項目における課題等を明らかにする。また、課題への対処等、実証において得られた成果をもとにノウハウを抽出することを目的とする。

4.5.2. 実証の手順

実証項目として以下を示す。実証における各フェーズについて、それぞれ手動及びツールを使用してのコストを抽出する。

表 4-7　各フェーズの実証内容

フェーズ	実証項目
初期導入	有償・無償ツールの価格、インストール及び設定、習熟に係る工数を計測する。
生成・共有	手動・有償・無償ツールでの SBOM 生成時間を計測し、精度比較を実施する。また医療機関への SBOM の提供・共有方法の検討を行う。
活用・管理	生成した SBOM をもとに、脆弱性の特定・管理の方法を検討し、脆弱性が特定された場合の対処方法についてのフローを検討する。 また、医療機関での SBOM 活用及び管理方法についてヒアリングを通して検討する。

4.5.3. 実証の対象システムの概要と構成

今回の実証の対象システムは、株式会社近畿レントゲン工業社（以下、「KRK」という。）で製造販売しているアーム型 X 線 CT 診断装置「KR-X SCAN」とした。今回 SBOM 実証では、システム全体構成を確認した上で、下図のとおり、従来 KRK で作成していたシステム構成を基としたシステム構成図を作成し、医療機器認証範囲及び実証範囲の特定を行った。

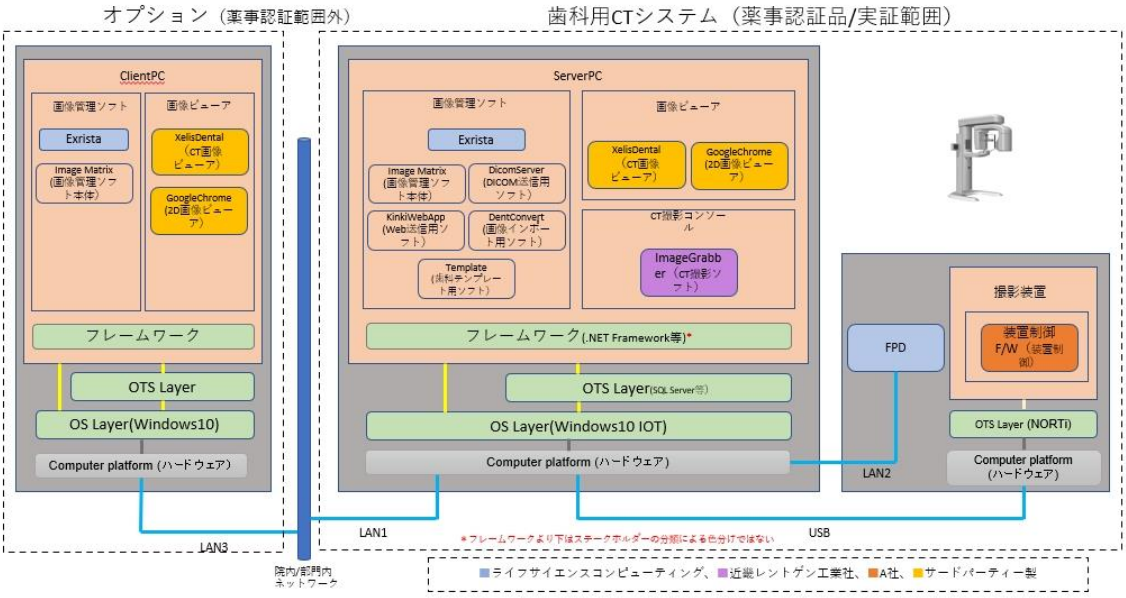


図 4-11　システム構成図

この装置は、人体の頭部の周りを回転しながら X 線撮影を行うことにより歯顎の 3 次元 X 線画像を生成するもので歯科診療に用いられている。当装置は主に X 線撮影を行う X 線発生部と X 線画像の受像部と、それらの回転機構で構成される。受像部で取得した X 線画像は Server PC に取り込まれ、専用のソフトウェア「Image Grabber」によって画像処理される。これによって生成された 3 次元 X 線画像は同じく Server PC の専用ソフトウェア「歯科用 CT」によってモニタに表示される。また、Server PC には、これらの一連の画像を管理するソフトウェアとして「歯科用 CT」がインストールされている。この他、オプション構成として Server PC の 3 次元 X 線画像を Client PC でも閲覧することができる機能を有している。Client PC は、歯科医院において、診療台等撮影装置と別の場所で画像を確認する場合に役立つが、医療機器のオプション部分であり、認証書では認証範囲外である。

従来のシステム構成図を見直すに当たり、KRK だけではなく、当装置のソフトウェア開発委託先である株式会社ライフサイエンスコンピューティング(以下、「LSC」という。)と共同で進めた。また、今回は実証事業では、一般社団法人日本医療機器産業連合会（以下、「医機連」という。）並びに有識者から助言を受けた。

KRK だけではシステム構成図の作成において特にオペレーティングシステム（以下、OS）を含めたフレームワーク及び off-the-shelf Software(以下、「OTS」という。)群を整理し、システム構造を把握することが困難であった。医機連、有識者等の指摘を受けてリテイクを繰り返すことにより、正確なシステム構成図の完成に至った。なお、システム構成図作成に要した工数は 8.50 人時間であった。対象医療機器の SBOM 生成のために透明性をもって改めて作成し見直し整理することが、SBOM を精査する上で役立つため、非常に重要なプロセスである。

実証範囲は医療機器の認証範囲である Server PC 部及び装置の制御プログラムとした。実証事業の時間的な制約から、Server PC 部のソフトウェアを対象に詳細調査を実施し、装置の制御プログラムは手動による SBOM 生成までに留めた。

4.5.4. 対象分野の前提条件・取引慣行等

薬機法では、医療機器においては製造業、販売業、修理業等の業の許認可について規定されている。薬機法に基づく業を行う際には各種の申請・届出が必要となっており[54]、厚生労働省又は各都道府県等の許可、登録、承認等を得なければ業を行うことができないとされている[55][56]。

医療機器の一般的な取引慣行について、関係法令・基準・ガイドライン等で示されている内容をもとにユーザー組織との関係、及びサプライヤーとの関係に分けてまとめたものを下表に示す。

表 4-8　医療機器の一般的な取引慣行

		ユーザー組織との関係	サプライヤーとの関係
契約形態・取引慣行等	契約類型	販売業者を介した製品売買 修理業者を介した製品保守	開発・保守委託、医療機器の修理業者への委託
	部品情報提供の要件	オープンソース及び市販のソフトウェア部品の透明性を確保するための SBOM の提供	最新の SBOM 提供及びソフトウェア部品の医療機器の製品寿命（END OF LIFE 以下「EOL」という。）、サポート終了（END OF SUPPORT 以下「EOS」という。）の開示
	第三者部品の申告		
	脆弱性の通知・修正	サイバーセキュリティ脆弱性又はインシデントが検知された場合の対応方法に関する通知及び修正	サイバーセキュリティ脆弱性が検知された場合の連絡及び修正
	納品形態	製品一式	実行モジュール一式
	損害賠償責任	契約による	契約による
	知的財産権の帰属	契約による	契約による
開発形態等	開発言語	ー	規定なし(サプライヤーにて選定)
	部品形態	ー	アプリケーション
	開発環境ツール	規定なし	規定なし(サプライヤーにて選定)
	ビルドツール	規定なし	規定なし(サプライヤーにて選定)
	構成管理ツール	規定なし	規定なし(サプライヤーにて選定)

今回の実証形態での取引慣行について、ユーザー組織(今回の場合、歯科医院を想定)と

[54] 厚生労働省、薬事関係法令に係る行政手続について
https://www.mhlw.go.jp/stf/seisakunitsuite/bunya/kenkou_iryou/iyakuhin/newpage_00843.html
[55] 薬機法、https://elaws.e-gov.go.jp/document?lawid=335AC0000000145
[56] 独率行政法人医薬品医療機器総合機構、医薬品の製造販売手順について：医薬品の製造販売手順について

KRK との関係、及びサプライヤー(今回の場合、主に LSC)と KRK との関係に分けてまとめて下表に示す。

表 4-9　今回の実証形態での取引慣行

		ユーザー組織との関係	サプライヤーとの関係
契約形態・取引慣行等	契約類型	販売業者を介した製品売買 修理業者を介した製品保守	開発・保守委託、 医療機器の修理業者（KRK※）
	部品情報提供の要件	なし	ソースファイル一式
	第三者部品の申告	なし	OTS 使用有無の連絡
	脆弱性の通知・修正	なし	なし
	納品形態	製品一式	実行モジュール一式 （インストーラ形式）
	損害賠償責任	契約による	契約による （発注額を限度とする）
	知的財産権の帰属	契約による	契約による （発注元に帰属）
開発形態等	開発言語	－	サプライヤーにて選定 （主に C#,C++,JavaScript）
	部品形態	－	アプリケーション
	開発環境ツール	規定なし	サプライヤーにて選定（主に Microsoft Windows での統合環境、Visual Studio,Visual Studio Code）
	ビルドツール	規定なし	サプライヤーにて選定（主に MSBuild,HAXE(JavaScript 用)）
	構成管理ツール	規定なし	GitHub,Bazaar(インストーラモジュール管理)

※KRK は製造販売業の登録のほか、医療機器の修理業の許可を得ている。また医薬品、医療機器等の品質、有効性及び安全性の確保等に関する法律施行令（昭和三十六年政令第十一号）、第 56 条医療機器の修理業の特例により、自ら製造する実証の対象医療機器については修理することは可能である[57]。

医療機器分野におけるセキュリティに関する主な規制については、下表のとおりである。

表 4-10　医療機器分野の規制要求事項

法令・基準・ガイドライン等	要求事項の概要
医薬品、医療機器等の品質、有効性及び安全性の確保等に関する法律（昭和 35 年法律第	サイバーセキュリティを含むリスクマネジメントが求められ、使用者に対する情報提

[57] 薬機法、医薬品、医療機器等の品質、有効性及び安全性の確保等に関する法律施行令：https://elaws.e-gov.go.jp/document?lawid=336CO0000000011_20221201_503CO0000000001

法令・基準・ガイドライン等	要求事項の概要
145 号）（以下、「薬機法」という。法）[58]	供や注意喚起を含めて最新の技術に立脚して医療機器の安全性を確保しなくてはならないとされている。
医薬品、医療機器等の品質、有効性及び安全性の確保等に関する法律第四十一条第三項の規定により厚生労働大臣が定める医療機器の基準（平成 17 年厚生労働省告示第 122 号）[59]	
医療機器におけるサイバーセキュリティの確保について（平成 27 年 4 月 28 日付け薬食機参発 0428 第 1 号・薬食安発 0428 第 1 号・厚生労働省大臣官房参事官（医療機器・再生医療等製品審査管理担当）・医薬食品局安全対策課長連名通知）[60]	製造販売業者は、不正なアクセス等が想定される医療機器については、サイバーリスクを含む危険性を評価・除去し、防護するリスクマネジメントを行い、使用者に対する必要な情報提供や注意喚起を含めて適切な対策を行うこととしている。
医療機器のサイバーセキュリティの確保に関するガイダンスについて（平成 30 年 7 月 24 日付け薬生機審発 0724 第 1 号・薬生安発 0724 第 1 号・厚生労働省医薬・生活衛生局医療機器審査管理課長・医薬安全対策課長連名通知）[61]	サイバーリスクに伴う医療機器の不具合等についても GVP 省令における安全性情報として取り扱い、販売業者・貸与業者や修理業者の協力のもと、医療機関と連携を取り、適切な市販後の安全確保が求められている。
N60:2020 Principles and Practices for Medical Device Cybersecurity(医療機器サイバーセキュリティの原則及び実践)	医療機器のサイバーセキュリティに関する国際整合を図るための一般原則とベストプラクティスを提供することを目的として、IMDRF により作成された。
国際医療機器規制当局フォーラムによる医療機器サイバーセキュリティの原則及び実践に関するガイダンスの公表について(周知依頼)」(令和 2 年 5 月 13 日付け薬生機審発 0513 第 1 号・薬生安発 0513 第 1 号・厚生労働省医薬・生活衛生局医療機器審査管理課長・医薬安全対策課長連名通知)[62]	製造販売業者は、市販前には、医療機器のサイバー攻撃に対する耐性が確保されるよう、設計及び開発を行い、市販後には、意図する使用環境における機器の運用、情報共有、脆弱性の修正、インシデントの対応等を適切に行う必要がある。また、医療現場において適正な管理がなされるよう、製造販売業者は、医療機関、使用者（以下、ユーザー）、規制当局及び脆弱性発見者等のステークホ

[58] 厚生労働省、医薬品、医療機器等の品質、有効性及び安全性の確保等に関する法律 https://elaws.e-gov.go.jp/document?lawid=335AC0000000145

[59] 厚生労働省、医薬品、医療機器等の品質、有効性及び安全性の確保等に関する法律第四十一条第三項の規定により厚生労働大臣が定める医療機器の基準 https://www.mhlw.go.jp/web/t_doc?dataId=81aa6953

[60] 厚生労働省、医療機器におけるサイバーセキュリティの確保について https://www.mhlw.go.jp/file/05-Shingikai-11121000-Iyakushokuhinkyoku-Soumuka/0000090664.pdf

[61] 厚生労働省、医療機器のサイバーセキュリティの確保に関するガイダンスについて https://www.mhlw.go.jp/hourei/doc/tsuchi/T211228I0070.pdf

[62] 厚生労働省、国際医療機器規制当局フォーラム(以下 IMDRF)による医療機器サイバーセキュリティの原則及び実践に関するガイダンスの公表について(周知依頼) https://www.mhlw.go.jp/hourei/doc/tsuchi/T200521I0040.pdf

法令・基準・ガイドライン等	要求事項の概要
	ルダーと必要な情報共有等を行い積極的に連携していくことが求められている。
医療機器のサイバーセキュリティの確保及び徹底に係る手引書について（令和 3 年 12 月 24 日付け薬生機審発 1224 第 1 号、薬生安発 1224 第 1 号）[63]	IMDRF ガイダンスの発行等の国際的な枠組みでの活動を踏まえて、旧ガイダンスを 置き換え、医療機器へのサイバー攻撃に対する国際的な耐性基準等の技術要件を我が国へ 導入して整備することを目的に、国立研究開発法人日本医療研究開発機構医薬品等規制調和・評価研究事業「医療機関における医療機器のサイバーセキュリティに係る課題抽出等 に関する研究」及び医機連において、医療機器のサイバーセキュリティに係る必要な開発目標及び技術要件等を検討し、別添のとおり、「医療機器の サイバーセキュリティ導入に関する手引書（以下、「手引書」という。)」として取りまとめられた。
N73DRAFT:2022‘Principles and Practices for Software Bill of Materials for Medical Device Cybersecurity（DRAFT DOCUMENT）‘医療機器のサイバーセキュリティのためのソフトウェア部品表の原則及び実践の原案（以下、IMDRF 追補ガイダンス案））[64]	IMDRF ガイダンスの追補文書で、SBOM の概要説明並びに SBOM の生成及び使用に対するベストプラクティスを提供することを目的として、IMDRF により検討されている。策定に向けてドラフト文書が公開され、意見公募が終了した。

医療機器分野において、医療機器の製造販売を規制する「医薬品、医療機器等の品質、有効性及び安全性の確保等に関する法律」（昭和 35 年法律第 145 号）（以下、「薬機法」という。）に紐づく「医薬品、医療機器等の品質、有効性及び安全性の確保等に関する法律第四十一条第三項の規定により厚生労働大臣が定める医療機器の基準」（平成 17 年厚生労働省告示第 122 号）（以下、「基本要件基準」という。）において、サイバーセキュリティを含むリスクマネジメントが求められ、使用者に対する情報提供や注意喚起を含めて最新の技術に立脚して医療機器の安全性を確保しなくてはならないこととされている。なお、2022 年度中にサイバーセキュリティ要件を明確化する一部改正について意見公募が実施され[65]、令和 5 年 3 月 9 日に「医薬品、医療機器等の品質、有効性及び安全性の確保等に関する法律第

[63]厚生労働省、医療機器のサイバーセキュリティの確保及び徹底に係る手引書について https://www.mhlw.go.jp/hourei/doc/tsuchi/T211228I0070.pdf

[64] IMDRF、N73DRAFT:2022 ‘Principles and Practices for Software Bill of Materials for Medical Device Cybersecurity（DRAFT DOCUMENT）‘医療機器のサイバーセキュリティのためのソフトウェア部品表の原則及び実践の原案（以下、IMDRF 追補ガイダンス案） https://www.imdrf.org/consultations/principles-and-practices-software-bill-materials-medical-device-cybersecurity

[65] e-gov, 医薬品、医療機器等の品質、有効性及び安全性の確保等に関する法律第四十一条第三項の規定により厚生労働大臣が定める医療機器の基準の一部を改正する件（案）に係る御意見の募集について https://public-comment.e-gov.go.jp/servlet/Public?CLASSNAME=PCMMSTDETAIL&id=495220262&Mode=0

四十一条第三項の規定により厚生労働大臣が定める医療機器の基準の一部を改正する件(厚生労働六七)」が官報に掲載[66]された。実証実施後の一部改正であるが、法改正の動向を注視しながら実証を実施した。

厚生労働省厚生労働省医薬・生活衛生局医療機器審査管理課長、厚生労働省医薬・生活衛生局医薬安全対策課長通知「医療機器のサイバーセキュリティの確保及び徹底に係る手引書について」では、我が国においても、医療機器製造販売業者に対して IMDRF ガイダンスを導入することが示されているため、本実証においても IMDRF ガイダンスを基に実証を進めた。また、IMDRF 追補ガイダンスが策定中であり、ドラフト文書では機器内部の共通コンポーネントのトレーサビリティ及び透明性確保のための手段として、SBOM について生成、管理活用について説明されているため、実証において参考文書とした。

4.4.5. 実証項目の特定

実証項目を下表に示す。

表 4-11　SBOM 実証における実施項目

フェーズ	項目	実証項目
初期導入	ー	有償、無償ツールの選定及びツール価格及びインストール、設定工数を算出する。
生成・共有	SBOM 生成主体	有償、無償、手動での SBOM 生成にかかる工数を算出する。
		サードパーティ製ソフトウェアの SBOM 提供をベンダーに依頼し、提供された SBOM の受入検査の工数を算出する。受入検査については、IMDRF 追補ガイダンス案に規定されている 8 要素満たせているかの確認とする。
		手動で作成した SBOM と有償、無償ツール（主に MEND SCA）で生成した SBOM で比較を行い、比較結果についての評価を実施する。
		手動 SBOM、有償ツールと無償ツールでの SBOM の比較を行い、比較結果についての評価を実施する。
		SBOM 生成主体で行った実証作業についての課題を抽出する。
	SBOM の項目	有償、無償ツール、手動及び提供されたサードパーティ製ソフトウェアの SBOM について、IMDRF 追補ガイダンス案 8 要素を満たしているかを確認し、確認に要した工数と評価を実施する。また課題を抽出する。
	共有方法	有償、無償ツール、手動での SBOM それぞれについて、開発委託先との共有方法を検討し、開発委託先との共有にかかるコストを机上で算出する。また課題を抽出する。
活用・管理	活用範囲	脆弱性マネジメントフローを作成し、フローに従い脆弱性の発見から修正に至るまでの工数を算出する。脆弱性の発見か

[66] 令和 5 年 3 月 9 日インターネット官報本紙
https://kanpou.npb.go.jp/20230309/20230309h00933/20230309h009330003f.html

フェーズ	項目	実証項目
		ら対処方法の特定までを実工数で計測し、それ以降に関しては机上で評価する。
	活用・管理方法	医療機関に SBOM を提供して導入する場合の留意事項を検討する。
	医療機関へのヒアリング項目検討	製造販売業者の視点から、SBOM 提供・活用に関してのヒアリング内容をまとめ、課題を抽出する。
	効果評価	SBOM の生成、管理活用で計測した工数をまとめ、評価分析を実施する。
	SBOM 適用範囲	実証できた項目を明確にし、SBOM 適用範囲に関しての改定案を検討する。
	取引契約における要求事項	開発委託先やサードパーティ製ソフトウェアのベンダーとの契約について、サイバーセキュリティに関する条項を検討する。

4.5.5. 実証結果

4.5.5.1. SBOM 初期導入

4.5.5.1.1. SBOM 管理ツールの選定基準

SBOM 管理ツールを選定するに当たり重要視した観点は以下のとおりである。

- SBOM を SPDX 形式で出力可否
- スニペット解析可否
- 価格

SBOM フォーマットは複数存在しているが、実証事業関係者と協議し、本実証では代表的なフォーマットの一つである SPDX 形式で出力可能なツールに絞ることとした。また、ソフトウェアのコードの一部をコピーして利用している場合もあるため、スニペット解析が可能か考慮した。また、有償ツールの場合、その価格差が大きいため、中小企業の資金力に応じたが購入検討可能な価格であり、今回の対象ソフトウェアの規模による費用対効果を考慮し選定した。医機連より大手企業では BlackDuck の導入実績はあるが、導入企業においては高額であるため、経営陣に説明し社内稟議を通すのは困難であることを確認した。

4.5.5.1.1.1. 有償ツール選定

有償ツールとしては、国内で入手可能な下表の 3 製品に絞って比較を行った。性能面では BlackDuck が最良であったが、価格が他と比べて 2 倍~10 倍以上高額であった。FOSSA はスニペット解析ができないため採用できないと判断した。MEND SCA については SPDX 出力のために別途アドオンツールが必要であるものの、特に大きな問題ではなく、また価格も BlackDuck と比較すると安価であるため MEND SCA を選定することとした。

表 4-12　有償ツール

ツール名	SPDX 出力	スニペット解析	価格
FOSSA	○	×	20 名程度 200 万円/年
MEND SCA	△（別途アドオンツールが必要）	○（軽微な改変のみ）	20 名から 336 万円/年
BlackDuck	○	○	ベーシック 600 万円/年 フルスペック 1300 万円/年 バイナリ解析込み 2600 万円/年

なお、MEND SCA に関しては、ライセンス費用に標準的なサポート内容も含まれるが、それ以外にも有償サポートを選択できる。参考に国内販売代理店である株式会社日立ソリューションズ（以下、「日立 SOL」という。）でのサポート内容及び価格を下表に示す。

表 4-13　日立 SOL での MEND SCA のサポート内容及び価格

分類	サポート内容	価格
無償サポート	ツールの障害や不具合及び MEND SCA の標準的な使い方に関わる問い合わせに対してメールで回答する。	MEND SCA のサブスクリプション費用に含まれる。
有償サポート	無償サポートの範囲に収まらないサポートについて、「OSS よろず相談」というソリューションメニューを用いて対応する。	チケット制 50 万円/5 チケット 以降 10 万/1 チケット
有償コンサルティング	PoC や実機調査、プロセス定義、プロトタイプ開発等を伴う相談等を受け付ける。	別途見積り。

4.5.5.1.1.2. 無償ツール選定

無償ツールとして、下表の 3 製品に絞って比較を行った。

3 製品ともスニペット解析が非対応であり性能に関しても同等ではあったが、本実証では SBOM 生成と脆弱性マネジメントを連動的に行うことができる Syft&Grype を選択した。

表 4-14　無償ツール

ツール名	SPDX 出力	スニペット解析	価格
FOSSology	○	×	無償
OSS Review Kit	○	×	無償
Syft&Grype	○	×	無償

また、本実証では GitHub に公開されている NewYork-Presbyterian Hospital が主体となって開発を行っている脆弱性マネジメントツール「Daggerboard」も評価することとした。その理由として、Daggerboard は 2018 年より医療機関における SBOM の検証テストをメーカーと連携して実施しており、これを用いて SBOM の脆弱性をスキャンすることで、SBOM の提

供を受けた医療機関の受入ツールとして活用できることが期待できるためである。さらに、Daggerboard は SBOM の内容を人間が読める形で確認するための脆弱性スキャンツールであるということが、Daggerboard インストールページ[67]に説明されている。

4.5.5.1.2. 実施内容とコスト評価

前項で選定した各ツールの初期導入においての実施内容と工数について次のようにまとめた。

MEND SCA の初期導入・設定・習熟に関して下表のように工数計測を実施した。

表 4-15　MEND SCA 初期導入・設定・習熟工数計測

実証項目	担当企業	結果
MEND SCA の初期導入	KRK/LSC	MEND SCA 本体はブラウザで動作する、いわゆる SaaS(Software as a Service)であるため、インストールの必要はないが、サブスクリプションの設定が必要である。初期設定に係る工数は 0.50 人時間であった。設定に関しては日立 SOL の支援を受けながら実施したが、一般的な SaaS ツールと同じような使い勝手であり、特に不明な点もなかったため、支援がなくても工数に変動はないものと考えられる。また、共有するクライアント設定に関しては、メールで送られてきた情報をもとにログインするだけであるため、工数は 0.00 人時間とした。
設定（環境整備）	KRK/LSC	MEND SCA でソースコードをスキャンするためのアドオンツールである[Unified Agent]をダウンロードし、任意の場所に保存する。また、SBOM を生成するためのアドオンツールである[SBOM Generator]をインストールする必要がある。上記 2 つをインストールするのに要した工数は 0.83 人時間であった。 同様の内容を LSC で行った場合の工数は 0.33 人時間であった。
習熟	KRK	日立 SOL より提供されたサンプルコードを用いて MEND SCA でのスキャン方法やスキャン結果の確認方法、実際に SBOM を生成するまでの工数を計測した。基本的にコマンドプロンプトを用いた操作となることと、オプション項目が多いため、日立 SOL の支援を受けながら作業を実施した。習熟に要した工数は 5.50 人時間であった。日立 SOL の支

[67]GitHub、’ nyph-infosec/daggerboard(Public)
https://github.com/nyph-infosec/daggerboard

実証項目	担当企業	結果
		援が無かった場合は、16.0 人時間の工数がかかると予想される。
	LSC	習熟に関しては、KRK から事前に教育訓練を受け、KRK が作成したマニュアルを見ながら作業を行ったため、習熟に要した工数は 2.00 人時間であった。

Syft&Grype の初期導入・設定・習熟に関して下表のとおり工数計測を実施した。

表 4-16　Syft&Grype 初期導入・設定・習熟工数計測

実証項目	担当企業	結果
Syft&Grype の初期導入	KRK	Syft は Windows、Linux 及び MacOS 用に公開されているが、Grype は Linux 及び MacOS 用のみでしか対応していない。本実証では Windows を使用したため、Windows 上に Linux 環境（VirtualBox Ubuntu）を構築してから Syft&Grype をインストールする必要があった。そのため Linux 環境の構築に 6.83 人時間と多くの工数を要した。 Linux 環境構築後、ターミナルを用いて特定のコマンドを実行し、Syft&Grype のインストールを行った。Syft のインストールには 0.35 人時間、Grype のインストールには 0.20 人時間の工数を要した。
設定（環境整備）	KRK	インストールの際に、インストール先のディレクトリに書き込み権限が無かったためインストールすることができず、権限を付与するのに手間取ったが、インストールが完了すれば特に設定等に困ることはなかった。その設定に 1.42 人時間を要した。 また、Windows と Linux 間のデータの受け渡しにファイル共有を行ったが、その設定に 0.53 人時間を要した。
習熟	KRK	習熟に関しては主に GitHub 上の Syft&Grype のインストールページから行った。日立 SOL より提供されたサンプルコードを Syft にかけて SBOM を出力し、Grype でサンプルコードの脆弱性情報のスキャンを行い、これらに要した工数を計測した。結果は 3.79 人時間であった。

Daggerboard の初期導入・設定・習熟に関して下表のとおり工数計測を実施した。

表 4-17　Daggerboard 初期導入・設定・習熟工数計測

実証項目	担当企業	結果
Daggerboard の初期導	KRK	現時点では Daggerboard は Windows 上での起動が

入		サポートされていないため、Daggerboard のインストールを Linux 環境上で行った。GitHub 上の Daggerboard ダウンロードページから「daggerboard_docker_image.tgz」、「DaggerBoard_Installer.bin」の 2 つのファイルをダウンロードし、Linux 上の任意の場所に保存する。その後特定のコマンドを実行し、インストールを行った。インストールに要した工数は 0.75 人時間であった。（Linux 環境構築にかかる工数は含まない）
設定（環境整備）	KRK	コンテナ型アプリケーション実行環境である Docker に読み込めば、Daggerboard はブラウザで起動するので特に設定等は必要なかった。
習熟	KRK	習熟は GitHub 上の Daggerboard のインストールページに掲載されている使用方法にて行った。Syft で作成した SBOM 及びサードパーティ製の SBOM のスキャン及び Daggerboard 上に表示された結果の脆弱性情報確認にかかる工数を計測し、その習熟に 9.42 人時間の工数を要した。（当初取り込めた Xelis Dental の SBOM に 2.35 人時間を、他の SBOM を安定的に取り込めるようになるまでに 7.07 人時間を要している。）

各ツールでの初期導入・設定・習熟の工数を比較した結果を下表に示す。

表 4-18　各ツールの初期導入・設定・習熟コスト比較

ツール名	初期導入	設定	習熟
MEND SCA	0.50 人時間(KRK)	0.83 人時間(KRK)	5.50 人時間(KRK)
		0.33 人時間(LSC)	2.00 人時間(LSC)
Syft&Grype	7.38 人時間(KRK)	1.95 人時間(KRK)	3.79 人時間(KRK)
Daggerboard	0.75 人時間(KRK)	0.00 人時間(KRK)	9.42 人時間(KRK)

初期導入に関しては、Syft&Grype が他のツールと比べて多くの工数を要した結果となった。これは Syft&Grype の Grype が Linux 環境を必要としており、Linux 環境構築のために多くの工数を要したためで、Linux 環境構築の工数(6.83 人時間)を除けば、1.00 人時間以下となり、他のツールとほぼ同等の工数であるといえる。

設定に関しては、MEND SCA は日立 SOL の支援を受けて実施したため、Syft&Grype に比べ少ない工数となっている。また、KRK と LSC 社とで 0.50 人時間の工数差があるが、これは KRK が最初に設定を行ったためであり、LSC 社は KRK が作成したマニュアルに従って実施したため、工数が短縮されたものと考える。また、Syft&Grype の設定時間が MEND SCA と比較して工数がかかったが、これは Syft&Grype の取扱いに関しての資料が少なく、外部支援もないことから情報収集して独学する時間を要したためである。

習熟に関しては、Daggerboard が最も多くの工数を要することになった。その要因として、Daggerboard は今回選択した他のツールよりも説明資料が少なく、日本語化された資料が存在しない上、外部支援もないため独学の作業となったためである。さらに、SBOM を安定的にスキャンできないという問題が発生し、その解決に多くの工数を要したことも一因に挙げられる。次いで MEND SCA が多くの工数を要する結果となったが、これはソースコードスキャンや SBOM 生成においての使用方法、オプション項目の把握の他、MEND SCA に表示される情報量や機能が多いためである。なお、LSC は、KRK が作成したマニュアルと教育訓練を受けたため、2.00 人時間と KRK よりも少ない工数で習熟することができた。最後に、Syft&Grype の習熟にかかる工数が一番少ない結果となった。これは Daggerboard と同様に独学での作業となった反面、情報の少なさから必要最低限の習熟しか行うことができなかったためであり、結果として習熟工数としては MEND SCA よりも短くなった。このことから習熟時間に関してはツールごとにできる内容に差があるためであり、単純比較は難しいと考える。

4.5.5.1.3. 課題、解決ノウハウ、留意点

各ツールの初期導入においての課題･解決ノウハウ･留意点について次のようにまとめた。

MEND SCA の初期導入・設定・習熟で得られた課題を下表に示す。

表 4-19　MEND SCA の初期導入・設定・習熟の課題

項目	課題	解決ノウハウ、留意点
初期導入	メイン画面に表示される情報や使える機能が多いため、取扱説明書だけでは充分に理解することが困難である。	MEND SCA のサポートを活用する。ただし、基本的にメールでの質問、回答となるためレスポンスは悪い。
	価格が高額で中小企業では導入が難しい。	現時点の価格帯では中小企業での導入は困難であるため、有償ツール導入に当たって何らかの補助金等の支援が必要と考える。あるいは無償ツールを使用するに当たってはツールの信頼性の向上が必要と考える。
習熟	[SBOM Generator]のオプション項目が多く、何を選択すればいいか分かりにくい。	[SBOM Generator]には基本となるテンプレートやデフォルトオプション等の提供はないため、取扱説明書に記載されているオプション項目から必要となる情報を選択してコマンドを作成する必要があるが、一度テンプレートを作成すれば、その後は若干の編集で再利用が可能なため、次回以降、時間短縮できる。LSC にはマ

項目	課題	解決ノウハウ、留意点
		ニュアルとあらかじめ作成したテンプレートを共有したため習熟の時間短縮になっている。

Syft&Grype の初期導入・設定・習熟で得られた課題を下表に示す。

表 4-20　Syft&Grype の初期導入・設定・習熟の課題

項目	課題	解決ノウハウ、留意点
初期導入	Linux 環境の構築に時間がかかる。	Linux 環境の構築が行われているかどうかでツールの導入にかかる工数に大きな差が出る。ただし、Linux 環境の構築に関する情報は書籍やインターネット上で豊富に得られる。
習熟	Syft&Grype に関する資料が少ない。	Syft&Grype の習熟に使用できるような資料が整備されておらず、基本的には GitHub 上の Syft&Grype のインストールページに記載されている説明を主に参照していくことになった。SBOM 生成や脆弱性の特定をしていく過程の中でトライ＆エラーを繰り返し、習熟していく必要があった。このことから、習熟のための資料が整備されていることがツール選択の基準の一つといえる。 なお、習熟の中で得られた内容を社内でマニュアル化し情報を共有すれば、習熟の時間を短縮することが可能である。

Daggerboard の初期導入・設定・習熟で得られた課題を下表に示す。

表 4-21　Daggerboard の初期導入・設定・習熟の課題

項目	課題	解決ノウハウ、留意点
初期導入	Linux 環境構築の必要がある。	現時点で Daggerboard は Linux 環境（VirtualBox Ubuntu）での動作確認しかされていないため、Linux 環境の構築が必要となり Grype と同様、導入の際に多くの工数を要することになる。
習熟	Daggerboard に関する資料がほぼ存在していない。	Daggerboard 習熟に使用できるような資料がほぼ存在せず、GitHub 上の

項目	課題	解決ノウハウ、留意点
		Daggerboard のインストールページ及び 2022/6/23-24 開催の Linux security summit で使用された資料「SBOM Ingestion and Analysis at NewYork-Presbyterian Hospital」[68]からしか情報を得ることができなかった。他の無償ツールと比べても圧倒的に資料が少なく、未だ不明点も多い。

[68]NewYork-Presbyterian Hospital
https://static.sched.com/hosted_files/ossna2022/27/opensourcesummit_daggerboard_embeddedfonts.pdf

4.5.5.2. SBOM の生成・共有

4.5.5.2.1. 実施内容とコスト評価

KRK のソフトウェアの SBOM 生成について、工数計測の実施結果及び評価結果を下表に示す。

表 4-22　各ツールの SBOM 生成に要した工数とその評価

使用ツール	概要	担当企業	手順の詳細	工数	評価
手動	歯科用 CT 及び装置の制御ソフトウェアの SBOM を手動で生成した。	LSC A 社	ソフトウェアで使用している Framework,lib 等を確認し、リンクしているものを目視で確認する。	LSC（歯科用 CT） 30.0 人時間 A 社（F/W） 1.50 人時間	OTS（OSS を含む既成ソフトウェア）等（構成レイヤー）の定義が不明確だったため、かなりの工数を要することとなった。部品数が多いソフトウェア等の正確な SBOM 作成は困難なものになると考えられる。
MEND SCA	歯科用 CT 及び装置の制御ソフトウェアのソースコードを MEND SCA でスキャンし SBOM を生成した。	KRK	MEND SCA のスキャンツールである Unified Agent でソースコードをスキャンし、SBOM 生成ツールである SBOM Generator で SBOM を生成した。	0.15 人時間	Unified Agent、SBOM Generator 共にコマンドでの操作となる。基本となるコマンドをテンプレートとして決めておけば、特にソースコードのスキャン及び SBOM 生成は短時間で実施できる。
Syft&Grype	開発委託先との取引契約上、歯科用 CT のソースコードを入手することが困難	KRK	KRK の実行環境上で Syft を使用し、サンプルコードをスキャンし SPDX 形式で SBOM を生成した。	0.27 人時間	SBOM 生成はコマンドでの操作となる。Syft には実行できるコマンドが多数存在するが、必要なコマンドは限られてくるため、必要なコマンドをあらかじめ把握していれば

使用ツール	概要	担当企業	手順の詳細	工数	評価
	であったため、日立 SOL より提供されたサンプルコードをスキャンして SBOM を生成した。				SBOM の生成に関しては短時間で実行可能である。 なお、本実証では左記の理由から歯科用 CT の代わりにサンプルコードで SBOM を生成したが、SBOM の生成時間はコードのサイズにより多少の違いはあるものの、サンプルコードでの工数評価で代用できると考える。

作成した SBOM が IMDRF 追補ガイダンス案に規定されている 8 要素を満たせているかを確認した結果を下表に示す。

表 4-23　各ツールの IMDRF 追補ガイダンス案 8 要素の確認工数

使用ツール	概要	担当企業	手順の詳細	工数	評価
手動	生成した SBOM が IMDRF 追補ガイダンス案の 8 要素を満たせているかを確認した。	KRK	手動及び各ツールで作成した SBOM を目視にて、IMDRF 追補ガイダンス案の 8 要素を満たせているかの確認を実施した。	2.00 人時間	「コンポーネントハッシュ」や「固有識別子」、「関係」等をどのように記載すればいいのか不明なため、現時点で 8 要素全てを網羅するのは難しい。
MEND SCA				1.00 人時間	IMDRF 追補ガイダンス案の 8 要素を網羅していることを確認した。
Syft&Grype				0.63 人時間	8 要素の内、「ソフトウェアコンポーネントのベンダー」及び「関係」が検出されず、8 要素全ての網羅はできなかった。また、「作成者名」が

使用ツール	概要	担当企業	手順の詳細	工数	評価
					SBOM ツールの提供者である Anchore,Inc となっていた。

サードパーティ製ソフトウェアのベンダーから提供された SBOM について、IMDRF 追補ガイダンス案に規定されている 8 要素を満たせているかを確認した結果を下表に示す。

表 4-24　サードパーティ製 SBOM の IMDRF 追補ガイダンス案 8 要素の確認工数

ソフトウェア名/ベンダー	概要	担当企業	手順の詳細	工数	評価
XelisDental/INFINITT	提供された SBOM が IMDRF 追補ガイダンス案の 8 要素を満たせているかを確認した。	KRK	提供された SBOM を目視にて、IMDRF 追補ガイダンス案の 8 要素を満たせているかの確認を実施した。	0.50 人時間	INFINITT 社が手動で作成し、提供された SBOM の内容を確認したところ、「コンポーネントハッシュ」項目の抜けがあったが、指摘後、修正された SBOM が提供され、IMDRF 追補ガイダンス案の 8 要素を全て網羅していることを確認した。
VivaLO8/Varex				0.50 人時間	Varex 社が Microsoft SBOM Tool で生成し、提供された SBOM の内容を確認したところ、IMDRF 追補ガイダンス案の 8 要素の内「タイムスタンプ」「ソフトウェアコンポーネ

ソフトウェア名/ベンダー	概要	担当企業	手順の詳細	工数	評価
					ント名」「コンポーネントハッシュ」「固有識別子」の 4 要素は確認できた。なお、「作成者名」は SBOM ツールの提供者である Microsoft となっていた。
NvidiaCUDA4/P 社				ー	本実証の期間においては、SBOM の提供を依頼した PC メーカーから SBOM が提供されなかったため、評価はできなかったが、本来であればサードパーティ製ソフトウェアより提供を受けられなかった場合においても、医療機器製造販売業者は構成管理上ソフトウェア検証が必要となる。

歯科用 CT について、MEND SCA で生成した SBOM と、手動で生成した SBOM の比較結果の一部を下表に示す。

表 4-25　手動と MEND SCA での検出結果

手動での検出			MEND SCA での検出		
ベンダー	ソフトウェア名	バージョン	実行環境	開発環境	ソースコード
Microsoft	.NET Framework	4.8	未検出	未検出	未検出
Microsoft	SQL Server 2012 Express Edition	11.0.2100.60(x64)Feb 10 2012 19:39:15	未検出	未検出	未検出
Microsoft	Visual Studio 2012 再頒布パッケージ	―	mfc110.dll msvcr110-11.00.51106.1.dll msvcp110-11.00.51106.1.dll	未検出	未検出
Apache Software Foundation	log4net	2.0.4	log4net-1.2.13.0.dll	未検出	未検出
Apache Software Foundation	log4net	2.0.12	log4net-2.0.12.0.dll	log4net.2.0.12.nupkg: アンインストールパッケージ log4net-2.0.12.0.dll	未検出
Microsoft	Microsoft.Bcl.AsyncInterfaces	5.0.0	Microsoft.Bcl.AsyncInterfaces-5.0.20.51904.dll	microsoft.bcl.asyncinterfaces.5.0.0.nupkg: アンインストールパッケージ Microsoft.Bcl.AsyncInterfaces-	未検出

手動での検出			MEND SCA での検出		
ベンダー	ソフトウェア名	バージョン	実行環境	開発環境	ソースコード
				5.0.20.51904.dll	
Microsoft	System.Buffers	4.5.1	System.Buffers-4.6.28619.01.dll	system.buffers.4.5.1.nupkg:アンインストールパッケージ System.Buffers-4.6.28619.01.dll	未検出
Microsoft	System.Memory	4.5.4	System.Memory-4.6.28619.01.dll	system.memory.4.5.4.nupkg:アンインストールパッケージ System.Memory-4.6.28619.01.dll	未検出
Microsoft	System.Numerics.Vectors	4.5.0	System.Numerics.Vectors-4.6.26515.06.dll	system.numerics.vectors.4.5.0.nupkg: アンインストールパッケージ System.Numerics.Vectors-4.6.25519.03.dll System.Numerics.Vectors-4.6.26515.06.dll	未検出
Independent JPEG Group	JPEGLIB	6b	未検出	mozjpeg-jpeg-ari	gdcm-gdcm-2.6.9
ルーヴァン・カトリック大学	OpenJPEG	2.0.0	OpenJpegDll.dll	openjpeg-version.2.0 bmabms2obj-ConvBMS OpenJpegDll.dll	openjpeg-wg1n6848 bmabms2obj-ConvBMS

<table>
<tr><th colspan="3">手動での検出</th><th colspan="3">MEND SCA での検出</th></tr>
<tr><th>ベンダー</th><th>ソフトウェア名</th><th>バージョン</th><th>実行環境</th><th>開発環境</th><th>ソースコード</th></tr>
<tr><td>OpenJS Foundation</td><td>jQuery</td><td>1.11.1</td><td colspan="3" rowspan="3">jquery-ui-i18n-1.8.18.js
jquery.effects.drop-1.8.17.min.js
jquery.ui.progressbar-1.8.18.min.js
jquery.ui.datepicker-1.8.18.min.js
jquery.effects.blind-1.8.17.min.js
jquery.effects.bounce-1.8.17.min.js
jquery.effects.clip-1.8.17.min.js
jquery.effects.core-1.8.18.min.js
jquery.effects.explode-1.8.17.min.js
…etc</td></tr>
<tr><td>Klaus Hartl</td><td>jQuery Cookie</td><td>1.4.1</td></tr>
<tr><td>Keith Wood</td><td>jQuery Keypad</td><td>2.1.1</td></tr>
</table>

Microsoft 社の.NET Framework や SQL Server 2012 Express Edition は MEND SCA では検出されなかった。その理由を MEND SCA の販売元である日立 SOL に問い合わせたところ、アーカイブやシンボリックリンクで実態がスキャン対象にない場合には検出は難しいとの回答を得た。.NET Framework や SQL Server 2012 Express Edition 等は、アーカイブしている仕様となるため、MEND SCA で SBOM を生成する場合には注意が必要となる。

また MEND SCA では、ソースコード、実行環境及び開発環境でスキャンした場合で SBOM 検出数に差が出ることが分かった。実証の結果としては、開発環境でのスキャンが他と比べて最も多く検出を行うことができたが、これは開発環境の方では、実際に製品には使用されないアンインストールパッケージが検出されてしまったためであり、実行環境でのスキャンが実態に近い結果となった。

Independent JPEG Group に関しては、ソースコードスキャンでは検出されている部品が、実行環境では検出されないという結果となった。

jQuery, jQuery Cookie, jQuery Keypad については、MEND SCA でスキャンをしたところ詳細な SBOM として検出される結果となった。これは開発者が認識している以上に MEND SCA が詳細に検出できるという結果であると考えられるが、前述のように MEND SCA でも検出されない部品があり、また MEND SCA もソースコードと実行環境でのスキャンとで結果が異なることから、より正確な SBOM 作成には、手動、MEND SCA（ソースコード、実行環境）の複数で SBOM を生成し、比較を行うことが望ましいという結果となった。

日立 SOL から提供されたサンプルコードについて、MEND SCA で生成した SBOM と Syft&Grype で生成した SBOM の比較結果を下表にまとめた。

サンプルコードを MEND SCA と Syft&Grype とでスキャンした結果、MEND SCA での検出数が 13 部品なのに対し、Syft&Grype では 8 部品の検出結果となり、Syft&Grype の方では 5 部品の検出漏れがあった。プログラムの規模としては小さいサンプルコードでの検証であるが、Syft&Grype だけでは、正確性のある SBOM 生成は無理があると考えられる。

表 4-26　MEND SCA と Syft&Grype での検出結果

MEND SCA での検出	Syft&Grype での検出
ant-1.5.2.jar	ant-1.5.2.jar
ant-optional-1.5.2.jar	ant-optional-1.5.2.jar
commons-fileupload-1.1.jar	commons-fileupload-1.1.jar
commons-io-1.1.jar	commons-io-1.1.jar
commons-logging-1.0.4.jar	commons-logging-1.0.4.jar
db-charmer-v1.6.10	未検出
hub-performance-probe-1.0	未検出
lucene-1.4.3.jar	lucene-1.4.3.jar
openssl-OpenSSL-fips-2_0_16	未検出
openvpn.exe	未検出
RuneKit.exe	未検出
xercesImpl.jar	xercesImpl.jar
xml-apis.jar	xml-apis.jar

4.5.5.2.2. 開発委託先との SBOM 共有方法

手動で作成した SBOM 及び各ツールで生成した SBOM を開発委託先と共有する方法について下表のとおり検討した。

手動で作成した SBOM 及び Syft&Grype で生成した SBOM を共有する場合、基本的には SBOM を電子ファイルとして受け取り、各社で個別に管理する必要がある。そのため、各社で文書管理方法を確立するとともに、SBOM が更新された場合の管理方法を確立する必要がある。

MEND SCA については SBOM を開発委託先とウェブ上で共有できるため、ファイル形式の SBOM のやり取りが不要である。

このことから開発委託先との SBOM 共有に関しては MEND SCA の方が有効であるといえる。

表 4-27　開発委託先との SBOM 共有方法検討

ツール名	共有方法及び評価	工数
手動作成	開発委託先で作成された SBOM を電子ファイルとして提供してもらう必要がある。提供された	文書管理：1.00 人時間

ツール名	共有方法及び評価	工数
	SBOM は「医療機器及び体外診断用医薬品の製造管理及び品質管理の基準に関する省令」（平成 16 年厚生労働省令第 169 号）（以下、「QMS 省令」という。）に基づいて、外部文書として社内で管理を行う必要がある。	/SBOM　提供ごと
MEND SCA	スキャン結果をウェブ上の同じインターフェースで閲覧・管理でき、実体のファイル形式で管理する必要がないため、共有に関して工数は発生しない。	0.00 人時間
Syft&Grype	MEND SCA のようにスキャン結果を共通のインターフェースで閲覧できる機能がないため、手動作成と同様に SBOM を電子ファイルとして提供を受けて、外部文書として社内で管理を行う必要がある。	文書管理：1.00 人時間 /SBOM 提供ごと

4.5.5.2.3. 医療機関への SBOM の提供方法

医療機関への SBOM 提供方法について、IMDRF 追補ガイダンス案の「5.3 SBOM の配布」の章にある表を参考に検討し、その結果を下表に示す。

表 4-28　医療機関への SBOM 提供方法検討結果

IMDRF 追補ガイダンス案に記載された配布方法	検討結果	評価
製造業者からの顧客向けセキュリティ文書に含める。	ソフトウェアによっては膨大な量となるため、紙媒体は非現実的である。医療機器の文書類は電子化が進んでいるため、セキュリティ文書の一部として電子ファイルでの提供がよいと考えるが、更新した SBOM の配布方法が課題となる。	△
製造業者が、別途（電子）文書で提供する。	上記のとおり SBOM については電子ファイルでの配布が現実的であるが、更新した SBOM の配布方法が課題となる。	△
ディスプレイ、（間接的な）参照又はダウンロードを通して医療機器からアクセス可能	あらかじめ医療機器のソフトウェアに SBOM のアクセスのためのユーザーインターフェースを実装した場合、医療機関側としても、更新ごと	○

IMDRF 追補ガイダンス案に記載された配布方法	検討結果	評価
	に配布されたSBOMの差替えを行うことなく、正しい SBOM を取得することが可能となる。	
医療機器の API からアクセス可能	医療機器のソフトウェアから、製造販売業者のリポジトリーにアクセスしダウンロードする方法が考えられるが、そもそもネットワーク環境がない場合は SBOM を取得できない。	△
製造業者が管理するリポジトリー	製造販売業者のサイトからSBOMをダウンロードできる方法が考えられる。ただし医療機関が自ら SBOM の取得をしなければならない。	○
集中型リポジトリー	一番理想的だと思われるが、現在のところ、このような仕組みはない。	×

この結果から、以下の 2 つの方法が最も現実的であると判断した。

・ディスプレイ、（間接的な）参照又はダウンロードを通して医療機器からアクセス可能

・製造業者が管理するリポジトリー

ただし理想的なのは、集中型リポジトリーであり、各製造販売業者が医療機器の SBOM を登録し、医療機関がアクセスしダウンロードするという方法が、製造業者、医療機関共に労力が少ないと考えられる。

4.5.5.2.4. 課題、解決ノウハウ、留意点

SBOM 生成・共有についての課題を下表に示す。

表 4-29　SBOM 生成・共有の課題

項目	課題	解決ノウハウ、留意点
[1] 手動での SBOM の作成	開発者が認識している部品しか検出できない。意図せずインストールされる部品等は、手動での検出が難しい。	手動では間接利用部品の特定が困難であるため、ツール使用によるソースコード解析、バイナリ解析、スニペット解析で部品を検出後、手動で検出した部品を合わせることによって検出漏れを極力なくすことができる。そのためツールとの併用が望ましい。

項目	課題	解決ノウハウ、留意点
	手動でSPDX形式のSBOMを作成するのが困難である。	最初から.tag や.json 形式の SBOM を手動で作成することは困難。EXCEL で SPDX 形式の SBOM を作成することが現実的だが、テンプレートとなるフォーマットが必要である。
[2] MEND SCA での SBOM の生成	実行環境、開発環境、ソースコードでスキャンした場合とで、検出される部品数が異なる。	基本的には実行環境でのスキャンが望ましいが、ソースコードでしか検出されない部品もあるため、実行環境、ソースコードの両方でスキャンしSBOMを生成することが望ましい。
	アーカイブやシンボリックリンクで実体がそこにない場合等は MEND SCA では検出されない。	.NET Framework 等、ソースコードに含まれずアーカイブされているソフトウェアの場合はツールで検出されないため、手動で作成した SBOM と比較し、最終的に一つにまとめる必要がある。
	検出されなかった部品を、MEND SCA に後から追加することができない。	手動、実行環境、ソースコードで検出される部品が違うため、1 つのソフトウェアに対して複数の SBOM を作成し比較する必要が生じる。そのため、MEND SCA 上で SBOM を編集できることが望ましいが、現在そういった機能はない。
[3] Syft&Grype での SBOM 生成	MEND SCA でのスキャン結果と比較して、検出漏れが多数ある。	Syft&Grype だけで運用するのは SBOM の正確性に欠けるため、単独で使用することは避けるべきである。
	IMDRF 追補ガイダンス案に規定されている 8 要素を満たせていない。	現時点では 8 要素全ては網羅されておらず、足りない項目を手動で追加していく必要がある。ただし、Syft&Grype は頻繁にアップデートがされているため、今後のアップデート次第では 8 要素全てが網羅される可能性もある。

項目	課題	解決ノウハウ、留意点
[4] サードパーティ製ソフトウェアのベンダーから入手したSBOM	医療系のソフトウェアを取り扱っているベンダーは、SBOM の重要性を理解しており、対応も速かった一方で、医療関係に直接かかわっていないベンダーは、SBOM 提示について認識してないため、SBOM の提供を受けられなかった。	SBOM の提示が受けられない場合であっても、製造販売業者が責任をもって構成管理を把握し、ソフトウェアの検証を行う必要がある。 今後、法制化され SBOM の認知度が上がれば、提供される可能性もある。
	提供された SBOM の内、IMDRF 追補ガイダンス案に規定されている 8 要素を満たせていないものがあった。	IMDRF 追補ガイダンス案の 8 要素を満たせていないベンダーは、SBOM 生成に Microsoft 社の SBOM Tool を使用していたため、今後、ツールのアップデートで解決される可能性もある。
[5] SBOM の共有方法（開発委託先）	既存の文書管理方法では SBOM を管理するのに不十分な可能性がある。	社内文書管理規定の見直し及び必要に応じて改定を要する場合もある。
[6] SBOM の共有方法（医療機関）	医療機関に提供する SBOM のファイル形式や提供した SBOM を医療機関がどのような方法で活用するのかが明確でない。	将来的に提供された SBOM を医療機関が機械判読するならば、製造販売業者と医療機関の間で、ある程度標準化された SBOM のフォーマットの確立が必要である。

4.5.5.3. SBOM の活用・管理

4.5.5.3.1. 実施内容とコスト評価

サイバーセキュリティにおいて、リアルタイムで更新されていく脆弱性に対応するには、いかに早く脆弱性を発見し対応していくかが重要となる。手動及び各ツールで作成した SBOM をもとに、脆弱性の特定と対処方法の特定までの工数を算出した。

本来は対象医療機器である歯科用 CT について、手動及び各ツールでの脆弱性の特定にかかる工数を比較検証することが望ましいが、歯科用 CT の開発環境に Linux 環境の構築ができないため、Syft&Grype で検証することが困難であった。そのため、手動及び各ツールで統一して評価が可能なサンプルコードを用い、これが対象医療機器に使用されているものと想定して検証を行った。サンプルコードで検出された「commons-fileupload-1.1.jar」について、各ツールでの CVSS のスコアと対処方法特定までの工数計測を実施した。

なお、歯科用 CT については、開発環境で利用可能であった MEND SCA で検出された脆弱性及び対処方法の机上評価を行った。今回の机上評価では、CVSS 値の高いソフトウェアを事例として取り上げる。MEND SCA より、歯科用 CT 内の「KinkiWebApp」という Web 送信用ソフト内の「System.Text.Encodings.Web-4.6.26515.06.dll」に CVE-ID: CVE-2021-26701, 深刻度:High, CVSS 値 9.8 の脆弱性が検出された。今回の実証においては、「System.Text.Encodings.Web-4.6.26515.06.dll」について脆弱性マネジメントフローに従って机上検討した。

サンプルコードについて、手動及び各ツールでの脆弱性の特定、深刻度評価及び対処方法特定の分析結果を以下に示す。また、MEND SCA については、対象医療機器「歯科用 CT」評価結果を示す。

4.5.5.3.1.1. 手動

手動で生成した SBOM の場合、部品 1 つ 1 つの脆弱性の有無について手作業で特定しなければならず、部品数に応じた工数がかかることになる。また、発見された脆弱性については深刻度評価を実施し、さらに対処方法の特定に至るまでも、全て手作業で実施するため、さらに工数がかかることとなる。また脆弱性情報に関してはリアルタイムで更新されていくということを考えると、手動での SBOM 運用は現実的ではないといえる。

4.5.5.3.1.2. MEND SCA

MEND SCA はスキャンしたプログラムの結果をクラウドにアップロードし、ウェブ上で検出された部品をリスト化し、リスト中に脆弱性が有る場合、脆弱性の深刻度（CVSS）も含めて視覚的に表示してくれる。また、発見された脆弱性に対し簡単な内容説明と情報源となる参考サイト、さらに対処方法に至るまでを提供する。仮に検出された脆弱性のある部品が、数世代前の古い OSS であった場合、MEND SCA は検出されたバージョンから、現在に至るまでの全ての対処方法を示すことができるほか、何をすれば脆弱性がない状態にできるのかが示されるため、脆弱性の発見から深刻度評価、対処方法の特定までを非常に迅速に行える。「commons-fileupload-1.1.jar」の脆弱性情報を例にすると、最新バージョンである ver.1.4 にアップデートすれば脆弱性が解消されると考えられる（表 4-25 参照）。

またクラウド方式であるため、開発委託先とも同じインターフェース上で情報共有できるため、その後の修正作業を考慮した場合、開発委託先との意思疎通もスムーズに行え、工数削減に大いに役立つものと思われる。

次に、対象医療機器の開発環境で検出された脆弱性について、脆弱性マネジメントフローに従って、机上検討を行い、その工数を算出した。歯科用 CT 内の「KinkiWebApp」という Web 送信用ソフト内の「System.Text.Encodings.Web-4.6.26515.06.dll」に CVE-ID: CVE-2021-26701, 深刻度:High, CVSS 値 9.8 の脆弱性が検出された。これはリモートを経由して攻撃者が title オプションを介して任意の Web スクリプト又は HTML を挿入する可能性があるとされる脆弱性である。これによって KinkiWebApp のソースコードの一部を外部から改ざんされる可能性があり、具体的には HTML で表示しているソフトウェアのインターフェース部の表示不良や撮影された画像が表示されないなどの影響が出るものと想定されるが、実際にサイバー攻撃を受けた場合、どのような症状となるのかの詳細な特定は難しい。また、この脆弱性に対しサイバー攻撃を受けた場合、前述のとおり画像診断が困難となる可能性が高いが、患者に危害を加えるものではない。この脆弱性の対処方法を MEND SCA で確認した結果、.NET Framework を 5.01 以降のバージョンにアップデートすれば解消されることが分かった(表 4-26 参照)。これに伴うソースコードの変更は不要であるため、対処工数としては.NET Framework 5.01 以降へのアップデート工数のみで、およそ 0.5 人時間と想定される。ただし他のサードパーティ製ソフトウェアへの影響も考慮し、全体的なシステムとしてのテストを実施する必要があるため、社内に構築されている検証機でテストを実施する。それに要する工数として別途 16 人時間程度かかると予想される。

4.5.5.3.1.3. Syft&Grype

Syft で作成した SBOM を Grype でスキャンすることで、検出された OSS の脆弱性及び深刻度も表示される。また、「.table」形式の SBOM の場合、全てではないものの対処方法の一部も一覧で表示される。ただし脆弱性が検出された「commons-fileupload-1.1.jar」について MEND SCA と比較した場合、Grype での脆弱性の対処方法は多少古く、MEND SCA が最新バージョンである ver.1.4 へのアップデートを示すのに対し、ver.1.3.3 へのアップデートまでの修正内容にとどまっている（表 4-25 参照）。そのため、Grype でのスキャン結果をもとに修正したとしても脆弱性が解消しきれていない可能性がある。これを回避するためには、そこから最新の脆弱性のないバージョンを特定するまでを手作業で行わなければならない。また、脆弱性の検出数も MEND SCA と比較して少ないため、MEND SCA の検証結果と同じレベルまでの修正を行うためには結果的に工数が発生することになる。

4.5.5.3.1.4. Daggerboard

Daggerboard では「.spdx」、「.xml」、「.cyclonedx」のファイル形式の SBOM の脆弱性を検出することができるが、MEND SCA や Grype と比較して明らかに検出される脆弱性が少なく、検出された CVE ID の情報も一番古いものになっている（表 4-25 参照）。さらに、脆弱性に対する対処方法もないため、CVE ID を検索し古いバージョンの脆弱性情報から地道に収集していかなくてはならない。「commons-fileupload-1.1.jar」の例で考えると、「CVE-2013-0248」を NIST NVD で検索すると、対処方法として ver.1.3 にアップデートを実施することで脆弱性が解消されるという情報が得られる。しかしながら、ver.1.3 の脆弱性情報をさらに調査すると、今度は ver.1.3 についての脆弱性が検出される。結果として、ver1.3 での脆弱性を調査し対処方法を特定し、また次のバージョンの脆弱性を調査する、ということを繰り返さなくてはならず、逆に最新版であるバージョンを調べ、その脆弱性を確認した方が早い結果となってしまう。そして上記の手順を繰り返したとしても、Grype 同様 ver.1.3.3 までのアップデートで脆弱性を発見できず、ver.1.4 のバージョンアップまでたどり着けないことになる。1 つの部品でも多くの工数を要するため、これを SBOM に記載されている部品全てに実施するのは現実的ではないと考えられる。

以上の点から、現時点では、Daggerboard は脆弱性情報を得て、そこから脆弱性を特定しその対処方法にいたるまでに役立てるツールとして用いることは不向きと考えられる。一方、当初意図したとおり、医療機関が脆弱性情報を得るための SBOM 受入の際のツールとして活用することが期待されるが、現時点では Daggerboard は他のツールと比べて検出漏れが多く脆弱性の発見を見逃す可能性もあり、ツール機能が十分ではないと考える。

表 4-30　ツール別の脆弱性の特定及び深刻度評価工数

使用ツール	概要	担当企業	手順の詳細	工数	評価
手動	手動で脆弱性の特定、深刻度評価を検証する。	LSC	手動で CVE サイトから脆弱性情報を入手し、脆弱性の深刻度評価と対処方法特定までの工数を計測した。	脆弱性の特定及び深刻度評価： 0.23 人時間 対処方法の特定： 0.54 人時間	手動で CVE サイトから脆弱性の深刻度を入手すること自体は、それほど時間はかからない。ただしそこから対処方法を追っていくことに時間を要することとなる。
MEND SCA	MEND SCA で脆弱性の特定、深刻度評価を検証する。	KRK	MEND SCA で検出された脆弱性のある 1 部品を対象に、脆弱性の特定と、深刻度評価及び対処方法特定までの工数を計測した。	脆弱性の特定及び深刻度評価： 0.08 人時間 対処方法の特定： 0.18 人時間	MEND SCA 上で CVE ID を始めとする脆弱性情報及び深刻度、またその解決方法までを示してくれる。これらは共有情報として開発委託先間でお互い管理できるため、月次レポート等の必要もなく、開発委託先との間で脆弱性情報の管理、監視がリアルタイムにできるという点において、専門の人員を割くことなく管理ができるというのは大きなメリットといえる。特に部品数が多いソフトウェアの場合には大きな助けとなる。 また、特筆すべきこととして MEND SCA には脆弱性情報として CVE ID とは別の独自の WS ID というものがある。これは CVE 識別番号管理サイトや NIST NVD にまだ公開されていない MEND SCA によって検証された既知の脆弱性に付与された ID であり、脆弱性情報収集の観点からも、より精度の高い脆弱性情報が検出されることが期待される。 なお MEND SCA は「CVSS v3 の値を用いて、

使用ツール	概要	担当企業	手順の詳細	工数	評価
					CVSS v2 のスコアレンジに従って分類している」という特殊な脆弱性の深刻度評価を採用しており、CVSS v3 スコアレンジでは Critical と表示されるところが High と表示される点には留意すべきである。
Syft&Grype	Syft&Grype で脆弱性の特定、深刻度評価を検証する。	KRK	Syft で生成した SBOM を Grype でスキャンし脆弱性を検出し、検出された 1 部品を対象に脆弱性の特定と、深刻度評価及び対処方法特定までの工数を計測した。	Grype スキャン： 0.12 人時間 脆弱性の特定及び深刻度評価： 0.39 人時間 対処方法の特定： 0.28 人時間	Syft で生成した SBOM を Grype でスキャンすることで OSS ごとの脆弱性がリスト化され、深刻度も確認できる。また table 形式で出力すれば検出された脆弱性に対し、対処方法の情報も出る。ただし、全ての情報がでるわけではないため、基本的には CVE 識別番号管理サイトや NIST NVD から CVE ID を検索し、脆弱性の内容及び対処方法を調べることとなる。また、Grype は CVE ID とは別に GHSA ID が割り当てられるものもある。これは GitHub Advisory Database という OSS のセキュリティリスクに関するアドバイザリーが記載されているデータベース用の独自の ID で、CVE ID と紐づけられ、対応する脆弱性の内容及び対処方法が記載されており、情報収集に非常に役立つ。
Daggerboard	Daggerboard で脆弱性の特定、深刻度評価を検	KRK	MEND SCA で生成した SBOM を Daggerboard でスキャンし、検出された 1 部	Daggerboard スキャン： 0.15 人時間	「.spdx」、「.xml」、「.cyclonedx」のファイル形式の SBOM を取り込むことができ、Grype 同様 OSS ごとの脆弱性がリスト化され、深刻度

使用ツール	概要	担当企業	手順の詳細	工数	評価
	証する。		品を対象に脆弱性の特定と、深刻度評価及び対処方法特定までの工数を計測した。	深刻度評価： 0.22 人時間 対処方法方の特定： 0.55 人時間	も確認できる。しかしながら、MEND SCA や Grype とは違い、脆弱性の対処方法の情報が出ないため、CVE ID や Daggerboard 上の脆弱性内容説明欄から対処方法を調査していくことになる。

各ツールで検出された「commons-fileupload-1.1.jar」の脆弱性の特定、深刻度評価及び対処方法についてのスキャン結果を示す。

表 4-31　ツール別の脆弱性の特定及び深刻度評価及び対処方法特定比較

Daggerboard			Syft&Grype			MEND SCA		
CVE ID 等	CVSS 値 深刻度	主な脆弱性の対処方法	CVE ID 等	CVSS 値 深刻度	主な脆弱性の対処方法	CVE ID 等	CVSS 値 深刻度	主な脆弱性の対処方法
CVE-2013-0248	4.3(CVSSv3) Medium	無し	CVE-2013-0248	3.3(CVSSv2) Low	無し	CVE-2013-0248	4.0(CVSSv3) Medium	1.3 へのバージョンアップ
			CVE-2014-0050	7.5(CVSSv2) High	無し	CVE-2014-0050	7.3(CVSSv3) High	1.3.2 へのバージョンアップ
			CVE-2016-1000031	7.5(CVSSv2) 9.8(CVSSv3) Critical	無し	CVE-2016-1000031	9.8(CVSSv3) High	1.3.3 へのバージョンアップ
			CVE-2016-3092	7.8(CVSSv2)	無し	CVE-2016-3092	7.5(CVSSv3)	1.3.2 へ

Daggerboard			Syft&Grype			MEND SCA		
CVE ID 等	CVSS 値 深刻度	主な脆弱性の対処方法	CVE ID 等	CVSS 値 深刻度	主な脆弱性の対処方法	CVE ID 等	CVSS 値 深刻度	主な脆弱性の対処方法
				7.5(CVSSv3) High			High	のバージョンアップ
			GHSA-vm69-474v-7q2w (CVE-2013-0248)	3.3(CVSSv2) Low	1.2.2 へのバージョンアップ	CVE-2013-2186	7.3(CVSSv3) High	1.3.1 へのバージョンアップ
			GHSA-xx68-jfcg-xmmf (CVE-2014-0050)	7.5(CVSSv2) High	1.3.1 へのバージョンアップ	WS-2014-0034	7.5(CVSSv3) High	1.4 へのバージョンアップ
			GHSA-7x9j-7223-rg5m (CVE-2016-1000031)	7.5(CVSSv2) 9.8(CVSSv3) Critical	1.3.3 へのバージョンアップ			
			GHSA-fvm3-cfvj-gxqq (CVE-2016-3092)	7.8(CVSSv2) 7.5(CVSSv3) High	1.3.2 へのバージョンアップ			
			GHSA-qx6h-9567-5fqw (CVE-2013-2186)	7.5(CVSSv2) High	1.3.1 へのバージョンアップ			

表 4-32　MEND SCA による対象医療機器の脆弱性の特定及び深刻度評価

CVE ID	CVSS 値・深刻度	主な脆弱性の対処方法
CVE-2021-26701	7.5(CVSSv2) 9.8(CVSSv3) High	.NET Framework を 5.01 以降へのバージョンアップ

4.5.5.3.2. 脆弱性マネジメントフロー

以下のとおり、医療機器に用いられるソフトウェアの脆弱性マネジメントフローを検討した。

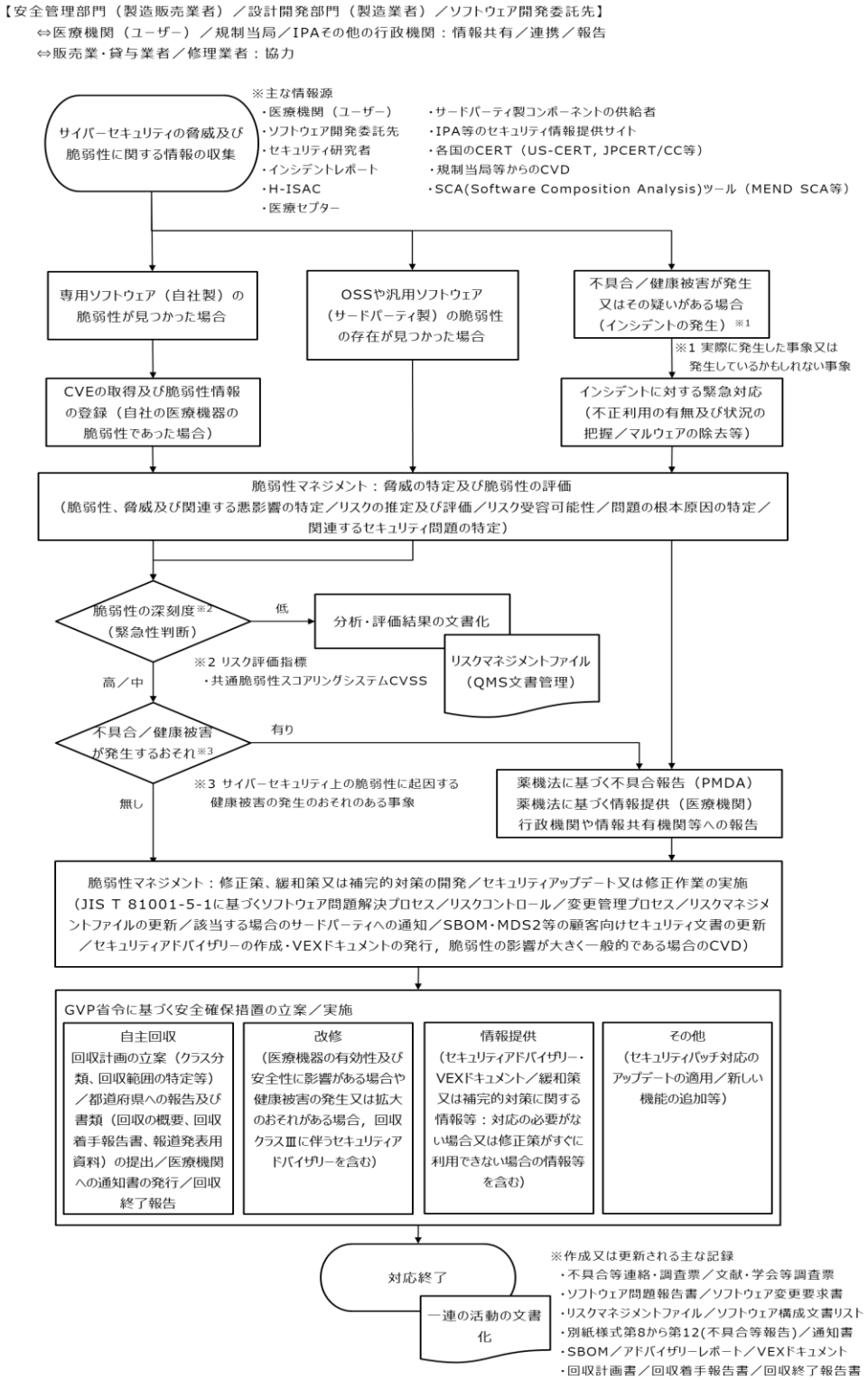


図 4-12　脆弱性マネジメントフロー

4.5.5.3.3. 脆弱性マネジメント工数の想定

SBOM は製品リリース時に顧客に情報共有のために渡される最新の製品ソフトウェア情報であり、SBOM は情報共有のためのツールである。日々報告される脆弱性に対する対応については IMDRF ガイダンスに従って管理を行うこととする。

本実証では、「図 4-3　脆弱性マネジメントフロー」に則って、脆弱性の発見又は認知からその対応の終了までに要する各作業項目に係る工数を机上で算出し、SBOM 活用効果について検証した。脆弱性マネジメントの想定シナリオとしては表 4-27 に示すサンプルコードが対象医療機器に使用されているとした場合と、表 4-28 に示す対象医療機器歯科用 CT の場合の 2 つのシナリオを想定して検討した。なお、脆弱性対応における改修及びパッチ適用については基本的には医療機器製販業者の指示に従って、プログラム医療機器（**SaMD**：Software as a プログラム医療機器（SaMD：Software as a Medical Device）　Medical Device）を除いて医療機器修理業者が行うこととされている。KRK では医療機器修理業についても許可を受けている。

表 4-33　サンプルコードにおける脆弱性マネジメントの想定シナリオ

サンプルコードにおける脆弱性マネジメントの想定シナリオ
すでに販売済みの対象医療機器において、30 部品程度の小規模なソフトウェアを対象に脆弱性調査を実施した結果、サードパーティ製のソフトウェアに CVSS 値が中程度(スコア値 4.0~6.9)の脆弱性が発見された。脆弱性がソフトウェアに与える影響を調査し、リスク評価を実施した結果、健康被害が発生するおそれはないものの、将来的に脅威となり得る可能性もあるため、改修（回収の内、医療機器を動かさずに修理、調整等を行うこと、又は、医療機器プログラムの場合は、品質、有効性及び安全性に問題のない新しいプログラムに置き換える又は修正すること）を行うことを決定した。なお、「医薬品・医療機器等の回収について」（平成 26 年 11 月 21 日薬食発 1121 第 10 号厚生労働省医薬食品局長通知）に基づく回収に係るクラス分類については、クラスⅡに該当するものと判断した。

また、各想定シナリオにおける作業項目の工数設定について検討した。共通の作業項目である社内会議、社内文書作成及び社内文書更新、社内・社外向け文書作成及び更新時におけるレビューについて、過去の経験からの平均的な工数を考慮して実証における標準的な工数を以下に示す。

表 4-34　本実証における文書作成・更新、レビュー、社内会議等共通項目の工数

作業項目	標準設定工数
社内会議	2.00 人時間
社内文書作成	1.50 人時間
社内文書レビュー※	4.00 人時間
社内文書更新	1.00 人時間

社内文書更新レビュー※	4.00 人時間
社外（医療機関、行政機関）向け文書作成	5.00 人時間
社外向け文書レビュー※	12.00 人時間

※本脆弱性マネジメント工数の算出における KRK のレビュー体制は以下のとおりとする。

1）脆弱性マネジメントフローの適切な段階において体系的なレビューを行う。このレビューは、次の事項を目的として実施する。

a） 作成又は更新した文書が妥当であるかどうかを評価する。

b） 必要な処置又は修正を明確にし、提案する。

2）レビューへの参加者には、経営陣、QMS 管理責任者、当該文書に関連する部門及び場合によっては開発委託先、専門家等を含め、全員を召集しての会議、個別の打合せやメール配信等を通して実施する。

また、各シナリオを想定した際の工数の考え方について、表 4 - 30、表 4 - 31 のとおり工数を設定した。文書作成、文書レビューについては作成する文書の難易度、分量によって異なる。また、当該工数は企業規模によって異なる場合があることに留意する必要がある。

サンプルコードの脆弱性対応の工数については、対象医療機器で用いられていないため、机上評価で医療機器への影響を評価することは難しい。手動及びツールによる工数比較のため、本実証では一医療機関に導入されている一医療機器についてソフトウェア改修を行うという想定で机上評価を行った。

表 4-35　脆弱性マネジメントにおける標準設定工数(サンプルコード)

作業項目	作業	標準設定工数
脆弱性情報の収集	脆弱性の調査	表 4-24 の工数を採用
	安全管理情報の検討	社内文書作成 1.50 人時間
脆弱性マネジメント	問題の調査	2.00 人時間
	ソフトウェア問題報告及びセキュリティアドバイザリー作成	社内文書作成 1.50 人時間
リスクアセスメント	リスク推定・評価	社内会議 2.00 人時間・社内文書更新 1.00 人時間
修正作業	ソフトウェア変更要求書	社内文書作成 1.50 人時間
	修正策（リスクコントロール手段）の開発	表 4-24 の工数+社内会議 2.00 人時間
	プログラム修正	ソフトウェア 1 部品修正 8.00 人時間
修正作業・修正策の実装	ソフトウェア設計仕様書、ソフトウェア要求仕様書	社内文書更新 1.00 人時間×2 文書
	修正策の影響特定	社内文書作成 1.5 人時間+社内文書更新 1.00 人時間×2 文

作業項目	作業	標準設定工数
		書
	ソフトウェアテスト結果報告書	検証作業 24.0 人時間+社内文書作成 1.50 人時間×2
	検証結果報告書、妥当性確認報告書	システム全体の検証 16.0 人時間+社内文書作成 1.50 人時間×2+社内文書レビュー4.00 人時間
リスクマネジメント	リスクマネジメントファイルの更新	社内会議 1.50 人時間
	トレーサビリティチャートの更新	社内文書更新 1.00 人時間
	残留リスクの有無	社内文書更新 1.00 人時間
脆弱性マネジメント	SBOM 更新	表 4-16 の工数を採用
	SBOM 更新レビュー	社外向け文書レビュー12.0 人時間
安全確保措置	安全確保措置の立案	社内会議 2.00 人時間
	改修計画の立案	社内文書作成 1.50 人時間
	都道府県への報告	社外（医療機関、行政機関）向け文書作成 5.00 人時間×2 文書+社外向け文書レビュー12.00 人時間×2 文書
	医療機関への情報提供	社内会議 2.00 人時間+社外（医療機関、行政機関）向け文書作成 5.00 人時間+社外向け文書レビュー12.00 人時間
	改修の実施	1 件当たりの実作業時間 1.00 人時間
	改修終了報告	社外（医療機関、行政機関）向け文書作成 5.00 人時間+社外向け文書レビュー12.00 人時間

また、実際に対象医療機器におけるソフトウェア脆弱性の対応を行う上での工数を以下のとおり想定した。脆弱性マネジメントフローに則って、脆弱性マネジメントの一連全体工数について机上評価を行った。ソフトウェア脆弱性 1 件におけるアップグレード 1 件の対応を行う上で算出根拠となる標準工数の考え方を以下表に示す。

表 4-36　脆弱性マネジメントにおける標準設定工数(対象医療機器のソフトウェア)

作業項目	作業	標準設定工数
脆弱性情報の収集	脆弱性の調査	表 4-24 の MEND SCA の工数

作業項目	作業	標準設定工数
		を採用
	安全管理情報の検討	社内文書作成 1.50 人時間
脆弱性マネジメント	問題の調査	2.00 人時間
	ソフトウェア問題報報告及びセキュリティアドバイザリー作成	社内文書作成 1.50 人時間
リスクアセスメント	リスク推定・評価	社内会議 2.00 人時間・社内文書更新 1.00 人時間
修正作業	ソフトウェア変更要求書	社内文書作成 1.50 人時間
	修正策（リスクコントロール手段）の開発	表 4-24 の MEND SCA の工数+社内会議 2.00 人時間
	プログラム修正	ソフトウェア 1 部品アップデート 0.50 人時間
修正作業・修正策の実装	ソフトウェア設計仕様書、ソフトウェア要求仕様書	社内文書更新 1.00 人時間×2 文書
	修正策の影響特定	社内文書作成 1.5 人時間+社内文書更新 1.00 人時間×2 文書
	ソフトウェアテスト結果報告書	社内文書作成 1.50 人時間×2
	検証結果報告書、妥当性確認報告書	医療機関を模擬した、社内での対象医療機器システム全体 16.0 人時間+社内文書作成 1.50 人時間×2+社内文書レビュー 4.00 人時間
リスクマネジメント	リスクマネジメントファイルの更新	社内会議 1.50 人時間
	トレーサビリティチャートの更新	社内文書更新 1.00 人時間
	残留リスクの有無	社内文書更新 1.00 人時間
脆弱性マネジメント	SBOM 更新	表 4-16 の MEND SCA の工数を採用
	SBOM 更新レビュー	社外向け文書レビュー12.0 人時間
安全確保措置	安全確保措置の立案	社内会議 2.00 人時間
	改修計画の立案	社内文書作成 1.50 人時間
	都道府県への報告	社外（医療機関、行政機関）向け文書作成 5.00 人時間×2 文書+社外向け文書レビュー 12.00 人時間×2 文書
	医療機関への情報提供	社内会議 2.00 人時間+社外（医

作業項目	作業	標準設定工数
		療機関、行政機関）向け文書作成 5.00 人時間+社外向け文書レビュー12.00 人時間
	改修の実施	1 件当たりの実作業時間 1.00 人時間
	改修終了報告	社外（医療機関、行政機関）向け文書作成 5.00 人時間+社外向け文書レビュー12.00 人時間

4.5.5.3.4. 脆弱性マネジメント工数の結果

手動及びツールごとにおける脆弱性マネジメント工数の算出結果を下表に示す。

表 4-37　サンプルコードでの脆弱性マネジメント工数の算出結果

<table>
<tr><th>項目・フェーズ</th><th>作業・文書</th><th>作業内容</th><th>MEND SCA</th><th>Syft& Grype</th><th>Daggerboard</th><th>手動 SBOM</th><th>SBOM 無し</th></tr>
<tr><td rowspan="2">脆弱性情報の収集</td><td>脆弱性の調査</td><td>対象ソフトウェアの SBOM をツールでスキャンし、脆弱性の有無を調査する。SBOM 無しの場合は手作業で調査を実施する。</td><td>0.08 人時間
（内訳：深刻度評価 0.08 人時間）</td><td>0.51 人時間
（内訳：スキャン 0.12 人時間+深刻度評価 0.39 人時間）</td><td>0.37 人時間
（内訳：スキャン 0.15 人時間+深刻度評価 0.22 人時間）</td><td>6.90 人時間
（内訳：CVE サイト調査 0.23 人時間×30 部品）</td><td>13.80 人時間
（内訳：CVE サイト、ソースコード精査 0.46 人時間×30 部品）</td></tr>
<tr><td>安全管理情報の検討</td><td>入手した安全管理情報を検討し、文献・学会等調査票を起票する。</td><td colspan="6">1.50 人時間（文書作成）</td></tr>
<tr><td rowspan="2">脆弱性マネジメント</td><td>問題の調査</td><td>開発委託先と協議し、当該脆弱性が製品のセキュリティに及ぼす影響を調査する。</td><td colspan="6">2.00 人時間(開発委託先への調査依頼し、調査報告を受ける)</td></tr>
<tr><td>ソフトウェア問題報報告及びセキュリティアドバイザリー作成</td><td>脆弱性情報の概要及び問題の調査結果等をソフトウェア問題報告作成、セキュリティアドバイザリーを作成する。</td><td colspan="6">1.50 人時間（文書作成）</td></tr>
<tr><td>リスクアセスメント</td><td>リスク推定</td><td>リスクマネジメント会議を招集しリスク推定を実施する。</td><td colspan="6">2.00 人時間（社内会議）
1.00 人時間（文書更新）</td></tr>
</table>

項目・フェーズ	作業・文書	作業内容	MEND SCA	Syft& Grype	Daggerboard	手動 SBOM	SBOM 無し
	リスク評価	受容可能性及び健康被害のおそれの有無を判断する。その結果をリスク分析表に記載する。					
修正作業	ソフトウェア変更要求書	問題の是正に必要な処置のための変更要求を、「ソフトウェア変更要求書」を用いて作成する。	1.50 人時間(文書作成)				
	修正策（リスクコントロール手段）の開発	開発委託先と脆弱性に対する対処方法を特定し、修正策を協議する。	2.18 人時間（内訳：対処方法特定 0.18 人時間+会議 2.00 人時間）	2.28 人時間（内訳：対処方法特定 0.28 人時間+会議 2.00 人時間）	2.55 人時間（内訳：対処方法特定 0.55 人時間+会議 2.00 人時間）	2.54 人時間（内訳：対処方法特定 0.54 人時間+会議 2.00 人時間）	2.54 人時間（内訳：対処方法特定 0.54 人時間+会議 2.00 人時間）
	プログラム修正	選択したリスクコントロール手段でプログラム修正を実施する。	8.00 人時間（ソフトウェア 1 部品修正）				
修正作業・修正策の実装	ソフトウェア設計仕様書、ソフトウェア要求仕様書	リスクコントロール手段をソフトウェア設計仕様書、ソフトウェア要求仕様書に含める。	2.00 人時間(文書更新 1.00 人時間×2)				
	修正策の影響特定	修正策の結果、やり直しが必要な全てのアクティビティを特定し、実装する。（ソフトウェア成果物一覧表、ソフトウェアトレーサビリティチャート、ソフトウェア発行連絡書、1 文書作成、2 文書更新）	3.50 人時間 （内訳：文書作成 1.50 人時間、文書更新 2.00 人時間）				

<table>
<tr><th>項目・フェーズ</th><th>作業・文書</th><th>作業内容</th><th>MEND SCA</th><th>Syft& Grype</th><th>Daggerboar d</th><th>手動 SBOM</th><th>SBOM 無し</th></tr>
<tr><td rowspan="3"></td><td>ソフトウェアテスト結果報告書</td><td>修正策を検証し、その結果をソフトウェアテスト結果報告書、ソフトウェア変更要求書に文書化する。</td><td colspan="5">27.0 人時間
（内訳：検証作業：24.0 人時間、文書作成：1.50 人時間×2）</td></tr>
<tr><td rowspan="2">検証結果報告書、妥当性確認報告書</td><td>システム全体での検証及び妥当性確認結果を報告書に記載。</td><td colspan="5">19.0 人時間
（システム全体の検証:16.0 人時間、文書作成：1.50 人時間×2）</td></tr>
<tr><td>妥当性確認報告書のレビュー。</td><td colspan="5">4.00 人時間（社内文書レビュー）</td></tr>
<tr><td rowspan="4">リスクマネジメント</td><td rowspan="2">リスクマネジメントファイルの更新</td><td>変更内容を分析して、危険状態の一因となる潜在的原因が新たに生じていないか、新たなソフトウェアリスクコントロール手段が必要かどうかを確認する。</td><td colspan="5" rowspan="2">1.50 人時間(社内会議)</td></tr>
<tr><td>ソフトウェアの修正によって既存のリスクコントロール手段の妨げとなる危険性を伴わないかを確認する。</td></tr>
<tr><td>トレーサビリティチャートの更新</td><td>危険状態からリスクコントロール手段の検証までのトレーサビリティを文書化する。</td><td colspan="5">1.00 人時間(文書更新)</td></tr>
<tr><td>残留リスクの有無</td><td>修正の結果、新たなリスク及び残留リスクがないかを確認し、リスクマネジメント報告書に記載する。</td><td colspan="5">1.00 人時間(文書更新)</td></tr>
</table>

項目・フェーズ	作業・文書	作業内容	MEND SCA	Syft& Grype	Daggerboard	手動 SBOM	SBOM 無し
脆弱性マネジメント	SBOM 更新	SBOM を更新する。（本実証での SBOM 生成工数を記載した。なお手動での更新に関しては、文書更新の工数を記載し、Daggerboard に関しては SBOM 生成ができないため、手動と同様に文書更新工数を記載している）	0.15 人時間	0.27 人時間	1.00 人時間 (社内文書更新)	1.00 人時間 (社内文書更新)	1.00 人時間 (社内文書更新)
		更新された SBOM のレビュー	12.0 人時間（社外向け文書レビュー）				
安全確保措置	安全確保措置の立案	安全性に係る不具合事象等を防止するために必要な措置を立案する。	2.00 人時間(社内会議)				
	改修計画の立案	改修計画を立案し、「改修計画書」を作成する。	1.50 人時間(社内文書作成)				
	都道府県への報告	「医療機器改修の概要」「改修着手報告書」「報道発表資料（該当する場合）」を作成し、都道府県に提出する。（本シナリオでは「報道発表資料」は該当しないため、「医療機器改修の概要」「改修着手報告書」の 2 文書とする。）	10.00 人時間 (行政機関向け文書作成 5.00 人時間×2)				
		「医療機器改修の概要」及び「改修着手報告書」のレビュー	24.00 人時間（社外向け文書レビュー12.00 人時間×2）				
	医療機関への情	対象範囲の特定、通知の方法の検討	2.00 人時間（社内会議）				

項目・フェーズ	作業・文書	作業内容	MEND SCA	Syft& Grype	Daggerboard	手動 SBOM	SBOM 無し
	報提供	使用の中止、不具合の発生を回避する方法や対処方法等の情報を提供する。	5.00 人時間(医療機関向け文書作成)				
		通知書のレビュー	12.00 人時間（社外向け文書レビュー）				
	改修の実施	現況確認、改修作業を実施する。（サービス要員が医療機関へ訪問し修正バッチファイルで修正し、動作確認する。）	1.00 人時間（1 件当たりの実作業時間）				
	改修終了報告	「改修終了報告書」を作成し、都道府県に提出する。	5.00 人時間(行政機関向け文書作成)				
		「改修終了報告書」のレビュー	12.00 人時間（社外向け文書レビュー）				
合計（人時間）			165.41 人時間	166.06 人時間	166.92 人時間	173.44 人時間	180.34 人時間

結果として、ツール別にそれぞれ MEND SCA が 165.41 人時間、Syft&Grype が 166.06 人時間、Daggerboard が 166.92 人時間、手動 SBOM が 173.44 人時間、SBOM 無しが 180.34 人時間となった。

まず、有償及び無償ツールを使用した場合の工数について比較する。無償ツールは有償ツールと比べて、数字上の工数としては 1.00 人時間以内の差に留まったものの無償ツールの深刻度評価及び対処方法で得られる情報の精度は低い可能性があり、脆弱性の対処方法に記載されているバージョンのソースコードを GitHub 等から入手し、これを再度スキャンして、対処方法として妥当かを確認する必要があるため、実際にはここに記載した以上の工数が発生する可能性が高い。

次に、SBOM 有りと SBOM 無しの場合の工数について比較する。十分に精査された SBOM 有りの場合と比べて、「基本要件基準」第 12 条第 2 項の規定（プログラムを用いた医療機器に対する配慮）で要求されている JIS T 2304:2017（医療機器ソフトウェア―ソフトウェアライフサイクルプロセス）に基づくソフトウェア構成管理による SBOM 無しの場合では、対象医療機器を構成するソフトウェア部品の粒度が SBOM に比べて粗いことも考えられる。そのため、ポータルサイト等から脆弱性の情報を発見又は認知したとしても、実際にその部品が対象医療機器のソフトウェ

アに使用されているかどうかを調査するためには、ソースコードを精査する必要がある。脆弱性情報の内容によりソースコードの精査工数にも影響があることから単純な机上算出は難しいが、SBOM 無しの場合、ソースコード精査工数を加味して手動 SBOM を用いた場合の 2 倍の工数とした。

本実証では脆弱性の発見及び認知について SBOM を活用した場合として工数算出を実施したが、本来、SBOM ツールの活用については脆弱性の発見及び認知の一手段に過ぎず、「図 4-3　脆弱性マネジメントフロー」に記載している主な情報源から幅広く入手する必要がある。医療機器分野において SBOM は医療機関及び製造販売業者等の関連するステークホルダーとの間で SBOM を共有することで透明性を維持し、相互に連携して医療機器のサイバーセキュリティを確保することを目的としている。そのため本項の工数算出結果については、あくまで一例であることに留意する必要がある。MEND SCA の脆弱性検出に基づいて対象医療機器における「脆弱性マネジメント工数算出結果を行った。対象医療機器については、現在、医療機関 117 施設に対して、117 台導入されている。今回のソフトウェア脆弱性について、施設毎に 1 件のパッチ適用の修正策の実施が必要となる。実際に医療機関で導入されている改修対象となる医療機器全てに対して実施する工数について、以下表に示す。

表 4-38　対象医療機器での脆弱性マネジメント工数の算出結果

項目・フェーズ	作業・文書	作業内容	工数
脆弱性情報の収集	脆弱性の調査	対象医療機器のソフトウェアのソースコードを MEND SCA でスキャンし脆弱性の有無を確認する。	0.08 人時間（内訳：深刻度評価 0.08 人時間）
	安全管理情報の検討	入手した安全管理情報を検討し、文献・学会等調査票を起票する。	1.50 人時間（文書作成）
脆弱性マネジメント	問題の調査	開発委託先と協議し、当該脆弱性が製品のセキュリティに及ぼす影響を調査する。	2.00 人時間(開発委託先への調査依頼し、調査報告を受ける)
	ソフトウェア問題報告及びセキュリティアドバイザリー作成	脆弱性情報の概要及び問題の調査結果等をソフトウェア問題報告及びセキュリティアドバイザリーを作成する。	1.50 人時間（文書作成）
リスクアセスメント	リスク推定	リスクマネジメント会議を招集しリスク推定を実施す	2.00 人時間（社内会議）

項目・フェーズ	作業・文書	作業内容	工数
		る。	1.00 人時間（文書更新）
	リスク評価	受容可能性及び健康被害のおそれの有無を判断する。その結果をリスク分析表に記載する。	
修正作業	ソフトウェア変更要求書	問題の是正に必要な処置のための変更要求を、「ソフトウェア変更要求書」を用いて作成する。	1.50 人時間(文書作成)
	修正策（リスクコントロール手段）の開発	開発委託先と脆弱性に対する対処方法を特定し、修正策を協議する。	2.18 人時間（内訳：対処方法特定 0.18 人時間+会議 2.00 人時間）
	プログラム修正	選択したリスクコントロール手段でプログラム修正を実施する。	0.50 人時間（ソフトウェア 1 部品アップグレード）
修正作業・修正策の実装	ソフトウェア設計仕様書、ソフトウェア要求仕様書	リスクコントロール手段をソフトウェア設計仕様書、ソフトウェア要求仕様書に含める。	2.00 人時間(文書更新 1.00 人時間×2)
	修正策の影響特定	修正策の結果、やり直しが必要な全てのアクティビティを特定し、実装する。（ソフトウェア成果物一覧表、ソフトウェアトレーサビリティチャート、ソフトウェア発行連絡書、1 文書作成、2 文書更新）	3.50 人時間 （内訳：文書作成 1.50 人時間、文書更新 2.00 人時間）
	ソフトウェアテスト結果報告書	修正策を検証し、その結果をソフトウェアテスト結果報告書、ソフトウェア変更要求書に文書化する。	3.00 人時間 （内訳：文書作成 1.50 人時間×2）
	検証結果報告書、妥当性確認報告書	医療機関を模擬した、社内での対象医療機器システム全体での検証を実施し、検証結果及び妥当性確認結果を報告書に記載。	19.00 人時間 （内訳：検証作業 16.00 人時間、文書作成 1.50 人時間×2）

項目・フェーズ	作業・文書	作業内容	工数
		妥当性確認報告書のレビュー。	4.00 人時間（社内文書レビュー）
リスクマネジメント	リスクマネジメントファイルの更新	変更内容を分析して、危険状態の一因となる潜在的原因が新たに生じていないか、新たなソフトウェアリスクコントロール手段が必要かどうかを確認する。	1.50 人時間(社内会議)
		ソフトウェアの修正によって既存のリスクコントロール手段の妨げとなる危険性を伴わないかを確認する。	
	トレーサビリティチャートの更新	危険状態からリスクコントロール手段の検証までのトレーサビリティを文書化する。	1.00 人時間(文書更新)
	残留リスクの有無	修正の結果、新たなリスク及び残留リスクがないかを確認し、リスクマネジメント報告書に記載する。	1.00 人時間(文書更新)
脆弱性マネジメント	SBOM 更新	SBOM を更新する。（本実証での SBOM 生成工数を記載した。なお手動での更新に関しては、文書更新の工数を記載し、Daggerboard に関しては SBOM 生成ができないため、手動と同様に文書更新工数を記載している）	0.15 人時間
		更新された SBOM のレビュー	12.0 人時間（社外向け文書レビュー）
安全確保措置	安全確保措置の立案	安全性に係る不具合事象等を防止するために必要な措置を立案する。	2.00 人時間(社内会議)
	改修計画の立案	改修計画を立案し、「改修計画書」を作成する。	1.50 人時間(社内文書作成)
	都道府県への報告	「医療機器改修の概要」「改修着手報告書」「報道発表資料（該当する場合）」を作成し、都道府県に提出する。（本シナリオでは「報道発表資料」は該当しないた	10.00 人時間 (行政機関向け文書作成 5.00 人時間×2)

項目・フェーズ	作業・文書	作業内容	工数
		め、「医療機器改修の概要」「改修着手報告書」の 2 文書とする。）	
		「医療機器改修の概要」及び「改修着手報告書」のレビュー	24.00 人時間（社外向け文書レビュー12.00 人時間×2）
	医療機関への情報提供	対象範囲の特定、通知の方法の検討	2.00 人時間（社内会議）
		使用の中止、不具合の発生を回避する方法や対処方法等のセキュリティアドバイザリー情報を提供する。	5.00 人時間(医療機関向け文書作成)
		通知書のレビュー	12.00 人時間（社外向け文書レビュー）
	改修の実施	改修作業を実施する。（サービス要員が医療機関へ訪問し修正パッチファイルで修正し、動作確認する。）	117.00 人時間(1.00 人時間（1 件当たりの実作業時間）×117 台)
	改修終了報告	「改修終了報告書」を作成し、都道府県に提出する。	5.00 人時間(行政機関向け文書作成)
		「改修終了報告書」のレビュー	12.00 人時間（社外向け文書レビュー）
合計（人時間）			249.91 人時間

4.5.5.3.5. 脆弱性マネジメントに関する分析

サンプルコード、対象医療機器のソフトウェアの脆弱性対応について、脆弱性マネジメントフローに則って机上検討を行ったが、想定したシナリオが異なるため、一概に比較することはできない。

MEND SCA を使用した場合の　サンプルコードと象医機器のソフトウェア脆弱性対応では、脆弱性マネジメントにおける修正作業、修正作業・修正策の実装については、対象医機器のソフトウェア脆弱性対処と比較して時間を要すると評価された。この理由として、サンプルコードでは脆弱性が確認されたファイルのバージョンアップに伴い、ソースコードの修正まで必要になることを想定していたのに対し、対象医療機器の場合は、.NET Framework のアップグレードのみでソースコードの修正が不要であったためである。

このことから修正を要する規模、修正による影響範囲や脆弱性の検出されたソフトウェア部品の使用箇所により影響を与える医療機器の機能等

によって、ソフトウェアの修正工数及び検証工数については大きく変動する可能性があると考える。

また、修正策がすぐに医療機関で対応できない場合等、必要に応じて医療機関にサイバーセキュリティにおける脅威の緩和策及び医療機器に影響する脆弱性を緩和するための IT 機器の構成、既知の脆弱性の悪用に対応する方法等もセキュリティアドバイザリーに含める必要もある。医療機関において修正策が対応可能でない場合、適切な脆弱性の緩和策又は補完的対策、例えばファイアウォールの設置等の実施を検討する必要がある。修正策又は緩和策の展開失敗について想定し、元の状態に戻す方法やその報告方法を事前に検討した上で、確立しておく必要がある。今回の机上評価では、対象医療機器のソフトウェア脆弱性について、インシデント発生を考慮していないが、実際に医療機器のソフトウェア脆弱性に起因してインシデントが発生した場合には、薬機法第 68 条の 10 第 1 項の規定により厚生労働省に対して報告することが義務づけられている。また、これらの報告については、薬機法第 68 条の 13 第 3 項の規定に基づき、平成 16 年 4 月より PMDA に対して報告することが義務づけられているため、医療機器製販業者は不具合の報告を行う必要がある[69]。

薬機法第 68 条の 9 では、医療機器製造販売業者は医療機器の製造販売をし、医療機器の使用によって保健衛生上の危害が発生し、又は拡大するおそれがあることを知つたときは、これを防止するために廃棄、回収、販売の停止、情報の提供その他必要な措置を講じなければならないとされている[70]。ソフトウェア脆弱性に起因する医療機器の自主的な回収、改修等について、医療機器製造販売業者が判断して実施する必要があるが、ノウハウの蓄積が少ないため医療機器製造販売業者で判断することが難しいと考えられる。行政機関等から判断の際の考え方や基準等が示され、研修会等で医療機器における脆弱性マネジメントの実践的な対応について教育が必要と考える。基本要件基準の一部改正により、セキュリティ要件が令和 5 年 4 月 1 日以降適用になるが、セキュリティ対策を実施するに当たりそのためのコスト負担、人材の確保についても補助金等がない場合、特に、中小企業の医療機器製販業者では資金力及び人的リソースの不足が予測されるため対応が難しい。

製品のライフサイクル全体における医療機器のセキュリティ対策については、「共同責任(Shared Responsibility)」のもと、医療機器産業界だけでなく、医療機関、規制当局、情報共有分析機関等全てのステークホルダーと連携可能な体制整備が求められる。

---

[69] 独立行政法人医薬品医療機器総合機構、企業からの報告（副作用・感染症・不具合報告）：https://www.pmda.go.jp/safety/reports/mah/0008.html

[70] 厚生労働省、医薬品等回収関連情報：https://www.mhlw.go.jp/stf/seisakunitsuite/bunya/kenkou_iryou/iyakuhin/kaisyu/index.html

4.5.5.3.6. 医療機器セキュリティポリシーにおけるサポート計画

製造販売業者は、IMDRF ガイダンスに基づく「TPLC に関するリスクマネジメント原則」に従って、サードパーティ製ソフトウェアの提供業者の倒産や買収による突然のサポート停止等を含め、脆弱性発生によるリスクがあることを理解した上で、医療機器製品の品質、有効性及び安全性が確保されるよう、製造販売業者の責任によって必要な対策を設計段階で講じる必要がある。サードパーティ製の開発ツールを含むサードパーティ製ソフトウェア部品のライフサイクルは、医療機器製品のライフサイクルに比べ短く、合致していることは少ない。このため、製造販売業者は、サードパーティ製ソフトウェア部品について、世代を超えた管理又は代替をあらかじめ計画することも必要となる。また、製造販売業者は、製品のセキュリティポリシー（以下、製品ポリシー）にしたがって、製品の具体的なサポート計画や手順書を取扱説明書どの顧客向け文書として整備する必要がある。医療機器においては製品の EOL よりも使用する汎用 OS 等ソフトウェア汎用部品の EOL が先行することも多い。この場合、サポート終了前に OS を変更する等のアップデート計画を市販前に行い、医療機関又は規制当局に開示できるようにしなければならない。

今回の実証対象機器である KR-X SCAN について、現行のサポート計画では、医療機器に用いられるソフトウェア製品の EOL、EOS について考慮して策定していなかった。本実証では、ソフトウェア部品の EOL、EOS を特定した上で、製品の TPLC を考慮したサイバーセキュリティに対応した計画への見直しを検討した。

まずは、現行のサポート計画を下表に示す。当該製品の耐用年数は 7 年としている。

手引書及び IMDRF ガイダンスでは、製造販売開始から EOL 及び EOS 並びにサポート終了後の段階ごとのセキュリティサポートの計画を医療機関に対し提供することを求めている。手引書「図 2　製品ライフサイクルにおけるレガシー医療機器の概念フレームワーク」を以下のとおり、引用する。

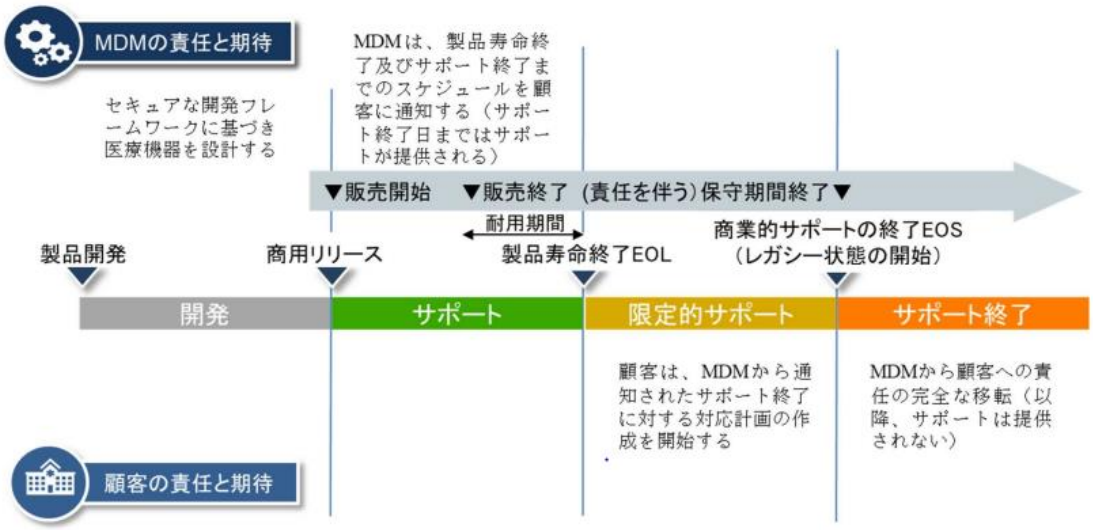


図 2　製品ライフサイクルにおけるレガシー医療機器の概念フレームワーク
（MDM：医療機器の製造販売業者）

（出所）「医療機器のサイバーセキュリティの確保及び徹底に係る手引書について（令和 3 年 12 月 24 日付け薬生機審発 1224 第 1 号、薬生安発 1224 第 1 号）」図 2 より抜粋

図 4-13　製品ライフサイクルにおけるレガシー医療機器の概念フレームワーク

このため製品を構成するハードウェア及びソフトウェア部品の EOL 及び EOS を考慮したサポート計画に見直す必要がある。なお、レガシー医療機器とは、「現在のサイバーセキュリティの脅威に対して合理的に保護できない医療機器」をいう。

まず、製造販売開始から EOL 及び EOS 並びにサポート終了後の段階ごとに、医療機関に提供する情報及び主なサポート内容を下表に示す。

表 4-39　各マイルストーンにおける医療機関に提供する情報

マイルストーン	提供する情報
製造販売開始	● セキュリティポリシー ● EOL 及び EOS 計画 ● アップグレード計画 ● 取扱説明書、セキュリティ文書 ● MDS2（Manufacturer Disclosure Statement for Medical Device Security）、SBOM ● サポート計画
EOL	● EOL の通知 ● EOS の計画 ● その他の更新情報
EOS	● EOS の通知 ● サポート計画を除くその他の更新情報

表 4-40　各ライフサイクルにおける医療機関に提供する主なサポート内容

期間	主なサポート内容
サポート期間（製造販売開始から EOL まで）	● ソフトウェア脆弱性の監視及び対応 ● 医療機関向け注意喚起のセキュリティアドバイザリーの提供 ● 医療機器のサイバーセキュリティ EOL 等、ライフサイクルの主要なマイルストーンの通知 ● 医療機関に対し、サードパーティによる機器部品のサポート終了及び EOL の事前通知
限定的なサポート期間（EOL から EOS まで）	● ソフトウェア脆弱性の監視及び対応 ● セキュリティアドバイザリーの提供 ● EOS 以降に当該医療機器がサポート対象外となることの医療機関への事前通知 ● 医療機関が関連する責任に備えるための十分な時間を確保できるように、EOS に関するスケジュールを継続的に通知
サポート終了後の期間（EOS 以降）	● 脆弱性情報の収集 ● セキュリティアドバイザリーの提供

サポート計画の見直しを検討するに当たって、ハードウェア部品の他、製品を構成する全てのソフトウェアコンポーネントの EOL 及び EOS を特定し、考慮する必要がある。しかしながら、本実証においては、提供終了日、サポート終了日が明確である代表的なソフトウェアコンポーネントである OS に絞って更新計画を立てることとした。現在、KR-X SCAN の OS には Windows10 IoT Enterprise LTSC 2019（以下、「Windows10 LTSC 2019」という。）を搭載している。また、2023 年からは Windows10 IoT Enterprise LTSC 2021（以下、「Windows10 LTSC 2021」という。）を搭載する予定としている。各 OS の EOL 及び EOS を日本マイクロソフト株式会社サイトから確認した結果を下表に示す。

表 4-41　OS の EOL 及び EOS の確認結果

OS	提供開始日	提供終了日（EOL）	延長された終了日（EOS）
Windows10 IoT Enterprise LTSC 2019[71][72]	2018 年 11 月 13 日	2024 年 1 月 9 日	2029 年 1 月 9 日
Windows10 IoT Enterprise LTSC 2021[73]	2021 年 11 月 16 日	2027 年 1 月 12 日	2032 年 1 月 13 日

上記表を踏まえると、現行のサポート期間としていた 2034 年までの計画では OS のサポート対応ができないため、計画の見直しが必要であることが分かった。Windows10 LTSC 2019 については、Windows10 LTSC 2021 へのアップグレード計画をたてた。Windows10 LTSC 2021 の後継となる OS 機が発売される場合には、Windows の後継機へのアップデートを実施する予定である。また Windows OS のサポートにおける、開始日から提供終了日までは、サポート期間として定期的にセキュリティアップデートが実施されるのに対し、提供終了日から延長された終了日の間はサポート期間ではあるが、前述のサポート期間よりもセキュリティアップデートの頻度は年々減っていくものと考えられ、製品の EOL 後は徐々にサイバーセキュリティリスクが高まっていくことを考慮する必要がある。

サポート計画の見直しについて下表に示す。それぞれの期間において医療機関へ提供する情報、主なサポートについては表 4-35 の期間の色分けと同様である。

表 4-42　Windows10 LTSC 2019 を搭載した KR-X SCAN のサポート計画

年	Windows10 LTSC 2019 搭載版	Windows10 LTSC 2021 搭載版	実施内容
2022	サポート期間（製造販売開始から EOL まで）		● 既存の Windows10 LTSC 2019 から Windows10 LTSC 2021 へのアップグレード計画を立案 ● Windows10 LTSC 2019 から Windows10 LTSC 2021 へのアップグレードを通知
2023			● Windows10 LTSC 2021 の更新を開始
2024			● Windows10 LTSC 2021 の更新の継続案内
2025		製造販売終了	● KR-X SCAN の EOL を通知 ● KR-X SCAN の製造販売を終了
2026		EOL	● EOL の継続通知
2027			
2028		限定的なサポート期間（EOL から EOS まで）	● Windows10 LTSC 2021 から後継 OS LTSC 版へのアップグレードを通知（予定）(※後継 OS が販売された場合)
2029			
2030			● KR-X SCAN の EOS の通知

[71] 「医療機器のサイバーセキュリティの確保及び徹底に係る手引書について」図 2 より引用
[72] https://learn.microsoft.com/ja-jp/lifecycle/products/windows-10-iot-enterprise-ltsc-2019
[73] https://learn.microsoft.com/ja-jp/lifecycle/products/windows-10-iot-enterprise-ltsc-2021

年	Windows10 LTSC 2019 搭載版	Windows10 LTSC 2021 搭載版	実施内容
2031		EOS	● EOS の継続通知
2032			
2033		サポート終了後の期間（EOS 以降）	● 医療機器レガシー状態になった場合を想定した補完的対策の実施検討（※後継 OS がない場合）
2034			
2035			

4.5.5.3.7. サポート計画の分析

現行のサポート計画については、現実的に対応が難しいためサポート計画を見直す必要があり、一方で製品ポリシー開示後、医療機関に対しても説明責任を果たす必要もある。さらに、医療機器がレガシー状態になった場合を想定した対応についても検討が必要である。ファイアウォール等の補完的対策によって、セキュアな状態を保証可能な場合は、その補完的対策を含めた構成において、レガシー医療機器とはみなされないため、限定的サポート期間の延長は可能であると考えられる。

現在、Microsoft 社より Windows10 LTSC 2021 の後継 OS に関して、Windows11 IoT Enterprise の情報はあるものの、医療機器に適した LTSC 版の情報については EOL、EOS 等を含めて明確に提供されていない。そのため医療機関に対して Windows の後継 OS への更新計画を含めた通知は難しいが、社内的には Windows10 LTSC 2021 の後継となる OS が発売されれば後継 OS にアップデートを実施する予定である。KR-X SCAN 製品ポリシーとおり、当初のサポートを実施できる可能性もある。なお、後継 OS への更新に伴い、PC 関連部品等のハードウェアも合わせて変更が必要になる場合がある。

製品の EOS 後も医療機関等で使用する場合、継続して使用する責任は医療機関に移転するため、あらかじめ医療機関との認識を共有することが重要である。ただし、EOS 後においても、医療機器において発生した不具合に関する情報収集義務（薬機法 68 条の 2 の 6 第 1 項）及び行政報告義務（薬機法 68 条の 10 第 1 項）は製造販売業者に残る。EOS 後の継続した使用に関しては、決して推奨されていないが、全てのステークホルダーが理解しておかねばならず、そのために製造販売業者は顧客との連携を行い、顧客への説明責任を果たす必要がある。また、サイバーセキュリティが設計開発段階で十分配慮されていない製品が、そのまま市場に存在している 場合は、既にレガシー状態となっている可能性があることに留意する必要がある。

上記のとおり、今回の実証においては OS のみを対象として検討を行ったが、製品に搭載されている全てのソフトウェア（OSS、サードパーティ製品を含む）やハードウェア部品を考慮した場合、EOL、EOS までの期間がより短くなる場合も考えられる。各ソフトウェア構成部品で EOL、EOS が開示されているわけではないため、その特定が困難な場合も多い。それらを総合的に考慮した上で実証機器である KR-X SCAN の製品ポリシー、EOL、EOS を設定する必要がある。

薬機法では、基本要件への適合性を示すために用いられている規格 JIS T 2304:2017 により構成管理は実施している。本実証においてシステム構成図を作成して全体像を把握した上で SBOM を活用することにより、ソフトウェア構成部品をより詳細に把握することがで

きた。製品のサポート計画を立てるに当たり、SBOM は製品の EOL、EOS の計画をたてる上で役立つと考えられる。

4.5.5.3.8. 医療機関へのヒアリング

医療機関のサイバーセキュリティ対策の実態について、日本医師会総合政策研究機構（日医総研）ワーキングペーパーNo.465「医療機器に関わるサイバーセキュリティの動向」[74]、No.453「病院・診療所のサイバーセキュリティ：医療機関の情報システムの管理体制に関する実態調査から」[75]等を参考として、製造販売業者と実証関係者がヒアリング内容を検討し、ヒアリング項目を設定した。

ヒアリングは、医療機関が医療機器導入の際に、SBOM 提供の影響、SBOM 提供を受けた場合の導入コスト、運用・管理コスト、管理体制等を明らかにし、製造販売業者と医療機関との SBOM の情報共有方法、及び脆弱性マネジメントにおける活用検討を目的とした。脆弱性マネジメントフローについては、製造販売業者の立場で事前に検討を行った。医療機関ヒアリングにおける要点を以下表に示す。

2022 年 11 月 30 日に東京大学医学部附属病院　企画情報運営部　新秀直先生に対してヒアリング調査を行った。

東京大学医学部附属病院新先生へのヒアリングの結果、得られた課題については、表 4-42「医療機関ヒアリングを通じての課題抽出（2022 年 11 月 30 日時点）」にまとめた。

表 4-43　医療機関ヒアリングにおける要点

項目	ヒアリング内容
SBOM の共有・評価	医療機関が希望する SBOM の提供方法、SBOM における構成要素、IMDRF 追補ガイダンス案 8 要素、また SBOM 以外に必要とする情報。 フォーマット形式等の SBOM 共有の際の留意事項等。 SBOM 提示の有無による医療機器導入の評価の影響。
SBOM の管理・運用	製造販売業者から提供された SBOM の医療機関における管理・運用方法。その際に係る想定されるコスト、工数、人員、体制確保、必要とされる支援。
SBOM の提供手段・頻度	脆弱性対応によるソフトウェアの修正、OS 更新、その他アップデート等により SBOM に変更・更新があった際の、製造販売業者からの提供方法、適切な頻度、タイミング等。
SBOM の活用・管理（脆弱性マネジメント・VEX 活用の検討）	脆弱性マネジメントにおける製造販売業者の期待される役割。 ソフトウェアの軽微な修正に留まる脆弱性が低い場合と緊急性の高い脆弱性が発見された場合の医療機関の対応の考え方。 ソフトウェアの更新や機器の使用停止等についての医療機関の考え方。 医療機器のソフトウェアの EOL を迎える場合のソフトウェア更新計画

[74] 坂口 一樹、堤 信之(2022),日本医師会総合政策研究機構ワーキングペーパーNo.465「医療機器に関わるサイバーセキュリティの動向」https://www.jmari.med.or.jp/result/working/post-3389/

[75] 坂口 一樹、堤 信之(2021), 日本医師会総合政策研究機構ワーキングペーパーNo.453「医療機器に関わるサイバーセキュリティの動向」https://www.jmari.med.or.jp/result/working/post-233/

項目	ヒアリング内容
	に関する情報提供方法、タイミング等。 VEX の導入の活用の可能性検討。 VEX 活用時における脆弱性マネジメント、セキュリティアドバイザリー機能等の必要な情報。 脆弱性が発見された場合の情報共有の際にステークホルダー（製造販売業者、規制当局、脆弱性発見者等）との情報共有の在り方。

4.5.5.3.9. 課題、解決ノウハウ、留意点

SBOM の活用・管理についての課題を下表にまとめた。

なお MEND SCA の SBOM の活用・管理については、MEND SCA 特有の課題は無かったものの、他のツールや SBOM 無しの場合等に共通する課題を下表の[1]に記載した。

表 4-44　SBOM 活用・管理の課題

項目	課題	解決方法、留意点
[1] SBOM ツール使用の有無や SBOM の有無に関わらない共通の活用・管理	医療機関に提出するためのセキュリティアドバイザリーが作成できない。	MEND SCA についてはユーザー向けレポート作成機能として各脆弱性の情報と対処方法をまとめて出力することができるものの、医療機関向けに求められるセキュリティアドバイザリーの体裁をとったものではないため、最終的には製造販売業者で作成する必要がある。 その他のツールには現時点でセキュリティアドバイザリーレポート作成機能は存在しなかった。
	ソフトウェアの EOL、EOS が特定できない。	どのツールにおいても EOL、EOS を特定する機能がない。MEND SCA の販売元に問い合わせたところ、技術的に今後も実現が困難であるとの回答を得たため、製造販売業者で調査していくしかないが、EOL、EOS の情報がないソフトウェア部品もあるため、設計開発時には、なるべくこうした部品を使用しないということについても考慮が必要となる。
[2] 手動 SBOM や SBOM 無しの場合の活用・管理	手動 SBOM、SBOM 無しの場合、脆弱性情報の入手のタイミングが難しく、時間もかかるため、個別に担当を置く必要がある。	脆弱性情報の入手を全て手動で実施することは工数的にも無理があり、ツールと併用して実施する方が現実的といえる。
	手動での SBOM 作成について、ツールを使用した場合と比較して、検出精度が低くなる可能性がある。	手動での SBOM の場合、どうしても検出漏れの懸念がある。検出精度を高めるためにツールと併用して SBOM を作成することが望ましい。

項目	課題	解決方法、留意点
[3] Syft&Grypeでの SBOM の活用・管理	Syft&Grype は、Syft で生成した SBOM でないと Grype で脆弱性のスキャンができない。またインポートするファイル形式によって情報も異なる。	Syft で生成した SBOM には CPE 情報があり、Grype はこの CPE 情報から脆弱性情報を検出しているからと考えられ、現時点での解決策は存在しない。
	脆弱性情報の信憑性が低い。	サンプルコードのスキャン結果ではあるが、Syft には検知漏れがあることが判明した以上、その SBOM から得られる脆弱性情報には不備がある可能性がある。 また、MEND SCA と比較して脆弱性情報も対応バージョンが古いことから完全なものとはいえない。 そのため、手動と Syft&Grype との組み合わせであれば、何とか使用できるレベルではあるかと思われる。ただ Syft&Grype の脆弱性情報の信憑性は低いため、手動での精査は必要となる。 ただし、手動で SBOM を修正するとしても Grype でスキャンできるようにするためには CPE 情報を追加する必要があるため、かなりの工数を要することが考えられる。
[4] Daggerboardでの SBOM の活用・管理	一部のサードパーティ製ソフトウェア Xelis Dental の SBOM しか Daggerboard に取り込めなかった。	当初、様々な形式で出力された SBOM のいずれも Daggerboard に取り込むことができなかったが、調査の結果、取り込むことができる SBOM のファイル形式が「.spdx」、「.xml」、「.cyclonedx」の 3 種類であり、SPDX の SBOM を読み込むには、ファイル形式も「.spdx」形式に変更しないと読み込めないことも判明した。 また、Package Version 名が空欄だと SBOM が読み込めないことが判明した。 結果として、Package Version 名の空欄箇所に「NONE」と記載し、ファイル形式を「.spdx」に変更した spdx-tag-value 形式の SBOM を取り込むことに成功した。
	「SBOM ANALYSIS」欄の脆弱性深刻度分布図の深刻度の数量と「Vulnerability Details」欄	Xelis Dental の SBOM において検出された脆弱性にかかる深刻度の数量が、「SBOM ANALYSIS」欄は CRITICAL が 7、HIGH が 9、MEDIUM が 9 だったの

項目	課題	解決方法、留意点
	の深刻度の数量が違う。	に対し、「Vulnerability Details」欄には CRITICAL が 6、HIGH が 11、MEDIUM が 7、LOW が 1 と違いがあった。
	深刻度「LOW」に CVSS SCORE「10」の評価がされている。	「SBOM ANALYSIS」欄の深刻度が「LOW」だったにも関わらず、同欄の CVSS3 SCORE では「10」と一番深刻度が高い評価がされていたため、深刻度評価の基準が不明である。これにより仮に低い深刻度を示していたとしても、手動での調査は必要となる。
	同じサンプルコードでも異なるツールで出力された SBOM の場合、Daggerboard 上で検出結果が異なる。	Daggerboard の検出結果は SBOM の精度にも依存することには留意する必要がある。

医療機器の製品ポリシーにおける TPLC を通じたサイバーセキュリティに係る課題を下表に示す。

表 4-45　製品ポリシーにおけるサポート計画等の課題

課題	解決方法、留意点
市販済みの製品について、ソフトウェアの EOL と EOS を考慮したサポート計画となっていない医療機器がある。	本実証における当該医療機器のように、すでに市販されている製品の中には IMDRF ガイダンスで規定されているように医療機器に用いられているソフトウェアの EOL、EOS を考慮した製品のセキュリティポリシーが策定されておらず、EOL 及び EOS について考慮が不十分な場合は製品のセキュリティポリシーについて見直す必要がある。医療機器がレガシー状態になった場合には補完的対策等の実施を具体的に検討し、顧客との連携を行い、顧客への説明責任を果たす必要がある。市販前の設計段階より JIT 2304 ソフトウェア構成管理をするほか、ソフトウェアの EOL、EOS を考慮して、製品のセキュリティポリシーを策定する必要がある。
医療機器に用いられているソフトウェア部品の EOL、EOS が	OTS の中でサードパーティ製ソフトウェアについては、設計段階で医療機器に用

課題	解決方法、留意点
明確でない場合がある。	いる際に、公表されている資料を確認、公表されていない場合はサードパーティに確認を取った上で利用する。OSS については、利用規約等を確認した上で、リスクについて評価した上で、利用可否を検討する。
医療機器に用いられているソフトウェア部品の EOL、EOS が早く、頻繁にバージョンアップする製品がある。	設計段階で医療機器に用いるソフトウェアについては、EOL、EOS については考慮して選択する必要がある。ソフトウェアを使用する場合のリスクについては評価した上で、対策も検討して顧客や規制当局に開示する必要がある。
医療機器に用いられているソフトウェア部品の中で、サードパーティ製ソフトウェアの SBOM 提示を受けられない場合がある。	サードパーティに SBOM 提示が受けられない場合は、現在の科学的な水準で対応可能な合理的な手法（手動又はツール等）で検証し、その検証結果を証拠として説明責任を果たせるようにしておく必要がある。
医療機器に対するサイバー攻撃及び脆弱性の影響は、時間経過に伴って変化する。市販前の設計段階で策定した製品ポリシーでは、リスクが受容可能な状態を適切に維持できない場合がある。	製造販売業者は、市販後の医療機器のサイバーセキュリティに係る対応として、薬機法に基づく不具合報告（サイバーセキュリティ上の脆弱性に起因する健康被害の発生のおそれのある事象に係る報告も含む）その他の市販後安全対策を実施する他、サイバーセキュリティに関係する行政機関への報告を行うことに加え、情報共有機関、脆弱性発見者等を含めた製品ライフサイクルの市販後プロセスに関与する全てのステークホルダーと連携したアプローチを行う必要がある。医療機器のサイバーセキュリティに関して、全てのステークホルダーが理解しておかねばならず、そのために製造販売業者は顧客との連携を行い、顧客への説明責任を果たす必要がある。
医療機器に使用されているソフトウェア部品の EOL、EOS について、明確でない製品がある。後継ソフトウェアの販売の有無、時期も不明な場合がある。	既に、医療機器導入時に医療機関に示した製品ポリシーに基づいて対応できない場合、製品ポリシーの見直し等を検討する必要がある。 社内的にはソフトウェア部品に関する情報収集を行い、個別のソフトウェアアップデート計画等を立てておき、医療機関に対応を説明できるように準備を進めておく必要がある。

課題	解決方法、留意点
医療機器の脆弱性マネジメントフロー、修正の手順は検討できたが、サイバーセキュリティ要件により業務拡大するため、実施のためのリソース（資金面・人員面等）の確保が難しい。	原資確保のため、脆弱性の対応コストを製品価格に上乗せする必要が出てくる。コスト負担を考えるにしても、負担額はまず製造販売業者が吸収することになるため大きな負担になる。特に施行直後は負担が大きくなることに留意する必要がある。

医療機関ヒアリングを通じて課題を抽出し、下表にまとめた。

表 4-46　医療機関ヒアリングを通じての課題抽出（2022 年 11 月 30 日時点）

課題	解決方法、留意点
SBOM 提供が制度上推奨された場合、医療機関及び製造販売業者における SBOM 導入、管理・活用コスト等について、負担が増加することとなるがコスト負担について検討がされていない。	現時点で製造販売業者側のコスト面での負担が大きく、SBOM 導入、管理コストを製品価格に上乗せする形となる可能性がある。また、医療機関側も価格の上昇した製品の購入に加え、同様に SBOM 導入、管理・活用のための管理体制の構築及び維持にコストが増加することが予想される。医療機器のサイバーセキュリティは医療機関と製造販売業者当事者だけでなく、社会的な仕組みの全体像を明確にした上で、コスト負担の仕組みを検討する必要がある。
医療機関に対する SBOM の提示方法、フォーマット形式等まだ決まっていない。	SBOM 形式については紙媒体より電子的に管理した方が効率的なため、電子的に提供する方がよい。フォーマット形式は規制当局が審査・承認又は認証における SBOM の取り扱いについて明確になり、フォーマット形式等代表的なフォーマット形式に標準化される場合にはそれに合わせる方針がよいと考えられる。
提供した SBOM を医療機関が適切に評価できる体制及び仕組みや人材がいないため、医療機関へ提供するセキュリティ文書として SBOM だけでは不十分である。	製造販売業者は SBOM の提供だけでなく、サイバーセキュリティ対策、脆弱性マネジメント、ソフトウェア脆弱性に関するアドバイザリー情報も含めて提供する必要がある。
医療機関で SBOM 管理する仕組みが検討できていない。	医療機器製販業者が SBOM 提示される際に、医療機器の資産管理台帳と紐づけて管理する必要がある。そのための人材確保も必要であり、対応するための管理工数も要する。有償ツールによっては SaaS 製品のため、共有が可能であるが医

課題	解決方法、留意点
	療機器との紐づけが必要である。 規制当局に審査・承認で SBOM を提示するのであれば、行政機関等で集中管理して医療機関がアクセスする方法が効率的と考えられるが、そのような検討は現時点ではされていない。
医療機関でセキュリティ対策の教育が、医学教育で実施されていないため、検討が難しい。	厚生労働省サイト「医療分野のサイバーセキュリティ対策について」ページに医療機関向けの教育実施等が行われている。ただし、医療機器のセキュリティ確保については、今後、通知発出されて対応について明らかになってから検討となる。 医療機関が実施すべき対策等について、具体的に示される必要がある。 医療情報システムの安全管理に関するガイドライン[76]では、6.5. 技術的安全対策 (7) 医療等分野における IoT 機器の利用で言及されているが、認知が広まっていない。「医療機関のサイバーセキュリティ対策チェックリスト」では、セキュリティ文書（例：SBOM 入手）等による検討が含まれている。
医療機関におけるセキュリティ対策について、管理体制整備が難しい。	情報システム部門と医療機器担当者が連携を図る必要がある。大規模な医療機関には情報システム部門はあるが、小規模な医療機関には情報システム部門がない場合もある。医療機器の安全確保と有効性維持を行っている臨床工学技士も大規模な医療機関には人員確保できているが、小規模な医療機関では難しいため、対応できる人員の確保が必要となる。
機器の脆弱性によるセキュリティインシデント対応については、医療安全の枠組みでは必要であるが、対応できる人材が医療機関にはいない可能性が高い。インシデントが発生した場合、アラートをあげるのが難しい。	製造販売業者が脆弱性を検知し、それが患者に危害を及ぼすかリスクアセスメントした上で、アドバイザリー情報を提供するとともに、対応を含めて実施する。 製造販売業者は、情報共有分析機関に参加する。医療 ISAC 等の情報共有分析機関や支援体制が整備されることが望ましい。
患者に健康被害を及ぼす可能性のある脆弱性が発見された場	使用を停止すると患者の生命に危険が及ぶような医療機器の場合は、このような

[76] 厚生労働省、「医療情報システムの安全管理に関するガイドライン　第 5.2 版（令和 4 年 3 月）」サイト https://www.mhlw.go.jp/stf/shingi/0000516275_00002.html

課題	解決方法、留意点
合、医療機関に対し医療機器の使用停止を通知する必要があるが、生命維持装置等直ちに使用を停止できない医療機器も存在する。	状況を想定した補完的対策等も設計時に盛り込んで計画をたてておく必要がある。緊急時に医療機器の使用を継続する場合は、設計時にたてた計画に基づいて補完的対策を実施するためのリスクコミュニケーションを医療機関に対して行う必要がある。
医療機関へは製品ポリシー提示しており、医療機器の更新計画、入れ替えのため、事前に通知する必要がある。	医療機関への医療機器の新規機種の入れ替え等を通知するタイミングについては医療機関が事業継続の対応がとれるよう導入医療機関の事情に合わせて、医療機器の価格、導入手続きも考慮する必要がある。（特に、高額な医療機器の場合、医療機関が入札等の手続きや財源確保に時間を要するため、早期に通知が必要な場合もある。）

4.5.5.4. 効果評価

実証で得られた、各ツールでの初期導入、SBOM 作成、共有方法、SBOM 活用・管理について下表に示す。

表 4-47　SBOM 作成、共有方法、活用・管理まとめ

項目	工数	効果評価
初期導入	MEND SCA：9.16 人時間 （内訳：インストール 0.50 人時間、設定：KRK 0.83 人時間、LSC 0.33 人時間、習熟：KRK 5.50 人時間、LSC 2.00 人時間）	初期導入に関しては、Syft&Grype の Linux 環境構築を除けば、ほぼ同様の結果となっている。 習熟に関しては MEND SCA と Daggerboard が多くの工数を要することになった。MEND SCA はできることが多く、学習すべき内容が他のツールと比較して多いためであり、Daggerboard は参考となる資料が他のツールと比べて特に少なく、基本的な操作を行えるようになるまでの情報収集にさえ苦労したためである。また、Syft&Grype も資料が少なく習熟に苦労したが、こちらはツールの使用するに当たり最低限の情報は得られた。このため、習熟に関して単純比較は難しいが、Syft&Grype と Daggerboard は最低限の操作に必要な分の習熟しか行っておらず、持
	Syft&Grype：13.12 人時間 （内訳：Linux 環境構築：6.83 人時間、インストール 0.55 人時間、設定：KRK 1.95 人時間、習熟：KRK 3.79 人時間）	

項目	工数	効果評価
	Daggerboard：10.17 人時間 （内訳：インストール 0.75 人時間、習熟 9.42 人時間）	っている機能を使いこなせるまでの習熟を必要としたら、これ以上に多くの工数を要すると考えられる。
SBOM 作成	手動での SBOM 生成：30.0 人時間	ツールでのスキャン工数は、MEND SCA と Syft&Grype では同じような工数となる。ただし検出精度を考えると Syft&Grype は検出漏れがあるため、その分 MEND SCA の方が信頼性が高いため、手動の SBOM との精査が必須と考えると MEND SCA 利用が全体の工数は少なくなると思われる。
	MEND SCA での SBOM 生成：0.15 人時間	
	Syft&Grype での SBOM 生成：0.27 人時間	
	サードパーティ製 SBOM の内容確認：0.50 人時間（IMDRF 追補ガイダンス案に規定されている 8 要素確認）	
共有方法	手動作成 SBOM の開発委託先との共有：更新管理：1.00 人時間/SBOM 提供ごと	手動作成の SBOM を社内で管理する必要があるため、社内システム構築と、提供された SBOM の更新管理に工数が発生する。
	MEND SCA での開発委託先との共有：0.00 人時間	MEND SCA の場合、委託元と開発委託先で情報の共有ができるため、工数は発生しない。ほかのツールの場合は、共有方法及び更新管理について検討が必要な分、MEND SCA 利用の方が共有に関して有利といえる。
	Syft&Grype での開発委託先との共有：	Syft&Grype は、MEND SCA のように、同じインターフェース上で共有できないため、

項目	工数	効果評価
	更新管理 : 1.00 人時間/SBOM 提供ごと	更新された SBOM ドキュメントを開発委託先より受け取り、社内で管理する必要があるため、社内システム構築と、提供された SBOM の更新管理に工数が発生する。
SBOM 活用・管理	手動の SBOM での脆弱性マネジメント：0.77 人時間（内訳：脆弱性の特定 0.23 人時間/1 部品、対処方法特定 : 0.54 人時間）	脆弱性の管理、修正を考えた場合、手動での SBOM の場合、1 部品ごとに脆弱性の有無を調査する必要があるため、部品数に応じた工数が必要となり、部品数が多いソフトウェアについては膨大な調査時間を要する。MEND SCA はリアルタイムで脆弱性情報を視覚的に管理でき、さらに脆弱性に対する評価、対処方法方を示してくれるため、他のツールに比べて、対処までの時間短縮になる。また、リアルタイムに更新されていく脆弱性情報の入手のために、MEND SCA であれば専門の人材を少人数で対応可能。脆弱性の修正に関しては、どのツールも同様にかかるため、情報入手のスピードと精度の点でメリットが大きい。
	MEND SCA の SBOM での脆弱性マネジメント：0.26 人時間（内訳：脆弱性の特定 0.08 人時間/1 部品、対処方法特定：0.18 人時間）	
	Syft&Grype の SBOM での脆弱性マネジメント : 0.79 人時間（内訳 : Grype スキャン 0.12 人時間、脆弱性の特定 0.39 人時間/1 部品、対処方法特定：0.28 人時間）	
	Daggerboard の SBOM での脆弱性マネジメント：0.92 人時間（内訳：Daggerboard スキャン 0.15 人時間、脆弱性の特定 0.22 人時間/1 部品、対処方法特定：0.55 人時間）	

4.5.5.5. SBOM 適用範囲（SBOM 活用モデル）の改訂案

本実証において、実証できた範囲を下表に示す。

表 4-48　本実証での実証範囲

		主な適用項目	コスト区分	コスト区分の判断理由	実証項目
生成・共有	(a)SBOM 作成主体	(a1)自社	小	自社開発で直接利用する部品を構成ファイル等から特定し、SBOM を生成する。コード改変部品を含む。	○
		(a2)サプライヤー（開発委託先）取引契約あり	中	取引契約のある開発委託先のソフトウェアで利用する部品の SBOM を生成する。	○
		(a3)サプライヤー（サードパーティ）取引契約なし	大	取引契約による SBOM の要件化できない OSS や既成部品ベンダーが SBOM を生成する。(b2)(c2)	△
	(b) 作成範囲	(b1)直接利用部品※1（開発主体が直接利用する部品）	小	開発者が直接利用する部品を構成ファイル等から特定し、ツール等で SBOM を生成する。	○
		(b2)間接利用部品※2（既製品等開発委託契約のない部品から再帰的に利用する部品）	大	既製品の提供者に対して、部品調達後に SBOM の提供を要求することは難しいため、その既製品の利用者は、既製品からさらに再帰的に利用される部品に対して、ツールを用いて SBOM を生成する。	○
	(c)検出・生成・検査手段	(c1)手動で特定（構成管理情報利用）・ツールで生成	小	直接利用する部品情報について構成ファイル等を用いて作成する。	○
		(c2)ツールで特定・生成・誤検知精査なし	中	ツールを用いて SBOM を生成し、精査は省略する。ツールの利用は再帰部品の SBOM 生成を主に想定するため商用ツールの利用を想定する。	○
		(c3)ツールで特定・生成・誤検知精査あり	大	商用ツールを用いて SBOM を生成し、ソースコードレビューを行い、誤検知、検出漏れの精査を行う。（再帰利用部品を含む）	○
		(c4)開発委託元が、開発委託先の作成した SBOM を独立に検査	大	開発委託元が、開発委託先の作成した SBOM を受け入れる際に、ツール等で独立して SBOM を作成する等して信頼性を検査する。	×
	(d) デー	(d1)標準フォーマット	中	SPDX 等の標準フォーマットで作成する。	○

		主な適用項目	コスト区分	コスト区分の判断理由	実証項目
	タ様式・項目	（SPDX、CycloneDX、SPDXLite 等）			
		(d2)大統領令におけるデータフィールドの最小要素を含む	中	大統領令におけるデータフィールドの最小要素を含む SBOM を生成する。	○
		(d3)上記を満たさない要素	小	独自の最小限の要素を作成する。	×
	(e) 活用範囲	(e1)脆弱性の特定	小	NVD,JVN 等の DB を対象として脆弱性の検索・特定を行う。	×
		(e2)脆弱性の深刻度評価	中	CVSS 値をベースとした深刻度を評価し、脆弱性対応の優先度を設定する。	○
		(e3)脆弱性の悪用可能性等の評価と対処	中	VEX 情報等を用いて悪用可能性、脆弱性対処の必要性を評価する。必要に応じて対処策等のアドバイザリーを発行する。	×
		(e4)ライセンス特定	中	ライセンスの特定と規約の取得を行う。	×
活用	(f) 活用主体	(f1)製品利用者	小	脆弱性が特定された場合、利用を中断し、ベンダーによる修正を待つ。業務中断コストも考慮すれば損害は大きい。	△
		(f2)最終製品ベンダー	中	利用者に脆弱性の発見を通知するとともに、直接開発者への修正依頼、修正後のビルド・利用者への提供を行う。必要に応じて当局、ISAC 等に報告する。	△
		(f3)各部品の開発者	大	開発者は、脆弱性の確認と修正を行い、最終ベンダーに修正版を提供する。（最終ベンダーが開発者である場合も含まれる。）必要に応じて当局、ISAC 等に報告する。	×

本実証を通じて SBOM の適用範囲について改定案を検討した。

医療機器分野において SBOM はセキュリティ文書の一部として医療機関に提供される。このことから多くの医療機器を導入している大型の医療施設等は、多くの SBOM の管理を行うこととなることを考慮すると、独自フォーマットよりも標準フォーマットでの SBOM 提供が望ましい。手動で標準フォーマット（SPDX）の SBOM 作成については、Excel 等の基本的なフォーマットがあれば作成は可能と考えられる。また(c1)について手動で特定しツールで SBOM を生成する項目については、今回の実証で使用したツールにこのような機能はないが、手動で作成した Excel シートを SPDX 形式のファイルに変換する SPDX Online Tool[77]というサイトがあるため適用可能としている。悪用可能性の評価については適用事例や情報も少ないため、現時点での適用は難しいと判断したが、脆弱性に関する深刻度は CVSS 値で評価できるため、部分適用（事例依存）としている。

このことから SBOM の適用範囲について、一部の活用法についての改定のみとなり、概ね現実的であることが確認できた。

SBOM 適用範囲を下表に示す。

表 4-49　実証項目の改定案

＊適用可能・必須、■不要又は対象外、部分適用（事例依存）、□実証できた項目

(a)SBOM 作成主体	(b)作成範囲	(c)生成方法	(d)生成項目	(e)活用法
(a1)自社	(b1)直接利用部品	(c1)手動で特定（構成管理情報利用）・ツールで生成	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記の一部のみ	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定

[77] SPDX Online Tool:https://tools.spdx.org/app/about/

(a)SBOM 作成主体	(b)作成範囲	(c)生成方法	(d)生成項目	(e)活用法
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
	(b2)間接利用部品	(c1)手動で特定（構成管理情報利用）・ツールで生成	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
				(e1)(e2)脆弱性・ライセンスの特定

(a)SBOM 作成主体	(b)作成範囲	(c)生成方法	(d)生成項目	(e)活用法
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
(a2)サプライヤー（開発委託先）取引契約あり	(b1)直接利用部品	(c1)手動で特定（構成管理情報利用）・ツールで生成	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記の一部のみ	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価

(a)SBOM 作成主体	(b)作成範囲	(c)生成方法	(d)生成項目	(e)活用法
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
	(b2)間接利用部品	(c1)手動で特定（構成管理情報利用）・ツールで生成	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
(a3)サプライヤー	(b1)開発者自身	(c1)手動で特定（構成管理情報利	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価

(a)SBOM 作成主体	(b)作成範囲	(c)生成方法	(d)生成項目	(e)活用法
（サードパーティ）取引契約なし		用）・ツールで生成	(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
	(b2)開発者以外（調達者、利用者）	(c1)手動で特定（構成管理情報利用）・ツールで生成	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価

(a)SBOM 作成主体	(b)作成範囲	(c)生成方法	(d)生成項目	(e)活用法
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価

4.5.5.6. 取引契約における要求条項等

開発委託先との契約や、サードパーティ製ソフトウェアベンダーとの契約について、サイバーセキュリティの観点から新たに契約事項に含める必要のある項目について検討した。

本実証の医療機器は長期にわたってサイバーセキュリティを意識した監視、管理が必要となるため、その開発においては使用部品の脆弱性やライフサイクルを考慮した部品選定を行う必要があり、またソフトウェアの更新ごとに SBOM についても更新を行う必要があることから、それらを開発委託先との契約事項として反映することが求められる。また、開発委託先を介さない直接契約のサードパーティ製ソフトウェアについても、開発委託先同様にソフトウェア更新時の SBOM の提供を求めるほか、脆弱性に対する監視及び発見時の連絡についても契約事項として反映する必要がある。

表 4-50　開発委託先及び直接契約における契約に含める条項

契約分類	契約内容
開発委託先との契約	ソフトウェア開発に使用する部品は、あらかじめセキュリティの脆弱性の有無を調査し、セキュリティの脆弱性がない物を選択する。また、開発委託製品のライフサイクルを考慮した部品選定を行う。
	開発の際に使用するサードパーティ製部品に関してリスト化し、ライセンス及びバージョンに関しても全て明らかにする。
	使用するサードパーティ製部品に変更があった場合は、SBOM を更新し、提供する。
	使用するサードパーティ製部品に脆弱性が発見された場合は、遅滞なく報告する。
直接契約（サードパーティ製ソフトウェアベンダー）	ソフトウェアのアップデートやセキュリティ対策等、ソフトウェアに変更のあった場合は遅滞なく報告し、更新された SBOM を提供する。
	常にソフトウェアの脆弱性を管理し、使用中のバージョンにおいて脆弱性が発見された場合は、脆弱性のある部品名、脆弱性の内容及び、脆弱性により起こりうる事象等を速やかに報告する。
	EOL や EOS といったソフトウェアのライフサイクルを明確にし、バージョンアップ等により EOL、EOS に変更があった場合は遅滞なく報告する。

4.6. 自動車分野

4.6.1. 実証の狙い・重点ポイント

自動車のコスト制約と安全性の要求を考慮し、サプライチェーン上の契約関係にある各開発者が自身の直接利用部品及び間接利用部品の SBOM を生成し、ティア 1 などがツールを用いて全体を管理することで、全体のコストを最小化することを想定する。

ツールの誤検知による SBOM 誤差に対する精査は、コスト評価に基づき実施の可否を考察する。

部品の粒度は、脆弱性 DB を用いた脆弱性特定などの目的を考慮して、部品の包含関係を含む全ての粒度の SBOM を保持することが有効か考察する。

SBOM 事業全体で事前に検討した SBOM の生成・活用範囲を示す SBOM 活用モデル（仮説）に対して、本実証では、具体的なコスト・効果を評価・比較することで、自動車分野で妥当と考えられる範囲について検討し、改定案を提示することを目的とする。

4.6.2. 実証の手順

下表の手順で実証を推進した。

表 4-51　実証手順

フェーズ	実施項目
SBOM 初期導入	・SBOM 生成ツールの環境構築、利用方法習得に関する工数計測と課題特定を行う。 ・SBOM 生成ツールとして、ティア 1 が Synopsis 社製 Black Duck とリックソフト社製 MEND SCA を使用し、ティア 2 がリックソフト社製 MEND SCA を使用する。 ・今回の実証実験では無償ツールは使用しない。実証期間が有限であること、また手動に対してツールを使用する優位性は有償ツールで検証可能であるためである。
SBOM の生成・共有	・ティア 1 とティア 2 で直接利用部品の SBOM を作成し、SBOM 作成の工数計測と SBOM 精査の工数計測、及び課題特定を行う。 ・SPDX 及び SPDX Lite の項目が米大統領令の最小要素に対応できるか確認する。 ・OEM、ティア 1、ティア 2 のデータ共有の可否について評価する。
SBOM の活用・管理	・ティア 1 で SBOM を集約し、脆弱性の特定と深刻度評価（CVSS）の実施可否、フィージビリティ、工数計測、課題特定を行う。 ・SBOM 作成時の部品情報の精査を省略し、脆弱性特定後に精査することで、コストを低減できるか評価する。 ・委託開発先に SBOM の提供、脆弱性の通知を契約で要件化できるか考察する。

フェーズ	実施項目
	・SBOM を用いて脆弱性が特定された場合の対処法の選択肢（発注者への通知、ソフトの改修、回帰テスト等）の検討、対処までの所要期間の計測、SBOM 活用による所要期間短縮効果を評価する。

4.6.3. 実証の対象システムの概要と構成

本実証では、自動車に搭載されることを想定したヒータコントローラ（エアコンの調整を行うスイッチ）の制御用組込システムを対象に SBOM 生成・活用を実施した。将来的に車載ソフトウェアにおいて OSS の利用が拡大した場合の SBOM のコスト・効果と、現状の一般的な車載ソフトウェアにおける SBOM のコスト・効果の両方を検証するため、「オープン系非リアルタイム OS ベースシステム」と「制御系システム」の 2 種類を用いて実証を行った。

4.6.3.1. 対象システム 1：オープン系非リアルタイム OS ベースシステム

対象システムは自動車に搭載されることを想定したヒータコントローラ （エアコンの調整を行うスイッチ）である。

ヒータコントローラのような制御系 ECU は情報系システムのソフトに比べサイズが小さく、OSS が使用されることは少ない。そのため、実証実験を行うために、オープン系非リアルタイム OS（今回は RaspberryPi OS を採用）を使用したシステムを作成し、SBOM を作成することとした。

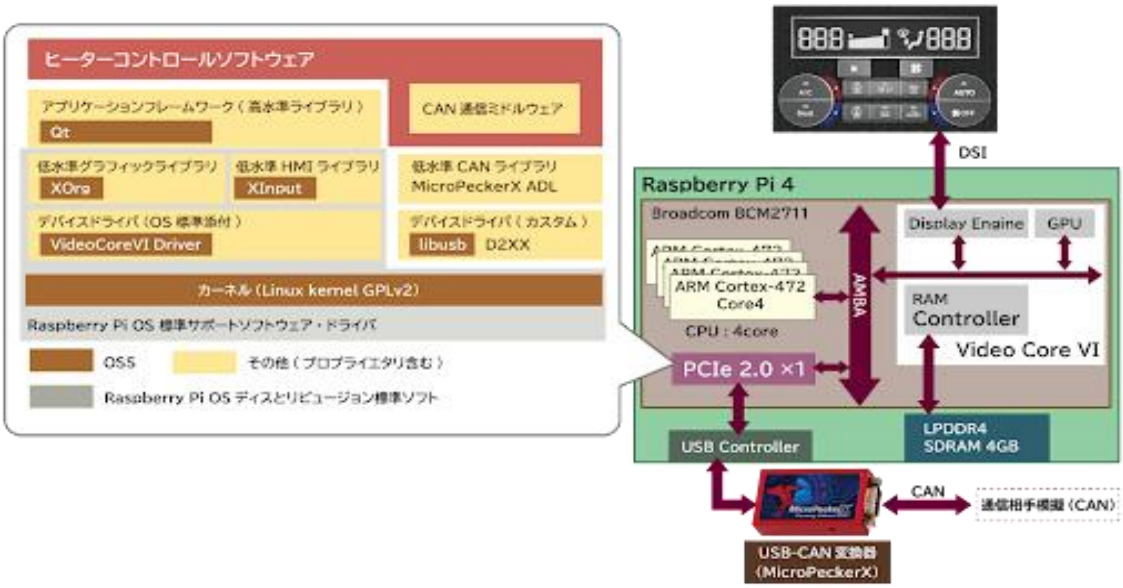


図 4-14　システム構成 1：オープン系非リアルタイム OS ベースシステム

SBOM 生成範囲のコンポーネントの開発言語は以下のとおりである。

表 4-52　オープン系非リアルタイム OS ベースシステムの開発言語

構成	要素	開発言語
Raspberry Pi OS 標準サポートソフトウェア・ドライバー	カーネル（Linux kernel GPLv2）	C 言語
	デバイスドライバー（OS 標準添付）	C 言語
ヒーターコントロールソフトウェア	Qt	C++
	起動スクリプト	Python

4.6.3.2. 対象システム 2：制御系システム（一般的なヒータコントローラの例）

車載開発への SBOM 活用について検証するために、一般的なヒータコントローラに即した構成のシステム構成 2 も用意した。OSS が全く含まれていないとシステム構成 1 との比較ができないため、組込向けリアルタイム OS とその他数点の OSS を含む構成とした。

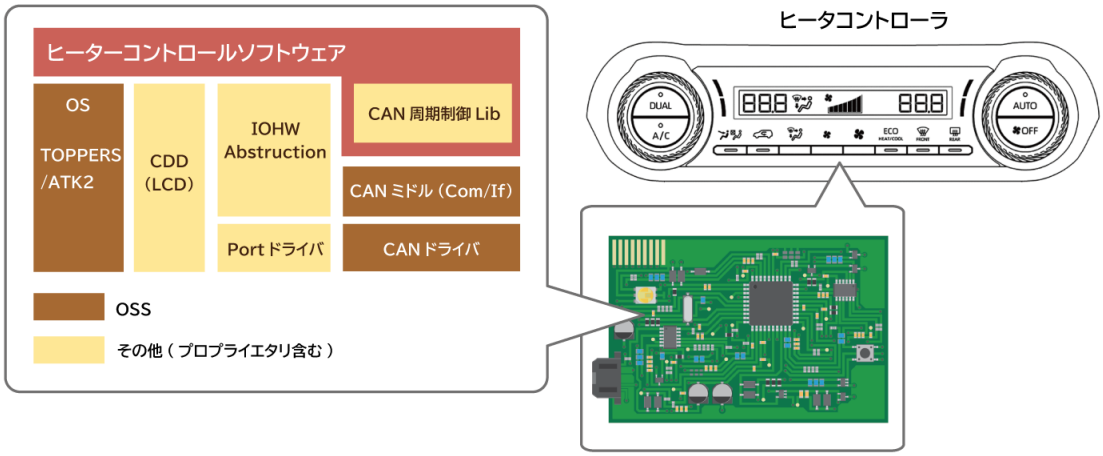


図 4-15　システム構成 2：制御系システム

SBOM 生成範囲のコンポーネントの開発言語は以下のとおりである。

表 4-53　制御系システムの開発言語

構成	要素	開発言語
OS	TOPPERS /ATK2	アセンブリ言語/C 言語
CAN ミドル（Com/If）	A-COMSTACK	C 言語
CAN ドライバー	―	C 言語
ヒーターコントロールソフトウェア	―	C 言語

4.6.3.3. システム構成 1（オープン系非リアルタイム OS ベースシステム）とシステム構成

2（制御系システム）のソフト開発の違いについて

オープン系非リアルタイム OS ベースシステムの場合は、設計者は直接利用するライブラリを把握して開発を開始する。OSS の依存関係はパッケージマネージャで管理されている。その場合は開発途中又は開発完了後に SBOM を作成してライセンス管理や脆弱性管理を行う。

制御系システムの場合は、使用する言語（C/C++/アセンブリ）にメジャーなパッケージマネージャは存在しないため、パッケージマネージャで部品を管理することがない。設計者が利用部品を全て決定して部品の管理を行い、開発を行う。開発するに当たって使用する部品を選定し、選定した部品以外は使用しない。

全ての部品を選定しているか否かが両システムの違いとなる。

全ての部品を選定できていることの補足として、今回想定した部品選定の手順は次のとおりである。まず対象システムのアーキテクチャを設計し、アーキテクチャ上の各モジュールの実現手段を決めていく。内製とするか、外製とするか、また外製の場合は商用部品を使用するか OSS を使用するかを決める。そして決めた部品に対してどのように組み込むかを検討して部品表を完成させる。

外製ソフトの場合も、コードを入手しコードレビューを行い、修正箇所を特定してコード修正を行うため、把握できていない部品はなく、部品間の依存関係は明確にされている。

ただし、バイナリ提供の商用部品やベンダーで品質保証されている商用部品の場合は、ベンダーに対して部品表提出とベンダー側の責任で品質保証することを契約で取り交わす。又は同等の情報開示を行うことを取り決めている。

4.6.3.4. 実証の進め方について

本実証では、狙いにより合致するオープン系非リアルタイム OS ベースシステムを用いて、SBOM 生成・活用時のコスト・効果の計測を行う。制御系システムでの SBOM 生成・活用時のコスト・効果の計測は行わないが、特有の事項があれば考察を加えることとする。

4.6.4. 対象分野の前提条件・取引慣行等

今回の実証実験ではティア 1・ティア 2 間での請負契約により実施した。自動車の制御系開発においてはティア 1 ティア 2 間でソースコードを納品することがあり、本実証においてはティア 1・ティア 2 間でバイナリデータだけでなく、ソースコードも納品している。これによりティア 2 の開発範囲についてもティア 1 がソースコードを参照し、SBOM 生成等に使用することが可能である。

また、市場での品質保証やサイバーセキュリティ維持のために必要な構成情報（ソフトウェア限定ではない)は、OEM、ティア 1、ティア 2 の関係者間で利用可能な状態としている。

SBOM 管理については、開発環境等に係る情報は大きく影響するため本実証及び同分野で想定される例についてまとめる。

表 4-54　開発環境等(注)

項目	本実証	業界において想定される主な例
開発言語	C++、C、Python	アセンブリ、C、C++等
部品形態	ミドルウェア、アプリケーション	ミドルウェア、アプリケーション
開発環境ツール	Qt Creator IDE	ティア 1 指示又は関係者で選定 (適用マイコン用の統合開発環境、エディタ等)
ビルドツール	GCC	ティア 1 指示又は関係者で選定 (適用マイコン用のコンパイラやアセンブラ等)
構成管理ツール	Git、Subversion	ティア 1 指示又は関係者で選定 (Subversion、Git 等)

(注)一例であり、これに限定したものではない

4.6.5. 実証項目の特定

本 SBOM 実証では、以下に示す項目を元に作業を実施した。

表 4-55　実施項目

フェーズ	実証項目
SBOM 初期導入	1. ツールの初期導入工数を計測するために次の工数を計測する。それぞれ、ティア 1 とティア 2 の工数を計測する。 ①ツールの環境構築工数 ②ツールの利用方法習得工数 ツールは、 ティア 1 : BlackDuck、MEND、ティア 2 : MEND とする。 2. ツール導入時の課題を抽出する。
SBOM の生成・共有	1. SBOM 作成工数に関する工数を計測する。それぞれティア 1、ティア 2 の工数を計測する。 ①SBOM 作成工数 ・ティア 2:直接利用部品を対象に SBOM の作成工数（手動及びツール使用） ・ティア 1:ソフト全体を対象に SBOM の作成工数 （手動及びツール使用、バイナリ解析の使用） ②ツール誤検知を検出するための精査工数 ③SPDX ファイルの出力工数 ④ティア 1 とティア 2 で共通ツールを使用した場合の SBOM 共有工数

フェーズ	実証項目
	⑤ティア 1 とティア 2 で異なるツールを使用した場合の SBOM 共有工数 2. 計測データをもとに次の項目の比較を行う。 ①手動とツール利用の工数比較 ②SBOM の検証及び正確性に関する比較 ③ソースによる SBOM とバイナリに対する SBOM の比較 ④SPDX 形式と SPDXLite 形式のファイルの米国大統領令 SBOM 最小要素との比較 3. ツール運用時の課題を抽出する。
SBOM の活用・管理	1. SBOM の粒度を比較するため次の項目を計測する・①脆弱性特定後に SBOM 詳細化（精査）して、深刻度評価を行う工数 ②SBOM 詳細化（精査）後に脆弱性特定及び深刻度評価を行う工数 ③ティア 1 及びティア 2 のそれぞれにおいて、SBOM による脆弱性監視の工数 ④常時監視及び、週 1 などの定期監視の場合の所要期間を計測 2. 上記データをもとに次の内容を比較する ①SBOM 精査のタイミングによる脆弱性特定と深刻度評価を行う際の工数比較 ②常時監視及び定期監視の所要期間比較

4.6.6. 実証結果

4.6.6.1. SBOM 初期導入

4.6.6.1.1. 実施内容とコスト評価

MEND を使用した場合、SBOM 初期導入には合計で 59 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりである。

表 4-56　MEND を使用時の SBOM 初期導入コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
環境構築工数	6.0	人時間	3.0	人時間	3.0	人時間	―
学習・体制構築工数	53	人時間	31	人時間	22	人時間	―

表 4-57　MEND 使用時の SBOM 初期導入コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
環境構築工数	環境整備工数	6.0	人時間	3.0	人時間	3.0	人時間	利用者を 1 人とした場合の PC へのインストールと設定の工数を計上した。SaaS 型ツールであるためサーバー構築工数は 0 である。

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
学習・体制構築工数	学習工数	5.0	人時間	3.0	人時間	2.0	人時間	利用者 1 人当たりの学習工数である。
	体制構築工数	48	人時間	28	人時間	20	人時間	―

また、Black Duck を使用した場合、SBOM 初期導入には合計で 54.3 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりである。

表 4-58　Black Duck 使用時の SBOM 初期導入コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
環境構築工数	10.3	人時間	10.3	人時間	―	人時間	ティア 1 のみ実施。

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
学習・体制構築工数	44	人時間	44	人時間	―	人時間	ティア 1 のみ実施。

表 4-59　Black Duck を使用時の SBOM 初期導入コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
環境構築工数	環境整備工数	10.3	人時間	10.3	人時間	―	人時間	ティア 1 のみ実施
学習・体制構築工数	学習工数	16	人時間	16	人時間	―	人時間	・ティア 1 のみ実施 ・利用者 1 人当たりの学習工数である。

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
	体制構築工数	28	人時間	28	人時間	ー	人時間	ティア 1 のみ実施

SBOM 手動生成の場合、SBOM 初期導入には合計で 55 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりである。

表 4-60　SBOM 手動生成時の SBOM 初期導入コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
環境構築工数	2.0	人時間	1.0	人時間	1.0	人時間	ー
学習・体制構築工数	53	人時間	31	人時間	22	人時間	ー

表 4-61　SBOM 手動生成時の SBOM 初期導入コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
環境構築工数	環境整備工数	2.0	人時間	1.0	人時間	1.0	人時間	ー
学習・体制構築工数	学習工数	5.0	人時間	3.0	人時間	2.0	人時間	一人当たりの学習工数である。
	体制構築工数	48	人時間	28	人時間	20	人時間	ー

各ツールの導入工数として、環境整備工数と学習工数を計測した。導入時のツールベンダーのサポート有無による比較も行った。今回はいずれのツールもサポートを受けての導入であったため、サポート有りの工数は実測値であり、サポート無しの工数はツールの公開情報のみを使用して導入することを想定した予測値である。

表 4-62　SBOM 導入工数

条件					工数
環境整備工数	MEND（SaaS）	サーバー構築	サポート有り	ティア 1	0 人時間
				ティア 2	0 人時間
			サポート無し	ティア 1	0 人時間
				ティア 2	0 人時間
		インストール/設定	サポート有り	ティア 1	3.0h/人
				ティア 2	3.0h/人
			サポート無し	ティア 1	6.0h/人
				ティア 2	6.0h/人
	Black Duck（オンプレミス）	サーバー構築	サポート有り		10 人時間
			サポート無し		20 人時間
		インストール/設定	サポート有り		0.3h/人
			サポート無し		0.5h/人
学習工数	MEND	サポート有り		ティア 1	3.0h/人
				ティア 2	2.0h/人
		サポート無し		ティア 1	4.0h/人
				ティア 2	3.0h/人
	Black Duck	サポート有り			16h/人
		サポート無し			30h/人

オンプレミス型のツールでは、ツールコストに加えてサーバーマシンの準備とサーバー構築工数が必要となるため、SaaS 型ツールに比べてベンダーサポート有りの場合に 10 人時

間多くの工数を必要とした。ベンダーサポート無しでは 20 人時間の工数を要すると推定した。

今回使用した 2 つのツールはいずれも導入サポート及びマニュアル類が整備されており、サポート窓口も用意されているため初期導入における躓きは無かった。

4.6.6.1.2. 課題、解決ノウハウ、留意点

4.6.6.1.2.1. SaaS 型ツールのセキュリティについて

実証上の課題として、SaaS 型ツールは自社のサーバー構築が必要ないため導入が容易であるが、その反面セキュリティにリスクがないかが懸念点として挙げられた。ツールによっては対象ソフトの構成情報などを外部に送信するものがあるので注意が必要である。

結果として今回使用したツールは双方とも、ハッシュ値のみが外部に送信され対象ソフトが外部送信されることはなく、セキュリティに配慮されていた。

解決ノウハウとして、ツール導入時にはツールが外部に送信する情報の有無及び仕様を確認するべきである。

4.6.6.2. SBOM の生成・共有

4.6.6.2.1. 実施内容とコスト評価

MEND を使用した場合、SBOM の生成・共有には 1OSS 当たり合計で 0.16016 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりである。

表 4-63　MEND を使用時の SBOM 生成・共有コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
情報取得工数	0.00016	人時間/OSS	0.000078	人時間/OSS	0.000078	人時間/OSS	―
精査工数	0.16	人時間/OSS	0.08	人時間/OSS	0.08	人時間/OSS	―
（作成ツール費用）	―	円	―	円	―	円	本実証では実証実験用の試用ライセンスで実施

表 4-64　MEND を使用時の SBOM 生成・共有コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
情報取得工数	部品特定工数	0.000078	人時間/OSS	0.000039	人時間/OSS	0.000039	人時間/OSS	SW 部品数は、ティア 1、ティア 2 ともに 766
	SBOM 生成工数	0.000078	人時間/OSS	0.000039	人時間/OSS	0.000039	人時間/OSS	作成した SBOM は、ティア 1、ティア 2 ともに(d1)標準フォーマット(SPDX)
精査工数	精査工数	0.16	人時間/OSS	0.08	人時間/OSS	0.08	人時間/OSS	SW 部品数は、ティア 1、ティア 2 ともに 766
（作成ツール費用）	（作成ツール費用）	―	円	―	円	―	円	本実証では実証実験用の試用ライセンスで実施

また、Black Duck を使用した場合、SBOM の生成・共有には 1OSS 当たり合計で 0.08058 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりである。

表 4-65　Black Duck を使用時の SBOM 生成・共有コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
情報取得工数	0.00058	人時間/OSS	0.00058	人時間/OSS	―	人時間/OSS	ティア 1 のみ実施
精査工数	0.08	人時間/OSS	0.08	人時間/OSS	―	人時間/OSS	ティア 1 のみ実施
（作成ツール費用）	―	円	―	円	―	円	既保有ツールを使用して実施

表 4-66　Black Duck を使用時の生成・共有コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
情報取得工数	部品特定工数	0.00057	人時間/OSS	0.00057	人時間/OSS	―	人時間/OSS	ティア 1 のみ実施
	SBOM 生成工数	0.0000009	人時間/OSS	0.0000009	人時間/OSS	―	人時間/OSS	ティア 1 のみ実施
精査工数	精査工数	0.08	人時間/OSS	0.08	人時間/OSS	―	人時間/OSS	ティア 1 のみ実施
（作成ツール費用）	（作成ツール費用）	―	円	―	円	―	円	既保有ツールを使用して実施

SBOM 手動生成の場合、SBOM の生成・共有には 1OSS 当たり合計で 1.00004 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりである。

表 4-67　SBOM 手動生成時の SBOM 生成・共有コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
情報取得工数	0.50004	人時間/OSS	0.25002	人時間/OSS	0.25002	人時間/OSS	―
精査工数	0.5	人時間/OSS	0.25	人時間/OSS	0.25	人時間/OSS	―
（作成ツール費用）	0	円	0	円	0	円	―

表 4-68　SBOM 手動生成時の SBOM 生成・共有コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
情報取得工数	部品特定工数	0.000038	人時間/OSS	0.000019	人時間/OSS	0.000019	人時間/OSS	SW 部品数は、合計、ティア 1、ティア 2 ともに 1571
	SBOM 生成工数	0.5	人時間/OSS	0.25	人時間/OSS	0.25	人時間/OSS	SBOM 形式はティア 1、ティア 2 ともに,(d1)標準フォーマット(SPDX)
精査工数	精査工数	0.5	人時間/OSS	0.25	人時間/OSS	0.25	人時間/OSS	SW 部品数は、合計、ティア 1、ティア 2 ともに 1571

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
（作成ツール費用）	（作成ツール費用）	0	円	0	円	0	円	ー

4.6.6.2.2. SBOM 作成工数

4.6.6.2.2.1. オープン系非リアルタイム OS ベースシステム

SBOM 作成を手動で行った場合とツールで行った場合の工数を計測した。

手動による SBOM 作成は次の 2 つの方法で計測した。方法 1 は、パッケージマネージャから出力した部品表を元に、各部品の情報を手動で追加していくという方法である。この方法だと全ての部品を確認しているため、精査及び SPDX 作成の工数も含んでいる。この場合は代表の部品について SBOM 作成した工数を実測し、全体の部品数を掛け合わせて算出した。方法 2 はパパッケージマネージャら出力した部品表にスクリプトを実行することでライセンス情報を追加する方法である。スクリプトの作成には 1 人時間を要している。

ツールの場合はツールの実行工数を計測した。

表 4-69　SBOM 作成工数（オープン系 OS 採用システム）

条件		総工数	部品数	1 部品当たりの工数
手動	ティア 1	（方法 1） 目視による手動作成 ：393 人時間	1571	0.25 人時間
		（方法 2） パッケージマネージャから部品表出力：0.03 人時間		0.000019 人時間
	ティア 2	（方法 1） 目視による手動作成 ：393 人時間	1571	0.25 人時間
		（方法 2） パッケージマネージャから部品表出力：0.03 人時間		0.000019 人時間
ツール	ティア 1	0.03 人時間	766	0.000039 人時間
	ティア 2	0.03 人時間	766	0.000039 人時間

4.6.6.2.2.2. 制御系システム

SBOM を手動で作成した場合とツールを使用して SBOM 出力した場合の工数を計測した。手動の場合は、ソフト開発当初に決定している OSS に関する情報を追加する工数を計測した。ツールの場合は、ツールの実行工数を計測した。

表 4-70　SBOM 作成工数（制御系システム）

条件		総工数	部品数	1 部品当たりの工数
手動	ティア 1	0.33 人時間	2	0.17 人時間

条件		総工数	部品数	1 部品当たりの工数
	ティア 2	0.33 人時間	2	0.17 人時間
ツール	ティア 1	0.03 人時間	0 （誤検出 2）	0.015 人時間
	ティア 2	0.03 人時間	0 （誤検出 2）	0.015 人時間

制御系システムについては、手動で作成した SBOM には設計者が決定した使用部品 2 つを記載して作成した。一方ツールで出力した SBOM には 2 件の部品が出力されていたが 2 件とも誤検出であり、実際の使用部品は特定されなかった。

使用する OSS は決まっているため、手動の場合も作成工数は 1 部品当たり 0.17 人時間とオープン系 OS 採用システムに比べて少なかった。

4.6.6.2.3. 精査工数

前項で作成した、オープン系非リアルタイム OS ベースシステムの SBOM 精査時間を計測した。計測は代表の部品に対する実測値に部品数を掛け合わせて算出した。

また、ツール使用の場合は、ツールのスキャン結果で「完全一致」となっているものを除いた部品数とした。

表 4-71　精査工数

条件		総工数	部品数	1 部品当たりの工数
手動	ティア 1	393 人時間	1571	0.25 人時間
	ティア 2	393 人時間	1571	0.25 人時間
ツール	ティア 1	22 人時間	268	0.08 人時間
	ティア 2	22 人時間	268	0.08 人時間

手動の場合は目視による手動生成の SBOM に対する精査を前提としている。手動で記載した項目が正しいか確認する工数である。ツールの場合は使用箇所の情報や、スニペット解析結果の場合のコード比較など判断材料となる情報を利用できるため工数が低減できている。

また、制御系システムの場合の精査工数は 0.03 人時間であり、オープン系非リアルタイム OS ベースシステムに比べて少なかった。これは設計者が初めに使用する部品を決定しており改めて情報の確認に工数を要することが無かったからである。

4.6.6.2.4. バイナリスキャンによる検出

オープン系非リアルタイム OS ベースシステムの対象ソフトを完全に一つのバイナリとし、そのバイナリに対してツールのバイナリスキャンを用いて OSS の検出数を測定した。ただし、ツールの 1 度にスキャンできる容量制限のため、バイナリを 5 つに分割したものに対してスキャンを行った。

その結果、バイナリスキャンで検出した部品数は、82 であった。

今回の実証においては、バイナリ解析で検出した部品数は通常スキャンに比べて 1 割程度であった。この結果を踏まえ、本実証におけるコスト・効果測定にはバイナリ解析時の測定結果は採用しなかった。SBOM 生成におけるバイナリ解析の位置づけ・期待効果、及びバイナリ解析実施時の最適な運用・ツール導入のあり方については継続検討が必要と考える。

4.6.6.2.5. SPDX ファイル作成工数

オープン系非リアルタイム OS ベースシステムのソフトについて、手動による SPDX ファイル作成工数とツールによる SPDX ファイル作成工数を計測した。

手動の工数は、1 つの任意の部品を抽出し、当該部品の情報を SPDX 形式で記述した SBOM を作成した際の工数に部品数をかけたものである。

ツールの工数はツールを実行させる作業時間を計測した。

表 4-72　SPDX 作成工数

条件		総工数	部品数	1 部品当たりの工数
手動	ティア 1	393 人時間	1571	0.25 人時間
	ティア 2	393 人時間	1571	0.25 人時間
ツール	ティア 1	0.03 人時間	766	0.000039 人時間
	ティア 2	0.03 人時間	766	0.000039 人時間

4.6.6.2.6. 米大統領令最小要素と SPDX,及び SPDX-Lite ファイルの比較

米国大統領令による SBOM 最小要素と SPDX 及び SPDX-Lite ファイルの比較を行った。比較した SPDX と SPDX-Lite 該当項目は SPDX バージョン 2.20 である。

表 4-73　米大統領令最小要素と SPDX 及び SPDX-Lite の比較

大統領令最小要素	SPDX の該当項目	SPDX Lite の該当項目
Supplier Name	3.5 Package Supplier	該当項目無し
Component Name	3.1 Package Name	3.1 Package Name

大統領令最小要素	SPDX の該当項目	SPDX Lite の該当項目
Version of the Component	3.3 Package Version	3.3 Package Version
Other Unique Identifiers	3.2 Package SPDX Identifier	3.2 Package SPDX Identifier
Dependency Relationship	7.1 Relationship	該当項目無し
Author of SBOM Data	2.8 Creator	2.8 Creator
Timestamp	2.9 Created	2.9 Created
Hash of the Component	3.10 Package Checksum	該当項目無し

米国大統領令 SBOM 最小要素に対して、SPDX ファイルで対応できることは確認できた。SPDX Lite の場合は、一部含まれていない項目に対して対応が必要である。

4.6.6.2.7. SBOM 共有工数

4.6.6.2.7.1. ティア 1 とティア 2 で同じツールを共有する場合（SaaS 型ツールの場合）

クラウド上で同じデータを共有できることで、情報を即座に共有できるため、工数は 0 となる。

4.6.6.2.7.2. ティア 1 とティア 2 で異なるツール間で情報共有する場合

4.6.6.2.7.2.1. 共有ファイルの入出力工数

異なるツール間では、各ツールの出力データに互換性がないため、共有できる形式のファイルを介して共有する必要がある。そのためのフォーマットとして、SPDX 若しくはそこから抜粋した SPDX-Lite 形式のファイルがあるが、今回使用したツールはいずれも上位形式である SPDX 形式のファイル出力に対応していることで、ツール間の情報共有の多くがファイルのやり取りで完結し、その他の作業が発生しない。したがって、対象ソフトに含まれる部品数に関わらず 0.03 人時間である。

4.6.6.2.7.2.2. SBOM 統合の工数

さらに、Black Duck の場合は手動で部品の追加が可能である。部品を SBOM に追加することで、ツールの脆弱性通知を利用できる。（ただし、メーカーのデータベースに登録されている部品の場合に限る）

手動で追加する工数は 1 部品当たり、約 0.08 人時間であった。

手動で追加する場合は、追加する部品それぞれに 0.08 人時間が必要となるが、Black Duck には入力の API が用意されており、この API を使用したスクリプトを用意することにより

自動化することができる。この場合は部品数に関わらず、スクリプトの実行工数（約 0.03 人時間）に低減できる。

4.6.6.2.8. 課題、解決ノウハウ、留意点

4.6.6.2.8.1. SBOM 作成時の、パッケージマネージャの部品数と、ツールで検出した OSS 数の関係について

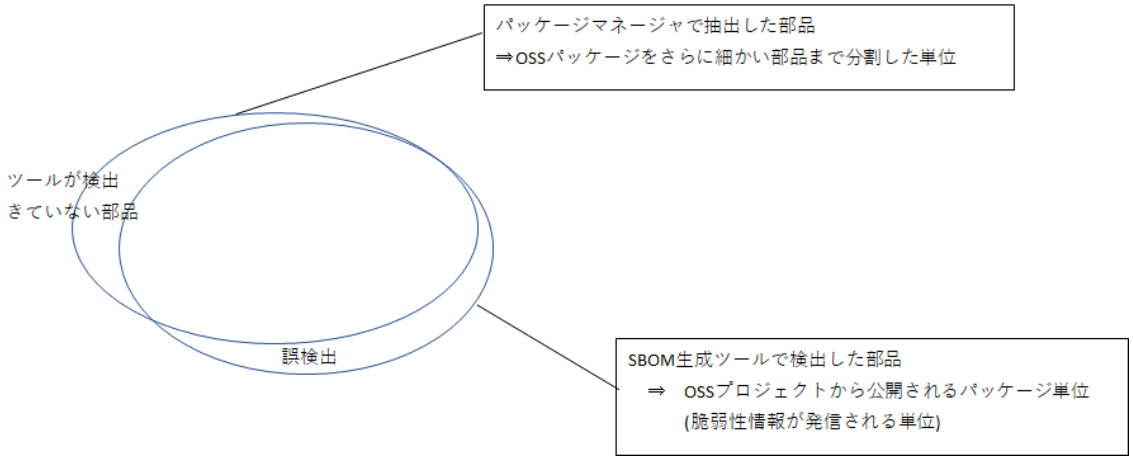


図 4-16　パッケージマネージャによる検出部品とツールによる検出部品の関係

本実証での課題として、SBOM 生成ツールで検出された部品とパッケージマネージャで抽出した部品が一致しないことが判明した。この理由としてはツールとパッケージマネージャの抽出範囲が異なることが挙げられる。両者の関係は上図のとおりである。ツールで検出された部品とパッケージマネージャで抽出した部品では粒度が異なるため、単純に数字の比較はできない。(パッケージマネージャの検出部品数とツールの検出部品数の差がそのまま、ツールで検出できなかった部品数というわけではない)

例として、SBOM 生成ツールで検出された OSS パッケージ「Scrot」内の 7 つの部品(giblib1、libc6、libimlib2、libx11-6、libxcomposite1、libxext、libxfixes3)がパッケージマネージャで抽出されている。Imlib2 の脆弱性（USN-5099-1）は実際に公開されており、これらのライブラリの粒度の部品情報を管理することは極めて重要である。

解決ノウハウとして、SBOM 生成ツールの仕様を把握し検出可能範囲を確認することで、部品表の差分について正しい判断を行うことができる。

4.6.6.2.8.2. パッケージマネージャで部品表を作ることが工数的に現実的であるか

今回の実証において手動生成の SBOM を作成する際にパッケージマネージャで抽出した部品表を元に作成したが、これが工数的に現実的であるかという課題について考える。

結論として、今回の場合はパッケージマネージャから SBOM を作成することについて工数的には現実的なものであった。作業内容は、事前にスクリリプトを作成して実行し、パッケージマネージャから部品表を抽出し、ライセンス情報の収集を行うというものであった。そのための作業工数はスクリプト実行工数の 0.03 人時間であり、ツールの実行工数と同等

であり、現実的な工数であった。

また、この SBOM の利点として、パッケージマネージャで管理されている細かい部品の粒度まで含まれているため、脆弱性情報に対してより細かい粒度の影響分析にも有効ではないかと考えられる。

ただし留意点として、今回の開発環境の OS は ubuntu であったが、ubuntu ではライセンス記述を共通化する取組がされており、共通化された仕組みに合わせてスクリプトを組むことにより、各部品のライセンス情報を収集することができた。そのような共通化がされていないとスクリプトを組むことが困難であると思われる。スクリプトを作成することができないと手動で SBOM を作成する必要があり、0.25 人時間/1 コンポーネントの工数が必要となり、今回の場合だと 393 人時間と大きな工数が必要となる。

4.6.6.2.8.2.1. SPDX ファイル出力工数に関する補足

現状では、パッケージマネージャから SPDX ファイルを自動で作成できないため、SPDX による情報共有を考えた場合に追加工数が必要になる（手動で 0.25 人時間/部品数）。そのため OSS の数が多いとツール利用の方に工数メリットがあると考えられる。

パッケージマネージャから SBOM を作成するのがコスト面で合理的かを判断する上では、使用 OSS 数と SPDX による情報共有の要否を踏まえた工数に留意すべきである。

4.6.6.2.8.2.2. 制御系システムに関する補足

制御系開発では一部の例外を除いて、多くの場合にパッケージマネージャを使用できないため、現状では上記のメリットを享受できない。しかし、制御系であっても開発環境内のビルドツールには構成情報が含まれているので、将来、その情報から部品リストを出力できるようにツールが対応できるようになると、開発者は意識しなくても SBOM を作れるような環境を得られる可能性がある。

このことを踏まえると、制御系開発における SBOM 作成に向けたノウハウとして、ビルドツール内の構成情報を日常の開発業務の中で正しく整備することが推奨される。

4.6.6.2.8.3. SBOM の精査についてのまとめ

課題として、本実証を進めるに当たって SBOM 精査にはいくつかの目的と方法があることが分かり、参加者の認識を合わせるために整理する必要が生じた。

解決ノウハウとして、SBOM 精査の目的と方法について検討した結果を次の図にまとめる。

検証したいこと		検証方法
ツールの使い方が正しいか		①ツールの実行Logにエラーが無いか確認
SBOMが正しいか	OSSを誤検出していないか	②SBOMのOSSが、対象ソフトに含まれているか確認 （ツールのマッチタイプで検証対象をフィルタリング可能） マッチタイプが完全一致でない場合はOSSが改変されている可能性が無いか確認 ③（②で判断がつかない場合）SBOMのOSS名を元にGitHubから入手したソースコードと対象ソフトを比較
	使用しているOSSが含まれているか	④対象のOSSが含まれているか、SBOMを確認
	改変されたOSSを検出できているか	⑤スニペット解析結果をもとに、対象ソースとOSSのソースを比較
	検出できていないOSSが無いか	⑥対象ソフトのCopyrightを抽出して検出したOSSがSBOMに含まれているか確認する等

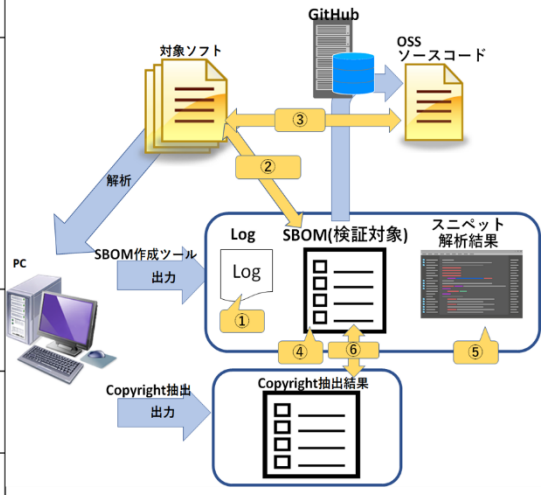


図 4-17　SBOM 精査まとめ

4.6.6.2.8.4. 制御系システムのツールスキャン結果について

本実証の課題として、制御系 ECU に対する SBOM 生成ツールの検出結果が 0 件であり、SBOM を生成できないという事項が発生した。

この理由としては、車載向けソフト部品については、ツールベンダーの OSS データベースへの登録が現時点ではまだ進んでいないのではないかと推測される。

この結果を踏まえると、ツールにより制御系 ECU の SBOM 生成を行う際のノウハウとして、事前に車載向けソフト部品を使用している可能性がないかを開発文書等から確認の上、使用可能性の高い部品につき SBOM 上で検知漏れ・誤検知が発生していないか重点的に精査することが推奨される。

現状ではこのような結果であったが、今後自動車業界での SBOM 生成ツールの利用が増え、ツールベンダーの OSS データベースが充実していくと、改善されていくのではないかと考える。

4.6.6.2.8.5. SBOM 共有時のコード流出について

同一ツールを利用した場合、即座に情報を共有でき、共有のための工数は 0 であった。しかし課題として、クラウドを介して情報を共有する際に、コードが流出するおそれがないかという懸念点があった。

結論としては、今回使用した MEND は、コードそのものをクラウド上にアップロードすることはなく、ツール画面上でもコードを閲覧することはできず、コード流出の心配がないことを確認できた。

ノウハウとしては、ツールを使用するに当たって、情報共有の仕様を確認しコード流出などのセキュリティ上の問題がないか確認すべきと考える。

4.6.6.3. SBOM の活用・管理

本章では全てオープン系非リアルタイム OS ベースシステムの SBOM を使用して計測を行っている。

4.6.6.3.1. 実施内容とコスト評価

MEND を使用した場合、SBOM の活用・管理には 1OSS 当たり合計で 4.48 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりである。

表 4-74　MEND を使用時の SBOM 活用・管理コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
脆弱性管理工数	4.16	人時間/OSS	2.08	人時間/OSS	2.08	人時間/OSS	―
ライセンス管理工数	0.32	人時間/OSS	0.16	人時間/OSS	0.16	人時間/OSS	―
（活用ツール費用）	―	円	―	円	―	円	本実証では実証実験用の試用ライセンスで実施

表 4-75　MEND を使用時の SBOM 活用・管理コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
脆弱性管理工数	脆弱性特定工数	0.16	人時間/OSS	0.08	人時間/OSS	0.08	人時間/OSS	―
	脆弱性対応工数	4.0	人時間/OSS	2.0	人時間/OSS	2.0	人時間/OSS	SW 部品数は合計、ティア 1、ティア 2 ともに 766 ・「特定された脆弱性の、自社 SW への影響有無（対策要否）判断までに要した作業」と「特定された脆弱性の深刻度評価」までに要した作業についての集計である。 ・「要対策脆弱性への対策検討・実施に要した

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
								作業」については脆弱性の内容と対策内容によって異なるため含んでいない。
ライセンス管理工数	ライセンス特定工数	0.16	人時間/OSS	0.08	人時間/OSS	0.08	SW 部品数は、合計、ティア 1、ティア 2 ともに 766	SW 部品数は、合計、ティア 1、ティア 2 ともに 766
	ライセンス違反対応工数	0.16	人時間/OSS	0.08	人時間/OSS	0.08	人時間/OSS	・「SBOM 上の OSS のライセンス上の規定を確認」と「自社開発 SW において、ライセンス上の規定に違反した使用形態となっている OSS の特定」までの集計である。 ・「違反状態の是正までに要した改修」については対策内容によって異なるため含んでいない。
（活用ツール費用）	（活用ツール費用）	－	円	－	円	－	円	本実証では実証実験用の試用ライセンスで実施

また、Black Duck を使用した場合、SBOM の活用・管理には 1OSS 当たり合計で 2.24 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりである。

表 4-76　Black Duck を使用時の SBOM 活用・管理コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
脆弱性管理工数	2.08	人時間/OSS	2.08	人時間/OSS	－	人時間/OSS	ティア 1 のみ実施
ライセンス管理工数	0.16	人時間/OSS	0.16	人時間/OSS	－	人時間/OSS	ティア 1 のみ実施
（活用ツール費用）	－	円	－	円	－	円	既保有ツールで実施

表 4-77　Black Duck を使用時の SBOM 活用・管理コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
脆弱性管理工数	脆弱性特定工数	0.08	人時間/OSS	0.08	人時間/OSS	－	人時間/OSS	ティア 1 のみ実施
	脆弱性対応工数	2.0	人時間/OSS	2.0	人時間/OSS	－	人時間/OSS	ティア 1 のみ実施。 ・「特定された脆弱性の、自社 SW への影響有無（対策要否）判断までに要した作業」と「特定された脆弱性の深刻度評価」までに要した作業についての集計である。 ・「要対策脆弱性への対策検討・実施に要した作業」については対策内容によって異な

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
								るため含んでいない。
ライセンス管理工数	ライセンス特定工数	0.08	人時間/OSS	0.08	人時間/OSS	－	人時間/OSS	ティア 1 のみ実施
	ライセンス違反対応工数	0.08	人時間/OSS	0.08	人時間/OSS	－	人時間/OSS	ティア 1 のみ実施。 ・「SBOM 上の OSS のライセンス上の規定を確認」と「自社開発 SW において、ライセンス上の規定に違反した使用形態となっている OSS の特定」までの集計である。 ・「違反状態の是正までに要した改修」については対策内容によって、異なるため含んでいない。
（活用ツール費用）	（活用ツール費用）	－	円	－	円	－	円	既保有ツールで実施

SBOM 手動生成の場合、SBOM の活用・管理には 1OSS 当たり合計で 5.16 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりである。

表 4-78　SBOM 手動生成時の SBOM 活用・管理コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
脆弱性管理工数	4.16	人時間/OSS	2.08	人時間/OSS	2.08	人時間/OSS	―
ライセンス管理工数	1.0	人時間/OSS	0.5	人時間/OSS	0.5	人時間/OSS	―
（活用ツール費用）	0	円	0	円	0	円	―

表 4-79　SBOM 手動生成時の SBOM 活用・管理コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
脆弱性管理工数	脆弱性特定工数	0.16	人時間/OSS	0.08	人時間/OSS	0.08	人時間/OSS	・左記工数に加え、脆弱性情報 DB の情報監視に 1.5 人時間/日の工数を別途要する。 ・SW 部品数は合計、ティア 1、ティア 2 ともに 1571
	脆弱性対応工数	4.0	人時間/OSS	2.0	人時間/OSS	2.0	人時間/OSS	・SW 部品数は、合計、ティア 1、ティア 2 ともに 1571 ・「特定された脆弱性の、自社 SW への影響有無（対策要否）判断までに要した作業」と「特定された脆弱性の深刻度評価」までに要した作業についての集計である。

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
								・「要対策脆弱性への対策検討・実施に要した作業」については脆弱性内容と対策内容によって異なるため含んでいない。
ライセンス管理工数	ライセンス特定工数	0.5	人時間/OSS	0.25	人時間/OSS	0.25	人時間/OSS	SW 部品数は、合計、ティア 1、ティア 2 ともに 1571
	ライセンス違反対応工数(注)	0.5	人時間/OSS	0.25	人時間/OSS	0.25	人時間/OSS	SW 部品数は、合計、ティア 1、ティア 2 ともに 1571 ・「SBOM 上の OSS のライセンス上の規定を確認」と「自社開発 SW において、ライセンス上の規定に違反した使用形態となっている OSS の特定」までの集計である。 ・「違反状態の是正までに要した改修」については対策内容によって異なるため含んでいない。
（活用ツール費用）	（活用ツール費用）	0	円	0	円	0	円	―

(注)ライセンス違反対応工数には、SBOM 上の各 OSS のライセンス内容の確認、及びライセンス違反している OSS の特定を行い、レビューを行う工数を含む。

4.6.6.3.2. 事前に全て精査済みの SBOM を元に深刻度評価を行う場合と、脆弱性検知後に SBOM を精査し深刻度評価をする場合の比較

ケース 1 を事前に全て精査済みの SBOM を元に深刻度評価する場合とし、ケース 2 を脆弱性検知後に SBOM を精査し、深刻度評価する場合とし工数を算出した。

計測するに当たって、脆弱性対応の流れは次のとおりとした。

ケース 1 では、事前に全て精査済みの SBOM を使用して、①ツールの脆弱性通知をトリガーに脆弱性情報と該当部品を特定し、③ツールのセキュリティ情報画面より CVE 情報と CVSS 情報を確認し自システムに対する CVSS 評価を実施。(その後は対応検討プロセスに移行)

ケース 2 では、事前精査なしの SBOM を使用して、①ツールの脆弱性通知をトリガーに脆弱性情報と該当部品を特定し、②該当部品が対象ソフトに含まれているか精査(誤検知の場合は SBOM を更新)し、③ツールのセキュリティ情報画面より CVE 情報と CVSS 情報を確認し自システムに対する CVSS 評価を実施。(その後は対応検討プロセスに移行)

コスト・効果評価においてはケース 1 を採用した。事前に精査を行った SBOM を使用することで脆弱性管理の効果を得られると考えたためである。

表 4-80　深刻度評価工数の SBOM 精査タイミングによる比較

ケース	ツールによる SBOM 作成工数	SBOM 精査工数	①脆弱性通知より該当部品を特定する工数	②SBOM 精査（更新）工数	③深刻度評価（CVSS）工数 (注)	合計工数
（ケース 1） 事前に全て精査済みの SBOM を元に深刻度評価	0.03 人時間	0.08 人時間/1 部品（22 人時間/280 部品）	0.08 人時間/1 通知	—	2.0 人時間	2.19 人時間/1 部品 （SBOM 全体の精査時間を含めると 24.19 人時間）
（ケース 2） 脆弱性検知後に SBOM を精査し、深刻度評価	0.03 人時間	—	0.08 人時間/1 通知	0.08 人時間/1 部品 0.16 人時間/1 部品（誤検知の場合）	2.0 人時間	2.19 人時間/1 部品

(注)深刻度評価工数には、ツールからリンクされた NVD 等の公開脆弱性情報と CVSS スコアを確認し、自システムへの CVSS 評価の実施及びリスク評価を行い、報告書作成とレビューの工数を含んでいる。車両に対してどのような対応をとるかを OEM が判断するための材料となるため、報告書作成とレビューを行う。

4.6.6.3.3. 脆弱性監視工数の比較

SBOM 無しの場合は、独自に管理している構成管理ツールを使用して部品の特定を行うため、SBOM 無し（独自形式）とする。SBOM 無し（独自形式）/手動作成の SBOM 利用/ツール作成の SBOM 利用の各監視工数を計測した。それぞれの脆弱性情報を監視作業は次のとおりとした。

SBOM 無し（独自形式）及び手動 SBOM の場合は、定期的に（毎日を想定）NVD などの脆弱性 DB の新着情報を監視し、新しい情報が公開されると初めに SIRT 管理部署でフィルタリングを行い脆弱性情報の件数を 5%～10%に絞り込む。次に該当する設計部署に情報を振り分け、各設計部署はその脆弱性情報が自ソフトに該当するかしないかを判断する。SBOM 無し（独自形式）の場合は、SBOM を用いて部品検索することができないため、事前に構成管理ツールにソフトの部品情報が登録、またそのソフトの開発を行った設計部署で検索することを前提として計測を行った。

手動 SBOM の場合は、作成されている SBOM を用いて該当するソフトが含まれるか検索を行った。

ツールの場合は、ツールの脆弱性監視機能を用いてツールの脆弱性通知をトリガーとして脆弱性の特定を行った。

表 4-81　脆弱性監視工数

条件		新着脆弱性情報のフィルタリング工数	脆弱性情報より該当ソフトの特定工数	深刻度評価（CVSS）工数
SBOM 無し（独自形式）		1.5 人時間/日	0.17 人時間/1 通知	2.0 人時間
手動作成の SBOM 利用		1.5 人時間/日	0.08 人時間/1 通知	2.0 人時間
ツール作成の SBOM 利用	ティア 1	0 人時間	0.08 人時間/1 通知	2.0 人時間
	ティア 2	0 人時間	0.08 人時間/1 通知	2.0 人時間

4.6.6.3.4. 週 1 回などの定期監視と常時監視の対応期間の比較

4.6.6.3.4.1. 脆弱性対応期間について

週 1 回などの定期監視の場合は、対応が最大 1 週間遅れてしまうためその期間は脆弱性が残存するリスクがある。

車載システムのようにユーザーの安全に関わるシステムの場合には極力迅速な対応が求められるため、常時監視を行う必要がある。

4.6.6.3.4.2. 常時監視工数について

常時監視の場合、ツールを使用しない場合は常に脆弱性情報を監視する必要があるため監視要員のアサインと稼働工数が必要となる。若しくは、監視用の内製ツールを用意する等の対応が必要になる。

ツールの場合、今回使用したツールはいずれも脆弱性監視機能を持っており、ツールの脆弱性通知をトリガーとして脆弱性対応を行うことができる。ツールの脆弱性監視機能は、NVD の情報を数時間ごとに監視し情報が最新に保たれている。NVD に脆弱性情報が公開されてから数時間後、遅くとも半日から 24 時間以内には情報として通知される。ツールの脆弱性監視機能を用いて、自動で監視が可能であり工数を低減できる。

4.6.6.3.5. 課題、解決ノウハウ、留意点

4.6.6.3.5.1. SBOM 精査のタイミングに関して

本実証内で、SBOM を脆弱性管理に使用する場合に、事前に精査するべきか脆弱性情報入手後に精査した方がよいかという課題があった。

工数について比較すると、特に SBOM に含まれる部品の数が多いほど事前精査の工数が大きくなる。事前精査はせずに脆弱性情報入手後に SBOM 該当箇所のみを精査する方が全体工数は抑えられる(ただし、SBOM に含まれている誤検出部品は取り除けていないため、誤通知に対する工数は余分に必要となる)。

しかし、事前に SBOM 精査を行わない場合は工数を抑えられるが、部品の検出漏れが発生していると、その部品が脆弱性監視の対象外となるおそれがある。その場合は脆弱性が発生してもそのリスクが残存したままとなる。そのため、自動車分野等の信頼性が必要なソフトにおいては事前に精査しておく方がよいと考えられる。

4.6.6.4. 効果評価

本実証の結果、脆弱性管理などにかかるコストの低減については効果が確認された。

一方、脆弱性残留リスクの低減、ライセンス違反リスクの低減、についての効果は確認されなかった。また、脆弱性対応期間の低減については対応期間が対象脆弱性によって異なるため計測できなかった。それぞれの効果測定結果と、算出に使用された情報は下表のとおりである。

表 4-82　効果測定結果

分類	効果測定			
	項目	結果		
				（前提条件）
リスク低減	脆弱性残留リスクの低減	805	個多くの OSS につき脆弱性リスクの管理が可能	手動作成時の、ツール利用時との比較
	脆弱性対応期間の低減	ー	日間短縮可能	SBOM 有りの場合の、SBOM なしの場合との比較。 測定不可。
	ライセンス違反リスクの低減	805	個多くの OSS につきライセンス違反リスクの管理が可能	手動作成時の、ツール利用時との比較
コスト低減	脆弱性管理などにかかるコストの低減	1183.9146	人時間の工数削減効果を確認	ツール利用時の、SBOM なしの場合との比較
				700 の部品（平均的な SW に含まれる OSS 数）を管理したと仮定した場合の値。 ツール費用は以下の理由で算入せず。 製品別に価格・料金体系が異なる 実証対象企業全社で使われるため、特定プロジェクトでのコスト按分が困難

表 4-83　効果の算出用情報一覧

分類	効果測定	効果算出用情報						
	項目	情報項目	ツール利用時		手動生成時		SBOM なしの場合	
			値	単位	値	単位	値	単位
リスク低減	脆弱性残留リスクの低減	OSS 特定数	766	個	1571	個	—	
	脆弱性対応期間の低減	脆弱性発見から対応完了までの必要期間	―	日	―	日	―	日
		（小計：ケース別の、SBOM なしの場合からの短縮期間）	―	日	―	日	—	
	ライセンス違反リスクの低減	OSS 特定数	766	個	1571	個	—	
コスト低減	脆弱性管理などにかかるコストの低減	初期導入工数	59	人時間	55	人時間	53	人時間
		1OSS 当たりの運用工数	4.64016	人時間/OSS	6.160038	人時間/OSS	6.340038	人時間/OSS
		一般的な SW での平均 OSS 利用数	700	個	700	個	700	個
		（小計：ケース別の管理工数）	3307.112	人時間	4367.0266	人時間	4491.0266	人時間
		（小計：SBOM なしの場合からの削減効果）	1183.915	人時間	124	人時間	—	

なお、脆弱性管理などにかかるコストの低減効果の算出において、小計として算出した「ケース別の管理工数」は以下の計算式で算出している。

$$初期導入工数 + 1OSS当たりの運用工数 \times 一般的なSWでの平均OSS数$$

4.6.6.4.1. 工数低減効果

4.6.6.4.1.1. 初期工数

SBOM 作成ツールを使用する場合、初期工数としてツールの導入工数が必要であるため、手動に比べて工数が多く必要となる。導入工数としてツールの環境整備（サーバー構築とソフトインストール作業）工数と学習工数をそれぞれ計上した。手動に比べ、初期導入工数として環境整備に要した工数は次のとおりである。

表 4-84　初期工数比較

ツール種別	サポート有無	工数
SaaS 型	有り	利用者一人当たり 3.0h
	無し	利用者一人当たり 6.0h
オンプレミス型	有り	10 人時間及び利用者一人当たり 0.3h
	無し	20 人時間及び利用者一人当たり 0.5h

また、学習工数は一方のツールでベンダーサポート有りの場合に、利用者一人当たり 2.0h～3.0h を要し、ベンダーサポート無しの場合に利用者一人当たり 3.0h～4.0h が必要であった。

より多機能のツールについて、ベンダーサポート有りの場合に利用者一人当たり 16h、ベンダーサポート無しの場合に利用者一人当たり 30h を要した。

4.6.6.4.1.2. 運用工数

各 SBOM 作成方法別に工数を集計した。「手動」はパッケージマネージャから部品表を作成し、手動で各部品の情報を入力した場合であり、「手動（スクリプト）は、パッケージマネージャから部品表を作成し、スクリプトを用いて各部品のライセンス情報を収集することで SBOM を作成した場合である。また、ツール利用の場合は「精査無し」がツールの出力結果をそのまま使用する場合であり、「精査有り」がツールの出力結果に対して精査を行った場合である。

4.6.6.4.1.2.1. SBOM を作成し、SPDX ファイルを出力する工数

SBOM を作成し、SPDX 形式のファイルを出力し情報共有できる形にするまでの工数を比較した。SPDX ファイルは米国大統領令最小要素も満たしている。

表 4-85　SPDX ファイル出力工数比較

SBOM 作成手段	工数	（参考：部品数 100 の場合）
手動	0.25 人時間/1 部品	25 人時間
手動（スクリプト）	0.03 人時間+0.25 人時間/1 部品	25.03 人時間
ツール（精査なし）	0.06 人時間	0.06 人時間

SBOM 作成手段	工数	（参考：部品数 100 の場合）
ツール（精査有り）	0.06 人時間+0.08 人時間/1 部品	8.06 人時間

手動の場合は部品数当たり 0.25 人時間を要している。スクリプトを使用した場合ライセンス情報を入手するまではスクリプトで自動化できたが、SPDX ファイルを出力するに当たって部品数当たり 0.25 人時間を要しており、この場合は手動に対して工数メリットは無かった。

ツールの場合、精査有りの場合に 0.06 人時間＋0.08 人時間の工数となっており、手動に比べて 1 部品当たり 0.17 人時間の工数低減となっている。これはツールの補助機能により、合致率 100%の部品を精査対象から除外し、各部品の詳細情報を利用でき、またスニペットの場合は対象コードの差分画面が表示されるなど、精査工数を低減するための仕組みを利用できるためである。部品数が多くなるほど、工数低減効果が大きくなる。

ツールを精査なしで使用する場合は、工数が 0.06 人時間であり、工数は非常に少なかった。しかし、現状のツールで誤検知や検出漏れは必ず含まれるため、ライセンス管理及び脆弱性管理のいずれにおいてもリスクが大きいため、おおよその OSS の数を把握したい場合などの利用に限定されると考える。

4.6.6.4.1.2.2. ライセンス情報のみの SBOM を作成する工数

表 4-86　ライセンス情報のみの SBOM 作成工数比較

SBOM 作成手段	工数	（参考：部品数 100 の場合）
手動	0.25 人時間/1 部品	25 人時間
手動（スクリプト）（精査無し）	0.03 人時間	0.03 人時間
手動（スクリプト）（精査有り）	0.03 人時間+0.25 人時間/1 部品	25.03 人時間
ツール（精査無し）	0.03 人時間	0.03 人時間
ツール（精査有り）	0.03+0.08 人時間/1 部品	8.03 人時間

手動（スクリプト）の場合、パッケージマネージャで部品表を作成した後に、ライセンス情報をスクリプトで収集して SBOM を作成したが、収集したライセンス情報の精査が必要であったため、0.03+0.25 人時間/1 部品の工数を要しており、ツールの場合も精査時間も含めると 0.03+0.08 人時間/1 部品を要している。

ライセンス情報の確認のために SBOM を活用する場合、ツール（精査有り）が目的を達成でき、工数低減効果が高い。

脆弱性特定のみに使用する場合、パッケージマネージャから出力した部品表を使用できるのであれば、手動（スクリプト）（精査無し）が目的を達成でき、工数低減効果が高い。パッケージマネージャを使用できない開発環境の場合は、ツール（精査有り）が目的を達成

でき工数低減効果が高い。

4.6.6.4.1.3. 脆弱性の深刻度評価工数

表 4-87　深刻度評価工数比較

条件		新着脆弱性情報のフィルタリング工数	脆弱性特定～深刻度評価工数
SBOM 無し（独自形式）		1.5 人時間/日	2.17 人時間/1 通知
手動作成の SBOM 利用		1.5 人時間/日	2.08 人時間/1 通知
ツール作成の SBOM 利用	ティア 1	0 人時間	2.08 人時間/1 通知
	ティア 2	0 人時間	2.08 人時間/1 通知

SBOM 無し（独自形式）の場合は、構成管理ツールにソフトの部品情報が登録されており、該当ソフトの設計部署で特定することを前提としている。

SBOM 無し(独自形式)と手動作成の SBOM の場合は、脆弱性情報 DB の情報をフィルタリングし、該当開発部署へ情報を振り分ける工数が必要である。(脆弱性情報 DB の公開情報は多数であるが、今回は情報取得を自動化し、情報のフィルタリングを手動で行うことを想定している。さらに情報のフィルタリング処理もキーワードマッチで自動化する等は省力化の方策として考えられる)

それに対してツール作成の SBOM を使用する場合は、ツールの脆弱性通知機能を使用することによりこれらの工数を削減できる。全体として、ツール使用の方が工数低減効果は大きい。

4.6.6.4.1.4. ツール導入工数に対する採算性について

ツールの導入には初期工数が必要であるが、扱う部品が多いほどツールによる工数低減効果が得られる。SPDX 形式の SBOM を作成した場合、どの程度の部品数で採算がとれるのかを次の表の値で検討した。なお、ツール費用は含んでいない。

表 4-88　ツール導入工数に対する採算性

SBOM 作成手段	初期工数		運用工数	
	環境構築	学習	SBOM 作成	精査
手動作成	—	—	0.25 人時間/1 部品	左記に含まれる
ツール使用	10 人時間	16 時間/人	0.06 人時間	0.08 人時間/1 部品

学習工数は 2 つのツールで異なったが、より大きい工数の方を使用し、1 人当たり 16 時間とした。この場合、利用者 1 人とすると 154 部品で工数低減効果が初期工数を上回った。さらに利用者が 1 人増えるごとに、部品数を 95 部品ずつ上乗せすると工数低減効果が初期

工数を上回る結果となった。

4.6.6.4.2. SBOM の精査

表 4-89 SBOM 精査工数比較

条件		工数	（参考：部品数 100 の場合）
手動作成の SBOM	ティア 1	0.25 人時間/1 部品	25 人時間
	ティア 2	0.25 人時間/1 部品	25 人時間
ツール作成の SBOM	ティア 1	0.08 人時間/1 部品	8.0 人時間
	ティア 2	0.08 人時間/1 部品	8.0 人時間

手動で作成した SBOM の精査には 1 部品当たり 0.25 人時間の工数を要したが、ツール作成の SBOM の場合には 1 部品当たり 0.08 人時間であり、工数低減効果があった。

ツール使用の場合には、検出した部品のバージョン情報などの表示、配布サイトへのリンク、該当 OSS との合致率の表示、スニペット上の差分表示、など、誤検知検出のための補助手段が使用できるため、手動による精査に比べて工数を低減することができた。

4.6.6.4.3. 部品の包含関係を含む全ての粒度の SBOM を保持することが有効か

今回の実証実験では、手動作成の SBOM として、パッケージマネージャの情報を利用して OSS パッケージに含まれる部品の粒度までの SBOM を作成した。（4.6.6.2.8.1）
現状のツールで生成している SBOM は OSS パッケージの粒度で作成される。NVD などの脆弱性情報 DB で公開される脆弱性情報の粒度はさまざまであるが、OSS パッケージの粒度の情報であれば、そのままツール生成 SBOM と紐づけられ、脆弱性の内容を確認することになる。脆弱性情報がより細かい部品の粒度の場合も OSS パッケージ単位で情報を紐づけ脆弱性の内容を確認することになる。

細かい部品の粒度までを含む SBOM であれば、OSS パッケージの粒度からそれに含まれる部品の粒度までを含むため、脆弱性情報 DB の様々な粒度の情報に、よりピンポイントに対応付けられ、該否判定と影響分析の作業もより正確にかつ期間も短縮できると考えられる。また、脆弱性を監視する際の工数についても、ツールの通知機能を使用できるため管理工数自体は変わらない。そのため SBOM の粒度は細かい方がより活用はしやすい。

将来的なツールの検出範囲として、OSS パッケージの粒度から、OSS に含まれる部品の粒度まで検出できるようになると、より脆弱性への対応期間が短縮されていくと考えられる。

4.6.6.4.4. 脆弱性対応期間の短縮効果について

SBOM 無し(独自形式)の場合及び手動 SBOM を使用して脆弱性監視する場合には、定期

的に脆弱性情報 DB から情報を入手する必要がある。そのため、脆弱性対応の開始に当たって、その定期確認の間隔分のリードタイムが必要となる。

一方、ツール生成の SBOM であれば、ツールの脆弱性通知受領後にすぐに初動を開始することができ、脆弱性対応開始のリードタイムが短縮される。(当然、効果の大きさは各ツールの脆弱性情報通知の早さにもよるため、ツールの仕様を確認する必要がある)

自動車分野など迅速な脆弱性対応が求められるシステムにおいては、対応遅れのリスクを軽減できることが、ツールのメリットとなる。

留意点として、ツールで SBOM を生成したとしても、そもそも SBOM に含まれていない部品については通知を受け取れない。脆弱性の対応期間短縮の効果を十分に得るためには、正確な SBOM の作成及び維持が必要である。

4.6.6.4.5. SBOM 活用の嬉しさについて

4.6.6.4.5.1. コンプライアンス上の嬉しさ

含まれている部品のライセンス情報を確認できるため、コンプライアンス上の過失を防ぐことができる。手動で作成した SBOM であってもツールで作成した SBOM であってもライセンス情報を明示し、各ライセンスへの対応漏れの防止に有効である。ツールを使用した場合、SBOM 作成工数を低減できるだけでなく、各ライセンスの内容の表示や、より注意が必要なライセンスが警告されるなどのサポート機能を活用できるため、ライセンスへの対応を行う者にとって有効である。

4.6.6.4.5.2. サイバーセキュリティ上の嬉しさ

手動の SBOM であってもツールで生成した SBOM であっても、SBOM の各部品と脆弱性情報 DB の情報を紐づけることにより、網羅的に脆弱性対応を行うことができる。
新しい脆弱性が発見された場合、手動の SBOM では定期的に脆弱性情報 DB を確認する作業が必要であり、定期の間隔の長さで、情報公開から脆弱性対応までのタイムラグが発生する。

ツールを使用した場合、ツールの常時監視により通知を受け取ることができ、脆弱性の確認漏れを防ぎ、初動が早くなりより迅速な対応が可能となる。

4.6.6.4.5.3. 情報共有の嬉しさ

ティア 1 からティア 2 へ開発を委託した際、開発元のティア 2 と SBOM を共有すれば、コンプライアンス上のリスクとサイバーセキュリティ上のリスクを共有することができる。特にソースコードの納入がない場合、ティア 1 の方で正確な SBOM を作成することは難しく、上記のようなリスクがないかを確認することが難しい。しかし、ティア 2 から SBOM が提供されれば同等の管理ができる。

4.6.6.5. SBOM 適用範囲（SBOM 活用モデル）の改訂案

本実証を踏まえ改定した SBOM 適用範囲を次の図に示す。緑は「適用可能」、黄は「部

分適用」、赤は「適用困難」、灰は「不要・対象外」を示す。

表 4-90　SBOM 適用範囲（SBOM 活用モデル）の改訂案

作成範囲	生成方法	生成項目	活用法
(b1) 直接利用部品	(c1) 手動で特定（構成管理情報利用）・ツールで生成	(d1) 標準フォーマット (SPDX、SPDXLite 等)	(e1) (e2) 脆弱性・ライセンスの特定
			(e3) (e4) 悪用可能性・深刻度の評価
		(d2) 大統領令最小要素を含む	(e1) (e2) 脆弱性・ライセンスの特定
			(e3) (e4) 悪用可能性・深刻度の評価
		(d3) 上記の一部のみ	(e1) (e2) 脆弱性・ライセンスの特定
			(e3) (e4) 悪用可能性・深刻度の評価
	(c2) ツールで特定・生成・誤検知精査なし	(d1) 標準フォーマット (SPDX、SPDXLite 等)	(e1) (e2) 脆弱性・ライセンスの特定
			(e3) (e4) 悪用可能性・深刻度の評価
		(d2) 大統領令最小要素を含む	(e1) (e2) 脆弱性・ライセンスの特定
			(e3) (e4) 悪用可能性・深刻度の評価
		(d3) 上記を満たさない要素	(e1) (e2) 脆弱性・ライセンスの特定
			(e3) (e4) 悪用可能性・深刻度の評価
	(c3) ツールで特定・生成・誤検知精査あり	(d1) 標準フォーマット (SPDX、SPDXLite 等)	(e1) (e2) 脆弱性・ライセンスの特定
			(e3) (e4) 悪用可能性・深刻度の評価
		(d2) 大統領令最小要素を含む	(e1) (e2) 脆弱性・ライセンスの特定

作成範囲	生成方法	生成項目	活用法
			（e3）（e4）悪用可能性・深刻度の評価
		（d3）上記を満たさない要素	（e1）（e2）脆弱性・ライセンスの特定
			（e3）（e4）悪用可能性・深刻度の評価
（b2）間接利用部品	（c1）手動で特定（構成管理情報利用）・ツールで生成	（d1）標準フォーマット（SPDX、SPDXLite 等）	（e1）（e2）脆弱性・ライセンスの特定
			（e3）（e4）悪用可能性・深刻度の評価
		（d2）大統領令最小要素を含む	（e1）（e2）脆弱性・ライセンスの特定
			（e3）（e4）悪用可能性・深刻度の評価
		（d3）上記を満たさない要素	（e1）（e2）脆弱性・ライセンスの特定
			（e3）（e4）悪用可能性・深刻度の評価
	（c2）ツールで特定・生成・誤検知精査なし	（d1）標準フォーマット（SPDX、SPDXLite 等）	（e1）（e2）脆弱性・ライセンスの特定
			（e3）（e4）悪用可能性・深刻度の評価
		（d2）大統領令最小要素を含む	（e1）（e2）脆弱性・ライセンスの特定
			（e3）（e4）悪用可能性・深刻度の評価
		（d3）上記を満たさない要素	（e1）（e2）脆弱性・ライセンスの特定
			（e3）（e4）悪用可能性・深刻度の評価

作成範囲	生成方法	生成項目	活用法
	（c3）ツールで特定・生成・誤検知精査あり	（d1）標準フォーマット（SPDX、SPDXLite 等）	（e1）（e2）脆弱性・ライセンスの特定
			（e3）（e4）悪用可能性・深刻度の評価
		（d2）大統領令最小要素を含む	（e1）（e2）脆弱性・ライセンスの特定
			（e3）（e4）悪用可能性・深刻度の評価
		（d3）上記を満たさない要素	（e1）（e2）脆弱性・ライセンスの特定
			（e3）（e4）悪用可能性・深刻度の評価
（b1）直接利用部品	（c1）手動で特定（構成管理情報利用）・ツールで生成	（d1）標準フォーマット（SPDX、SPDXLite 等）	（e1）（e2）脆弱性・ライセンスの特定
			（e3）（e4）悪用可能性・深刻度の評価
		（d2）大統領令最小要素を含む	（e1）（e2）脆弱性・ライセンスの特定
			（e3）（e4）悪用可能性・深刻度の評価
		（d3）上記を満たさない要素	（e1）（e2）脆弱性・ライセンスの特定
			（e3）（e4）悪用可能性・深刻度の評価
	（c2）ツールで特定・生成・誤検知精査なし	（d1）標準フォーマット（SPDX、SPDXLite 等）	（e1）（e2）脆弱性・ライセンスの特定
			（e3）（e4）悪用可能性・深刻度の評価
		（d2）大統領令最小要素を含む	（e1）（e2）脆弱性・ライセンスの特定
			（e3）（e4）悪用可能性・深刻度の評価

作成範囲	生成方法	生成項目	活用法
		(d3) 上記を満たさない要素	(e1) (e2) 脆弱性・ライセンスの特定
			(e3) (e4) 悪用可能性・深刻度の評価
	(c3) ツールで特定・生成・誤検知精査あり	(d1) 標準フォーマット (SPDX、SPDXLite 等)	(e1) (e2) 脆弱性・ライセンスの特定
			(e3) (e4) 悪用可能性・深刻度の評価
		(d2) 大統領令最小要素を含む	(e1) (e2) 脆弱性・ライセンスの特定
			(e3) (e4) 悪用可能性・深刻度の評価
		(d3) 上記を満たさない要素	(e1) (e2) 脆弱性・ライセンスの特定
			(e3) (e4) 悪用可能性・深刻度の評価
(b2) 間接利用部品	(c1) 手動で特定（構成管理情報利用）・ツールで生成	(d1) 標準フォーマット (SPDX、SPDXLite 等)	(e1) (e2) 脆弱性・ライセンスの特定
			(e3) (e4) 悪用可能性・深刻度の評価
		(d2) 大統領令最小要素を含む	(e1) (e2) 脆弱性・ライセンスの特定
			(e3) (e4) 悪用可能性・深刻度の評価
		(d3) 上記を満たさない要素	(e1) (e2) 脆弱性・ライセンスの特定
			(e3) (e4) 悪用可能性・深刻度の評価
			(e1) (e2) 脆弱性・ライセンスの特定

作成範囲	生成方法	生成項目	活用法
	（c2）ツールで特定・生成・誤検知精査なし	（d1）標準フォーマット（SPDX、SPDXLite 等）	（e3）（e4）悪用可能性・深刻度の評価
		（d2）大統領令最小要素を含む	（e1）（e2）脆弱性・ライセンスの特定
			（e3）（e4）悪用可能性・深刻度の評価
		（d3）上記を満たさない要素	（e1）（e2）脆弱性・ライセンスの特定
			（e3）（e4）悪用可能性・深刻度の評価
	（c3）ツールで特定・生成・誤検知精査あり	（d1）標準フォーマット（SPDX、SPDXLite 等）	（e1）（e2）脆弱性・ライセンスの特定
			（e3）（e4）悪用可能性・深刻度の評価
		（d2）大統領令最小要素を含む	（e1）（e2）脆弱性・ライセンスの特定
			（e3）（e4）悪用可能性・深刻度の評価
		（d3）上記を満たさない要素	（e1）（e2）脆弱性・ライセンスの特定
			（e3）（e4）悪用可能性・深刻度の評価

4.6.6.5.1. SBOM 適用範囲の変更箇所

「(a1)自社」についての変更箇所は次のとおりである。

- 直接利用部品について、「(c3)ツールで特定・生成・誤検知精査有り」の項目について、「(e3)(e4)悪用可能性・深刻度評価」を部分適用とし、「 (e1)(e2)脆弱性・ライセンスの特定」を適用可能とした。
- 「(b2)間接利用部品」について、部分適用とした。

「(a2)サプライヤー(開発委託先)取引契約あり」についての変更箇所は次のとおりである。

- 「(b1)直接利用部品について、c1,c2,c3 全てにおいて、「(e3)(e4)悪用可能性・深刻度評価」を部分適用とし、「 (e1)(e2)脆弱性・ライセンスの特定」を適用可能とした。
- 「(b2)間接利用部品」について、部分適用とした。

4.6.6.5.2. 変更理由

- ツール生成の SBOM に誤検知が含まれる可能性があるので、十分に SBOM の効果を得るためにはツールで特定した部品の誤検知精査は必要と判断したため、誤精査有りの項目を含めた。
- 間接利用部品に対しても、パッケージマネージャを利用するなどして手動による特定も一部実施可能であると判断したため、間接利用部品について部分適用とした
- ティア 2 においても、対象ソースコードを所持しておりティア 1 と同等の SBOM 作成が可能と判断したため、ティア 1 同様の項目選択とした。

4.6.6.6. 取引契約における要求条項等

サプライチェーンにおける取引契約において委託先などに対して、主にティア 1 とティア 2 間の契約における SBOM 関連要求事項をまとめる。

この点、前項で論じた SBOM 適用範囲（SBOM 活用モデル）も、取引契約における要求事項として重要な論点の 1 つである。作成範囲・生成方法・生成項目・活用法について、どの範囲までを要求条項とするかについては、個々の取引先への委託内容の他、自動車業界の場合は納品先の OEM の要件や各種法規対応上必要な対応等を踏まえ、契約上明確化することが推奨される。

本項では、SBOM 適用範囲（SBOM 活用モデル）で論じた事項以外の重要な要求条項についてまとめる。

4.6.6.6.1. 開発中の作業について

開発開始前に次のような項目について合意しておいた方がよいと考える。

- 使用する OSS 等のソフト部品。契約時点で使用する OSS 等のソフト部品を特定するのが難しい場合、使用を認める OSS のホワイトリストや、OSS 利用時の取引先間での合意フロー、使用する OSS の選定基準（OSS ポリシー）等の取り決めにより代替

することも考えられる。

- SBOM の生成・更新タイミングやルール、その他開発中の SBOM 運用のあり方。
- 監視する脆弱性 DB。
- 脆弱性検知時の報告タイミング/報告内容

4.6.6.6.2. 開発終了後の作業について

これまでの品質保証の仕組みを参考に当事者間の取り決めによる。

ティア 1 は脆弱性の監視を行う必要がある。ティア 2 はティア 1 との協議に応じて対応する。

また、開発完了後の脆弱性検出時の対応や、OTA アップデート時の SBOM 作成・管理についてのルール(差分のみ作成するか、全体を作り直すか)等については事前に OEM とティア 1 で合意しておくことが望ましい。

4.6.6.6.3. SBOM データの共有について

自動車の制御系においてはソースコードを含めて納品することがあり、この場合は全てティア 1 でツールを使った解析を行うことが可能である。この場合も、ツールの出力結果をティア 2 に公開し、ティア 1 ティア 2 双方で誤検出や検出漏れの有無を一度確認することが推奨される。

納入されるソフトの一部又は全体がバイナリの場合はティア 1 で全ての対応ができないため、バイナリ部分の責任がティア 2 であること、不具合や脆弱性への対応作業を保証してもらうこと、またティア 1 としての対応のために SBOM データは提出してもらうこと、などを事前に合意しておく必要がある。

4.6.6.6.4. 今後の課題について

自動車分野において、脆弱性を監視し特定するために SBOM の重要性が高まっていくと考えられる。

また、SBOM を活用していくに当たっては、SBOM の作成や、SBOM の維持管理等の機能や責任をサプライチェーンの中でどのように持つべきか、今後も継続的に検討されていくと考えられる。

4.7. ソフトウェア分野（トレンドマイクロ）

4.7.1. 実証の狙い・重点ポイント

自社にて製品開発を行うソフトウェアベンダーかつ、製品のサポート OS や使用プログラミング言語が幅広い企業の立場から、全社的な SBOM の効率的かつ持続可能な作成・管理方法について検討するとともに、今後同業種での SBOM 活用に向けた技術的または運用面での利点と、課題および留意点について考察する。

特に SBOM の作成・管理に関しては、昨今のアジャイル開発や DevOps のスピード感に合わせた対応が大きなコスト負担なしに可能かどうかにも注目した。

また、SBOM の活用においては自社内およびエンドユーザ双方にとって、SBOM が脆弱性対応の迅速化および管理の効率化に寄与するために必要な考慮、プロセスについて検討を実施した。

4.7.2. 実証の手順

本実証においては、トレンドマイクロ開発の企業向け製品である Trend Micro Safe Lock 2.0[α]（以下、TMSL）を対象として SBOM の作成や共有、活用管理を検討することとした。共有、活用、管理においては、TMSL だけでなく、トレンドマイクロの複数製品において SBOM が生成されていると仮定して検討を行った。

SBOM 生成は、すでにトレンドマイクロで ソフトウェア構成分析（Software Composition Analysis; SCA）ツールとして導入済みであった有償ツール（ツール名は非公開。以下 SCA ツール）を利用することとした。SBOM の活用実証においては、特定の OSS に新しい脆弱性が発見された場合を仮定し、該当脆弱性に関するサプライヤーとしての影響度調査の対応速度および効率性に変化があるか検証するため、以下の三つの手法で必要工数と時間を算出し、比較を行った。

A) SBOM ファイルを用いた場合
B) SCA ツールの UI を用いた場合
C) 手動で確認を実施した場合

また、製品顧客から SBOM の提供を求められた場合を仮定し、顧客への提供形態、プロセス、および提供の際の注意点について検討を行った。

各実証項目においては、SBOM の作成や活用による効率性について基本的には作業時間をベースにコスト評価を行った。補足情報として作業時間を金額に換算する場合には昨年度の実証に倣い、一般社団法人日本ニアショア開発推進機構の「エンジニア単価情報 2021 年版レポート」を参照した。技術レベルごとのエンジニア単価の対応は下表の通りである。なお、一時間当たりの単価を月額参考価格から計算するにあたっては、一ヶ月の業務時間を 160 時間として算出した。

[α] TREND MICRO および Trend Micro Safe Lock は、トレンドマイクロ株式会社の登録商標です。本ドキュメントに記載されている各社の社名、製品名およびサービス名は、各社の商標または登録商標です。

表 4-91　人的コストの算出法

技術レベル	説明	一時間当たりのエンジニア単価（円）
初級者	開発エンジニアまたは IT エンジニアとして 3 年未満の経験があるエンジニア。役割別エンジニア単価情報 2021 の「IT アーキテクト系」エンジニアのレベル 2 に相当	4,820
中級者	開発エンジニアまたは IT エンジニアとして 3-7 年以上の経験を持つ。エンジニアとして自立して業務が可能なレベル。役割別エンジニア単価情報 2021 の「IT アーキテクト系」エンジニアのレベル 3 に相当	5,350
上級者	アーキテクトなど、技術的な方向性を決定する上位エンジニア。役割別エンジニア単価情報 2021 の「IT アーキテクト系」エンジニアのレベル 4 に相当	6,350

4.7.3. 実証の対象システムの概要と構成

本実証の対象とする TMSL は、Windows OS 向けのホワイトリスト型のエンドポイントセキュリティ製品である。Windows 2000 から Windows 10 までの幅広い OS をサポートしており、制御機器や組み込み端末などの特定用途機器や、クローズド環境下の端末、レガシー OS など従来型のエンドポイントセキュリティ製品が利用できない環境向けに提供されている。

管理対象端末にインストールされたエージェントは、管理コンソールで管理者により設定された実行可能なアプリケーションのリストや、ふるまい検知機能によって、マルウェアや悪意のあるファイル、未許可のアプリケーションの実行や、それらによるシステムの変更を防ぐ。管理コンソールでは、上記のポリシーの設定の他、アプリケーションのブロックといったエージェント側でのイベントの確認などの管理者機能を提供している（下図参照）。

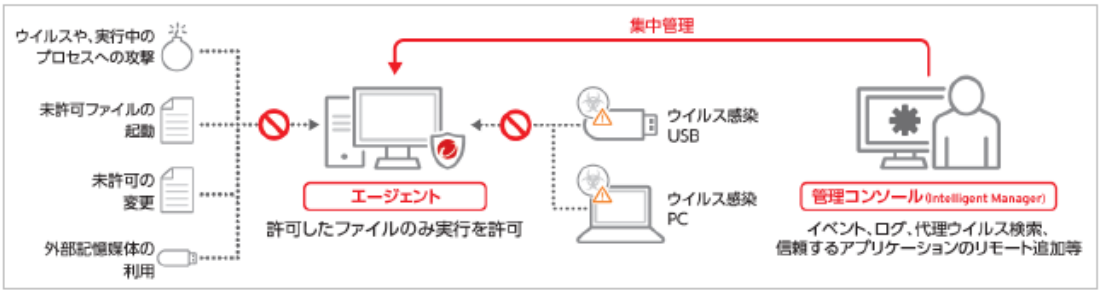


図 4-18　TMSL の基本機能

TMSL のシステム構成および SBOM の作成範囲については下図の通りである。下図に記載のある挙動監視エンジンは自社開発のコンポーネントであるが、TMSL の開発チームとは別の開発チームが担当しており、今回の SBOM　作成の範囲からは除外した。また OSS や .NET Framework、Visual Studio のランタイムライブラリを利用しているものの、社外の第三者に開発を委託している部分は存在しない。

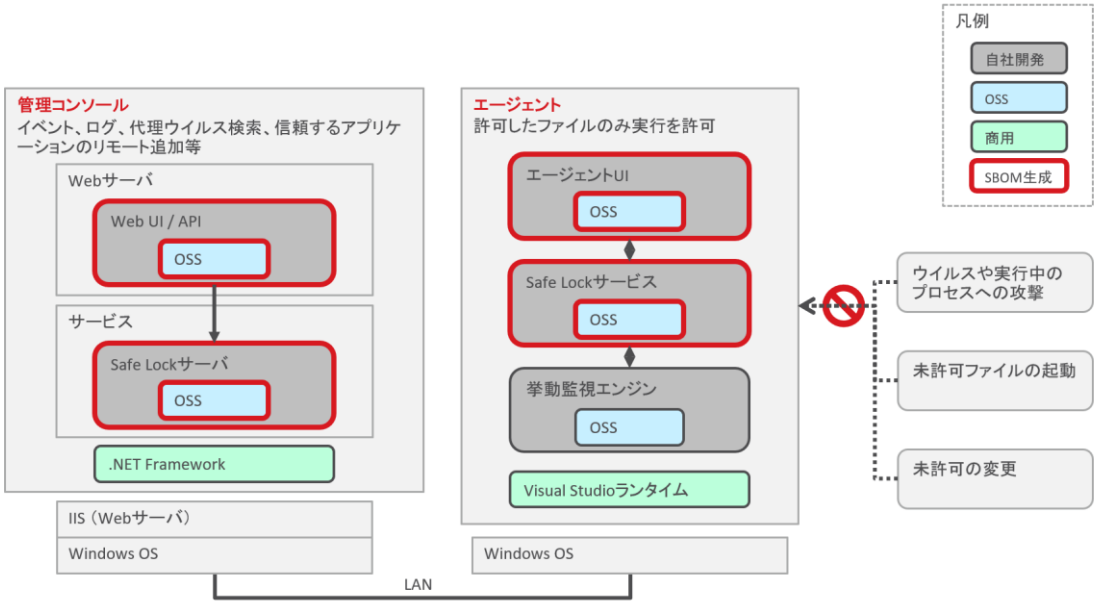


図 4-19　TMSL の構成と SBOM 作成範囲

SBOM 生成にあたってはフォーマットを多くのツールがサポートしている SPDX とした。出力ファイル形式は .json 形式を選択した。また、SBOM 生成範囲のコンポーネントの開発言語は以下の通りである。ファイルフォーマットおよび下記言語のいずれも SCA ツールのサポート対象であることを確認している。

表 4-92　TMSL の開発言語

構成	要素	開発言語
管理コンソール	Web UI / API	C#, JavaScript, PHP
	Safe Lock サーバ	C#
エージェント	エージェント UI	C++
	Safe Lock サービス	C++

4.7.4. 対象分野の前提条件・取引慣行等

トレンドマイクロでは OSS および一部の有償ライブラリを調達して製品開発を実施しているものの、開発の一部を外部に委託して開発することは原則として行っていない。したがって、本項では自社開発のソフトウェアベンダーとユーザ間の契約形態および取引慣行等について下の表に示す。

表 4-93　契約形態および取引慣行について

	本実証における形態	業界において想定される主な形態
契約類型	既製品ライセンス契約	既製品ライセンス契約
部品情報提供の要件	サポート期間中の製品について、契約ユーザ（場合によっては購入検討中の見込み顧客）に無償提供（予定）	契約ユーザに対して無償提供（ファイルまたはユーザが自ら取得できるコマンド等の提供）
第三者部品の申告	OSS および有償ライブラリについては製品同梱のテキストファイルにてライセンス情報を提供	OSS および有償ライブラリのライセンス形態に合わせた情報提供（ウェブページからの参照、電子ファイルにて提供など）
脆弱性の修正	● トレンドマイクロが修正すべきと判断した脆弱性に対して、修正プログラムを契約中のユーザに提供する（対象はサポート期間中の製品）。 ● 原則 SaaS の製品を除き、ユーザ側での適用作業が必要。	自社が修正すべきと判断した脆弱性に対して、修正プログラムの提供。原則対象はサポート期間中の製品だが、影響度が極めて高い場合にはサポート対象外でユーザが多い製品・バージョンにも修正が提供される場合がある（例：Microsoft 社のサポート終了済み OS に対する対応など）
脆弱性の告知	● トレンドマイクロのサポートウェブページでの告知や担当者へのメールなどで実施。 ● 悪用事例があるなど、特に注意喚起が必要なものについては、有償サポート契約ユーザの担当者に電話で連絡するなど、特別な告知を実施する場合もある。 ● JPCERT と脆弱性公表に関して連携している。	サポートウェブページなどで告知
納品形態	● バイナリパッケージを電子納品。一部製品はインターネットから自動アップデート機能を使用してプ	バイナリパッケージのインターネット上での配布。SaaS 製品は、ライセンスの提供のみを行い、製品

	ログラムを配信する。SaaS 製品は、ライセンスの提供のみを行い、製品コンソールへのアクセス権を付与する。ハードウェアアプライアンス製品の場合にはソフトウェアは機器組み込みにて提供。 ● クラウドサービスの Market Place などでは仮想マシンイメージやコンテナとして OS 含めて提供される場合もある。	コンソールへのアクセス権を付与。 ハードウェア製品の場合にはソフトウェアは機器組み込みにて提供
損害賠償責任	使用許諾契約総覧には以下のような記載が存在する。 ● 物理的なインストールメディアの欠陥に関しては購入後一定期間内で無償交換 ● 上記を除き、ライセンス製品、マニュアルまたはスタンダードサポートに関しては一切の保証を行わない ● ライセンス製品、マニュアル、スタンダードサポート、またはサポートが受けられないことに起因して生じた損害に関しては、一切責任を負わない	トレンドマイクロと同等の賠償責任規定を持つ会社が多い印象
知的財産権の帰属	トレンドマイクロ	自社に帰属

また、開発形態等に関して、トレンドマイクロおよび業界内の主な状況を下表に示す。製品によりサポート OS や開発言語は異なり、トレンドマイクロにおいても幅広いサポート OS や開発言語を用いている。

表 4-94　開発言語、環境およびツールに関する情報

項目	トレンドマイクロにおける形態	業界において想定される主な形態
開発言語	C / C++ / Objective-C / Kotlin / Go / C# / PHP / Java / JavaScript / Python / Lua など	Python/ Java / Go / JavaScript / Rust / Swift / Objective-C / C / C++ / C# など
部品形態	OS、OS ライブラリ、Web アプリケーションフレームワーク、データベースサービスなど SaaS の場合はクラウドサービス提供のサービスなどを含む	同左
開発環境ツール	Visual Studio シリーズ、Eclipse、	同左

	Android Studio、 XCode など	
ビルドツール	（非公開）	CI/CD Pipeline：Jenkins、 Circle CI、 Gitlab、 Bamboo
構成管理ツール	（非公開）	Github（Github.com / Github EE）、Gitlab、 Team Foundation Server、subversion

4.7.5. 実証項目の特定

SBOM の生成から活用・管理までの各フェーズにおける実証項目を以下に記述する。

4.7.5.1. SBOM 初期導入

すでに OSS の脆弱性管理用に導入していた SCA ツールで SBOM を作成するために、必要な追加設定や必要な学習に関する工数について実測した。また、過去同ツールを全社導入した際の工数および導入コストを社内で調査することで、SBOM 作成のために初めて有償ツールを導入した場合の総合的な工数およびコストについて評価した。

この情報をもとに、全社的に有償ツールを導入するにあたり、検討すべき項目や注意点について考察を実施した。

4.7.5.2. SBOM 生成

ソフトウェアベンダーとして、自社で販売するすべての製品での自社開発部分および使用 OSS に対する SBOM を作成するにあたり、必要なコストと課題について確認した。また、自社の他開発チーム開発部分のコンポーネント（挙動監視エンジン）を、疑似的に「外部サプライヤーから調達するコンポーネント」として仮定し、SBOM 作成予定や、これらの情報を含んだ SBOM を生成する際に想定される方法や課題について検討した。

SBOM の作成に当たっては前項で記載の通り有償ツールを用いた。なお製品ソースコードレポジトリにて、製品で使用している OSS のソースコードのバージョン管理を一緒に実施することで、間接利用部品に関する情報がツールにより取得できると仮定し、作業を実施した。

出力された SBOM の正確性については、大統領令最小要素をすべて含んでいるか念のため確認するとともに、複数の角度から検証を実施した。検証する項目および手法については下表に記載の通りであった。なお、便宜上、これ以後自社開発部分のコード内で直接参照・依存している OSS や他社開発のライブラリを「トップレベルコンポーネント」、トップレベルコンポーネントが直接的または間接的に参照・依存している他の OSS やライブラリを「サブレベルコンポーネント」と呼ぶ。

表 4-95 生成された SBOM の正確性に関する検討項目

検討項目	説明・手法

自動検出における OSS の過検知率	ツールによる OSS の自動検出において、実際には使用していない OSS を検知した割合。自動検出の結果を、手動で確認することにより算出した。有償ツールによる間接利用部品のカバレッジ検出の正確性に関係。
バージョン情報の正答率	正しく検出されたコンポーネントのうち、そのバージョン情報も正しく検出されたものの割合。脆弱性管理においては、使用されているコンポーネントの名前だけでなく、バージョン情報も重要なため、正確性を確認する必要があると考え、この項目を確認した。
トップレベルコンポーネントにおける自動検出率	手動にて確認している製品のトップレベルコンポーネント（OSS）のうち、ツールによる自動検出で検出可能だったコンポーネントの割合。有償ツールによる間接利用部品のカバレッジ検出の正確性に関係。
トップレベルコンポーネントにおけるツールの捕捉率	対象製品で使用されているトップレベルの OSS について、ツールで管理可能であったコンポーネントの割合。自動検出で正しく補足できたコンポーネント、および SCA ツールのコンソールから開発エンジニアが手動で情報追加が可能だったコンポーネントからなる。

4.7.5.3. SBOM のユーザへの共有

トレンドマイクロでは、基本的に不特定多数のユーザへ製品を提供しているという事実を踏まえ、SBOM の情報をファイルとしてユーザ企業に提供する場合に必要なプロセスおよび考慮事項について検討した。検討に当たっては、製品ユーザ企業およびパートナー企業に対して、SBOM 活用に関する実情や期待、課題についてヒアリングを実施し、その回答を参考とすることとした。また、SBOM に誤りが発見された場合に必要な対応についても合わせて検討した。具体的な検討項目は以下の通りである。

表 4-96　顧客への SBOM 提供に関する検討事項

検討項目	詳細
提供形態	ファイル形式、一般公開の有無 製品および修正プログラムのリリース形態や頻度による SBOM 提供の可否
提供条件	有償とするか否か、特定の契約を持つユーザのみに制限するかなど
顧客との契約条項の修正有無	SBOM に誤りがあった場合の免責事項を含める必要があるか

加えて、SBOM を作成・提供することによってもたらされる、顧客およびベンダー双方における価値および課題について考察した。

4.7.5.4. SBOM 活用・管理

活用・管理においては、1）ソフトウェアベンダーとしての自社製品の SBOM 管理、運用の主体となる場合、2）当社顧客が当社から提供された SBOM 管理運用の主体となる場合の 2 つのパターンが想定される。この 2 つのパターンそれぞれにおいて、主に脆弱性調査の観点での SBOM の活用・管理に関する実証を行った。

ベンダーとしての活用においては、開発チーム視点での対応と PSIRT チーム視点での対応の双方について検討を実施した（チームによる脆弱性調査フローの違いを以下の図に示す）。

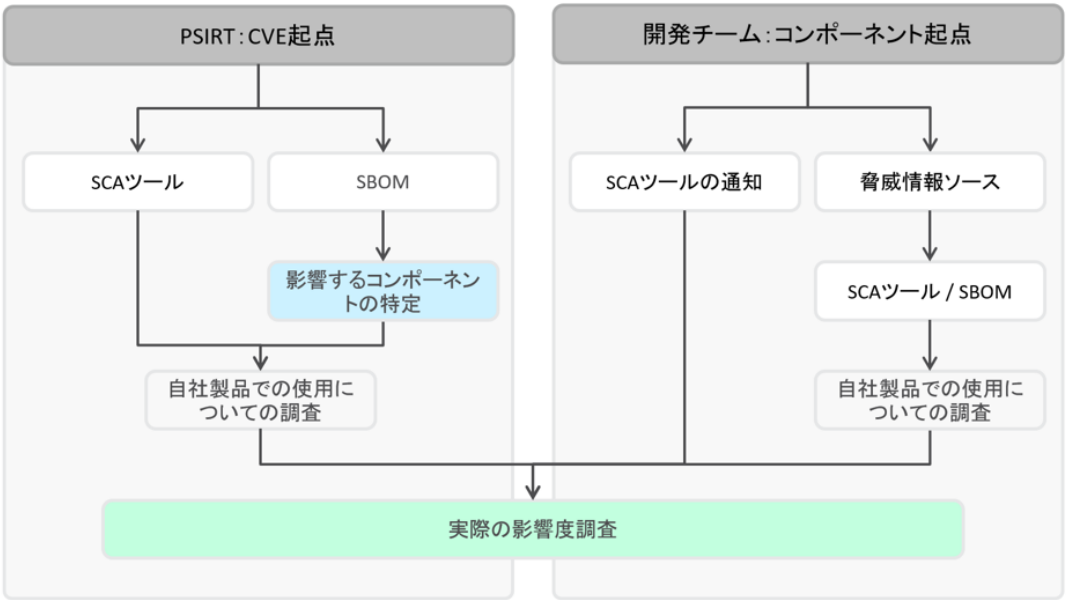


図 4-20　チームによる脆弱性調査フローの違い

いずれの場合も新たに脆弱性が確認された状況を念頭に置き、過去 3 年の間に実際に報告された脆弱性 3 件を例として、下表に示す手法で自社開発の製品における脆弱性の影響度を調査したときの工数を計測・比較した。

表 4-97　比較検証した脆弱性の調査手法

主体	手法
開発チーム	SCA ツールからの通知により脆弱性調査を開始する場合
PSIRT チーム	SBOM を蓄積した自社開発の SBOM 管理システムを使用した場合
	SCA ツールのコンソールを用いた場合
	過去使用していた自社開発の OSS 管理システムから OSS のリストを抽出して実施した場合

PSIRT チーム主体での調査で用いた、自社開発の SBOM 管理システムの主要機能を下表に示す。

表 4-98　社内 SBOM 管理システムの機能

機能	詳細
製品情報登録・管理	SBOM を提供する製品・バージョン情報の登録および管理を行う。以下の情報からなる。 ● 製品名 ● 製品バージョン ● 製品種別（オンプレミス、SaaS、共通モジュールなど） ● 担当チーム名 ● サポート終了日
SBOM 管理	製品のリリースパッケージと対応して作成された SBOM を保管する。情報の紐づけのため、下記の情報を付加情報として保持する。 ● 対応する製品パッケージのリリース日 ● 製品名、バージョンおよびビルド番号 ● 対応するリリースパッケージの保存先（リポジトリマネージャ内） ● SCA ツール上でのプロジェクト名およびバージョン ● SBOM 作成ツールの名称 ● リリースパッケージ作成に使用したパッケージングツールの名称およびバージョン
SBOM 検索	各種情報から該当する SBOM ファイルを検索する。以下の情報を検索に使用できる。 ● 製品名 ● 製品種別 ● 製品バージョン・ビルド番号 ● サポート状況（サポート中か否か） ● コンポーネント名・バージョン
API 連携機能	SBOM の登録等を自動化するための API の提供

また、実際の検証でサンプルとして使用した脆弱性は下表の通りである。

表 4-99　評価に用いた脆弱性と TMSL における実際の影響度

CVE ID	コンポーネント	製品での使用	攻撃の成立
CVE-2019-15903	libexpat	あり	あり
CVE-2018-1285	Apache Log4net	あり	あり
CVE-2022-3786	OpenSSL 3.x	なし	なし

ユーザとしての活用・管理においては、新たな脆弱性が確認された場合の対応に関して、SBOM の活用による利点や課題にサプライヤーとの違いがみられるかどうか考察した。加えて、サプライヤー側の製品リリース形態（リリース頻度、パッチ, Hotfix など）による SBOM の作成管理、および使用に関する課題の違いについて整理し、考察した。

4.7.6. 実証結果

4.7.6.1. SBOM 初期導入

4.7.6.1.1. 実施内容とコスト評価

SCA ツールを使用した場合、SBOM 初期導入には合計で 120 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表の通りである。

表 4-100　SCA ツールを使用時の SBOM 初期導入コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
環境構築工数	120	人時間	0	人時間	0	人時間	
学習・体制構築工数	0	人時間	0	人時間	0	人時間	

表 4-101　SCA ツールを使用時の SBOM 初期導入コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
環境構築工数	環境整備工数	120	人時間	0	人時間	0	人時間	本実証開始前にツールが導入済みだったため、社内聞き取り調査の結果を記載した。数年前の出来事で学習および体制構築工数などに分けることができなかったため、環境整備工数にすべてまとめている。
学習・体制構築工数	学習工数	0	人時間	0	人時間	0	人時間	
	体制構築工数	0	人時間	0	人時間	0	人時間	

4.7.6.1.2. 課題、解決ノウハウ、留意点

トレンドマイクロでは、 SBOM 作成機能を持つツールがすでに導入済みであったため、ツール未導入の企業においては、ツールの選定作業とコストについても念頭に入れる必要がある。詳細な内訳は記載できないが、例として、過去実施したトレンドマイクロでの導入比較要素について下表に記載した。

表 4-102　SBOM 作成ツール検討時の確認項目例

検討項目	検討内容	結果概要および考察
開発言語のサポート範囲	現存のすべての開発プロジェクトで使用されている開発言語をサポートしているかを確認する	複数ツールを用いることも可能だが、ツールの挙動や出力結果の差異を吸収する必要が出るため、単一ツールですべてのプロジェクトをカバーできることが望ましい。トレンドマイクロでは導入した SCA ツールによって主要プロジェクトをすべてカバーできることを確認
OSS 検出精度	実際に複数製品プロジェクトにおいてツールを使用し、トップレベルコンポーネントの検出精度を確認する。また誤検出の程度についても確認する。	プロジェクトまたは開発言語などにより、同一ツール内でも検出率に違いがあった。あるプロジェクトでは検出率が低いツールが、特定プロジェクトでは非常に成績が良い場合もあった。ある会社のプロジェクトでの検出精度が他社にそのまま適用できるとは限らないため、ツール導入時には可能な限り自社プロジェクトにて検証を行い、結果を確認の上で判断することが望ましい。
脆弱性検出能力	実際に複数製品プロジェクトにおいてツールを使用し、脆弱性の検出力を確認する。	同上
各種開発ツールとの連携	自社で使用している開発ツール（ビルドツールなど）との連携が可能か確認する。	SBOM 生成の自動化など効率を考える上では、すでに自社で活用しているツールとの連携機能も重要になる可能性が高い。
システム形態	同ツールでオンプレミス版と SaaS 版が選択できる場合に、どちらを選択す	オンプレミス版の製品を導入した場合には、ライセンスの他にサーバの維持管理費用が発生する（ SaaS 版ではライセンス料に含まれる）。まずは小規模に導入し、の

	るか検討する。	ちに拡張していく場合には、柔軟にライセンス数等を変更可能な SaaS 版の方が、一般的に導入のハードルは低い。

4.7.6.2. SBOM の生成・共有

4.7.6.2.1. 実施内容とコスト評価

SCA ツールを使用した場合、SBOM の生成・共有には 1OSS あたり合計で 1.077 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表の通りである。

表 4-103　SCA ツールを使用時の SBOM 生成・共有コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
情報取得工数	0.002	人時間/OSS	0	人時間/OSS	0	人時間/OSS	
精査工数	1.075	人時間/OSS	0	人時間/OSS	0	人時間/OSS	
（作成ツール費用）	（非公開）	円	0	円	0	円	

表 4-104　SCA ツールを使用時の SBOM 生成・共有コスト内訳

小計	内訳				
項目	項目	結果			備考
		合計	ティア 1	ティア 2	

		値	単位	値	単位	値	単位	
情報取得工数	部品特定工数	0.002	人時間/OSS	0	人時間/OSS	0	人時間/OSS	SCA ツールでは OSS の特定および SBOM の生成が一度に行われるため、内訳記載は不可能である。まとめて部品特定工数に記載。 対象ソフトウェアで SBOM 作成までにかかった時間を、自動検出されたコンポーネントの数（49。過検出を含む）で割った値を記載。
	SBOM 生成工数	0	人時間/OSS	0	人時間/OSS	0	人時間/OSS	SCA ツールでは OSS の特定および SBOM の生成が一度に行われるため、個別の特定は不可能である。まとめて部品特定工数に記載。 (d1)標準フォーマット（SPDX、SPDXLite 等）で (d2)大統領令最小要素を含む
精査工数	精査工数	1.075	人時間/OSS	0	人時間/OSS	0	人時間/OSS	SBOM 全体での精査工数合計を、最終的に SBOM に含められたすべてのコンポーネント数（40）で割った値
（作成ツール費用）	（作成ツール費用）	（非公開）	円	0	円	0	円	

4.7.6.2.1.1. SBOM の生成

SBOM 初期導入におけるコストの詳細は下表の通りである。導入済みの SCA ツールにて SBOM の生成を新たに始めるにあたり、ライセンスの追加やプロジェクト側の設定変更などは不要であった。なおドル建てでの支払いものは、1 USD = 140 円として換算している。

表 4-105　SBOM 作成ツール導入コスト

大項目	小項目	作業者のレベル	作業コスト（時間）	金額（円）	補足

SBOM 作成ツール導入	年間ライセンス料	–	-	（非公開）	オンプレミス製品を使用。800 ライセンス分。 通知機能や、リスク管理機能、コードスニペット解析機能を含む
	SBOM 生成のための追加コスト	–	0	0	SBOM 作成機能使用に関する追加コストはなし
システム運用費用	AWS リソース使用量（1 年あたり）	-	-	16,800,000	
社内問い合わせ担当	専任担当者の設置（1 か月あたり）	中級者	1920	10,272,000	専任担当者を 1 名配置。開発エンジニアからの問い合わせ対応やサーバの死活監視などを実施

SCA ツールを用いた SBOM の生成は、コンソールの UI から簡単に実施が可能であり、作成後は同様に UI からダウンロードが可能だった。生成された SBOM ファイルには大統領令の最小要素がすべて含まれていた。OSS のライセンス情報に関しても SCA ツールで各コンポーネントのライセンス情報が確認でき、また同じライセンス情報が SBOM に記載されることを確認した。

次に、最初に生成した SBOM に記載されたコンポーネント情報を確認した。結果、ソースコードを同じコードレポジトリに保存している OSS に関して、サブコンポーネントが期待通り自動検出されることがわかった。一方で、 OSS 自動検出による過検知ならびに検出漏れも存在していた。ほとんどのものは SCA ツールのデータベース上に情報が登録されており、コンソールから手動で追加が可能だったが、一つのコンポーネントはデータベースに情報が存在していなかった。このコンポーネントに関しては、最終的にカスタムコンポーネントとして自社の SCA ツールシステムに情報を登録することで、コンソールから手動で追加することが可能となった。

疑似的に「外部サプライヤーから調達するコンポーネント」とした挙動監視エンジンについては、現状では SBOM を作成していないことがわかった。トレンドマイクロのポリシーとしては、2023 年以降共通モジュールに対しても SBOM を作成することを義務付ける方針であることは確認したため、SBOM 作成開始後には 製品の SBOM に含めていく方向となるであろう。

なお、SCA ツールによる自動検出をソースコードレポジトリに対して実施しているため、製品モジュールのビルド時にスタティックリンクされる Microsoft Visual Studio のランタイムライブラリは予想通り検出はされなかった。ただし、コンポーネント情報をコンソールから手動追加可能であることが確認できたため、情報を追加した。また、コンポーネント名が正しく検出されているものの中にも、バージョン情報が誤っている OSS が複数確認された。ただし、こちらはすべて手動でバージョン情報の修正が可能だった。

これらのコンポーネントの手動追加およびバージョン情報修正を実施後に、再度 SBOM を生成したところ、修正内容が正しく内容に反映されることを確認した。

SCA ツールにおけるコンポーネントの検出結果の詳細を下表にまとめて記載する。

表 4-106　SCA ツールでの OSS の検出結果と SBOM への出力

自動検出の状態	検出結果	コンポーネント種別
SCA ツールで自動検出されたコンポーネント	誤検出だったコンポーネント： **15**　*(z)*	
	自動検出されたコンポーネント： **34**	トップレベルコンポーネント： **22**　*(a)*
		サブコンポーネント： **12**　*(b)*
SCA ツールで自動検出されなかったコンポーネント	SCA ツールのデータベースから情報を手動追加可能だったコンポーネント： **5**	トップレベルコンポーネント： **2**　*(c)*
		サブコンポーネント： **3**　*(d)*
	カスタムコンポーネントとして情報を手動追加したコンポーネント： **1**	トップレベルコンポーネント： **1**　*(e)*
		サブコンポーネント： **0**　*(f)*
その他	バージョン情報に誤りがあったコンポーネント： **5**　*(y)*	

測定数値をもとに計算した SCA ツールの正確性データを下表に示す。

表 4-107　SCA ツールの正確性

項目	式	値
作成した SBOM に含まれるコンポーネント数　（TTL）	= a + b + c + d + e	**40**
うちトップレベルコンポーネント（TTLt)	= a + c + e	**25**
うちサブレベルコンポーネント（TTLs）	= b + d	**15**
自動検出における OSS の過検知率　（%）	= z / (a + b + z)	**30.61%**
バージョン情報の正答率（%）	= (TTL - y) / TTL	**87.18%**
トップレベルコンポーネントにおける SCA ツールの捕捉率（%）	= a / TTLt	**88.00%**

自動検出における過検知は 30 % を超えるものの、トップレベルコンポーネントの捕捉率は 88 % に達していた。

一方、上記の SBOM 生成およびその正確性の確認に要した時間とコストの詳細は下表の通りである。

表 4-108　SBOM 作成および正確性の確認に要した時間

大項目	小項目	作業者レベル	作業コスト（時間）	金額（円）	補足
SBOM 作成	SCA ツールコンソールより実施	初級者	0.1	482	コンソールより指定の作業を実施し、コンソールからファイルをダウンロード実施するまでの時間
SBOM の正確性	手動管理のリスト	初級者	2	9,640	別に存在する社内の DB 情報から手動で抽出
	SCA ツールで管理できるコンポーネントの確認	初級者	2	9,640	SCA ツールによるスキャン＋手動追加
	誤検出されたコンポーネントの確認	中級者	15	80,250	手動による精査
	手動追加されたリスト（実測）	中級者	4	21,400	手動管理リストとの差分を追加
	手動追加不可のコンポーネントへの対応方法の調査	中級者	15	80,250	SCA ツールの DB への追加方法の調査と対応可否の確認、カスタムコンポーネント機能の調査と対応まで
	バージョン情報の確認と修正	中級者	5	26,750	手動によるバージョンの更新

今回の実証対象ではソフトウェア・コンポジション解析に活用できるパッケージマネージャーの構成情報などが利用できないため、誤検出されたコンポーネントがプロジェクトに含まれていない事を確認するためには手動でのソースコード精査を行う必要があった。誤検出に見えるコンポーネントが、あるコンポーネントのサブレベルコンポーネントとして含まれている可能性や、プロジェクトソースコード内にスニペットとして含まれている可能性が考えられ、これらを否定するために広範囲なソースコード精査を実施したことから、作業に時間がかかっている。

なお、サブレベルコンポーネントに関する情報の正確性担保のために、OSS の深さに関する精査を 3 つの OSS に対して試しに実施したところ、一つの OSS に対して平均 4 時間が必要であった。すべてのコンポーネントで深さに関する精査を実施する場合 TMSL では 170 時間以上が必要

であると想定され、SBOM 生成のために実施する作業としては大きな負荷がかかるため、今回の計測からは除外した。

4.7.6.2.1.2. SBOM のユーザへの共有

SBOM に関心が高いと想定される国内のインフラ・製造業関連の一部顧客、およびトレンドマイクロ製品の国内販売パートナー企業に対して SBOM 活用に関するヒアリングを実施し、計 8 社から回答を得た。内訳は下図の通りである。すでに SBOM の活用を実施している企業は存在しておらず、三分の一程度が検討中であった。何らかの検討・対応を開始している企業は、自動車関連事業を営む企業が多かった。

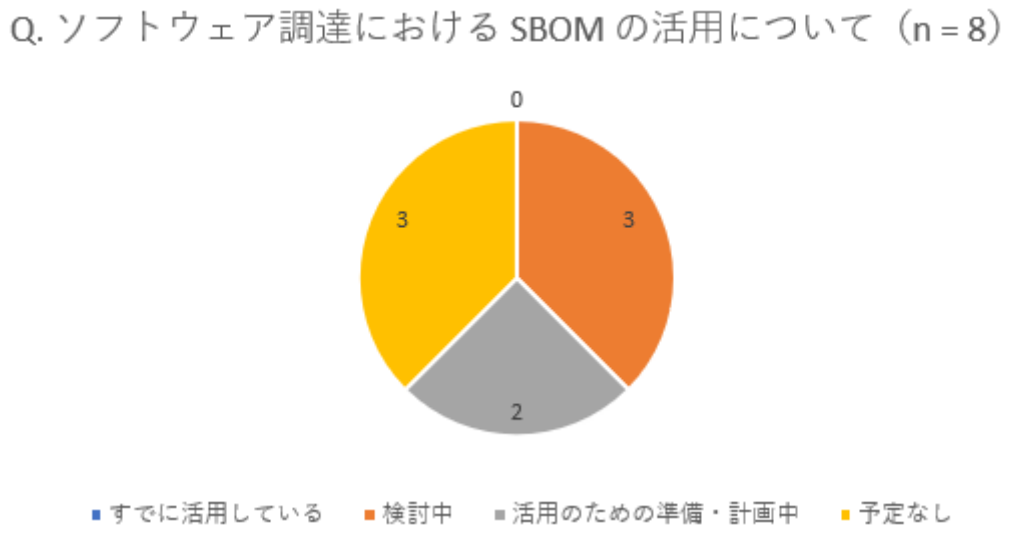


図 4-21　SBOM 活用に関するヒアリング結果

SBOM ファイルの提供形式について追加質問したところ、自由記述にて 5 社から回答を得た。下図が示す通り、SPDX または SPDXlite を期待する回答が大部分だった。

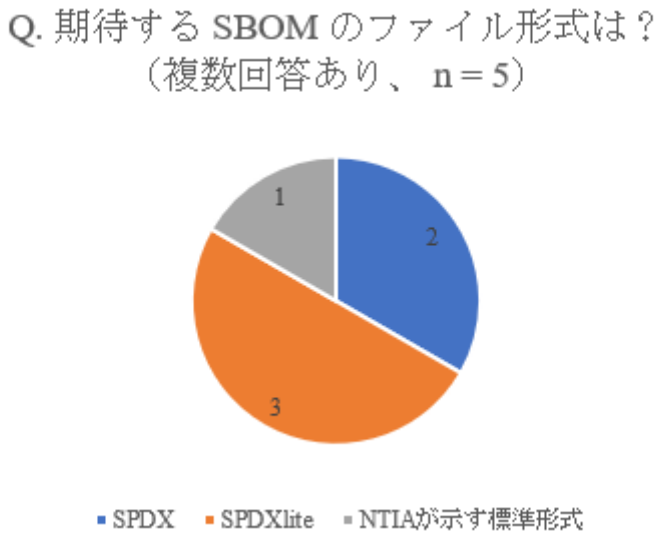


図 4-22　SBOM ファイルの提供形式

その他、SBOM 活用においてのユーザ側での期待や、課題を同様に自由記述にて質問し、得られた回答を下表にまとめた。各社状況により、回答は様々であったが、SBOM の具体的な管理方法や、情報の取り扱いなどに課題や不安を感じていることがわかった。

表 4-109　ユーザ側での SBOM 活用に関する期待と課題

カテゴリ	項目	内容
期待	SBOM 活用への期待	● 構成情報の共通化および共通管理が望めるところ（パートナー）
	ベンダーへのリクエスト	● 利用ライセンスの記載とパッケージ版数、できれば版数に紐づいた脆弱性の一覧がわかるとよい。依存関係の情報も重要（製造業）
課題	運用	● コストが膨大（パートナー） ● 脆弱性管理が現実的には難しい（パートナー） ● 簡単に使用できる SBOM 管理ツールが見つけられていない。ライセンス管理、脆弱性管理も同時に行いたい（製造業）
	標準化	● 深さ、粒度、範囲などに関して標準化が必要（パートナー） ● 名寄せができないため異なるソースやパッケージマネージャーから取得したコンポーネント名が紐づけられない（パートナー） ● 詳細な仕様が固まっていないので何をすればいいか不明確（製造業）
	SBOM による情報提供	● SBOM で情報を提出してもらえるのか、確実な情報を出してもらえるのか不安（製造業） ● 3rd Party 製のソフトウェアが OSS を利用していた場合に、利用している OSS が見えない（製造業）
その他	ユーザの現状	● 自社で管理ツールを開発中（パートナー） ● ベンダーからの SBOM は共通フォーマットの SPDXlite に変換したのちに結合して管理する予定（製造業） ● SBOM については、どこかで検討をしなければならないと思っている。きっかけとしては log4j などの脆弱性。ただし、どの程度運用が大変なのかが分からず、まだまだ検討の序の口の部分。とにかく情報が欲しいという状況（製造業）

上記ヒアリング結果も踏まえ決定した、現時点での SBOM 提供に関する方針を下表に記載する。

表 4-110　SBOM 提供に関する現時点での方針

検討項目	小項目	詳細
提供形態	項目なし	SPDX 形式のファイルを直接送付
提供条件	リリース形態と SBOM 提供	当社サイトで今後公開する、オンプレミス製品のメジャー・マイナーリリース、Critical Patch, Patch パッケージに対応する SBOM を作成・保管する。ただし、SBOM の活用詳細などがまだ確定していない状況を鑑み、要件が不確定な SBOM ファイルが広く拡散するのを防ぐ目的から、 SBOM ファイル自体は一般公開せず、問い合わせに応じて提供する。
	SBOM 提供有償化の有無	原則、製品 readme や技術情報の提供と同様、当社製品顧客や見込み客に無償で提供する。場合により提供時に金銭以外の条件を設定する可能性がある。
契約	条項の修正	現在の企業向けオンプレミス製品の使用許諾契約には、SBOM に誤りがあった場合に関係する記述が存在しない。現状、SBOM 提供の問い合わせがないため、SBOM 記載の内容に誤りがある場合の免責や対応については、提供時に付加情報として説明する予定。今後問い合わせ数の増加が見込まれる場合には、個別の説明が不要となるよう、条項の修正を検討する。
その他	SBOM の記載に誤りがあった場合の修正	SCA ツール上で修正が必要・可能な項目については、修正後 SBOM を再作成する。過去リリースしたパッケージに紐づく SBOM については、ツール側での設定を修正して再作成することが難しいと想定されるため、 SBOM ファイル自体を直接編集する必要がある。 SBOM を外部公開している場合には、修正したファイルで差し替えを実施する。問い合わせベースでの共有の場合には、内容の重要度によってはファイルを配布したユーザに、修正済みのファイルを提供する必要があるだろう。

4.7.6.2.1. 課題、解決ノウハウ、留意点

4.7.6.2.1.1. ライセンスコストとその低減策

ライセンス料が高額であったが、これは製品やコンポーネントの開発プロジェクトの数ならびに開発エンジニアが多いトレンドマイクロの事情を過分に反映したものである。トレンドマイクロでも開発エンジニア全員にはライセンスは付与しておらず、各プロジェクトの代表者のみが使用する形にしているが、プロジェクトの管理方法や SCA ツールにアクセスする必要がある社員を減らすなど運用を工夫すれば、ライセンス料を抑えることは可能だと考えられる。

なお、今回の検証では使用しなかったためコストには含めていないが、 SCA ツール のバイナリ解析を用いる場合には別途追加費用が必要である。バイナリ解析のオプションを SCA ツールに登録されているライセンスの一部のみに追加することが仕組み上できないため、トレンドマイクロではバイナリ解析の使用に関しては、わずかな数のライセンスだけを登録した別の環境を用意している。これは、一部のプロジェクトのみで利用する予定の追加ライセンス料を圧縮するための措置であり、サーバの管理コストは増えるものの、全ライセンスにオプションを付加した場合の増加コストよりは低くなるだろう。

4.7.6.2.1.2. ツールのデータベースに存在しない OSS への対応

実証の結果、今回使用した SCA ツールでは、ツールのデータベースに含まれていないコンポーネントは、自動検出だけでなくコンソールからの手動追加もできないことが分かった。データベースに SBOM に含めたいコンポーネント情報が含まれていない場合には SCA ツールベンダーのサポートに依頼することで、コンポーネント情報の追加が可能である。ただし、この SCA ツールのデータベースに追加できる OSS には条件があった。今回の対象製品には、この条件に当てはまる OSS が存在していたため、SCA ツールのサポート部門に問い合わせを行ったものの、やはりデータベースへの追加は対応不可能という回答であった。

データベースに登録可能な条件を満たさない OSS に対しては、代わりに自分たちでコンポーネント情報を追加する機能を活用可能だった。なお、カスタムコンポーネントの登録作業は、特別な権限を持った管理者が実施する必要があるため、通常のプロジェクトメンバーでは対応ができなかった。ツールのデータベースにない OSS を製品で使用する場合の対応プロセスについて社内であらかじめ定義しておくと混乱を避けられるだろう。

4.7.6.2.1.3. OSS の深さレベルの精査と SBOM の正確性

SBOM の理想は、「製品内で使用している OSS 内部で使われている OSS も含めて、すべてのコンポーネントの情報が網羅されていること」である。一方で、特に C/C++ においては、他言語ではソフトウェア構成分析に活用できるパッケージマネージャーが存在しな

いことなどから、一般的に正確な SBOM の自動作成はより困難であるとされている[γ]。実際今回、3 つのトップレベルコンポーネントを選択して検証した、ツールにおけるサブレベルコンポーネントの検出率は下表の通りであった。

表 4-111　サブコンポーネントの検出率

OSS 名	手動で確認したサブレベルコンポーネント数	自動検出可能だったサブレベルコンポーネント数	検出率
protobuf-csharp-port	3	3	**100.00%**
angularjs	9	6	**66.67%**
php	2	2	**100.00%**
合計	14	11	**78.57%**

自動検出できなかったサブレベルコンポーネントは、すべてコンソールより手動追加が可能であったものの、少なくても現状の手法ではすべてを自動で検出することは難しいと考えられる。

また、自動検出の有無は、コンポーネントの深さレベルにより決定しているわけではないことも、今回の検証から確認できた。例として、SBOM に含まれる angularjs とそのサブコンポーネントの構成を下図に示す。二階層目以降の深さのコンポーネントについても自動検出可能であることが確認できた一方で、一階層目のコンポーネントが自動検出されずに、二階層目が自動検出される場合もあった（下図参照）。

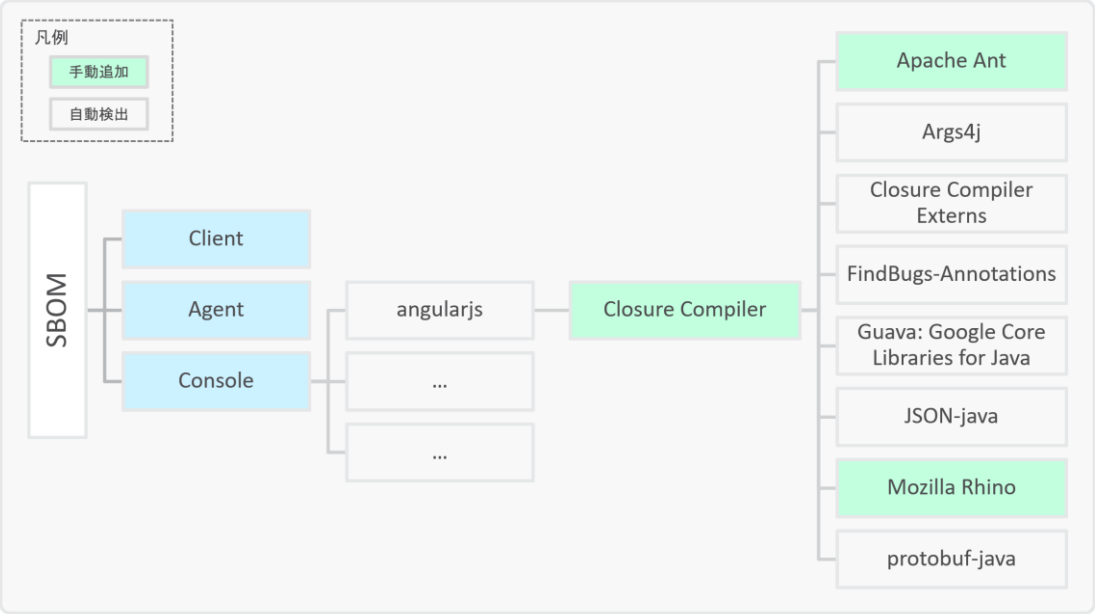


図 4-23　angularjs のコンポーネント構成と検出結果

したがって、「コンポーネントの深さレベルの正確性が特定の階層まで保たれている」と

[γ] Synopsys. “Using the Black Duck C/CPP Tool”. 2021. https://community.synopsys.com/s/article/Using-the-Black-Duck-C-CPP-Tool, （参照 2022-12-27)

明言することも、実際に手動精査を実施しない限りは難しいであろう。

サブレベルコンポーネントについての手動精査を行う場合には、いくつか方法が考えられる。今回の実証では、以下の列挙する手法によって確認を実施した。

A) 対象コンポーネントのソースコードやドキュメントを手動で精査することにより、使用しているサブコンポーネントを列挙
B) サブレベルコンポーネントが見つかった場合、再帰的にそのサブレベルコンポーネントの精査を行う
C) パッケージソースコードが git で管理されている場合、submodule として組み込まれているコンポーネントをサブレベルコンポーネントとして抽出
D) パッケージマネージャーを利用している場合、その管理情報を参照する事でサブコンポーネントを抽出

しかしながら、今回選択した 3 つの OSS では C、D の条件に当てはまるサブレベルコンポーネントは存在しなかった。また、コンポーネントごとにサブレベルコンポーネントの管理形態は異なっており、系統だったやり方が確立しづらい面もあった。平均して精査完了には 4 時間がかかっており、全コンポーネントにて手動精査を実施する場合には、非常に多くの時間とコストがかかる可能性もある。初回の SBOM 作成時にすべてを精査して SCA ツールに情報を登録すれば、二度目以降はすべてを見なおす必要はないかもしれないが、少なくてもトップレベルコンポーネントが更新された場合には、内部のサブレベルコンポーネントに変更がある可能性があり、再度精査が必要になるであろう。

現状では、正確性の度合いと、開発コストおよびリリース時間の増加はトレードオフである。例えば、脆弱性を含む重大な問題の修正リリースを計画する場合に、そのパッケージに対応する「正確な SBOM」 が準備できるまでリリースを遅らせるべきか否か、という議論もあると想定される。SBOM の活用目的の一つである「ユーザによる脆弱性への速やかな対処」を目標とした場合に、何を優先させるべきなのかにはさらなる議論が必要だと考えられる。

4.7.6.2.1.4. ソフトウェア構成と SBOM 上のコンポーネントの関係

TMSL のソースコードはサーバ、エージェントなどのシステム構成要素に分かれたコードレポジトリで管理されており、当初はそのすべてのコードレポジトリを一つの SCA ツール上のプロジェクトに紐づけて管理をしていた。その状態で SBOM を出力した場合には、コードレポジトリごとにコンポーネントの依存関係が記載されると想定していたが、予想とは異なり、「検出されたコンポーネントがどのコードレポジトリから検出されたかたどれる情報は SBOM に含まれない」という結果となった。(SBOM ファイル上では、「name プロパティが SCA ツールのプロジェクト名、バージョンが SCA ツールのバージョン名で記述される package」の SPDXID と各コンポーネントの依存関係のみが表現されていた)。

製品開発チームにとっては、脆弱性が含まれる場所にかかわらずその影響度に応じた対応をすべきである一方で、ユーザにとっては、「脆弱性がシステムのどの部分に存在するか」という情報は、自社ビジネスへの影響度の見積もりや、公式の修正が提供されるまでの間の次善策を検討するにあたって重要であると想定される。例えば、サーバ・クライアントアー

キテクチャのシステムにおいて、脆弱性がサーバ側にあるのか、クライアント側にあるのかによって、ビジネスへの影響診断の結果や取り得るリスク回避の手段は異なるであろう。

このような情報を、現在の SBOM に追加することが可能か調査したところ、この SCA ツール上でのソースレポジトリとプロジェクトの設定方法により、SBOM 中のコンポーネントの構成情報が変わることが分かった。

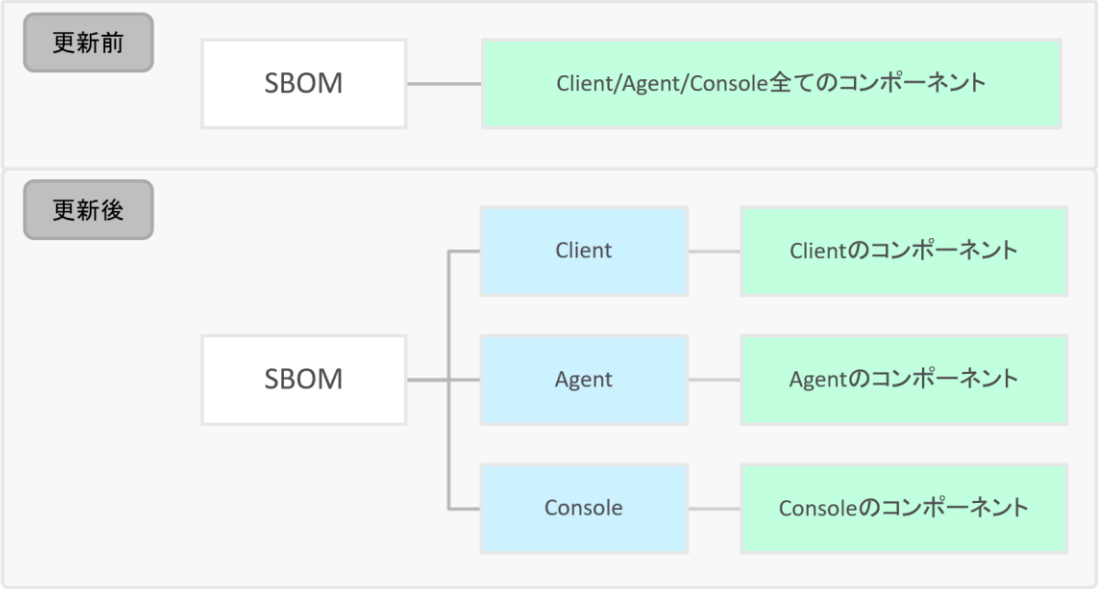


図 4-24　SCA ツール上のプロジェクト階層構造

上図の「更新後」の設定のように、コードレポジトリとシステム構成ごとに 「Client」「Agent」「Console」という SCA ツール上 のプロジェクトを分けて作成したうえで、親プロジェクトである「SBOM」にこの 3 つのプロジェクトを追加したところ、Client, Agent, Console それぞれで利用されている OSS がどれに当たるのか、SBOM の relationships プロパティの情報からたどることが可能となった（下図参照）。

```
"name": "tmsl-sustain",
"SPDXID": "SPDXRef-package-aa233243-b706-4197-9c48-3bf3718ffa0c",
"versionInfo": "sbom_agent",

"name": "tmsl-sustain",
"SPDXID": "SPDXRef-package-c2135f0e-664a-4f46-9247-ed2deae0c764",
"versionInfo": "sbom_client",

"name": "Protobuf",
"SPDXID": "SPDXRef-package-aee3e972-2b15-4652-afdd-9520420ec888",
"versionInfo": "3.21.1",

"files": [],
"relationships": [
    {
        "relationshipType": "DEPENDS_ON",
        "spdxElementId": "SPDXRef-package-b285311d-f9fa-45f2-a63b-b5d289482217",
        "relatedSpdxElement": "SPDXRef-package-b7da174d-1296-4bf1-a4c4-a15d7c1c30cc"
    },
    {
        "relationshipType": "DEPENDS_ON",
        "spdxElementId": "SPDXRef-package-aa233243-b706-4197-9c48-3bf3718ffa0c",
        "relatedSpdxElement": "SPDXRef-package-aee3e972-2b15-4652-afdd-9520420ec888"
    },
    {
        "relationshipType": "DEPENDS_ON",
        "spdxElementId": "SPDXRef-package-c2135f0e-664a-4f46-9247-ed2deae0c764",
        "relatedSpdxElement": "SPDXRef-package-aee3e972-2b15-4652-afdd-9520420ec888"
    },
```


図 4-25　プロジェクト構成変更後の SBOM（抜粋）

上図では例として protobuf コンポーネントの情報を図示した。protobuf は TMSL サーバとクライアントの通信機能（sbom_agent プロジェクトで管理）および TMSL クライアント（sbom_client プロジェクトで管理）の双方で使用されている。プロジェクト構成変更前は、relationships プロパティ内で protobuf の SPDXID は一つしか確認できなかったが、プロジェクト構成変更後は、上図のように一つのコンポーネントに sbom_client, sbom_agent の両方が依存している事実を示すことができた。
システムの単位でコードレポジトリのフォルダが分かれていれば、この手法で管理することが可能である。SCA ツール内のプロジェクト構成を決定する際には、自社内だけでなく、ユーザにとって必要な情報がより表現できるような検討を行うとよいだろう。また、サーバとクライアントを全く別のプロジェクトとして管理し、SBOM も独立して作成するのも一つの手段であると考えられる。自社のユーザにとってどの単位で SBOM を管理するのが望ましいかも検討した上で決定するのが好ましい。

4.7.6.2.1.5. 外部サプライヤーから調達するコンポーネントの管理

別開発チームが作成したコンポーネントや、外部サプライヤーから調達するコンポーネントについても SBOM に含める場合には、前項と同様に、該当の SCA ツールプロジェクトを追加することによって達成できると考えられる。

ただし、この場合には、この SCA ツールの場合には、同じ SCA ツールのサーバで該当プロジェクトを管理し、別開発チームまたは外部委託部分の SCA ツール上のプロジェクトに対するアクセス権限を開発チームの SCA ツールプロジェクト管理者が持っている必要があ

る。オンプレミス版使用の場合には社外開発チームが使用するシステムとの連携が難しい可能性があるが、 SaaS 版を用いればハードルは下がると考えられる。

もう一つの手法として、 OSS の管理と同様、ソースコードを外部サプライヤーから受け取り、自社のソースコードレポジトリで管理する方法が考えられる。ただし、この場合はソースコードが納品物として提供されている必要がある。また自動検出の結果を確認し、最終的にツール側で修正するのも最終サプライヤー側の作業となると想定される。

4.7.6.2.1.6. 顧客目線での SBOM 内の情報整理

作成された SBOM には、コンポーネント情報だけでなく SCA ツール上で開発チームが設定した情報が含まれる可能性がある。この情報が顧客にとって活用しやすい情報を十分に含んでいるかどうかは、よく検討する必要がある。

例えば、プロジェクト名やバージョン情報には注意が必要だろう。SCA ツール上プロジェクト名やバージョン番号を設定できることがあるが、設定時期的に、通常は製品開発中に使われている情報が登録されるだろう。だが、開発中とリリース後で製品名やバージョン名が変更されることは、製品開発の現場ではよく見る光景である。開発チームにとって SCA ツールは社内用ツールである認識が強いため、製品 UI やドキュメント上の製品名やバージョンには気を払っても、内部でのみ使用するプロジェクト管理ツールにまでその情報を同期しているチームはそこまで多くはないと想定される。 SBOM を受け取った顧客が、使用している製品・バージョンのファイルであると簡単に認識できるようにするために、外部向けの情報を設定するようにすべきだろう。

```
{
    "name": "tmsl-sustain",
    "creationInfo": {
        "created": "2022-12-08T06:23:20.160Z",
        "licenseListVersion": "3.13",
        "comment": "",
        "creators": [
        ]
    },
    "documentDescribes": [
        "SPDXRef-package-b959fc4b-38f5-4fd4-9902-a9d021303b7d"
    ],
    "packages": [
        {
            "name": "tmsl-sustain",
            "SPDXID": "SPDXRef-package-b959fc4b-38f5-4fd4-9902-a9d0211
            "versionInfo": "sbom",
            "licenseDeclared": "NOASSERTION",
            "licenseConcluded": "NOASSERTION",
            "homepage": "NOASSERTION",
            "supplier": "NOASSERTION",
            "copyrightText": "NOASSERTION",
            "filesAnalyzed": false,
            "downloadLocation": "NOASSERTION",
            "externalRefs": [
```


SCAツールで設定したプロジェクト名

SCAツールで設定したバージョン名

図 4-26　SBOM に出力される SCA ツール上の設定情報例

一方、特にオンプレミス製品においては製品名やバージョンだけでなく、ビルド番号も重要である。一つのバージョン内でも、OS のアップデートや機能追加、バグ修正のために、新たなパッケージが、別のビルド番号にてリリースされるからである。ただし、今回使用した SCA ツールにおいて、ビルドサーバで製品のビルドを作成する際に決定されるビルド番号を versionInfo プロパティに埋め込む方法は見つからなかった。ユーザへの提供を考える際には、ファイル名に対応するビルド番号を含めるなど、ファイル単体を確認した際に、どのパッケージに対する SBOM なのか、容易に判断ができる方法を準備しておくと、ユーザ側でののちの混乱を防ぐことができるだろう。

4.7.6.2.1.7. 特定のリリース形態における SBOM 提供の課題

製品の保守フェーズにおいては、問題の修正提供の目的で、製品を構成するモジュールの一部のみを切り出して Hotfix やパッチなどとして顧客に提供する場合がある。修正による影響を狭め、検証時間とリスクを低減するための手法であり、ユーザにも開発チーム側にも好まれる傾向がある。このようなリリース形態の場合、開発言語によっては、提供予定の一部のモジュールのみで使用されている OSS を自動で検出し、SBOM を作成することは簡単ではないと想定されるため、リリース形態によらず SBOM を作成可能かどうかは、ソフトウェアベンダー側で確認の上で、ユーザとのコミュニケーションを行う必要がある。

また、仮にベンダー側で一部モジュールのみに対応する SBOM を生成可能であったとし

ても、その SBOM を渡されたユーザ側では別の課題が生じる。ユーザ側では、「現在のシステム構成で使用されている OSS」の情報をもとに脆弱性やライセンスの確認をしたいと考えるであろう。ただ現状では、システムにパッチを適用するようにユーザ側で SBOM の内容の一部のみを更新するよい方法がない。特に既存ソフトウェアの場合は、ユーザ側でどのパッチを適用するか最終的に判断することになるため、ベンダー側の持つ最新の SBOM に対応したシステム構成とユーザ側のシステム構成が同一であるという保証もなく、ユーザ環境ごとに SBOM を準備するのも困難である。ユーザ側ではインストール時に使用したパッケージの SBOM と適用済みのパッチに対応した SBOM をそれぞれ取得・管理して、「いずれかがシステムで使用されている可能性のあるコンポーネント」として影響範囲の絞り込みに利用するのが現実的かもしれない。

また、実際の製品開発や保守を考えると、サプライヤー側では外部への新しい製品パッケージのリリースのたびに対応する SBOM を生成する作業が発生することになる。特にアジャイル開発で頻繁にリリースを実施する場合には、リリースのたびに人手を介在させて SBOM の作成や正確性の担保、共有をする時間を持つことが困難である。 CI/CD への統合による作成の自動化の他、SBOM ファイルの共有など SBOM のライフサイクル全般において可能な限り自動化ができる仕組みを整えることが望ましい。

4.7.6.3. SBOM の活用・管理

4.7.6.3.1. 実施内容とコスト評価

SCA ツールを使用した場合、SBOM の活用・管理には 1OSS あたり合計 0.502 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表の通りである。

表 4-112　SCA ツールを使用時の SBOM 活用・管理コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
脆弱性管理工数	0.5	人時間/OSS	0	人時間/OSS	0	人時間/OSS	
ライセンス管理工数	0.002	人時間/OSS	0	人時間/OSS	0	人時間/OSS	
（活用ツール費用）	0	円	0	円	0	円	

表 4-113　SCA ツールを使用時の SBOM 活用・管理コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
脆弱性管理工数	脆弱性特定工数	0	人時間/OSS	0	人時間/OSS	0	人時間/OSS	SCA ツール側から通知可能なため、脆弱性の判定までの作業時間は 0 である
	脆弱性対応工数	0.5	人時間/OSS	0	人時間/OSS	0	人時間/OSS	実際の脆弱性修正までの評価は実施していない
ライセンス管理工数	ライセンス特定工数	0.002	人時間/OSS	0	人時間/OSS	0	人時間/OSS	ライセンス情報の SCA ツール上での確認のみ実施
	ライセンス違反対応工数	0	人時間/OSS	0	人時間/OSS	0	人時間/OSS	今回は検証せず
（活用ツール費用）	（活用ツール費用）	0	円	0	円	0	円	SBOM 作成ツールと共通

4.7.6.3.1.1. SBOM 活用とコスト

サプライヤー側での各調査手法において、脆弱性の影響度確定までに必要な作業は下図の通りであった（SCA ツールの通知から始まる場合には、実際の影響度調査のみが関係することとなるため割愛)。「実際の影響度調査」に当たる作業は、手法によらずすべて同一であり、開発エンジニアの手動作業が必要な領域であった。

図 4-27　脆弱性調査手法と作業内容

単一製品内での影響度を調査するのにかかった詳細な時間及び工数を以下に示す。記載に当たっては、製品チーム主体の調査に加え、PSIRT 視点での対応の情報も含めた。

表 4-114　単一製品での影響度調査における作業時間
（単位：時間/人、平均は値が計測されたもののみで算出）

主体	手法	CVE ID	コンポーネント特定	自社製品使用調査	影響度調査
製品チーム	SCA ツール通知	CVE-2019-15903	0.00		0.5
		CVE-2018-1285	0.00		0.5
		CVE-2022-3786	0.00		-（使用なし）
		平均	0.00		0.5
PSIRT	SBOM	CVE-2019-15903	0.08	0.08	0.5
		CVE-2018-1285	0.08	0.08	0.5
		CVE-2022-3786	0.08	0.08	-（使用なし）
		平均	0.08	0.08	0.5
	SCA ツールコンソール	CVE-2019-15903	0.08		0.5
		CVE-2018-1285	0.08		0.5
		CVE-2022-3786	0.08		-（使用なし）
		平均	0.08		0.5
	手動	CVE-2019-15903	0.08	0.42	0.5
		CVE-2018-1285	0.08	0.42	0.5
		CVE-2022-3786	0.08	0.08	-（使用なし）
		平均	0.08	0.31	0.5

SCA ツールの通知機能は、メールまたは Slack, Microsoft Teams といったコラボレーションツールとの連携によって、開発チームの担当者に脆弱性に関する情報を自動で知らせることができる。したがって、担当者は通知を受け取った時点で、担当製品で使用されているコンポーネントのいずれかに脆弱性が報告されたことを把握できる状態である。そのため、追加調査なく、実際の影響度調査に移ることが可能であった。

PSIRT チーム起点での調査においては、SBOM を活用した場合および SCA ツールのコンソールから実施した場合を比較したところ、 SCA ツールを用いた手法の方が、作業工数が少なかった。SCA ツールではコンポーネント情報だけでなく、CVE ID での検索機能が存在しているため、CVE ID から影響のある OSS について調査する時間の分だけ工数が短縮されたためである。また、自社作成の SBOM 管理システムと CVE の詳細情報において名寄せされていないコンポーネントがあったところも、確認に時間を要する原因となった。ただし、CVE ID の公開後、該当 ID ベースの情報が SCA ツールコンソール上で使用可能になるまでのリードタイムは不確定であり、ID ベースの情報がない場合には SBOM と同様の方法でコンポーネントの特定を実施する必要があるため、この工数差は縮まる可能性が高い。

一方、両手法とも、手動での調査結果と比較した場合には、大幅に時間が短縮されていた（平均作業時間において、手動調査の 25.8%～51.6%）。手動調査においては社内で以前使用していた OSS 管理システムと CVE の詳細情報において名寄せされていないコンポーネントがあったことが、時間がかかった主な原因である。SCA ツールを利用した調査はユーザ

企業側では不可能であることを考慮すると、ユーザ企業においては SBOM を活用した調査がもっとも現実的かつ効率性が高いといえるだろう。ただし、影響度の判定に関しては、正確に実施しようとした場合に、開発チームによる詳細調査（PoC コードを用いた再現確認や、ソースコードレベルでのチェックなど）が必須となる。手法によらず同じ作業が必要なこの詳細調査の工数の割合は、どの手法でも全体の 5 割を超えるため、全体でみると、手法の違いによる効率性の変化の影響は小さくなった。

複数製品に対して、コンポーネントの製品使用について調査したところ、SCA ツールのコンソールの利用では、製品数の増加によらず、同程度の時間で結果が出力された一方、自社開発の SBOM 管理システムでは製品数により比例的に検索時間が増加する傾向があった。また手動調査の場合も、製品数に比例して合計調査時間が増加した。

なお、ライセンス管理に関しては、OSS のライセンス情報の管理と一覧の取得に 同じ SCA ツールを利用し、この情報をもとに法務部にレビューを依頼するプロセスを採用している。そのため、今回の工数の計算からは除外した。

4.7.6.3.1.2. SBOM 管理のコスト

SBOM を管理するためのシステムは現在開発・構築中であり、開発が終了していない。最終的なコストが確定していないため、今回情報は割愛した。

4.7.6.3.1. 課題、解決ノウハウ、留意点

4.7.6.3.1.1. ソフトウェアベンダーにおける OSS 管理ツールと SBOM での調査の比較

製品開発において特定コンポーネントの使用を判断する場合、今回使用したような SCA ツールのコンソール、または SBOM を活用することで手動よりも大きく工数を削減できることが確かめられた。SCA ツールの方が社内 SBOM 管理システムに比べ、UI が洗練されており、またツールベンダー独自の脆弱性に関する分析情報が含まれることもある。また、SBOM に含まれる情報は SCA ツール上で表現されている情報の一部であることから、現在のソースコードレポジトリに関する情報については、SCA ツールの方がより情報量が多いであろう。

一方で、今回の検証において、SBOM を用いる方が効率がよいと確認できたケースは二点あった。一点目は「その時々の状態のスナップショットを記録できる」という点である。今回使用した SCA ツールでは、最新のソースレポジトリの情報をもとに調査を行うことになる。SaaS 型製品では、基本的に常に最新ビルドが使用されるが、オンプレミス型製品で同一バージョンの製品であっても顧客が使用するビルドに幅や揺らぎがある場合がほとんどである。脆弱性の影響を受けるコンポーネントが存在している製品名・バージョン名だけでなく、対象のビルド番号やリリース時期を絞りこむことができれば、顧客とのコミュニケーションや、ユーザ側での実作業の必要性の確認などをより効率的に実施可能であり、SBOM 活用に利があると考えられる。

二点目は「実際に顧客にリリースされているパッケージの判別が容易である」ことである。開発中のものなど、現状外部リリースされていない開発プロジェクトも含めて SCA ツールで管理していること、また SCA ツール上のプロジェクト内の製品名が内部名になっている

場合が多いことなどから、トレンドマイクロの PSIRT チームにとっては、実際のユーザ環境で影響がある製品がどれにあたるのかを把握することが難しいという課題があった。製品開発チームとしては、どの開発フェーズにあっても脆弱性の問題は同様に注意を払うべきものではあるが、会社全体では実際の顧客に影響があるものについて優先度を高くして対応すべきである。外部リリース実績があるもののみ SBOM ファイルを管理し、その情報をもとに影響度を調査することで、顧客への影響にフォーカスした調査がしやすくなるだろう。
一方、VEX に含まれるような実際の影響度の判断においては、SBOM・有償ツールによる自動判断は難しく、現状は開発エンジニアの手動確認に頼らざるを得なかった。この部分にかかる工数が多いほど、ツールや SBOM 導入による全体的な工数圧縮効果は低減すると考えられる。ただし、この SCA ツールのコンソール上では、 CVSS 3.x の現状評価の値や、攻撃コードが発見された日付など、NVD には含まれない情報も確認することができた。また、コンソール上で確認できる脆弱性の重要度は、これらの付加情報も考慮に入れて設定されていることがわかった。日々多くの脆弱性が発見される中では、どのように優先度をつけて対応していくかは重要な課題である。これらの付加情報は、「実際の攻撃に使用される可能性が高い脆弱性」を判別し、作業の優先度を決定するために有益であると考えられる。

4.7.6.3.1.2. SBOM 管理の範囲と社内 SBOM 管理システムについて

これまでのところ SBOM 管理に適したツールを発見できなかったことから、トレンドマイクロでは現在社内で SBOM 管理システムを開発中であり、一部運用を開始している。

同システムでは、SBOM の検索など活用の目的以外に、現状 SCA ツールではカバーできない部分の管理目的でも活用している。例えば、今回使用した SCA ツールでは生成した SBOM ファイルはレポートの一種として生成日時の記録や、ファイルの保存は可能であったが、内容の検索や詳細な管理はできなかった。また、InstallShield など、サポートされていないパッケージングツールの情報やビルド番号の情報、対応するリリースパッケージとの紐づけについても、この SCA ツールで管理する方法が見つからなかった。これらも脆弱性による顧客への実影響を効率よく調査するのに重要な情報である。そのため、現在では社内 SBOM 管理ツールで情報を集約管理している次第だが、本来は SBOM 作成ツールとアーティファクトレポジトリやビルドツールが連携し、ユーザへの SBOM 提供を見越した統合管理できる体制が本来は望ましい（一部、DevOps 向けの自動化ソリューションでは、関連情報も含めて管理できると謳うものもある[δ]ようだが、本検証内では詳細を確認できなかった）。また、API 連携機能は、CI/CD 内での連携や、作業の効率化のために活用することが望ましい機能であると考えられる。

4.7.6.3.1.3. SBOM 活用・運用の体制について

トレンドマイクロにおいては、 SBOM に関連するプロセスの導入、管理などは PSIRT に

[δ] JFrog Ltd. “ It’s Time to Get Hip to the SBOM”. 2021. https://jfrog.com/blog/its-time-to-get-hip-to-the-sbom/ （参照 2022-12-27)

相当するチームが実施した。脆弱性対応という観点を踏まえると、他企業でも PSIRT またはそれに類するチーム主導で管理を行うのが望ましいと考えられる。また、作成された各製品の SBOM を PSIRT にて活用することで、実際に顧客環境で利用されている OSS の絞りこみに必要な工数が削減され、より効率的な脆弱性対応や監視が可能になると考えられる。

しかしながら、会社の規模や状況によっては PSIRT が存在していない場合もあるだろう。その場合でも、脆弱性の対応は一定のポリシーの下に行われることが望ましい。代わりに SBOM 運用の主体となりえるチームの一つは、品質管理チームであろう。会社横断で品質管理を担うチームが存在するようであれば、品質管理の一環として、成果物としての SBOM の定義や管理、SBOM 活用による脆弱性の対応を行うことで、一定のポリシーの基での運用が可能になると考えられる。

会社横断での品質管理チームがいない場合には、まずは特定の製品チームで SBOM ないしは OSS 管理ツールを導入し、運用・活用のノウハウを蓄積するところから始めるのがよいと考えられる。その後、経験をベストプラクティスとしてまとめて横展開し、チームごとに SBOM を活用することで、一元管理ではないものの、おおむね同じポリシーの基での活用が可能になると考えられる。

4.7.6.3.1.4. ユーザ側での SBOM 活用

ユーザ側での SBOM 活用は、ソフトウェアベンダーにおける PSIRT 視点での調査と一部共通すると考えられる。ユーザとして採用可能な脆弱性の調査手法について下表に記載した。

表 4-115　ユーザとしての脆弱性の調査手法

手法	内容
SBOM	SBOM を蓄積した社内システムまたは、ファイルを直接検索して、自社内のシステムに該当の脆弱性を持つコンポーネントが使用されているか判断する。 PSIRT 視点で社内 SBOM 管理システムを使用し、自社製品での使用について調査するまでの流れと同等。
手動	システム内に使用されているコンポーネント情報を何等かのシステムで管理しているとみなし、その情報をもとに脆弱性を判断する。

ユーザ企業側では、現状は手動検査で調査の基となるコンポーネント情報を持つことがほぼ不可能と考えられる。ベンダーに問い合わせるしか手段はなく、回答が得られるかも含めて、ユーザ企業側ではコントロールできる部分が少ない。正確な情報を持つ SBOM を活用した場合には、 PSIRT 視点での調査結果と同様、自社のシステム内に脆弱性を持つコンポーネントが使用されているかどうかまでを、短期間で判定することが可能だと考えられる。

ただし、ベンダーと異なり、ユーザ企業側で脆弱性の実際の影響度までを判断することは、PoC コードが動作するか確認するなどの別手法がない限り困難である。したがって、実際の影響度を厳密に調査したい場合には、やはりベンダーに確認する必要があるところは変

わらない点は注意する必要がある。また、SBOM の理想は「対応に必要なコンポーネント情報が漏れなく、正しく網羅されているもの」であることは疑いないものの、その理想にすぐにたどり着くのは困難と考えられることは前述の通りである。SBOM により得られた情報は、「ある程度確度があるものの完璧ではない」という実情を踏まえた上で、「影響が判明するまでシステムの使用を制限する」といった具体的なリスク低減策まで踏み込んで実行できるのかなど、事前に SBOM 活用の結果を想定・検討し、社内で合意形成をしておくことが必要だと考えられる。

4.7.6.4. 効果評価

本実証の結果、脆弱性残留リスクの低減、脆弱性対応期間の低減、ライセンス違反リスクの低減、脆弱性管理などにかかるコストの低減については効果が確認された。それぞれの効果測定結果と、算出に使用された情報は下表の通りである。

表 4-116　効果測定結果

分類	効果測定			
	項目	結果		
				（前提条件）
リスク低減	脆弱性残留リスクの低減	15	個多くの OSS につき脆弱性リスクの管理が可能	ツールを使用することで、サブレベルコンポーネントの自動検出が一部可能であったため、トップレベルコンポーネントのみを手動で管理していた状態（25）よりも多く脆弱性リスクを管理できた
	脆弱性対応期間の低減	1	日間短縮可能	ツールを活用することで、脆弱性のあるコンポーネントの自社製品への影響確定までの時間は短縮可能であった。特定された脆弱性の悪用可能性有無以降の調査は共通だったため、差異はない。
	ライセンス違反リスクの低減	15	個多くの OSS につきライセンス違反リスクの管理が可能	ツールを使用することで、サブレベルコンポーネントの自動検出が一部可能であったため、トップレベルコンポーネントのみを手動で管理していた状態（25）よりも多くライセンス違反リスクを管理できた
コスト低減	脆弱性管理などにかかるコストの低減	97.7	人時間の工数削減効果を確認	1OSS あたりの運用工数の差はわずかであったが、OSS の数が多い状況では、初期導入工数のコスト増加を補って余りあるコスト削減が見込めることがわかった 700 の部品（平均的な SW に含まれる OSS 数）を管理したと仮定した場合の値。 ツール費用は以下の理由で算入せず。 製品別に価格・料金体系が異なる 実証対象企業全社で使われるため、特定プロジェクトでのコスト按分が困難

表 4-117　効果の算出用情報一覧

分類	効果測定	効果算出用情報						
	項目	情報項目	ツール利用時		手動生成時		SBOM なしの場合	
			値	単位	値	単位	値	単位
リスク低減	脆弱性残留リスクの低減	OSS 特定数	40	個	—	個	—	
	脆弱性対応期間の低減	脆弱性発見から対応完了までの必要期間	17	日	—	日	18	日
		（小計：ケース別の、SBOM なしの場合からの短縮期間）	1	日	—		—	
	ライセンス違反リスクの低減	OSS 特定数	40	個	—		—	
コスト低減	脆弱性管理などにかかるコストの低減	初期導入工数	120	人時間	—	人時間	0	人時間
		1OSS あたりの運用工数	1.58	人時間/OSS	—	人時間/OSS	1.89	人時間/OSS
		一般的な SW での平均 OSS 利用数	700	個	—	個	700	個
		（小計：ケース別の管理工数）	1225	人時間	—	人時間	1323	人時間
		（小計：SBOM なしの場合からの削減効果）	97.7	人時間	—	人時間	—	

なお、脆弱性管理などにかかるコストの低減効果の算出において、小計として算出した「ケース別の管理工数」は以下の計算式で算出している。

$$初期導入工数 + 10OSSあたりの運用工数 \times 一般的なSWでの平均OSS数$$

SBOM または SBOM 作成にも利用可能な OSS 管理ツールを活用することで、脆弱性とライセンスの管理をより効率的に実施できることが確かめられた。

また、ユーザ企業においては、これまでは特定の OSS の脆弱性やライセンスについて確認したいときは、ベンダーに問い合わせることがほぼ唯一の手段であった。SBOM の活用は、そのための専用のシステムが必要になると想定されるものの、ユーザ企業にとっては自社内で影響調査を実施する場合に活用できる手段を増やせるという大きな利点がある。ユーザが自分で調査できれば、その分ベンダー側への問い合わせも減ると考えられるため、問い合わせ対応の工数を考慮に入れると、全般多岐な脆弱性対応コストをさらに削減できる可能性がある。

ただし、脆弱性管理のコスト削減においては、OSS の利用数が少ない場合、または小規模な製品が対象の場合、初期導入工数のコストをカバーするまでの削減は見込めない可能性がある。また、今回計測に含めた作業以上の正確性の精査を実施した場合にも、コスト削減効果は減少することになる。正確性の向上を求めるほどベンダー側の負担が大きくなる一方で、正確性を過度に高く見積もると、ユーザ・ベンダー双方で、脆弱性の検出漏れなどが発生するおそれがある。正確性の担保をどこまで求めるのか、正確性の程度の認識を業界でどうすり合わせていくのかが、今後の検討課題となるであろう。 NTIA の “The Minimum Elements For a Software Bill of Materials"[ε] にも、”starting today is better than waiting for perfection” とあるように、完璧を待つのではなく、今の課題を少しでも解決するために何が寄与できるかという観点で議論が進められるとよいのではないだろうか。

[ε] NTIA. “The Minimum Elements For a Software Bill of Materials”. 2021. https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf,（参照 2022-12-12）

4.7.6.5. SBOM 適用範囲（SBOM 活用モデル）の改訂案

作成主体が自社の場合の、SBOM 適用モデル案検討結果を下表に示す。

表 4-118　作成主体が自社の場合の SBOM 適用モデル案検討結果
緑は「適用可能」、黄は「部分適用」、赤は「適用困難」を示す

作成範囲	検出・生成・検査手段	生成項目	活用法	活用主体
（b1）直接利用部品	（c2）ツールで特定・生成・誤検知精査なし	（d1）標準フォーマット（SPDX、SPDXLite 等）	（e1）（e2）脆弱性・ライセンスの特定	（f1）製品利用者
				（f2）最終製品ベンダー
			（e3）（e4）悪用可能性・深刻度の評価	（f1）製品利用者
				（f2）最終製品ベンダー
		（d2）大統領令最小要素を含む	（e1）（e2）脆弱性・ライセンスの特定	（f1）製品利用者
				（f2）最終製品ベンダー
			（e3）（e4）悪用可能性・深刻度の評価	（f1）製品利用者
				（f2）最終製品ベンダー
	（c3）ツールで特定・生成・誤検知精査あり	（d1）標準フォーマット（SPDX、SPDXLite 等）	（e1）（e2）脆弱性・ライセンスの特定	（f1）製品利用者
				（f2）最終製品ベンダー
			（e3）（e4）悪用可能性・深刻度の評価	（f1）製品利用者
				（f2）最終製品ベンダー
		（d2）大統領令最小要素を含む	（e1）（e2）脆弱性・ライセンスの特定	（f1）製品利用者
				（f2）最終製品ベンダー
			（e3）（e4）悪用可能性・深刻度の評価	（f1）製品利用者
				（f1）製品利用者
（b2）間接利用部品	（c2）ツールで特定・生成・誤検知精査なし	（d1）標準フォーマット（SPDX、SPDXLite 等）	（e1）（e2）脆弱性・ライセンスの特定	（f1）製品利用者
				（f2）最終製品ベンダー
			（e3）（e4）悪用可能性・深刻度の評価	（f1）製品利用者

				（f2）最終製品ベンダー
		（d2）大統領令最小要素を含む	（e1）（e2）脆弱性・ライセンスの特定	（f1）製品利用者
				（f2）最終製品ベンダー
			（e3）（e4）悪用可能性・深刻度の評価	（f1）製品利用者
				（f2）最終製品ベンダー
	（c3）ツールで特定・生成・誤検知精査あり	（d1）標準フォーマット（SPDX、SPDXLite 等）	（e1）（e2）脆弱性・ライセンスの特定	（f1）製品利用者
				（f2）最終製品ベンダー
			（e3）（e4）悪用可能性・深刻度の評価	（f1）製品利用者
				（f2）最終製品ベンダー
		（d2）大統領令最小要素を含む	（e1）（e2）脆弱性・ライセンスの特定	（f1）製品利用者
				（f2）最終製品ベンダー
			（e3）（e4）悪用可能性・深刻度の評価	（f1）製品利用者
				（f2）最終製品ベンダー

前項にも記載の通り、「(c2) ツールで特定・生成・誤検知精査なし」において、特に脆弱性特定において作業効率化が見込まれることから「適用可能」と判断した。トレンドマイクロのセキュリティポリシー上もこの部分の SBOM 作成・活用は開発プロセスの一部としてすでに取り入れられており、妥当であると考えている。「(e3) (e4) 悪用可能性・深刻度の評価」に関しては、SBOM に含められる有用な情報がほぼなく、結局は手動での確認が必要であることから、SBOM と外部情報を組み合わせることで可能になるという部分的な適用になると考えられる。

「(c3) ツールで特定・生成・誤検知精査あり」については、直接部品か間接部品かで判断が分かれた。今回の検証においては、直接部品・間接部品双方で、すべてを自動検出できないケースが確認できた。直接部品については、開発チームが自覚的にコンポーネントを利用するため、正しい直接部品のリストと突き合わせることは、よほど大規模なプロジェクトでない限りは難しくはないだろう。一方で間接部品については、比較すべき「正しい」情報を確かめる効率的な方法がなく、精査を行うには時間と労力がかかることがわかった。特にアジャイル開発などでは、開発スピードに合わせた精査を行うことは極めて困難だと考えられる。そのため直接部品については「部分適用」、間接部品においてはすべてにおいて「適用困難」と判断した。

ただし、ツールの活用においては、各企業においてその導入・運用コストが認められるかどうかにも実現性が大きく影響される。トレンドマイクロでは有償ツールを全ての開発プロジェクトに対して展開して使用しているが、その分ライセンス料や運用コストも高額となっている。これだけの投資が他の企業で同様に可能かどうかは難しい問題である。SaaS 版の製品を用いて初期投資を減らし、自社にとって特に重要な製品や、SBOM の需要が高いと思われる製品に対してのみを対象とするところから始め、投資効果を見極めるのも一つの手段であろう。

ユーザ企業にとっては、サプライヤーから SBOM を取得することによって、自社に適さないライセンスや、特定のコンポーネントの情報などを効率よく調べる手段が増えることにはなった。ただし、ユーザ企業に適した SBOM 管理ツールは現状なく、自社でシステム開発が可能な企業以外は、SBOM の活用または管理はまだ難しい状態だと考えられる。ユーザ企業の運用状況を加味した管理ツールの開発や、SBOM 情報の整理が今後の業界全体での SBOM 活用には重要となると考えられる。

なお、トレンドマイクロでは外部への開発委託を基本的に実施していないため、開発委託先への SBOM 要件化に関する実践例や対応予定は現状ない。また、OSS や既成部品ベンダーに関しては、使用者一社からの要望で対応されるとは現実的には考えにくく、業界全体での SBOM 活用の広がりから作成者側での積極的な対応が促されるような状況がない限り、要件に含めることは難しいと考えられる。ソースコードが公開・共有されているコンポーネントに関しては、自社のソースコードレポジトリにて該当ソースコードを管理し、SBOM 作成ツールを利用することが次善策となるだろう。

4.7.6.6. 取引契約における要求条項等

ソフトウェアサプライヤーからユーザへの SBOM 提供にあたっては、製品の技術情報の提供などと同様に、検討をすべきだと考えられる。

ユーザに対する SBOM 提供に関する、トレンドマイクロでの方針の現状について、改め

て下表に記載する。

表 4-119　SBOM 提供に関する現時点での方針

検討項目	小項目	詳細
提供形態	項目なし	SPDX 形式のファイルを直接送付
提供条件	リリース形態と SBOM 提供	当社サイトで今後公開する、オンプレミス製品のメジャー・マイナーリリース、Critical Patch, Patch パッケージに対応する SBOM を作成・保管する。ただし、SBOM の活用詳細などがまだ確定していない状況を鑑み、要件が不確定な SBOM ファイルが広く拡散するのを防ぐ目的から、SBOM ファイル自体は一般公開せず、問い合わせに応じて提供する。
	SBOM 提供有償化の有無	原則、製品 readme や技術情報の提供と同様、当社製品顧客や見込み客に無償で提供する。場合により提供時に金銭以外の条件を設定する可能性がある。
契約	条項の修正	現在の企業向けオンプレミス製品の使用許諾契約には、SBOM に誤りがあった場合に関係する記述が存在しない。現状、SBOM 提供の問い合わせがないため、SBOM 記載の内容に誤りがある場合の免責や対応については、提供時に付加情報として説明する予定。今後問い合わせ数の増加が見込まれる場合には、個別の説明が不要となるよう、条項の修正を検討する。
その他	SBOM の記載に誤りがあった場合の修正	SCA ツール上で修正が可能な項目については、修正後 SBOM を再作成する。過去リリースしたパッケージに紐づく SBOM については、ツール側での設定を修正して再作成することが難しいと想定されるため、SBOM ファイル自体を直接編集する必要がある。 SBOM を外部公開している場合には、修正したファイルで差し替えを実施する。問い合わせベースでの共有の場合には、内容の重要度によってはファイルを配布したユーザに、修正済みのファイルを提供する必要があるだろう。

トレンドマイクロでは他の技術情報の提供と同様、SBOM の提供を有償で行う予定はないが、企業によってはツールの導入などのコストを考えて、製品またはサポート費用として上乗せするかどうか、検討が必要な場合もあると考えられる。また、現状の契約内容に SBOM にあてはまる条項があるか、特に誤りがあった場合にどういった法的リスクが考えられるかは、検討の上対応が必要な内容だと考えられる。

関連分野でサプライヤーとの契約において想定される契約事項については下表に記した。

表 4-120　サプライヤーとの契約において期待される契約事項

部品種別	小項目	詳細
直接利用部品	SBOM 提出の要件化	● ソースコードが納品物として含まれる場合には、自社での SBOM 作成が可能と考えられる。

		その場合契約上、自社での SBOM 作成に問題がないかは確認の必要がある ● ソースコードが納品されない場合には、委託先で SBOM 作成が可能かどうか議論し、成果物の一つとして追加すべく交渉する必要がある。 ● 委託先に SBOM 作成を依頼する場合には、SBOM 作成ツールの導入・維持コストが課題になる可能性がある。コスト負担についても必要に応じて契約に記載すべき ● 作成した SBOM の内容精査を実施する場合には、委託先の開発チームに確認作業を依頼するのか、する場合にはその業務について契約でカバーされているか確認が必要 ● SBOM 提供または内容精査に当たっては、内部利用の OSS について正確な情報を記述するには大きなコストがかかることを両者理解した上で、正確性の担保の粒度に関して認識合わせをし、対応範囲について事前に合意しておくことが望ましい
	第三者部品・OSS 利用の申告	SBOM による提供が不可能な場合には、自社の調達ポリシーに合った情報提供について、別途契約に含めることが望ましい
	脆弱性の通知・修正申告	● SBOM 提供の有無にかかわらず、脆弱性の通知や報告、対応に関して確認し、合意内容を契約に盛り込んでおくことが望ましい ● 特に脆弱性の報告に対する応答や修正に関して自社のポリシーや方針が確立されている場合には、委託先の対応がそのポリシーに沿うことが可能かを確認しておくべき
	ライセンス違反等の責任関係・損害賠償	契約内で定義し、合意しておくことが望ましい
間接利用部品	SBOM 提出の要件化	● 一般的に、サプライヤー側で SBOM を提供することを決定している場合を除き、要件化や依頼は困難 ● 間接利用部品のライセンス条項を確認し、自社でソースコードやバイナリからの SBOM 作成が可能な場合には、自社で一定程度対応が可能
	第三者部品・OSS の利用の申告	
	脆弱性の通知・修正申告	● 既成部品ベンダーのコンポーネントや OS などの場合には、ライセンス契約内の記述、または会社のポリシーにて定められている可能性が高い。不明瞭な場合にはベンダーに確認した上で、

		自社の対応を検討すべき ● OSS の場合には、一般的に確約された対応プロセスは存在しない。原則的に自社にあった対応を依頼することは不可能と考えるべき
	ライセンス違反等の責任関係・損害賠償	● 既成部品ベンダーのコンポーネントや OS などの場合には、契約書内での記述を確認する必要がある ● OSS 内部でのライセンス違反や、自社のライセンスポリシー違反などによる、責任追及や損害賠償は一般的に困難

4.8. ソフトウェア分野（コラボスタイル）

4.8.1. 実証の狙い・重点ポイント

- 対象ソフトウェアの SBOM を生成する際の運用コストを、その精度も含めて手段毎に測定・比較し、ソフトウェアベンダーが SBOM を日常的に活用する際の課題点を洗い出す。
- 対象ソフトウェアがサードパーティのミドルウェア等に依存する場合、サードパーティからの SBOM の入手方法や、その他代替手段があるかを確認し、課題点を洗い出す。
- 対象ソフトウェアの脆弱性検出に、SBOM やその出力ツールが有用であるかを測定・比較する。
- SBOM 事業全体で事前に検討した SBOM の生成・活用範囲を示す SBOM 活用モデル（仮説）に対して、本実証では、具体的なコスト・効果を評価・比較することで、ソフトウェア分野で妥当と考えられる範囲について検討し、改定案を提示することを目的とする。

4.8.2. 実証の手順

実証の手順は以下のとおりとする。

1. 対象システムの構成と、SBOM 生成対象を特定する。
2. ソフトウェア分野における契約形態、SBOM の利用範囲を整理する。
3. 初期導入時の体制構築、環境整備、ツール検証を行う。
4. SBOM 生成の工数計測、比較評価、課題抽出を行う。
5. SBOM 生成結果の精査、工数計測、比較評価、課題抽出を行う。
6. SBOM の活用範囲、活用主体の検討課題抽出を行う。
7. 管理・運用方法（SBOM の更新等）の工数計測、課題抽出、責任関係の検討を行う。

実証に参加したメンバーの経験、スキルレベルは以下のとおり。

表 4-121　実証に参加したメンバーの経験、スキルレベル

担当者	役割	スキルレベル・経験
担当者 A	プロジェクト管理者	・10 年以上のソフトウェア開発経験、インフラ、ネットワーク構築経験
担当者 B	作業担当者	・10 年以上のソフトウェア開発経験、インフラ、ネットワーク構築経験

なお、本実証を実施した 2 名の担当者の双方が 10 年以上のソフトウェア開発経験、インフラ、ネットワーク構築の経験を持つメンバーである。そのため、より経験が浅い人物が SBOM を利用した際にどのような課題があるかについての検証は、本実証では行えていない点に留意が必要である。

4.8.3. 実証の対象システムの概要と構成

実証の対象システムはワークフローシステム（業務フロー管理システム）の「コラボフロークラウド[78]」とする。本システムの利用シーンや導入顧客については、導入事例[79]ページで確認できる。

対象システムは AWS 上で稼働するクラウドサービスであり、ミドルウェアとして Adobe 社の「ColdFusion」、データベースに OSS の「PostgreSQL」が稼働している。

今回の実証の対象範囲としては、「ColdFusion」を含めることでサードパーティがライセンスするミドルウェアの SBOM が入手可能かを検証する。また、ミドルウェアとは独立してコラボフロー側で独自に採用している「OSDC」等を対象として SBOM がツールにて抽出可能であるかを確認する。

また、クライアント側は Web ブラウザがあれば他にソフトウェアのインストールを必要としないが、サーバーサイドから JavaScript ファイルがダウンロードされるため、それらのライブラリの SBOM がツールで抽出可能かについても確認する。

SBOM 生成範囲のコンポーネントの開発言語は以下のとおりである。ファイルフォーマット及び下記言語のいずれも Mend のサポート対象であることを確認している。

表 4-122　コラボフロークラウドの開発言語

構成	要素	開発言語
コラボフロー（業務フロー管理システム）	業務フロー管理サーバー	ColdFusion（CFML)
	OSDC（PDF 作成ソフトウェア）	不明
ColdFusion（アプリケーションサーバー）	Tomcat/JavaEE（サーバーサイド）	Java
ユーザーインターフェース（ブラウザ）	（全体）	JavaScript

[78] https://www.collabo-style.co.jp/
[79] https://www.collabo-style.co.jp/case/

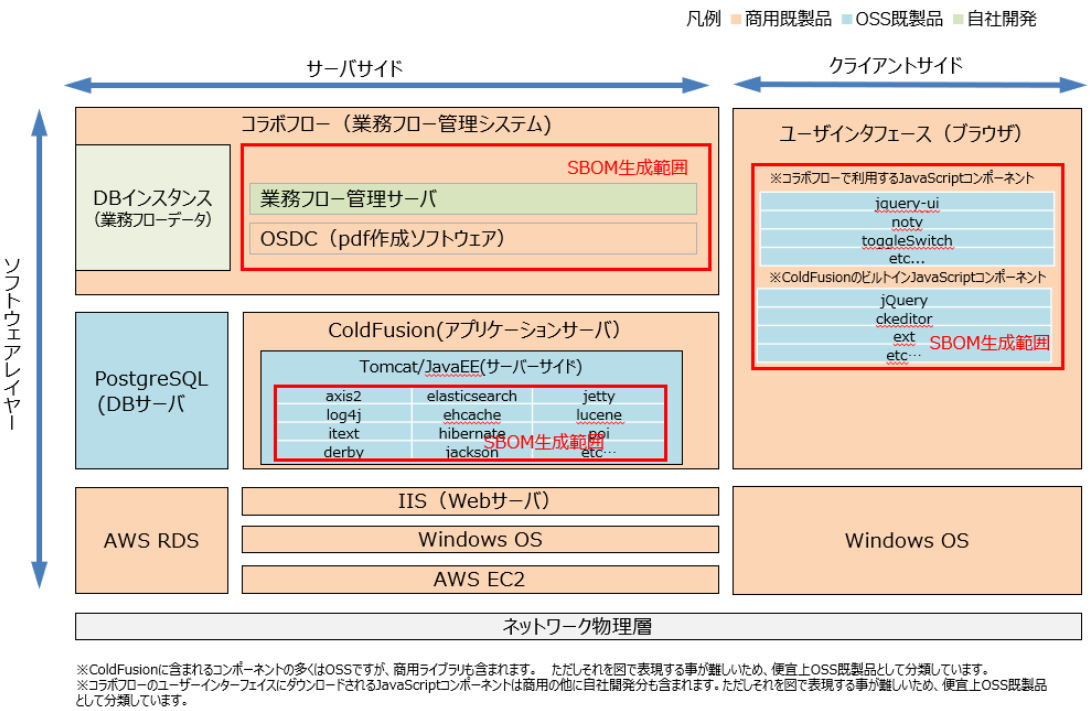


図 4-28　対象システムの構成と SBOM 生成範囲

4.8.4. 対象分野の前提条件・取引慣行等

対象分野で SBOM に関連し必要となる要件、取引慣行などの前提条件を、以下のとおり特定した。コラボスタイルでは OSS 及び一部の有償ライブラリを調達して製品開発を実施しているものの、開発の一部を外部に委託して開発することは原則として行っていない。したがって、本項では自社開発のソフトウェアベンダーとユーザー間の契約形態及び取引慣行等について下の表に示す。

表 4-123　対象分野の前提条件・取引慣行等

契約形態・取引慣行等	本実証における形態	業界において想定される主な形態
契約類型	既製品ライセンス契約（委託契約は行っていないため、ユーザー企業との契約のみ）	既製品ライセンス契約。(請負/準委任での開発委託契約を行っているケースも多いと思われる)
部品情報提供の要件	規約上、利用者への構成情報・部品情報の提供義務無し。要求された場合でも現状は非開示。今後 SBOM 提供が一般的になれば開示を検討するが、その場合でも SBOM が脆弱性の洗い出し材料として利用されないように対策は必要と考える。	弊社と同様のケースが多いと想定。
第三者部品の申告	規約上、利用者への告知義務なし。ただし OSS のライセンス条項に基づき、第三者部品は、ライセンス情報ファイルに利用しているライブラリを記載。	弊社と同様のケースが多いと想定。
脆弱性の通知・修正	規約上通知義務や、修正義務はない。 発見された脆弱性は、実際の悪用を防ぐため、全ての関連サービスの修正が終わってからリリースノートで公開される。 万が一第三者により脆弱性が一般に広く公開されるなど早急なユーザーの協力が必要となる場合は、ユーザーへの個別連絡や対策を行うこともある。	弊社と同様のケースが多いと想定。
納品形態	クラウドサービスであるため、メールによるお客様環境開設通知を以って納品とする。	弊社と同様のケースが多いが、追加で紙の納品書を送るケースもあると想定される。
損害賠償責任	クラウド規約に従った本サービスの提供に関して、お客様に発生したいかなる損害について、一切の責任を負わない。	弊社と同様のケースが多いと想定するが、SLA に基づき、一定の損害賠償責任を負う場合もある。
知的財産権の	ベンダー保有・使用権のみ提供。	弊社と同様のケースが多いと

帰属		想定。

SBOM 管理については、開発環境等に係る情報は大きく影響するため本事象及び同分野で想定される例についてまとめる。

表 4-124　開発環境等

項目	本実証	業界において想定される主な例
開発言語	ColdFusion(CFML), JavaScript	サーバーサイドでは nodejs(express),PHP,GO,Ruby 等 クライアントサイドは JavaScript ベースのフレームワーク （React, Vue.js 等）
部品形態	OS、システムライブラリ、コンパイラライブラリ、ミドルウェア、アプリケーション、データベースサービス、メールサービス、Web サービス	OS、システムライブラリ、コンパイラライブラリ、ミドルウェア、アプリケーション、データベースサービス、メールサービス、Web サービス等。基本的には自社と大きく変わらない想定。
開発環境ツール	VisualStudioCode	VisualStudioCode
ビルドツール	webpack	クライアントサイドでは webpack や babel が多いと想定。
構成管理ツール	Bitbucket(git)	Git による管理が多いと想定。

4.8.5. 実証項目の特定

本 SBOM 実証では、以下に示す項目を元に作業を実施した。

表 4-125　SBOM 実証フェーズ毎の実証項目

フェーズ	実証項目
基本方針 （SBOM 活用モデルを実施する際のコスト計測、実現性、課題を確認する。）	・サプライチェーンにおける最終開発者（Sier 又は機器ベンダー）を SBOM 作成の主体とし、委託契約のある開発者の SBOM 生成は、実証のコスト評価に基づく実運用での作成可否を考察する。 ・SBOM を生成できない委託先開発者から再帰的に利用される部品（間接利用部品）について、最終開発者等がツールにより SBOM を生成し、その精査の可否、コストについて評価する。 ・SBOM の先行企業が参加できる場合、SBOM の信頼性を高めるための精査方法やコストについて評価する。

<table>
<tr><th colspan="2">フェーズ</th><th>実証項目</th></tr>
<tr><td colspan="2">実証主体（想定）</td><td>SIer、サプライヤー(ソフトウェア受託開発者又はパッケージソフトベンダー)</td></tr>
<tr><td colspan="2">対象ソフト</td><td>コラボフロー（業務フロー管理 SaaS）</td></tr>
<tr><td colspan="2">初期導入</td><td>・ツールの選定、インストール、利用方法習得の工数などのコストの評価、課題の特定を行う。</td></tr>
<tr><td rowspan="6">生成・共有</td><td>SBOM 作成主体（ユーザー／ベンダー／サプライヤー／サードパーティ）</td><td>・サプライヤーが、自社開発部分の SBOM を生成することのフィージビリティを確認するため、生成コストの計測と課題を確認する。
・CSP、サプライヤーの両者による計測を想定するが、サプライヤーの実証が難しい場合、CSP などが代替して計測する。</td></tr>
<tr><td>対象範囲（直接利用する部品のみ／再利用等も含む）</td><td rowspan="2">・CSP、サプライヤーが、自身の開発部分の直接利用部品の SBOM を生成し、コストを評価する。
・SBOM を作成できないサプライヤーや既製品から再帰的に利用される部品（間接利用部品）は CSP が有償ツール等を用いて一括生成し、コストを評価する。
・直接利用部品については、無償ツールにより SBOM 生成の可否を検討することで、コストの低減が可能か評価する。
・間接利用部品について、OSS などソースコードレビューによりツールで生成した SBOM の誤検知など精査し、コストを評価する。</td></tr>
<tr><td>作成手段（無償／有償の別、ツールの種類、機能など）</td></tr>
<tr><td>SBOM のデータ形式</td><td>ツール選択肢と入出力を考慮し、妥当な標準形式を選定する。SPDX 等を想定し、CSP,サプライヤーで統一可能か検討する。</td></tr>
<tr><td>SBOM の項目</td><td>ツールにより、大統領令最小要素をカバーできるか確認する。</td></tr>
<tr><td>共有方法（ファイル形式、受入検査等）</td><td>・ツールの SBOM 入出力の可否を確認する。
・サプライチェーンでツールの共通化の可否、課題を検討する。
・サプライチェーンでデータ形式が統一できない場合、ツールによるデータ変換の検討。
・SBOM 受入側は、独自に SBOM 生成し比較するなど、SBOM の信頼性検査の要否を検討する。</td></tr>
</table>

フェーズ		実証項目
活用・管理	活用範囲（調達時のリスク評価、脆弱性対応、資産管理、供給者管理）	・脆弱性特定、脆弱性深刻度評価(CVSS)、悪用可能性評価(VEX)、ライセンス特定を実施し、コスト、課題を評価する。 ・SBOM 誤検知など信頼性の誤差を許容し、脆弱性特定の絞り込み後に SBOM 精査を行うことでコスト低減できるか評価する。
	SBOM の管理・運用の責任主体	・OSS 含めライセンス契約に準拠した部品の申告は必要 ・契約条項として SBOM の要求が可能か考察。 ・最終製品ベンダーの部品情報、SBOM の責任について考察。
	管理・運用方法など	CSP、サプライヤー、ユーザー企業について、脆弱性検出時の対処（通知、利用中断、修正など）を洗い出し、SBOM 有無によりリスク低減効果（対処時間の短縮など）を評価する。
主な実証項目に適合するツール		Mend, Yamory

4.8.6. 実証結果

実証項目に基づき、SBOM 初期導入、SBOM の生成・共有、及び SBOM の活用・管理等について以下のとおり実証を行った。

4.8.6.1. SBOM 初期導入

4.8.6.1.1. 実施内容とコスト評価

Mend を使用した場合、SBOM 初期導入には合計で 62 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりである。

表 4-126　Mend を使用時の SBOM 初期導入コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
環境構築工数	8	人時間	8	人時間		人時間	・ツールのインストール方法調査、実施、簡易的な動作確認(4h) ・ツールのバッチ処理作成・検証など、SBOM 出力自動化に関する環境整備(4h)
学習・体制構築工数	54	人時間	54	人時間		人時間	・SBOM の概念、必要性、大まかなフォーマット等を理解するための学習・教育(4h) ・ツールのバッチ処理作成・検証など、SBOM 出力自動化に関する環境整備(30h) ・SBOM の概念、必要性、大まかなフォーマット等を理解するためのドキュメント作成(4h)

							・ツールのインストール、使用方法手順書整備(初期セットアップ)(4h) ・ツールを運用するための手順書整備(12h)

表 4-127　Mend を使用時の SBOM 初期導入コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
環境構築工数	環境構築工数	8	人時間	8	人時間		人時間	・ツールのインストール方法調査、実施、簡易的な動作確認(4h) ・ツールのバッチ処理作成・検証など、SBOM 出力自動化に関する環境整備(4h)
学習・体制構築工数	学習工数	34	人時間	34	人時間		人時間	・SBOM の概念、必要性、大まかなフォーマット等を理解するための学習・教育(4h) ・ツールのバッチ処理作成・検証など、SBOM 出力自動化に関する環境整備(30h)
	体制構築工数	20	人時間	20	人時間		人時間	・SBOM の概念、必要性、大まかなフォーマット等を理解するためのドキュメント作成(4h) ・ツールのインストール、使用方法手順書整備(初期セットアップ)(4h) ・ツールを運用するための手順書整備(12h)

また、Yamory を使用した場合、SBOM 初期導入には合計で 38 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりであ

る。

表 4-128　Yamory を使用時の SBOM 初期導入コスト小計

小計							
項目	**結果**						**備考**
	合計		**ティア 1**		**ティア 2**		
	値	単位	値	単位	値	単位	
環境構築工数	2	人時間	2	人時間		人時間	・ツールのインストール方法調査、実施、簡易的な動作確認(2h) ・ツールのバッチ処理作成・検証など、SBOM 出力自動化に関する環境整備(0h：自動化不可のため対象外)
学習・体制構築工数	36	人時間	36	人時間		人時間	・SBOM の概念、必要性、大まかなフォーマット等を理解するための学習・教育(4h) ・ツールのバッチ処理作成・検証など、SBOM 出力自動化に関する環境整備(16h)

表 4-129　Yamory を使用時の SBOM 初期導入コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
環境構築工数	環境整備工数	2	人時間	2	人時間		環境整備工数	・ツールのインストール方法調査、実施、簡易的な動作確認(2h) ・ツールのバッチ処理作成・検証など、SBOM 出力自動化に関する環境整備(0h：自動化不可のため対象外)
学習・体制構築工数	学習工数	20	人時間	20	人時間		学習工数	・SBOM の概念、必要性、大まかなフォーマット等を理解するための学習・教育(4h) ・ツールのバッチ処理作成・検証など、SBOM 出力自動化に関する環境整備(16h)
	体制構築工数	16	人時間	16	人時間		体制構築工数	・SBOM の概念、必要性、大まかなフォーマット等を理解するためのドキュメント作成(4h) ・ツールのインストール、使用方法を運用するための手順書整備(初期セットアップ)(4h) ・ツールのインストール、使用方法を運用するための手順書整備(8h)

SBOM 手動生成の場合、SBOM 初期導入には合計で 64 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりである。

表 4-130　SBOM 手動生成時の SBOM 初期導入コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
環境構築工数	0	人時間	0	人時間		人時間	無し
学習・体制構築工数	64	人時間	64	人時間		人時間	無し

表 4-131　SBOM 手動生成時の SBOM 初期導入コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
環境構築工数	環境整備工数	0	人時間	0	人時間		人時間	無し

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
学習・体制構築工数	学習工数	0	人時間	0	人時間		人時間	無し
	体制構築工数	64	人時間	64	人時間		人時間	・SBOM の概念、必要性、大まかなフォーマット等を理解するためのドキュメント作成(4h) ・ツールを用いずに SBOM を生成する、脆弱性管理を行う、ライセンス管理を行うための手順確立・手順書整備」(60h)

SBOM を生成するための環境を整備するに当たり、当初はツールとして株式会社アシュアードが提供する「Yamory」を検討した。しかしながら検証の結果、弊社環境では Yamory による SBOM 抽出ができないことが判明したため、Mend を採用することとなった。

表 4-132　SBOM 実証対象ツール

製品名	**Yamory**	**Mend**
URL	https://yamory.io/	https://www.mend.io/
運営会社	株式会社アシュアード	White Source Ltd.
製品概要	IT システムの脆弱性対策とリスク管理を一つのプラットフォームで実現する国産の脆弱性管理クラウド。	プロダクトが依存している OSS を特定・既知の脆弱性を通知し、利用しているライセンスのコンプライアンスを管理するソフトウェア構成分析ツール。
提供方式	SaaS	SaaS
弊社での SBOM 出力	不可。詳細は「4.8.6.2.2 課題。解決ノウハウ、留意点」一つめの課題「パッケージマネージャの必要性」に記載	可。詳細は「4.8.6.2.2 課題。解決ノウハウ、留意点」一つめの課題「パッケージマネージャの必要性」に記載

初期導入に必要となるコストとして発生した、SBOM を生成するための前提知識の学習やインストール、文書整理など、初期導入に関する作業工数は以下のとおりであった。

表 4-133　SBOM 初期導入コスト

作業ステップ	作業の内容	作業時間
SBOM 前提理解	SBOM の概念、必要性、大まかなフォーマット等を理解するための学習・教育	4 時間
	SBOM の概念、必要性、大まかなフォーマット等を理解するためのドキュメント作成（担当者変更時の継続運用のため必要）	4 時間 （未実施：想定される工数）
利用方法理解とツールのセットアップ（Yamory）	ツールのインストール方法調査、実施、簡易的な動作確認	2 時間
	ツールを継続して運用するための手順書整備(初期セットアップ)	4 時間 （未実施：想定される工数）
利用方法理解とツールのセットアップ（Mend）	ツールのインストール方法調査、実施、簡易的な動作確認	4 時間
	ツールのインストール、使用方法を運用するための手順書整備(初期セットアップ)	4 時間 （未実施：想定される工数）

初期導入作業時には以下の点に留意する必要がある。

- SBOM の概念やフォーマット等は現時点であまり一般的ではないため、概念や前提を理解し、SBOM のフォーマットについての概要を把握するための前提理解のステップを設けた。
- 組織では常に担当者の変更が起こり得ることから、担当が変更となった場合にもスムーズに引継ぎができるドキュメントを整備しておく必要がある。そのため、それらのコストを本工数として計上した。（今回の実証では実際にはドキュメントの整備は行わず、想定される工数を計上するのみとした。）

また、SBOM を定期的に生成するためのツールの学習、出力結果の評価、文書整理等に関する作業工数は、以下のとおりであった。

表 4-134　SBOM 定期生成、評価、運用コスト

作業ステップ	作業の内容	作業時間
ツール利用方法調査（Yamory）	ツール自体の使用方法を調査（試行と動作確認）	8 時間
	ツールによる SBOM 出力方法の調査、試行、動作確認（直接部品のみ）	4 時間 （結果的には出力不可であったが、確認・調査に必要となった工数を計上）
	ツールによる SBOM 出力方法の調査、試行、動作確認（再利用部品を含む）	4 時間 （未実施：パッケージマネージャが存在する環境での想定工数）
	ツールのバッチ処理作成・検証など、SBOM 出力自動化に関する環境整備	対象外 （SBOM は Web から手動ダウンロードするため対象外）
	ツールを継続して運用するための手順書整備	8 時間 （未実施：想定される工数）
ツール利用方法調査（Mend）	ツール自体の使用方法を調査（試行と動作確認）	8 時間
	ツールによる SBOM 出力方法の調査、試行、動作確認（直接部品のみ）※誤検知、検出漏れの確認工数も含む。	12 時間
	ツールによる SBOM 出力方法の調査、試行、動作確認（再利用部品を含む）※誤検知、検出漏れの確認工数も含む。	12 時間 ※調査の結果、OSS のライブラリとして認識された部品の再利用部品を検出するのは不可と判明。
	ツールのバッチ処理作成・検証な	4 時間

作業ステップ	作業の内容	作業時間
	ど、SBOM 出力自動化に関する環境整備	
	ツールのインストール、使用方法を運用するための手順書整備	12 時間

4.8.6.1.2. 課題、解決ノウハウ、留意点

SBOM 初期導入における課題としては、検証前に SBOM ツールの機能を正確に把握するのが難しい、ということが挙げられる。

「4.8.6.1.1.実施内容とコスト評価」で示したとおり、弊社環境ではある程度検証が進んだ段階で Yamory による SBOM 抽出ができないことが判明した。この原因はツール選定時に yamory-cli の「ホストスキャン」がソースコード解析を行うものと誤認したためであった。

マニフェストファイルを用いてソフトウェア開発を行っている場合は、多くの SBOM ツールが一般的なマニフェストファイルによる解析に対応しているため、ツールによる SBOM 検出ができない可能性は低い。

しかし、ソフトウェア開発にマニフェストファイルを使用しておらず、ソースコード解析が必要になる場合や、マニフェストファイルを利用している場合でも一般的ではない形式で管理されている場合は、SBOM ツールでの解析ができない場合もあるため、事前にメーカーに相談し、使用している言語でのソースコード解析が可能かどうかを確認しておくことで、不要な検証コストが発生しないように対策することができる。

また、SBOM 初期導入における実施内容とそのコスト評価の詳細を以下に示す。

SBOM は現時点で広く知られている概念ではないため、前提知識として SBOM の概念や前提、及び SBOM のフォーマットについての概要を把握するため、4 時間の工数が必要となった。

また、実際に学習を行った結果、SBOM の担当者には前提条件として対象者がソフトウェア開発に一定以上関わった経験（2～3 年以上）があることが望ましいとの所見を得た。

これは、SBOM の概念理解、活用のためには、基本的なソフトウェアの構造やセキュリティに関する知識等が必要となるためである。

SBOM 管理ツールの初期セットアップについては、双方共にホストサーバーは SaaS 形式で提供されるため、解析した SBOM 情報の収集先となるサーバーをセットアップする必要は無く、インストールは容易であった。

以下にインストールが必要となったツールを示す。

表 4-135　SBOM を生成するために必要なツール

製品名	Yamory	Mend
ホストサーバー	SaaS 形式で提供されるためインストール不要	SaaS 形式で提供されるためインストール不要
構成分析ツール	クライアント側ツール	クライアント側ツール

製品名	**Yamory**	**Mend**
	yamory-cli を利用する	Unified Agent を利用する
SBOM 生成ツール	SBOM はホストサーバーからダウンロードするため専用のクライアントツールは存在しない。	クライアント側ツール Mend SBOM Generator を利用する

それぞれのツールの概要を確認し、インストールと簡単な動作確認、ツールを継続して運用するための初期セットアップ分の手順書整備を行うために、Yamory で 6 時間、Mend で 8 時間が必要となった。なお、Yamory よりも Mend の方がわずかに時間を要しているのは、Yamory の SBOM 生成がホストサーバーに Web ブラウザでアクセスしダウンロードするという簡易的なものであるのに対し、Mend は Mend SBOM Generator という専用の出力クライアントを持つためである。

4.8.6.2. SBOM の生成・共有

4.8.6.2.1. 実施内容とコスト評価

Mend を使用した場合、SBOM の生成・共有には 1OSS 当たり合計で 0.0945 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりである。

表 4-136　Mend を使用時の SBOM 生成・共有コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
情報取得工数	0.0045	人時間/OSS	0.0045	人時間/OSS		人時間/OSS	・SBOM 定期出力のための設定作業工数」の一連の作業に含まれるが 1OSS 辺りの工数は算出していない。 対象の部品数：22 点 ・Mend の初期設定が整っている状態での部品特定は手動で処理を走らせて、検出を待って確認するのみとなり、工数は 0.1h 程度 ・Mend SBOM Generator によって(d1)標準フォーマット（SPDX 形式）のデータが出力されるため、情報追記、整形の必要は無く、工数は 0 となる。
精査工数	0.09	人時間/OSS	0.09	人時間/OSS		人時間/OSS	対象の部品数：22 点 SBOM 定期出力のための設定作業工数「ツールによる SBOM 出力方法の調査、試行、動作確認（直接部品のみ）」に誤検知、検出漏れの確認工数を含んでいる(1h*2)
（作成ツール	3,360,000	円		円		円	ライセンス体系(プラン)及び開発者数に応じた見積。費用は年額（2022 年 12 月現在）

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
費用）							

表 4-137　Mend を使用時の SBOM 生成・共有コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
情報取得工数	部品特定工数	0.0045	人時間/OSS	0.0045	人時間/OSS		人時間/OSS	・SBOM 定期出力のための設定作業工数」の一連の作業に含まれるが 1OSS 辺りの工数は算出していない。 対象の部品数：22 点 ・Mend の初期設定が整っている状態での部品特定は手動で処理を走らせて、検出を待って確認するのみとなり、工数は 0.1h 程度
	SBOM 生成工数	0	人時間/OSS	0	人時間/OSS		人時間/OSS	・Mend SBOM Generator によって(d1)標準フォーマット（SPDX 形式）のデータが出力されるため、情報追記、整形の必要は無く、工数は 0 となる。
精査工数	精査工数	0.09	人時間/OSS	0.09	人時間/OSS		人時間/OSS	対象の部品数：22 点 SBOM 定期出力のための設定作業工数「ツールによる SBOM 出力方法の調査、試行、動作確認（直接部品のみ）」に誤検知、検出漏れの確認工数を含んでいる(1h*2)

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
（作成ツール費用）	（作成ツール費用）	3,360,000	円		円		円	ライセンス体系(プラン)及び開発者数に応じた見積。費用は年額（2022 年 12 月現在）

また、Yamory を使用した場合、SBOM の生成・共有には 1OSS 当たり合計で 0.0945 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりである。

表 4-138　Yamory を使用時の SBOM 生成・共有コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
情報取得工数	0.0045	人時間/OSS	0.0045	人時間/OSS		人時間/OSS	対象の部品数：22 点 ・SBOM 定期出力のための設定作業工数の一連の作業に含まれるが 1OSS 辺りの工数は算出していない。 ・Yamory の初期設定が整っている状態での部品特定は手動で処理を走らせて、検出を待って確認するのみとなり、工数は 0.1h 程度
精査工数	0.09	人時間/OSS	0.09	人時間/OSS		人時間/OSS	対象の部品数：22 点 ツールによる SBOM 出力方法の調査、試行、動作確認（直接部品のみ）に誤検知、検出漏れの確認工数を含んでいる(1h*2)

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
（作成ツール費用）	2,000,000	円		円		円	ライセンス体系(プラン)及び開発者数に応じた見積。費用は年額 180 万円に初期費用 20 万円を含んだもの（2022 年 12 月現在）

表 4-139　Yamory を使用時の生成・共有コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
情報取得工数	部品特定工数	0.0045	人時間/OSS	0.0045	人時間/OSS		人時間/OSS	対象の部品数：22 点 ・SBOM 定期出力のための設定作業工数の一連の作業に含まれるが 1OSS 辺りの工数は算出していない。 ・Yamory の初期設定が整っている状態での部品特定は手動で処理を走らせて、検出を待って確認するのみとなり、工数は 0.1h 程度
	SBOM 生成工数	0	人時間/OSS	0	人時間/OSS		人時間/OSS	yamory-cli によって独自形式のデータが出力される。本形式のデータは大統領令最小要素を満たすため、それ以上の要求が無ければ情報追記、整形の必要は無く、工数は 0 となる。
精査工数	精査工数	0.09	人時間/OSS	0.09	人時間/OSS		人時間/OSS	対象の部品数：22 点 ツールによる SBOM 出力方法の調査、試行、動作確認（直接部品のみ）に誤検知、検出漏れの確認工数を含んでいる(1h*2)

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
（作成ツール費用）	（作成ツール費用）	2,000,000	円		円		円	ライセンス体系(プラン)及び開発者数に応じた見積。費用は年額 180 万円に初期費用 20 万円を含んだもの（2022 年 12 月現在）

SBOM 手動生成の場合、SBOM の生成・共有には 1OSS 当たり合計で 0.333 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりである。

表 4-140　SBOM 手動生成時の SBOM 生成・共有コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
情報取得工数	0.333	人時間/OSS	0.333	人時間/OSS		人時間/OSS	OSS の情報を構成ファイルや Web 等で確認し、部品表への追記を行う。
精査工数	0	人時間/OSS	0	人時間/OSS		人時間/OSS	手動作業となるので、誤検知検出漏れの特定作業はない。
（作成ツール費用）	0	円		円		円	無し

表 4-141　SBOM 手動生成時の SBOM 生成・共有コスト内訳

小計	内訳

項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
情報取得工数	部品特定工数	0.083	人時間/OSS	0.083	人時間/OSS		人時間/OSS	OSS の情報を構成ファイルや Web 等で確認し、部品表への追記を行う。
	SBOM 生成工数	0.25	人時間/OSS	0.25	人時間/OSS		人時間/OSS	大統領令最小要素を対象として工数を算出。Web 等による情報の確認と追記。
精査工数	精査工数	0	人時間/OSS	0	人時間/OSS		人時間/OSS	手動作業となるので、誤検知検出漏れの特定作業はない。
（作成ツール費用）	（作成ツール費用）	0	円		円		円	無し

本実証における SBOM のデータ形式、項目、共有方法の妥当性、実現性について以下のとおり検証を行った。

表 4-142　SBOM 手動生成時の SBOM 生成・共有コスト内訳

作業の内容	実証項目	実証結果
SBOM のデータ形式	SPDX 形式での出力が可能かについて検証する。	•Mend では SPDX 形式での出力が可能。 •Yamory は独自形式の CSV 出力。
SBOM の項目	ツールにより、大統領令最小要素をカバーできるか確認する。	Mend,Yamory 共に大統領最小要素をカバーしている。 【Mend】 日立ソリューションズ様に確認：NTIA による SBOM の最小構成要素(全 7 要素)の内、6 要素については明確に出力をサポートしている。しかし「Other Unique Identifiers」だけは、CPE や PURL 等の「SBOM 構成要素と脆弱性情報との紐づけに活用できる情報」が出力されるべきとされる説もあるため、これらの情報が必要と考える場合は、手動で補足する必要がある。 　なお、CPE や PURL 等が無くても、Mend によって脆弱性の検知が可能であるため、当該情報が無くとも脆弱性管理実務上の問題はないと考える。 【Yamory】 株式会社アシュアード様に確認：Visional 社からのプレスリリース[80]のとおり、NTIA による SBOM の最小構成要素(全 7 要素)に対応している。 Yamory から出力される SBOM 情報は、現状 PURL 形式ではないが PURL に含まれている情報は持っているため、必要であれば手動で生成することは可能である。 なお、CPE や PURL 等が無くても、Yamory によって脆弱性の検知が可能であるため、当該情報が無くとも脆弱性管理実務上の問題はないと考える。
共有方法（ファイル形式、受入	SBOM を社内共有する方法	Mend,Yamory 共にブラウザで部品表と脆弱性情報の両方を確認可能であるため、SBOM をダウンロードしてファイル共有するのではなく、必要なユーザーがサービスにログインし部品表を

[80] https://prtimes.jp/main/html/rd/p/000000407.000034075.html

作業の内容	実証項目	実証結果
検査等）		確認する運用が最も利便性が高い。
	サードパーティ製ミドルウェアの SBOM 入手	ミドルウェアベンダーである Adobe 社からの SBOM 入手は不可であった。「4.8.6.2.2 課題、解決ノウハウ、留意点」三つめの課題「サードパーティ製ミドルウェアの SBOM 入手」に詳細を記載。

なお、今回出力された SBOM が SPDX に準拠しているかについて、項目レベルでのチェックは行っていない。これは、Mend の SBOM 生成ツール「Mend SBOM Generator」[81]が公式に SPDX 形式又は CycloneDX 形式の SBOM レポートを生成するためのツールとして提供されていることから、詳細なチェックは必要ないと判断したためである。

[81] https://github.com/whitesource-ps/ws-sbom-generator

部品の誤検知・検出漏れについては、自社でスプレッドシートにより管理しているコンポーネントのリストと、検出された部品が一致するかについて目視で確認を行った。
その結果、自社管理部品と検出された部品が一致することが確認された。

表 4-143　対象のコンポーネント数

	自社管理	Mend による検出
コンポーネント数	22	22

ただし、自社管理でも再利用部品の管理までは行っておらず、また Mend では再利用部品までの検出ができないことから、再利用部品の管理については課題が残っている。
また、弊社では使用されている部品数が 22 点と少なく、確認は容易であったが、一般的にはより多くのライブラリを利用している可能性が高い。多くのライブラリを利用し、かつパッケージマネージャを利用していない場合は、チェックの工数はより多くなると考えられる。
なお、構成情報が取得できたツールは Mend のみであったため、Yamory の部品の誤検知、検出漏れチェックは実施していない。

上記の作業により、実際に SBOM を継続運用するための設定が完了し、担当者に前提知識が備わっている状態となった。
その上で、SBOM を定期的に出力するための運用に必要となる作業時間を計測した。

表 4-144　SBOM の運用コスト

作業ステップ	作業の内容	作業時間
Yamory	SBOM を Web 画面からダウンロードする必要があるため、必要な時に画面にアクセスし、手動でダウンロードする作業が必要となる。	0.5 時間
Mend	バッチ出力が可能であるため、SBOM を定期的に生成するコストはほぼゼロに近い。	0 時間

上表に示したとおり、Yamory では SBOM を Web 画面からダウンロードする必要があるため、SBOM を定期的に出力するためには SBOM 出力画面にアクセスし、手動でダウンロードを行う必要がある。

一方 Mend では Mend SBOM Generator による SBOM 出力が可能であり、本ツールはバッチ的な実行が可能であることから、SBOM 出力自動化に関する環境整備が事前になされていれば、SBOM の定期的な出力コストはほぼゼロに近い結果となった。

4.8.6.2.2. 課題、解決ノウハウ、留意点

SBOM を継続的に生成、共有する上での課題は三つ挙がった。一つめの課題は「パッケージマネージャの必要性」である。弊社で利用している「ColdFusion」の開発環境では、パッケージマネージャが用意されていない。そのため、ソフトウェア部品情報の抽出にパッケージマネージャが必須となるツールでは、部品情報を抽出することができなかった。

一方 Mend は抽出ツールである Unified Agent がパッケージマネージャに依存せず、ソースコード分析による部品抽出が可能であるため、弊社環境でのソフトウェア部品を検出することが可能であった。

表 4-145　検証に利用したツールと SBOM 出力可否

製品名	Yamory	Mend
弊社での SBOM 出力	不可	可
SBOM 出力可否の理由	ソフトウェア部品情報の抽出にパッケージマネージャが必須となるため（弊社環境ではパッケージマネージャが存在しない）	抽出ツールである Unified Agent がパッケージマネージャに依存せず、ソースコード分析による部品抽出が可能であった。
部品の再帰解析	同上	再帰分析自体は可能であったが、部品が OSS と認識された場合はそれ以上の解析を行わずにそこで再帰が中止されるため、実際に抽出を確認できたのは直接利用部品と同じレベルまでであった。

現在、多くの場合は、開発環境にパッケージマネージャが存在していると考えられるが、弊社のようにパッケージマネージャを利用しない環境も一定数存在すると思われる。また、Mend でもソースコードからの直接利用部品の解析は可能であるものの、再帰的に含まれる部品の抽出を行うことは不可であった。

二つめの課題は「間接利用部品の検出工数の算定」である。間接利用部品を検出する方法として、OSS のライブラリを全てダウンロードし、別途 Mend による SBOM 生成を実施する方法が考えられるため、OSS のリポジトリーを取得し、Mend Unified Agent による解析と SBOM 出力を行うための工数を算出した。
検証の結果、必要な工数は 1.5 時間程度であった。

表 4-146　OSS ライブラリ毎の間接利用部品の検出工数

作業ステップ	作業時間
Mend による構成解析	0.5 時間
Mend SBOM Generator による SBOM 出力	0.5 時間
SBOM 出力の自動化に関する設定	0.5 時間

つまり、OSS ライブラリの間接利用部品を洗い出すための初期設定には、弊社の場合で部品数 22×1.5 時間 = 33 時間が必要となる。

また、コンポーネントのバージョンが上がった際に、利用するコンポーネントが増えた場合のメンテナンスも必要となるため、初期コスト以外にもコンポーネントにアップデートがある度に継続的に工数が必要となるところが大きな課題と考える。

これら再利用部品の管理工数は実現不能なほど大きいわけではないが、再利用部品を多く含むメジャーなライブラリでは、脆弱性が発見された場合は開発者コミュニティにより比較的迅速に修正が行われるため、弊社のように部品数が少ない場合は、かけるコストに見合った効果を得ることが難しいと考えられる。

なお、Yamory の運用会社である株式会社アシュアード様からは、本件について「Yamory は将来的にもソースコード解析を行うことは検討しておらず、利用しているソフトウェアを手動で登録できるようにすることを検討している」との回答を得ている。

このように、自動化されたアプローチだけでソフトウェア情報を解析できない領域については、手動での管理を交えながら部品の管理を行っていくのも有効な手法と考えられる。

三つめの課題は「サードパーティ製ミドルウェアの SBOM 入手」である。実証の対象システムには、Adobe 株式会社が提供するミドルウェア「ColdFusion[82]」が含まれるため、ColdFusion の SBOM を Adobe 社から入手可能かについて確認を行った。

なお、ColdFusion の日本での総販売代理店は株式会社サムライズ[83]であり、サポートなどの問い合わせ窓口も同社に集約されている、そのため、本実証では同社に対し SBOM の入手に関する問い合わせを行った。

その結果、以下の点で SBOM の提供が不可となるとの回答を得た。

- Coldfusion の Program Manager にコンタクトしたが、Program Manager 自身に「SBOM」に関する認識が無かった。
- SBOM の概要を説明することで理解が得られ、同社のセキュリティチーム及び法務チームに確認を行ってもらったが、双方ともに特に SBOM に関する認識が無かった。
- 結果として、現状提供可能な ColdFusion の SBOM は存在していないことが判明した。

なお、上記作業の工数（現状提供可能な ColdFusion の SBOM は存在していないことが判明するに至るまでの、各種連絡・折衝工数）は、4/人時間であった。

また、Adobe 社で公開しているサードパーティソフトウェアに関する通知及び/追加条件[84]は、対象ソフトウェアに含まれるサードパーティライブラリを推測するのに利用できる。ただし本ドキュメントはライセンス記載義務に準拠するために作成されたもので、全てのライブラリが網羅されているわけではないため、利用の際には注意が必要である。

Adobe 社はクリエイティブ・デザイン、ビデオ編集ツール、AcrobatPDF 等で市場でもトップシェアを持つ著名な企業であるが、同社のプロダクトの中でもマイナーな存在である ColdFusion が対象だったとは言え、提供可能な SBOM が存在しないことを考えると、現時

---

82 https://www.adobe.com/jp/products/coldfusion-family.html
83 https://www.samuraiz.co.jp/index.html
84 サードパーティソフトウェアに関する通知及び/追加条件

点で商用のサードパーティ製品の SBOM を安定的に入手するのは困難ではないかと考えられる。

4.8.6.3. SBOM の活用・管理

4.8.6.3.1. 実施内容とコスト評価

Mend を使用した場合、SBOM の活用・管理には 1OSS 当たり合計で 0.5075 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりである。

表 4-147　Mend を使用時の SBOM 活用・管理コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
脆弱性管理工数	0.5	人時間/OSS	0.5	人時間/OSS		人時間/OSS	対象の部品数：22 点 詳細は以下に記載 ・4.8.6.4.効果評価「脆弱性の特定」深刻度・悪用可能性評価 ・脆弱性の確認は Mend のダッシュボードで状況を確認するのみとなるので、0 に近い。
ライセンス管理工数	0.0075	人時間/OSS	0.0075	人時間/OSS		人時間/OSS	・ライセンス特定は Mend のダッシュボードで状況を確認するのみとなるので、0 に近い。 ・全 22 点のライセンス違反チェックに必要な工数は 0.166h 程度と想定。
（活用ツール費用）	3,360,000	円		円		円	ライセンス体系(プラン)及び開発者数に応じた見積。費用は年額（2022 年 12 月現在）

表 4-148　Mend を使用時の SBOM 活用・管理コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
脆弱性管理工数	脆弱性特定工数	0	人時間/OSS	0	人時間/OSS		人時間/OSS	対象の部品数：22 点 Mend のダッシュボードで状況を確認するのみとなるので、0 に近い。
	脆弱性対応工数	0.5	人時間/OSS	0.5	人時間/OSS		人時間/OSS	詳細は以下に記載 ・4.8.6.4.効果評価「脆弱性の特定」深刻度・悪用可能性評価
ライセンス管理工数	ライセンス特定工数	0	人時間/OSS	0	人時間/OSS		人時間/OSS	対象の部品数：22 点 ライセンス特定は Mend のダッシュボードで状況を確認するのみとなるので、0 に近い。
	ライセンス違反対応工数	0.0075	人時間/OSS	0.0075	人時間/OSS		人時間/OSS	全 22 点のライセンス違反チェックに必要な工数は 0.166h 程度と想定。
（活用ツール費用）	（活用ツール費用）	3,360,000	円		円		円	ライセンス体系(プラン)及び開発者数に応じた見積。費用は年額（2022 年 12 月現在）

また、Yamory を使用した場合、SBOM の活用・管理には 10SS 当たり合計で 0.5075 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりである。

表 4-149　Yamory を使用時の SBOM 活用・管理コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
脆弱性管理工数	0.5	人時間/OSS	0.5	人時間/OSS		人時間/OSS	詳細は以下に記載 4.8.6.4.効果評価「脆弱性の特定」「深刻度・悪用可能性評価」 ・対象の部品数：22 点 脆弱性の確認は Yamory のダッシュボードで状況を確認するのみとなるので、0 に近い。
ライセンス管理工数	0.0075	人時間/OSS	0.0075	人時間/OSS		人時間/OSS	対象の部品数：22 点 詳細は以下に記載。 4.8.6.4.効果評価「ライセンス違反の特定」 ・ライセンス特定は Yamory のダッシュボードで状況を確認するのみとなるので、0 に近い ・全 22 点のライセンス違反チェックに必要な工数は 0.166h 程度と想定。
（活用ツール費用）	2,000,000	円		円		円	ライセンス体系(プラン)及び開発者数に応じた見積。費用は年額 180 万円に初期費用 20 万円を含んだもの（2022 年 12 月現在）

表 4-150　Yamory を使用時の SBOM 活用・管理コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
脆弱性管理工数	脆弱性特定工数	0	人時間/OSS	0	人時間/OSS		人時間/OSS	対象の部品数：22 点 ・脆弱性の確認は Yamory のダッシュボードで状況を確認するのみとなるので、0 に近い。
	脆弱性対応工数	0.5	人時間/OSS	0.5	人時間/OSS		人時間/OSS	詳細は以下に記載 4.8.6.4.効果評価「脆弱性の特定」深刻度・悪用可能性評価
ライセンス管理工数	ライセンス特定工数	0	人時間/OSS	0	人時間/OSS		人時間/OSS	対象の部品数：22 点 詳細は以下に記載。 4.8.6.4.効果評価「ライセンス違反の特定」 ・ライセンス特定は Yamory のダッシュボードで状況を確認するのみとなるので、0 に近い
	ライセンス違反対応工数	0.0075	人時間/OSS	0.0075	人時間/OSS		人時間/OSS	・全 22 点のライセンス違反チェックに必要な工数は 0.166h 程度と想定。
（活用ツール費用）	（活用ツール費用）	2,000,000	円		円		円	ライセンス体系(プラン)及び開発者数に応じた見積。費用は年額 180 万円に初期費用 20 万円を含んだもの（2022 年 12 月現在）

SBOM 手動生成の場合、SBOM の活用・管理には 1OSS 当たり合計で 0.7246 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりである。

表 4-151　SBOM 手動生成時の SBOM 活用・管理コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
脆弱性管理工数	0.6496	人時間/OSS	0.6496	人時間/OSS		人時間/OSS	詳細は以下に記載 4.8.6.4.効果評価「脆弱性の特定」「深刻度・悪用可能性評価」
ライセンス管理工数	0.165	人時間/OSS	0.165	人時間/OSS		人時間/OSS	詳細は以下に記載 4.8.6.4.効果評価「ライセンス違反の特定」 ・全 22 点のライセンス違反チェックに必要な工数は 0.165h 程度と想定。
（活用ツール費用）	0	円		円		円	無し

表 4-152　SBOM 手動生成時の SBOM 活用・管理コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
脆弱性管理工数	脆弱性特定工数	0.136	人時間/OSS	0.0136	人時間/OSS		人時間/OSS	詳細は以下に記載 4.8.6.4.効果評価「脆弱性の特定」
	脆弱性対応工数	0.5136	人時間/OSS	0.5136	人時間/OSS		人時間/OSS	詳細は以下に記載 4.8.6.4.効果評価「深刻度・悪用可能性評価」
ライセンス	ライセンス特定	0.09	人時間/OSS	0.09	人時間/OSS		人時間/OSS	詳細は以下に記載

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
管理工数	工数							4.8.6.4.効果評価「ライセンス違反の特定」
	ライセンス違反対応工数	0.075	人時間/OSS	0.0075	人時間/OSS		人時間/OSS	・全 22 点のライセンス違反チェックに必要な工数は 0.166h 程度と想定。
（活用ツール費用）	（活用ツール費用）	0	円		円		円	無し

4.8.6.3.2. 課題、解決ノウハウ、留意点

SBOM 活用・管理における課題、解決ノウハウ、留意点については、以下「4.8.6.4 効果評価」の作業毎に「評価・課題・解決ノウハウ・留意点」として示した。

4.8.6.4. 効果評価

本実証の結果、脆弱性対応期間の低減、脆弱性管理にかかるコストの低減については効果が確認された。

一方、脆弱性残留リスクの低減、ライセンス違反リスクの低減にかかるコストの低減についての効果は確認されなかった。それぞれの効果測定結果と、算出に使用された情報は下表のとおりである。

表 4-153　効果測定結果

分類	効果測定			
	項目	結果		
				（前提条件）
リスク低減	脆弱性残留リスクの低減	0	個多くの OSS につき脆弱性リスクの管理が可能	差は生じず
	脆弱性対応期間の低減	1	日間短縮可能	手動作成時の、ツール利用時との比較
	ライセンス違反リスクの低減	0	個多くの OSS につきライセンス違反リスクの管理が可能	差は生じず
コスト低減	脆弱性管理などにかかるコストの低減	804	人時間の工数削減効果を確認	手動作成時の、ツール利用時との比較

分類	効果測定		
	項目	結果	
			（前提条件）
			700 の部品（平均的な SW に含まれる OSS 数）を管理したと仮定した場合の値。 ツール費用は以下の理由で算入せず。 製品別に価格・料金体系が異なる 実証対象企業全社で使われるため、特定プロジェクトでのコスト按分が困難

表 4-154　効果の算出用情報一覧

分類	効果測定	効果算出用情報						
	項目	情報項目	ツール利用時		手動生成時		SBOM なしの場合	
			値	単位	値	単位	値	単位
リスク低減	脆弱性残留リスクの低減	OSS 特定数	22	個	22	個	—	
	脆弱性対応期間の低減	脆弱性発見から対応完了までの必要期間	1	日		日	1	日
		（小計：ケース別の、SBOM なしの場合からの短縮期間）	0	日	1	日	—	
	ライセンス違反リスクの低減	OSS 特定数	22	個	22	個	—	
コスト低減	脆弱性管理などにかかるコストの低減	初期導入工数	62	人時間	0	人時間	62	人時間
		1OSS 当たりの運用工数	0.6	人時間/OSS	0	人時間/OSS	1.06	人時間/OSS
		一般的な SW での平均 OSS 利用数	700	個	700	個	700	個
		（小計：ケース別の管理工数）	483	人時間	0	人時間	804	人時間
		（小計：SBOM なしの場合からの削減効果）	321	人時間	804	人時間	—	

なお、脆弱性管理などにかかるコストの低減効果の算出において、小計として算出した「ケース別の管理工数」は以下の計算式で算出している。

$$\textit{初期導入工数} + 1\textit{OSS当たりの運用工数} \times \textit{一般的なSWでの平均OSS数}$$

以下は製品に含まれる部品から脆弱性を特定する作業を行うためのコストの詳細である。

表 4-155　脆弱性特定作業 1 回毎のコスト

	手動作業時	ツール利用時
作業内容	スプレッドシートで管理されている部品表毎に、脆弱性が発見されていないかについて JVN iPedia 等の脆弱性データベースで確認を行う。	Mend のダッシュボードで検出された脆弱性を確認する。
コスト	0.3 時間	ゼロに近い

なお、本作業では作業 1 回毎に、部品数 22 点のチェックを実施している。部品毎に分解して工数算出することも考えられるが、チェック毎の時間のばらつきを考慮し、作業 1 回を報告単位とした。

以下に、本作業における評価・課題・解決ノウハウ・留意点等を示す。

- 脆弱性の特定頻度を上げることで、脆弱性発見のリードタイムは削減できるが、手動作業ではそれなりに作業コストがかかるため、頻度を上げづらい。Mend のような SBOM に対応したツールがあることで、脆弱性発見までのリードタイムの短縮と、作業漏れ、ミスによる残留リスクが低減される。
- 弊社での管理部品は少なく、手動作業時のコストは比較的低く抑えられているが、管理部品が多いソフトウェアではコストは増大するため、メリットはより大きくなると考えられる。
- 手動作業が必要となる場合、工数との兼ね合いから月 1 回程度の実施となるのが実情と考えられる。また、多くの場合は定期的なチェックの他に脆弱性に関するニュースをウォッチすることで、なるべく脆弱性感知の機会を増やしていると想定される。

以下は、特定された脆弱性の深刻度、及び悪用可能性を評価するための作業コストの詳細である。

表 4-156　特定された脆弱性の深刻度、及び悪用可能性を評価作業コスト

	手動作業時	ツール利用時
作業内容	スプレッドシートで管理されている部品表毎に、が発見されていないかについて JVN iPedia 等の脆弱性データベースで CVSS v3 による深刻度スコアと、ベンダーから提供されている対策についての確認を行い、自社管理用に記録する。 当該情報を元に、自社サービスにおける脆弱性への対策と、その悪用可能性を評価する。	Mend のダッシュボードで CVSS 情報、及びアドバイザリー情報を確認。 当該情報を元に、自社サービスにおける脆弱性への対策と、その悪用可能性を評価する。
情報収集に関するコスト	0.3 時間 / 作業 1 回 （全 22 点の部品全て）	ゼロに近い
脆弱性への対策、悪用可能性評価に関するコスト	0.5 時間 / 1 部品毎	0.5 時間 / 1 部品毎

なお、脆弱性への対策、悪用可能性評価に関するコストでは、部品毎に必要となる作業時間が多いため、一つの部品に関する作業時間を報告単位とした。

悪用可能性の評価では、CVSS v3 の各評価を参考に、自社サービスにおいて優先的に対策を実施しなければならない脆弱性を検討することを前提とした。

優先度は自社サービスにおける悪用可能性と修正の容易性（既にパッチが適用されているか否か等）を考慮して算出されるため、PSIRT チーム（現状 3 名）による合議による決定が必要となる想定である。

本作業のコストは 3 名による 1 時間の会議で 6 部品を評価可能であると想定し、1 部品につき 0.5 時間のコストがかかるものとした。

以下に、本作業における評価・課題・解決ノウハウ・留意点等を示す。

- 悪用可能性の調査頻度を上げることで、悪用可能性把握のリードタイムは削減できるが、手動作業ではそれなりに作業コストがかかるため、頻度を上げづらい。Mend のような SBOM に対応したツールがあることで、悪用可能性把握までのリードタイムの短縮と、作業漏れ、ミスによ

る残留リスクが低減される。

- 弊社での管理部品は少なく、手動作業時のコストは比較的低く抑えられているが、管理部品が多いソフトウェアではコストは増大するため、メリットはより大きくなると考えられる。
- 悪用可能性については、手動作業、ツール利用の双方で人力による悪用可能性評価が必要となるため、ツールによる時間短縮があったとしても毎日評価を実施することは困難である。工数との兼ね合いから考えて、手動作業、ツール利用のどちらも月 1 回程度の実施となるのが実情と考えられる。
- Mend では部品毎に発見された脆弱性と、その CVE ID 及び危険度ランク、CVSS 3 のスコア等が表示される。詳細では悪用可能性の指標等が表示され、適用すべきアップデート情報が提供されるため、手動作業時と比較して悪用可能性情報と対策情報の収集を効率化することが可能であり、有用性が高い。ただし悪用可能性の評価については、手動作業時でもツール利用時でも作業内容が変わらないため、作業時間は同じとなった。
- 今回の実証では VEX については検証を行っていないが、今後 VEX が広く普及すれば、本項での対象とした自社での悪用可能性評価結果を VEX 情報として配布し、取引先やユーザーが SBOM を元にした脆弱性、悪用可能性確認を行う際のコストを低減できると思われる。

以下は、利用している部品が、必要なライセンスに準拠しているかを確認するための作業である。

表 4-157　利用部品のライセンス準拠確認作業 1 回毎のコスト

	手動作業時	ツール利用時
作業内容	スプレッドシートで管理されている部品表毎に、それぞれの Web サイト等でライセンスを確認する。パッケージマネージャがあれば一括でライセンスチェックが可能なツールが提供されている場合もある。	Mend のダッシュボードでライセンスの状況を確認する。
情報収集に関するコスト	2 時間/ 作業 1 回	ゼロに近い

なお、本作業では作業 1 回毎に、部品数 22 点のチェックを実施している。部品毎に分解して工数算出することも考えられるが、チェック毎の時

間のばらつきを考慮し、作業 1 回を報告単位とした。

以下に、本作業における評価・課題・解決ノウハウ・留意点等を示す。

- ライセンスの管理に関しては、脆弱性の特定、深刻度・悪用可能性評価と比べ、チェックの頻度は非常に少なくてよい。これは部品のライセンスが変わる頻度が、ライブラリに脆弱性が発生する頻度よりも少ないからである。そのため、作業 1 回のコストの差は大きいものの、他の二つに比べて SBOM ツールを用いるメリットは少ない。
- 弊社ではソフトウェア開発の外部委託を行っていないが、外部委託を行った場合に予期せぬライブラリが使われていないか、ライセンスに違反していないかをチェックする必要性が高い企業では、SBOM ツールによるライセンスチェックはより効果的であると考えられる。

4.8.6.5. SBOM 適用範囲（SBOM 活用モデル）の改訂案

SBOM 適用範囲について、以下のとおり改定案を作成した。

表 4-158　SBOM 適用範囲(ソフトウェア)

SBOM 作成主体	作成範囲	検出・生成・検査手段	生成項目	活用法
(a1)自社	(b1)直接利用部品	(c1)手動で特定（構成管理情報利用）・ツールで生成	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記の一部のみ	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価

SBOM 作成主体	作成範囲	検出・生成・検査手段	生成項目	活用法
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
	(b2)間接利用部品	(c1)手動で特定（構成管理情報利用）・ツールで生成	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価

SBOM 作成主体	作成範囲	検出・生成・検査手段	生成項目	活用法
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
(a2)サプライヤー（開発委託先）取引契約あり	(b1)直接利用部品	(c1)手動で特定（構成管理情報利用）・ツールで生成	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価

SBOM 作成主体	作成範囲	検出・生成・検査手段	生成項目	活用法
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
	(b2)間接利用部品	(c1)手動で特定（構成管理情報利用）・ツールで生成	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価

SBOM 作成主体	作成範囲	検出・生成・検査手段	生成項目	活用法
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
(a3)サプライヤー（サードパーティ）取引契約なし	(b1)開発者自身	(c1)手動で特定（構成管理情報利用）・ツールで生成	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価

SBOM 作成主体	作成範囲	検出・生成・検査手段	生成項目	活用法
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
	(b2)開発者以外（調達者、利用者）	(c1)手動で特定（構成管理情報利用）・ツールで生成	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価

以下に適用範囲毎の当初仮説との比較結果を示す。

表 4-159　(a1)自社 >(b1)直接利用部品

検出・生成・検査手段	改定の理由
(c1)手動で特定（構成管理情報利用）・ツールで生成	変更なし
(c2)ツールで特定・生成・誤検知精査なし	Mend,Yamory 共に深刻度評価を CVSS で示す機能が利用可能であるため、脆弱性・ライセンスの特定が可能。 VEX 情報は利用できないため、悪用可能性・深刻度の評価は不可として、当初仮説とおりとなった。 ただし、d3 は d1,d2 を満たさないフォーマットを出力するのがむしろ難しく、出力するメリットもないと考えられるため「不要又は対象外」とした。
(c3)ツールで特定・生成・誤検知精査あり 【不要又は対象外】→【部分適用】	当初仮説では「不要又は対象外」となっていた。これは「マニフェスト解析ではほぼ誤検知はあり得ないため、検証の必要がない」との理由によるものだが、弊社環境ではマニフェストファイルを使った解析ができないことからソースコード解析を利用しており、この場合は一定の誤検知精査が必要となる。そのため、本項は「部分適用」とした。（誤検知精査のコストは精査をどこまで行うかで変わるが、例えば最初のみ全体チェックし、その後のチェックは大きな変更があった時のみとすればコストを抑えた運用も可能であるため、評価は「適用困難」ではなく「部分適用」と判断している。） d3 は d1,d2 を満たさないフォーマットを出力するのがむしろ難しく、出力するメリットもないと考えられるため「不要又は対象外」とした。

表 4-160　(a1)自社 >(b2)間接利用部品

検出・生成・検査手段	改定の理由
(c1)手動で特定（構成管理情報利用）・ツールで生成	変更なし
(c2)ツールで特定・生成・誤検知精査なし 【適用可能】→【適用困難】	間接利用部品については Mend を用いても抽出することができない。また「4.8.6.2.2 課題、解決ノウハウ、留意点」二つめの課題「間接利用部品の検出工数の算定」で示した

	とおり、個別の OSS リポジトリーを手動取り込みし、継続的にチェックし続けるのは困難である。
(c3)ツールで特定・生成・誤検知精査あり	同上

表 4-161　(a2)サプライヤー（開発委託先）取引契約あり ＞(b1)直接利用部品

検出・生成・検査手段	改定の理由
(c1)手動で特定（構成管理情報利用）・ツールで生成 【部分適用】→【適用可能】	ソフトウェア業界では自社製品をサポートし続ける必要がある関係上、委託先からの納品形式をバイナリ納品とするケースは少なく、ソースコード納品とするケースがほとんどであると想定される。 ソースコード納品であれば自社で Mend 等のツールを利用することで部品の管理が可能であり(a2)と(a1)には実質的な差はないと考えられるため、(a2)と(a1)は全て同じ評価とした。 ※弊社ではソフトウェアの開発委託を行っていないが、業界では取引契約委託先があるベンダーも多いと想定し、取引契約がある場合を想定しての評価を行った。
(c2)ツールで特定・生成・誤検知精査なし 【部分適用】→【適用可能】	同上
(c3)ツールで特定・生成・誤検知精査あり 【不要又は対象外】→【部分適用】	同上

表 4-162　(a2)サプライヤー（開発委託先）取引契約あり ＞(b2)間接利用部品

検出・生成・検査手段	改定の理由
(c1)手動で特定（構成管理情報利用）・ツールで生成	変更なし
(c2)ツールで特定・生成・誤検知精査なし 【部分適用】→【適用困難】	ソフトウェア業界では自社製品をサポートし続ける必要がある関係上、委託先からの納品形式をバイナリ納品とするケースは少なく、ソースコード納品とするケースがほとんどであると想定される。 ソースコード納品であれば自社で Mend 等のツールを利用することで部品の管理が可能

	であり(a2)と(a1)には実質的な差はないと考えられるため、(a2)と(a1)は全て同じ評価とした。 ※弊社ではソフトウェアの開発委託を行っていないが、業界では取引契約委託先があるベンダーも多いと想定し、取引契約がある場合を想定しての評価を行った。
(c3)ツールで特定・生成・誤検知精査あり 【部分適用】→【適用困難】	同上

表 4-163　(a3)サプライヤー（サードパーティ）取引契約なし ＞(b1)開発者自身

検出・生成・検査手段	改定の理由
(c1)手動で特定（構成管理情報利用）・ツールで生成 【適用困難】→【不要又は対象外】	サードパーティの提供するミドルウェア等を利用する場合（弊社の場合は Adobe 社の ColdFusion）、サードパーティベンダーがどのような SBOM の管理を行っているかは知る余地がないため、(a3)(b1)は全て「不要又は対象外」とした。
(c2)ツールで特定・生成・誤検知精査なし 【適用困難】→【不要又は対象外】	同上
(c3)ツールで特定・生成・誤検知精査あり 【適用困難】→【不要又は対象外】	同上

表 4-164　(a3)サプライヤー（サードパーティ）取引契約なし ＞(b2)開発者以外（調達者、利用者）

検出・生成・検査手段	改定の理由
(c1)手動で特定（構成管理情報利用）・ツールで生成	変更なし
(c2)ツールで特定・生成・誤検知精査なし 【適用困難】→【部分適用】	「4.8.6.2.2 課題、解決ノウハウ、留意点」三つめの課題「サードパーティ製ミドルウェアの SBOM 入手」で示したとおり、ベンダーからの SBOM 入手は不可であったが、Mend を用いた ColdFusion の SBOM 生成、及び脆弱性検出、悪用可能性評価、ライセンス確認は（直接利用部品については）可能であった。 評価としては「部分適用」ではなく「適用可能」とすることも考えられるが、直接利用部品は検出可能であるものの、間接利用部品の検出が困難であることを考慮して、「部

	分適用」とした。
(c3)ツールで特定・生成・誤検知精査あり 【適用困難】→【部分適用】	同上

4.8.6.6. 取引契約における要求条項等

ユーザー企業、開発委託先、サードパーティのそれぞれについて、SBOM に関連し契約に盛り込むべき要求条項があるかを以下のとおりまとめる。

ユーザー企業向けのライセンス契約などについては、現時点で SBOM に関連する規定を設ける必要はないと考える。理由は以下のとおり。

- SBOM はユーザー企業への認知は現時点ではあまり広がっておらず、SBOM が入手可能であることが企業の信頼を高め、販売が促進されることにはつながらないと考える。
- 弊社では経産省提供の「クラウドサービスレベルのチェックリスト」を元に、サービスの開発体制、クラウドの運用状況を 59 項目で示したセキュリティチェックシートを公開している。ユーザー企業はこれら全てを総合的に確認し評価するため、SBOM が入手可能であることだけが重要なわけではない。
- ユーザー企業が SBOM を入手したとしても、それを有効活用するためのツールが少なく、利用するためのハードルが高い。

開発委託先については、現在弊社では他社への開発委託を行っていないため、契約面で SBOM に関する規定変更は考えていない。また、今後開発を外部に委託することになった場合も、SPDX 等の SBOM フォーマットでの納品は必ずしも必要ではないと感じている。その理由は以下のとおり。

① ソフトウェア業界での開発の外部委託は、小さい単位で開発を進め、変更点を委託先が都度プルリクエストを出し、委託元が承認してマージする形で開発が行われることが多い。

② プルリクエストには変更点の解説があり、どのような変更が行われたかが説明される。また DIFF ツールによってソースの差分がチェック可能で、委託元ベンダーはそれを確認しながらソースコードレビューを行うため、サービスにどのような部品が利用されたかについては把握できている状態となる。（このような納品形態をとるのは、SaaS ではサービスが継続する限りソースコードを自社で保守し、成長させ続けるため、内容を完全に理解しておく必要があることと関連があると推定する。）

③ 現状、委託先が SBOM のツール、知識、ノウハウを持たないことが多く、委託先に SBOM 納品を依頼することが容易ではない（委託先に対して SBOM の必要性、ツールの利用方法、必要に応じて出力のサポートを行う必要があり、コストがかかる。）

④ SBOM 納品によって受け取った SBOM は何らかのツールにインポートして利用することが望ましいが、現状そのようなツールが一般的ではない。

⑤ ソフトウェア業界では納品をリポジトリーへのコミットとマージで受け付けた後、CI/CD のプロセスで自動的に SBOM が生成されるような仕組みを整備する（SBOM を自社生成する）ケースが増えると考えられる。そのため、ソフトウェア業界では開発委託先から SBOM を納品物として受領する形態だけでなく、SBOM を自社で生成する運用も多く採用されるようになると想定する。

ただし「4.8.7.2.今後の取組の方向性整理」の①で示すとおり、今後 SBOM の認知が広まり、委託先が SBOM 生成ツールの知識やノウハウを持つことが一般的になれば、開発者自身である委託先が SBOM を低コストで生成・共有できるようになり、サプライチェーン全体での部品管理がよりスムーズになることが期待される。

サードパーティベンダーについては、ライセンス規約に基づき同一のソフトウェアを多くのユーザーに提供しているため、一般的に個別契約を結ぶことはなく、SBOM 活用のために特別な契約を締結することは難しいことから、現状、SBOM を入手するための契約面での対策は考えていない。

4.8.7. 実証成果と課題の整理

SBOM の活用や SPDX 等の形式での納品を受けることで得られるメリットについてまとめる。

① 今回の実証ではサードパーティベンダーからミドルウェアの SBOM を入手することはできなかったが、将来的に入手可能になれば、脆弱性の特定が迅速になるメリットが得られる。（しかしながら後述の「4.8.6.2.2.課題、解決ノウハウ、留意点」三つめの課題で記載のとおり、迅速な脆弱性対策にはミドルウェアベンダー側での早期対応が必要となるため、SBOM が必須というわけではない。）

② 弊社では開発委託を行っておらず、委託する場合でも運用ポリシー上ソースコード納品が必須であるため、開発委託先から SPDX 等の形式での SBOM の提供を受けるメリットは少ない。ただ、ソースコード納品を受けられないような契約形態の場合は、自社での部品管理を正当に行い、製品のセキュリティを守るためにも、SBOM 納品を必ず受けられるような体制、契約条項の見直しを行うことが必須になると考える。

③ Mend や Yamory などの脆弱性検知、部品管理ツールは SBOM を出力可能であるというメリット以外にも、部品の検出を自動化し作業工数を大きく削減する、脆弱性発見のリードタイム短縮と検出漏れの低減できる等のメリットがあり、非常に有用である。

4.8.7.1. 検討すべき課題

今回の実証を通じて感じた、SBOM 普及のための課題は次のとおりである。

① 「4.8.6.2.2.課題、解決ノウハウ、留意点」の一つめの課題「パッケージマネージャの必要性」で示したとおり、パッケージマネージャを利用しない開発環境では、ツールによる部品抽出が困難となる。ツールによってはソースコード分析が可能であるが、ソースコード分析はパッケージマネージャによる分析よりも誤検知が発生する可能性が高く、その検証に一定のコストがかかることについて留意する必要がある。

② 「4.8.6.2.2.課題、解決ノウハウ、留意点」の一つめの課題「パッケージマネージャの必要性」で示したとおり、ソースコード分析による部品抽出では、間接利用部品の再帰的な検出は困難であった。このようなケースではライブラリの開発元から SBOM が入手可能となり、SBOM をインポート・管理可能なツールが普及することで、間接利用部品

を効果的に管理できるようになると期待される。

なお、ライブラリの開発元から SBOM が入手できない場合の対策として、利用者数が多くメジャーなライブラリを採用することが考えられる。メジャーなライブラリでは、間接利用部品に脆弱性が発見された場合にも、開発チームにより早期の脆弱性修正が行われる可能性が高いこと、また、間接利用部品が脆弱性を含む場合に、早期に脆弱性の存在がアナウンスされる可能性が高いため、脆弱性が発見された場合の安全性が高まる。

③ 「4.8.6.2.2.課題、解決ノウハウ、留意点」三つめの課題「サードパーティ製ミドルウェアの SBOM 入手」で示したとおり、今回の実証ではサードパーティベンダーからの SBOM が入手不可であった。

ミドルウェアベンダーからの SBOM が入手できないことはリスクではあるが、実際に致命的な脆弱性が発見された場合は、ミドルウェアベンダーは迅速に対応を実施する場合が多い。例えば log4j のケースではメーカーの Adobe からは翌営業日には脆弱性が含まれる旨の情報が開示され、設定などによる回避方法が公開された。また、その後 1~2 営業日で修正パッチも提供された。

従って、ミドルウェアを利用する際は提供ベンダーが継続してサポートや脆弱性の修正を提供できる体制を持っているかについて確認するべきである。

なお、SBOM によって部品特定ができていれば検出はより早くなるが、ミドルウェアの設定変更などの対策については、利用者側で勝手に動くのは実際には困難である（ミドルウェアの詳細まで把握できていないため、独自判断で変更を行うことでサービスを停止されてしまうなどのリスクがある）そのため、現実問題としてはミドルウェアの SBOM を入手していたとしても、ユーザー側ができることは限られる点に留意する必要がある。

④ 「4.8.6.6.取引契約における要求条項等」の「開発委託先」で示したとおり、ソフトウェア業界では委託先と密にコミュニケーションをとる開発スタイルを採用していることが多い。その状況化で SBOM 納品の有効性が、バイナリ納品などと比べて下がることについて考慮が必要である。

⑤ 「4.8.6.2.2.課題、解決ノウハウ、留意点」一つめの課題「パッケージマネージャの必要性」で示したとおり、パッケージマネージャを利用しない開発環境では、ツールによる部品抽出が困難となる。ツールによってはソースコード分析が可能であるが、ソースコード分析はパッケージマネージャによる分析よりも誤検知が発生する可能性が高く、その検証に一定のコストがかかることについて留意する必要がある。

⑥ ソフトウェアベンダーは多くの顧客に同一のソフトウェアを提供しているため、提供した SBOM が顧客以外の第三者に流出し、当該 SBOM が攻撃者に利用されるリスクがある。（例えば特定の部品に対するゼロデイ攻撃が可能となった場合に、SBOM が攻撃者の手にあることで、攻撃者が攻撃対象を容易に絞り込むことが可能となると想定される。）

また、実証の一環として、弊社サービスの利用ユーザーに対し、簡単な SBOM に関する認知調査を行った。以下にヒアリングの結果を示す。

表 4-165　SBOM に関する認知調査①

業種	流通業
SBOM に関する認知	なし
社員数	1000 名～2000 名規模
お客様が想定する SBOM の利用方法	脆弱性の管理のため、利用している部品について個別に台帳を作って管理している。コラボフローについては SaaS であるため不要かも知れないが、自社内の開発ソフトウェアに関しては、メーカーから SBOM を入手できるようになればよいと思う。
お客様が想定する課題、所見	無し

表 4-166　SBOM に関する認知調査②

業種	専門・技術サービス
社員数	2000 名～3000 名規模
SBOM に関する認知	あり 一部の製品でオープンソースを使用しており、部品管理が課題となったため、ソフトウェア脆弱性管理ツールとして Yamory を利用している。 ただし SBOM は出力していない。
お客様が想定する SBOM の利用方法	利用する予定なし
お客様が想定する課題、所見	SBOM という「ログ」が取れるだけではあまり意味がないと思っている。必要なのはそれによって「どう行動すべきなのか？」が分かること。 現状は SBOM だけもらっても一般的なユーザーは何に活用すればよいのかわからないであろうし、自分達としても活用方法が定まっていない。 今後 SBOM を活用するためのツールが一層増えることを期待している。

また、SAJ で OSS の DB 構築の要件定義（補助金事業）を行った際に、SBOM の議論などを実施している。本資料を参考資料として提出する（OSS の脆弱性情報管理に関する戦略策定報告書（FIX 版・表紙付き）.pdf）

4.8.7.2. 今後の取組の方向性整理

今後の取組の方向性について以下に示す。

① 「4.8.7.実証成果と課題の整理」の②で示したとおり、弊社では開発委託を行っていないことから、開発委託先からの SBOM 提供について現状取り組む課題はない。
また、今後開発委託を実施する際も「4.8.6.6 取引契約における要求条項等」の「開発委託先」で示したとおり、ただちに SBOM 納品が必要になるとは考えていない。

しかしながら、今後 SBOM の認知が広まり、委託先が SBOM 生成ツールの知識やノウハウを持つことが一般的になれば、開発者自身である委託先が SBOM を低コストで生成・共有できるようになり、サプライチェーン全体での部品管理がよりスムーズになることが期待される。

また、「開発委託先」の⑤で示したように、CI/CD によってソースコードをマージする際に委託元で SBOM の自動生成を行うケースも多くなると考えられる。その場合も生成主体は委託元になるものの、結果的にはサプライチェーン全体での部品管理の効率化は高まると期待される。

② 「4.8.7.実証成果と課題の整理」の③で示したとおり、Mend や Yamory 等のツールの利便性は高く、弊社の規模でも導入にはメリットがあると考えられる。ただし現状管理する部品数が少なく、手動での管理でも対応可能な範囲であるため、ツールのライセンス費用や保守費用を含めたコスト評価を実施し、導入を検討する。

また、開発体制や利用するミドルウェアを見直し、パッケージマネージャが利用可能な環境に開発をシフトすることで、よりツールの効果を得られるようになるため、開発体制の移行についても今後の検討とする。

③ 「4.8.6.6.取引契約における要求条項等」の「ユーザー企業」で示したとおり、SBOM のユーザー企業への認知がまだそれほど広がっていないことから、ユーザー向けへの SBOM 提供については、現状すぐに取り組む課題はない。

しかし今後 SBOM の認知度が高まり、SBOM を分析、活用するツールが広く浸透すれば、SBOM の利用価値が上がり、SBOM の提供を希望するユーザーも増えると考えられる。そのような状況になった場合は、適切なタイミングでユーザーに SBOM を提供することが企業としても望ましいと考える。

（上記状況となった場合、先行して開発委託先、販売店等の間では SBOM の共有を行っている可能性が高い。SBOM の配布コストは、対象を増やしても大きくは変わらないと考えられるため、配布対象にユーザーが加わっても配布コスト増加は大きな課題にはならないと想定している。）

④ 今回の実証ではサプライチェーンの関係者が「ユーザー」「委託先」「サードパーティベンダー」に限られていたが、製品の「販社」も SBOM の入手を希望する可能性があることを考慮すべきと考える。これは製品を販売する大手代理店が、製品の「販売責任」を負っており、log4j のように広く使われている部品に致命的な脆弱性が発見された際に、自分たちが販売した製品にその影響があるかについて、早期に把握したいと考えるケースが多いためである。（実際、log4j 脆弱性が発見された際には、早い時点で多くの販社から弊社製品への影響について問い合わせがあった。）

4.9. ソフトウェア分野（さくらインターネット）

4.9.1. 実証の狙い・重点ポイント

実証の狙い

システムに含まれるソフトウェアの脆弱性を迅速に把握し、セキュリティリスクが放置されることを未然に防ぐことはシステムの運用において優先度の高い項目である。しかしながら、システムは多様な機器から構成されており、規模に応じて機器数も増える。ひとくちに脆弱性の迅速な把握といっても通常は少なくないコストを払う必要があり、なおかつ抜けや漏れが発生しやすいという問題がある。逆にいえば、この迅速性・網羅性・効率性を向上させることができれば、脆弱性把握の品質を改善し、システムの安全性を向上させることができる。本実証では、自社のサービス基盤を対象に、SBOM の利用がこの脆弱性把握の品質改善につながるどうかを検証することが目的である。

重点ポイント

脆弱性管理の観点から見ると、SBOM とそれに対応する脆弱性情報を特定したものは、ある時点において脆弱性スキャンを行った記録であるとみなすことができ、当該時点での脆弱性の有無を証明する証跡にもなり得る。SBOM とはこうした記録・証跡がある程度共通化されたフォーマットであると考えた時、脆弱性管理の内容を他者に確認させる(あるいは自らが他者のそれを確認する)事が容易になることが期待できる。実際には、他者との SBOM の共有が必要となる状況は現時点で発生していないが、必要になるものと本実証では想定し、他者からの SBOM 入手及び自らによる SBOM 作成を実施する。その上でどのような用途や利点が考えうるかについて検討する。

4.9.2. 実証の手順

本実証では主に自社の事業、すなわち IaaS クラウドを始めとする各種ホスティングサービスの基幹ネットワークで使用されている既製のネットワーク機器(中間層ルータ)及びそれらの管理用サーバーを対象としている。この 2 つは以下の点で特性が異なる。

既製のネットワーク機器(中間層ルータ)…機器メーカーが製造した製品を販売代理店(販社)から購入して使用する。機器に含まれるソフトウェアは一部に OSS を利用している場合もあるが、大部分は機器メーカーが独自に開発したものであり、そのソースコードなどは購入者であっても見ることができない。そのため機器に含まれる脆弱性の全てをメーカーより先に察知することは困難で、購入者は原則としてメーカー側からの通知によって脆弱性について知ることとなる。この時、やり取りの仲立ちとなるのが販社であり、購入者は販社を通じて脆弱性を修正するためのパッチなどを受け取ることもある。このような事情があるため、SBOM の提供についても販社を通じてやり取りを行う。具体的には現時点で SBOM を提供可能であるかどうか、可能であればそのまま SBOM の提供を受け、不可能であれば提供の予定についてさらに問い合わせる。

実施の流れは以下のとおり。

1. 対象となる機器を選定する
2. 問い合わせ内容を作成する
3. 機器販社への問い合わせを行う

4. 問い合わせ結果について精査・検討する

基幹ネットワーク管理サーバー…各サービスとインターネットとを結ぶネットワークを基幹ネットワークと呼ぶ。この基幹ネットワークを管理するシステムの内、先述の中間層ルータなど機器の監視などを行うサーバーを本実証の対象とする。このサーバーは一般的な Linux サーバーであり、これに対してクラウドサービス型脆弱性管理ツール yamory を用いたホストスキャンを実施し、SBOM を生成する。生成された SBOM が目的に合致するか否かを検証する。

実施の流れは以下のとおり。

1. yamory について学習する
2. yamory の導入先を選定する
3. yamory の導入を行う
4. yamory によるホストスキャンを実施する
5. yamory 上で SBOM を生成する
6. 生成した SBOM を精査する
7. 各種の検討を行う
   - 7.1. SBOM の活用範囲について
   - 7.2. SBOM の管理・運用の責任主体について
   - 7.3. SBOM の管理・運用の方法について

4.9.3. 実証の対象システムの概要と構成

本実証では、IaaS を始めとするクラウドサービスの一部を対象とする。サービスはデータセンターに物理的な実体がある。以下は対象システムを含む基盤の全体像である。この内、対象システムは基幹ネットワークと呼ばれるデータセンター内ネットワークの一部で、具体的には「中間層ルータ」と「基幹ネットワーク管理システム」がこれにあたる。前者はデータセンター内のネットワークトラフィックを効率的にやり取りする役割を持つ機器で、商用既製品を用いている。後者は基幹ネットワーク内の機器を監視・管理するための機器で、一般的な Linux サーバーに監視などを行うための OSS を導入している。

SBOM実証 対象システム構成 (さくらインターネット株式会社)

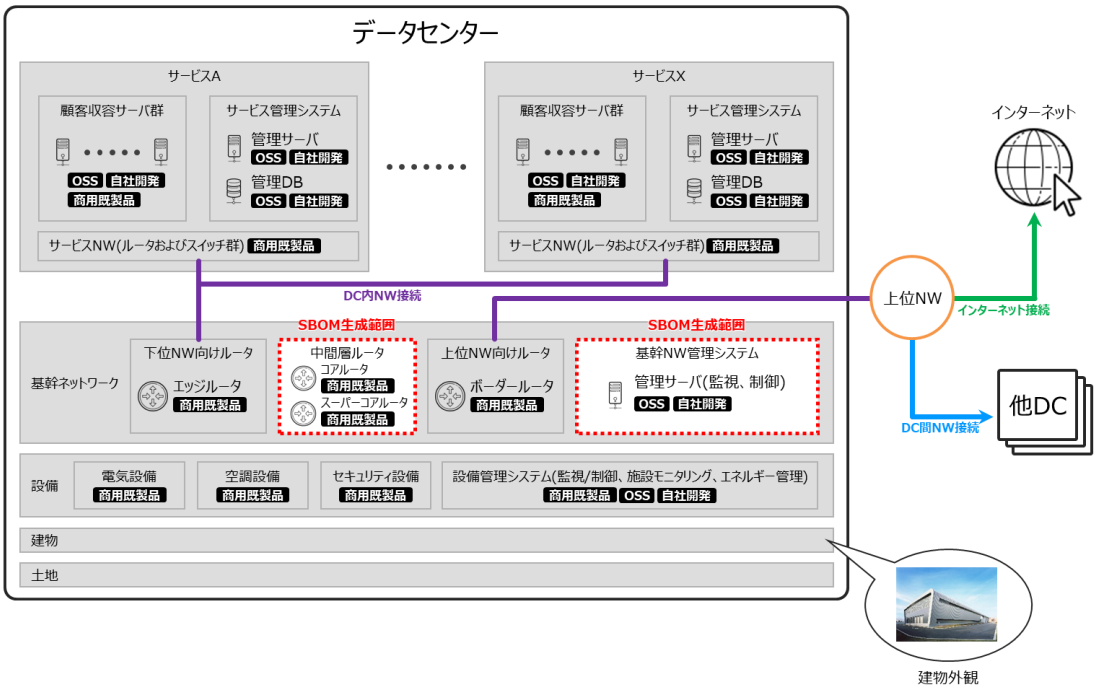


図 4-29　SBOM 実証 対象システム構成(さくらインターネット株式会社)

4.9.4. 対象分野の前提条件・取引慣行等

表 4-167　当社における前提条件及び取引慣行等

		本実証における形態（さくらインターネット）	業界において想定される主な形態を列挙
契約形態・取引慣行等	契約類型	既製品ライセンス契約など	個別のケースで異なる
	部品情報提供の要件	サプライヤーとの間：規定なし 顧客との間：　規定なし	同左
	第三者部品の申告	サプライヤーとの間：規定なし 顧客との間：　規定なし	同左
	脆弱性の通知・修正	既製品の場合、保守条件に基づく	同左
	納品形態	機器組込など	同左
	損害賠償責任	既製品の場合、保守条	同左

		本実証における形態（さくらインターネット）	業界において想定される主な形態を列挙
		件に基づく	
	知的財産権の帰属	既製品の場合、ベンダー保有・使用権のみ提供	個別のケースで異なる
開発形態等	開発言語	Java, Python, Go など	個別のケースで異なる
	部品形態	ミドルウェア、アプリケーション	個別のケースで異なる
	開発環境ツール	特になし	個別のケースで異なる
	ビルドツール	Java: Ant Go: Mage	個別のケースで異なる
	構成管理ツール	Ansible など	個別のケースで異なる

4.9.5. 実証項目の特定

本実証の実証項目は下記のとおりである。

表 4-168　本実証の実証項目

フェーズ	ソフトウェア分野　実証項目（案）
導入	・ツールの選定、インストール、利用方法習得の工数などのコストの評価、課題の特定
作成・提供	・サプライヤーが、自社開発部分の SBOM を生成することのフィージビリティを確認するため、生成コストの計測と課題を確認する。 ・CSP、サプライヤーの両者による計測を想定するが、サプライヤーの実証が難しい場合、CSP などが代替して計測する。
	・CSP、サプライヤーが、自身の開発部分の直接利用部品の SBOM を生成し、コストを評価する。 ・SBOM を作成できないサプライヤーや既製品から再帰的に利用される部品（間接利用部品）は CSP が有償ツール等を用いて一括生成し、コストを評価する。 ・直接利用部品については、無償ツールにより SBOM 生成の可否を検討することで、コストの低減が可能か評価する。 ・間接利用部品について、OSS などソースコードレビューによりツールで生成した SBOM の誤検知など精査し、コストを評価する。
	ツール選択肢と入出力を考慮し、妥当な標準形式を選定する。SPDX 等を想定し、CSP,サプライヤーで統一可能か検討する。
	ツールにより、大統領令最小要素をカバーできるか確認する。

フェーズ	ソフトウェア分野　実証項目（案）
	・ツールの SBOM 入出力の可否を確認する。 ・サプライチェーンでツールの共通化の可否、課題を検討する。 ・サプライチェーンでデータ形式が統一できない場合、ツールによるデータ変換の検討。 ・SBOM 受入側は、独自に SBOM 生成し比較するなど、SBOM の信頼性検査の要否を検討する。
運用・管理	・脆弱性特定、脆弱性深刻度評価(CVSS)、悪用可能性評価(VEX)、ライセンス特定を実施し、コスト、課題を評価する。 ・SBOM 誤検知など信頼性の誤差を許容し、脆弱性特定の絞り込み後に SBOM 精査を行うことでコスト低減できるか評価する。
	・OSS 含めライセンス契約に準拠した部品の申告は必要 ・契約条項として SBOM の要求が可能か考察。 ・最終製品ベンダーの部品情報、SBOM の責任について考察。
	CSP、サプライヤー、ユーザー企業について、脆弱性検出時の対処（通知、利用中断、修正など）を洗い出し、SBOM 有無によりリスク低減効果（対処時間の短縮など）を評価する。

4.9.6. 実証結果

4.9.6.1. SBOM 初期導入

yamory を使用した場合、SBOM 初期導入には合計で 10.5 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりである。

表 4-169　yamory を使用時の SBOM 初期導入コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
環境構築工数	6.5	人時間	6.5	人時間	-	人時間	
学習・体制構築工数	4.0	人時間	4.0	人時間	-	人時間	

表 4-170　yamory を使用時の SBOM 初期導入コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
環境構築工数	環境整備工数	6.5	人時間	6.5	人時間	-	人時間	yamory のユーザー登録、初期設定、導入先サーバー選定、導入先サーバー管理者との打合せ、導入手順の作成、yamory-cli の導入を実施
学習・体制構築工数	学習工数	1.0	人時間	1.0	人時間	-	人時間	アシュアード社による yamory 製品説明会の実施
	体制構築工数	3.0	人時間	3.0	人時間	-	人時間	社内協議の実施

SBOM 手動生成による実証は行っていない。

SBOM の初期導入については、yamory の導入を実施した。導入タスクの内訳と所要の工数は以下のとおりである。

表 4-171　導入タスクの内訳と所要工数

フェーズ	タスク	主担当者	サーバー管理者
yamory の導入	メーカーによる yamory 勉強会	1.0h	-
	ユーザー登録・初期設定	1.0h	-
	導入先サーバーの選定	1.0h	-
	導入先サーバー管理者との打合せ	1.0h	1.0h
	導入手順の作成	2.0h	-
yamory-cli の導入	yamory-cli の導入	-	0.5h
工数計		6.0h	1.5h

関わった人員の水準は以下のとおりである。

- 主担当者: 当社において中級レベルの技術者。
- サーバー管理者: 当社において上級レベルの技術者。

4.9.6.1.1. 実施内容とコスト評価

実証についてのタスクの内訳と所要工数は以下のとおりである。

表 4-172　実証におけるタスクの内訳と所要工数

実施内容	タスク	主担当者	調達担当者	サーバー管理者
機器販社への問い合わせ	調達担当者との打合せ	1.0h	1.0h	-
	質問内容の作成	-	2.0h	-
	販社 A とのやり取り	-	2.0h	-
	販社 B とのやり取り	-	3.0h	-
	問い合わせ内容の共有	0.5h	0.5h	-
ホストスキャンの実施	実施手順の作成	2.0h	-	--
	ホストスキャンの実施	-	-	0.5h
	ホストスキャンが実施されたかの確認	2.0h	-	-
ホストスキャン結果の確認	実施結果の精査	11.0h	-	-
	SBOM の生成	0.5h	-	-
	SBOM の精査	8.0h	-	-
合計		20.0h	8.5h	0.5h

4.9.6.1.2. 課題、解決ノウハウ、留意点

4.9.6.1.2.1. ツール選定について

本実証では、株式会社アシュアードが開発・提供するクラウド型脆弱性管理サービス「yamory」を選定した。ツールの選定については、SBOM を利用することが前提である以上 SBOM 出力に対応している点は必須だが、出力ファイルの形式やインポート機能の有無など、SBOM 関連機能の詳細については事前に確認することが望ましい。

SBOM の生成が可能な脆弱性管理スキャナについては OSS として無償で利用可能なものも含め複数あるが、開発元が本事業への協力に応じており、技術的なサポートや今後のロードマップについての詳細なやり取りが可能であること、本実証で行いたいホストスキャンの機能を備えていることが選定の主な理由となった。

4.9.6.1.2.2. 導入先の選定について

導入先は yamory が動作する環境であること、なるべく実運用に近い環境であることを主眼に検討を行った結果、基幹ネットワーク管理システムのサーバーが該当した。ただし、いわゆる本番環境への導入は不測の事態を引き起こしかねないことを考慮し、本番環境と同一のステージング環境に導入することとした。ステージング環境での実行結果は本番環境

でのそれと同一である。

4.9.6.2. SBOM の生成・共有

4.9.6.2.1. 実施内容とコスト評価

yamory を使用した場合、SBOM の生成・共有には 1OSS 当たり合計で 0.00058 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりである。

表 4-173　yamory を使用時の SBOM 生成・共有コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
情報取得工数	0.00058	人時間/OSS	0.00058	人時間/OSS	-	人時間/OSS	
精査工数	0.01277	人時間/OSS	0.01277	人時間/OSS	-	人時間/OSS	
（作成ツール費用）	0	円	0	円	-	円	

表 4-174　yamory を使用時の SBOM 生成・共有コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
情報取得工数	部品特定工数	0	人時間/OSS	0	人時間/OSS	-	人時間/OSS	対象 OS:Ubuntu 18.04 部品(OSS)数:861 所要時間: 0.5h yamory によるホストスキャン時に自動で SW 部品特定が行われるため、工数発生せず
	SBOM 生成工数	0.00058	人時間/OSS	0.00058	人時間/OSS	-	人時間/OSS	生成した SBOM 形式:d2
精査工数	精査工数	0.01277	人時間/OSS	0.01277	人時間/OSS	-	人時間/OSS	対象 OS:Ubuntu 18.04 部品(OSS)数:861 所要時間: 11.0h

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
（作成ツール費用）	（作成ツール費用）	0	円	0	円	-	円	試用ライセンスのため費用発生せず

なお、SBOM 手動生成による実証は行っていない。

4.9.6.2.1.1. yamory から生成した SBOM の内容

初期導入を行った後 yamory のホストスキャンを実施した。その結果、比較的対応優先度の高い脆弱性が 1 件検出された。脆弱性について管理者と協議し、アップデート等の対策を講じることとした。

続いて SBOM の生成を行った。SBOM の生成は yamory の管理画面からダウンロードする形で行う。ダウンロードされた SBOM のファイル形式は CSV である。

4.9.6.2.1.2. SBOM の内容精査

生成した SBOM の情報は、ファイル名とファイル内容に分かれて存在する。

ファイル名は以下のような形式である。

20221116145132_user_software_list.csv

これを意味のある単位に分解すると、

20221116145132	_	user	_	software_list	.csv
日付と時刻	セパレータ	スキャンの実行ユーザー	セパレータ	固定値	拡張子

となる。

また、ファイルの内容は以下のような形式である(先頭の 2 行を例示)。

```
チーム,ホスト,OS,ソフトウェア,バージョン,識別子
"さくらインターネット(株)_SBOM 実証実験",isk-zab-test,"Ubuntu 18.04",accountsservice,0.6.45-1ubuntu1.3,"01GHZ3ZPHA07VE1WY2GN311QXF"
```

先頭行は見出し行であり、2 行目以降は見出しに応じた内容となっている。なお、識別子は yamory の内部形式である。

OSS の誤検知・検知漏れについても確認した。具体的には、出力された SBOM の情報と対象ホストのパッケージリストを突合した。その結果、誤検知・検知漏れや、OSS の情報に関する誤り・漏れはないことが確認できた。

4.9.6.2.1.3. SBOM の内容に関する開発元への問い合わせ内容

上記の SBOM について開発元のアシュアード社の担当者に対して大統領令 8 要素を満たしているかについて問い合わせた。その結果、下記の回答を得た。

●最小要件の SBOM データフィールドと yamory の対応状況
・サプライヤー → ホストスキャンの場合は情報として保持できていないため出力できていない。アプリライブラリスキャンの場合はパッケージリポジトリー情報を記述する。
・コンポーネント名 → コンポーネント名を記述
・コンポーネントのバージョン → バージョンを記述
・識別子 → yamory の識別子を記述
・依存関係 → ホストスキャンの場合は直接依存のみ。アプリライブラリスキャンの場合は直接依存か、関節依存かを記述。
・SBOM の作成者 → CSV ファイルのファイル名に作成者を記述
・SBOM 作成のタイムスタンプ → CSV ファイルのファイル名にタイムスタンプを記述

その上で、情報がファイル名とファイル内容に分散している点についてはファイル内容への統合すること、また SPDX や CycloneDX といったフォーマットへの対応を要望した。

4.9.6.2.2. 課題、解決ノウハウ、留意点

4.9.6.2.2.1. SBOM 作成主体について

今回の実証では、対象となったシステムの管理者に依頼する形で脆弱性スキャンと SBOM の作成を行った。当社の事業基盤であるデータセンターには 4.8.3.で示したように今回の対象以外にも多くのシステム・機器が存在し、それぞれの特性に精通した複数の管理者が担当している。それぞれのシステム・機器の稼働に与える影響を回避(あるいは最小化)しながら脆弱性スキャンや SBOM 作成を適切に行うには、管理者に委託することは理にかなっている。

しかしながら各管理者は委託された作業を実施するだけであり、実際には作業を委託する側が SBOM 作成の主体であるといえる。以下は当社における想定である。

SBOM 作成の主体は、日常的には脆弱性スキャンや SBOM の作成を適切な相手に依頼し、結果を集約・精査する。問題が発見された際には関係者に情報を共有し、必要な対処に着手する起点となる役割を担う。脆弱性には軽微なものから重大なものまで様々あるが、顧客への影響が重大な場合、社内関係者はそれぞれ下記のような対応を行う。

- 開発者/運用担当者: 問題を修正するための対応策や、稼働中のサービスやシステムのメンテナンスを検討・実施する。
- 社外的なアナウンスを行う担当者: 脆弱性それ自体やメンテナンスなどで顧客に影響が発生する場合、その告知内容や手段を検討・実施する。
- 調達担当者: 脆弱性が外部から調達した機器やシステムなどに起因する場合、調達先に対する問い合わせや修正の要請などを行う。
- 経営者: 事業運営や企業としてのレピュテーションにどのような影響があるかを検討し、対応について検討・実施する。

利益相反の可能性を排除するため、理想的には上に挙げたいずれの立場にも属することなく、独立した組織が主体を担うことが望ましい。

SBOM 作成の主体は、上記のような対応が速やかに行われるために迅速・正確な情報共有を行う。そのためには集約する脆弱性や SBOM の内容を正しく理解し、問題が発見された場合の影響範囲を正しく把握することが求められる。その前提として各組織において SBOM が必要とされる背景を含めた正しい知識が備わっており、SBOM への対応が事業に貢献するという意識が共有されている必要がある。この前提を揃えるには、日頃から社内教育・啓蒙を行うと共に、対応する業務フローの整備を行うといった地道な準備が求められる。

4.9.6.2.2.2. 対象範囲の決定

対象範囲の決定に当たっては、何を対象範囲とするか否かの基準が必要となる。SBOM が普及していない現時点では、理念として何を対象範囲に「すべき」かに重きを置くよりも、現実として何を対象範囲に「できる」かに重きを置く方が検討を進めやすい。以下では、当社の状況を例に取り、主に実現可能性を念頭に置いて対象範囲の決定基準について検討する。

まず、現時点でソースコードや構成情報にアクセスできるソフトウェアは、問題なくSBOM を生成可能といえる。具体的には、

1. 自社がソースコードや構成情報を保有するもの(例:自社開発のソフトウェア)
2. ソースコードや構成情報が公開されているもの(例:OSS)
3. 契約などによってソースコードや構成情報にアクセス可能なもの(例:開発を委託したソフトウェア)

である。既に見たように、ツールを用いて SBOM 作成を自動化することは工数の面で大きな利点があり、活用しない手はない。加えてツールによって発見された脆弱性を、該当するソフトウェアのアップデートによって修正することも、ツールの API 連携機能次第では自動化が可能である[85]。

逆にソースコードや構成情報にアクセスできないソフトウェアは自力で SBOM を生成できないため、ソフトウェアやソフトウェアを含む機器の提供元に SBOM の提供を受ける必要がある。システム基盤の一部として利用している外部のソフトウェア(有償 OS やパッケージソフトウェア)やクラウドサービス(SaaS や PaaS など)はこれに該当する。

当社において重要といえるのは、保有しているデータセンターを始めとする物理基盤である。ソースコードや構成情報にアクセスできないソフトウェアが大量にあり、かつそれらに問題があった場合、事業に大きな影響が出る。以下に具体例としてデータセンター、サーバーについて述べる。

近代的なデータセンターでは、各種設備が基本的に電子制御、すなわち組み込まれたソフトウェアの働きによって動作している。典型的には、

電源設備 …分電盤などに制御ソフトウェアが組み込まれている。遠隔からの監視や制御に対応している場合もある。データセンター内に電力を供給している。

空調設備 …熱源機器、空気調和機などに制御ソフトウェアが組み込まれている。遠隔からの監視や制御に対応している場合もある。主にサーバー類を冷却している。

セキュリティ設備 …入場ゲートや電子錠システムに制御ソフトウェアが組み込まれている。許可された人間のみがデータセンターに立ち入れるようにしている。

監視設備 …各種計測を行うセンサーや監視カメラに制御ソフトウェアが組み込まれており、それらを集約・閲覧するためのソフトウェアもある。

ビル管理システム …上記の各種設備を遠隔から監視したり制御したりするソフトウェアである。

上記いずれの設備やシステムのごく一部であっても、脆弱性を突いた攻撃や悪意のあるソフトウェアが混入する[86]などして動作に異常をきたした場合、事業や顧客に大きな悪影響が出る可能性がある。大規模なデータセンターでは、個々の機器の総数は数万台にのぼる。脆弱性を悪用、あるいは悪意のあるソフトウェアが混入された場合の影響を考えると、電子的に制御される殆ど全ての設備について SBOM の生成・活用を実施「すべき」であるが、設備の供給側・使用者側双方に莫大なコストが発生するため現実的ではなく、SBOM が一般化するまでは相当な割合の設備を SBOM 生成・活用の対象外とせざるをえない。対策としては、仮に対象外となった設備に脆弱性が見つかる、ないし悪意のあるソフトウェアの混入が判明した場合にその設備を切り離したり、代替設備に切り替えたりといったことが比

[85] ただし、自動でのアップデートがシステムやアプリケーションの意図しない動作を引き起こさないよう考慮する必要がある(アップデート処理の前には必ず管理者による承認を経るようにするなど)。
[86] ネットワークに接続しているか否かに関わらず、一定時間経過後に破壊的な結果をもたらす動作をするよう予めプログラムされているようなものも含む。

較的容易に可能となるような設備設計や運用体制を整備することが考えられる。

また、システムの物理基盤であるサーバーにもソフトウェアが多く含まれるが、ソースコードや構成情報にアクセスすることが難しいと考えられる構成要素も多数存在する。サーバーに関していえば、

サーバー本体 …OS より下位レイヤーの制御ソフトウェア。UEFI, BIOS, BMC など。

ストレージ関連 …HDD, SSD などのディスク類や HBA/RAID カード、エキスパンダー等のファームウェア。

内蔵デバイス …NIC, GPU などのファームウェア。

などが該当する。これらも同様に深刻な悪影響の原因となり得る。大規模なデータセンターでは、個々の機器の総数は数万台にのぼる。ストレージや内蔵デバイスなどはサーバーに搭載されていても、サーバーとは別途調達したものを追加した場合には SBOM もそれぞれの調達経路に合わせて入手する必要がある。また、これらサーバーに含まれるソフトウェアはそれぞれのバージョンを固定して調達することが難しい場合も多く、同一機種であっても調達時期が異なれば搭載しているソフトウェアのバージョンが異なる、即ち SBOM に記載されるべき内容も異なるといったケースもある。加えて調達後に運用上の都合で自らがファームウェアを更新する場合もある。完璧を期すならば一台一台のサーバーごとに SBOM が作成・管理・更新されなければならない。

また、ソフトウェア構成の複雑性は比較的少ないとはいえ、ネットワーク機器にもサーバーと基本的に同様のことがいえる。

このように、大規模な物理基盤を SBOM の導入対象とした場合、SBOM の数が膨大にのぼることで、SBOM を提供する側(メーカーなど)・SBOM を受け取り管理する側(当社)ともに対応コストも膨大になる。この工数をいかに少なくできるかが当社のような組織においては重要であり、SBOM のデータ形式や項目、共有方法はこの点を最優先に検討する必要がある。

4.9.6.3. SBOM の活用・管理

4.9.6.3.1. 実施内容とコスト評価

yamory を使用した場合、SBOM の活用・管理には 1OSS 当たり合計で 0.00116 人時間の工数が発生した。実証項目別のコストの小計と内訳は下表のとおりである。

表 4-175　yamory を使用時の SBOM 活用・管理コスト小計

小計							
項目	結果						備考
	合計		ティア 1		ティア 2		
	値	単位	値	単位	値	単位	
脆弱性管理工数	0.00116	人時間/OSS	0.00116	人時間/OSS	-	人時間/OSS	
ライセンス管理工数	0	人時間/OSS	0	人時間/OSS	-	人時間/OSS	
（活用ツール費用）	0	円	0	円	-	円	

表 4-176　yamory を使用時の SBOM 活用・管理コスト内訳

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
脆弱性管理工数	脆弱性特定工数	0	人時間/OSS	0	人時間/OSS	-	人時間/OSS	対象 OS:Ubuntu 18.04 部品(OSS)数:861 yamory によって自動で脆弱性が特定されるため工数発生せず
	脆弱性対応工数	0.00116	人時間/OSS	0.00116	人時間/OSS	-	人時間/OSS	対象 OS:Ubuntu 18.04 部品(OSS)数:861 所要時間: 1.0h

小計	内訳							
項目	項目	結果						備考
		合計		ティア 1		ティア 2		
		値	単位	値	単位	値	単位	
								サーバー管理者に対して発見された脆弱性について報告し、対策について協議を実施
ライセンス管理工数	ライセンス特定工数	0	人時間/OSS	0	人時間/OSS	-	人時間/OSS	ライセンス管理を実施していないため、工数発生せず
	ライセンス違反対応工数	-	人時間/OSS	-	人時間/OSS	-	人時間/OSS	ライセンス管理を実施していないため、工数発生せず
（活用ツール費用）	（活用ツール費用）	0	円	0	円	-	円	試用ライセンスのため費用発生せず

なお、SBOM 手動生成による実証は行っていない。

脆弱性の特定に関しては、SBOM そのものを利用したわけではないが、主担当者がツール(yamory)上で脆弱性を確認し、サーバー管理者との打合せを実施。脆弱性の悪用可能性や深刻度を評価し、当該ソフトウェアのアップデートを行うこととした。なお、ライセンス管理関連の作業は実施していない。

4.9.6.3.1.1. SBOM 提供側としての具体的な想定

当社の場合、SBOM の提供は B2B, B2P での取引において NDA などの契約を結んだ上で開示する形が典型的であると考える。SBOM を広く公開することは ASM[87]として論外であるし、B2C での取引では、データセンターで稼働しているサービスの利用者たる個人がデータセンターの利用 OSS・脆弱性有無を把握することを求めるケースは実態上極めて稀であり、SBOM に対するニーズがほとんどないと想定される。

B2B で提供する場合、原則として個別契約に依存するが、ポリシーとして SBOM の対象範囲、引き渡しのタイミング、引き渡しの手段について定めておく必要がある。

SBOM の対象範囲 …基本的に顧客が利用するサービスに関連するものに限定するが、サービス単位か、あるいは顧客が関係する設備や機器の範囲にするかは要検討である。

引き渡しのタイミング …四半期に 1 回、年に 1 回のような形で定期的に更新するか、あるいは提供相手が任意のタイミングを指定できるようにするか、など。

引き渡しの手段 …ファイルを送りつけるような形式は、量が膨大になった際に非効率的である。リポジトリーのようなものを用意し、顧客は認証を通過した上で自らに関係のある SBOM をダウンロードできる形式が望ましいのではないか。

4.9.6.3.1.2. SBOM 要求側としての具体的な想定

当社の物理基盤には、脆弱性や悪意のあるソフトウェアが含まれていた場合、事業や顧客に対して深刻な事態になり得る設備・機材・システムが多く存在するため、可能な限り SBOM による管理を導入したい。

しかしながら現時点では、実証でもその一端を垣間見たように、調達先から SBOM が提供されず、提供される見通しも立たないケースの方が多いと見られる。これについては定期的に SBOM への対応を問い合わせるなどしながら普及を待つしかないと考える。

仮にあらゆる対象について SBOM が取得可能だったとしても、あまりにも対象が多いため、その精査や管理に対応するためのコストを捻出できないという別の問題もある。

期待としては、受注/発注や納品/検収、構築から運用に至るまでの各フェーズで SBOM のやり取りと脆弱性のチェックなどが一貫して自動で行われるようなシステムが登場することである。

4.9.6.3.2. 課題、解決ノウハウ、留意点

4.9.6.3.2.1. SBOM 作成の日常的な運用と自動化

今回行った yamory によるホストスキャンは、脆弱性管理の一環として日常的に行う運用

[87] Attack Surface Management

業務であるが、yamory の機能性によって「ついでに」SBOM の作成も行うことができる。仮にこの機能性がなく、SBOM 作成を独立した業務として行わなければならない場合の作業コストを省けると考えれば、非常に有用といえる。またコマンドラインツールであるため cron や Jenkins などを用いて定期的に自動実行することも容易である。この「ついでに」SBOM を作成し、さらに自動化することで、日常的な運用業務に SBOM 作成を組み込みつつ、運用コストを最小化することができる。

4.9.6.3.2.2. 実証を通じて明らかになった認識不足

実証前は、SBOM は専ら自組織内の脆弱性を特定・管理するための仕組みであるという理解だったが、サプライチェーンの中で自らも SBOM を提供し、提供先の脆弱性の特定・管理の一端を担う役割がある点に気付いた。SBOM を提供するからにはその品質を管理・改善する必要があり、そのことが自組織内の脆弱性特定・管理の品質を高めることにもつながると認識を改めた。

4.9.6.3.2.3. SBOM を時系列管理することの重要性

当社が顧客に IaaS クラウド等のサービスを提供するに当たり、提供期間は長い場合で 10 年以上に及ぶ。この間、顧客が使用するサーバーやネットワーク機器のソフトウェアアップデートを行うことは珍しくなく、その目的がソフトウェアの脆弱性修正である場合もある。この場合、顧客がサービスを利用し始めてから修正されるまでの間には脆弱性が存在していたことを意味しており、SBOM による脆弱性の追跡が一般に普及した際には「いつからいつまで脆弱性が存在していたか(脆弱性に晒されていたか)」を把握するニーズが高まるものと考える。

このニーズに応えるには、ソフトウェアの構成がアップデートによってどのように変化したかを履歴化しておく必要がある。そのためには対象の SBOM を最新のものだけでなく、少なくとも構成に変更が生じる度に記録・保存し、時系列で管理することが望ましい。
時系列で管理された SBOM が IaaS サービスの顧客に対して提供されていれば、IaaS の上に何らかの Web サービス(SaaS など)が稼働していた時、Web サービスの顧客に対して脆弱性がどのような影響をもたらしたか(あるいはもたらさなかったか)を説明することが可能となり、自社の信頼性を高めることにつながる。また、時系列の SBOM と外部情報を突合すれば、提供側が脆弱性をどの程度放置したか(あるいはしなかったか)が明確になり、サービスの利用側が提供側を評価する尺度のひとつとなる。

4.9.6.4. 効果評価

本実証におけるそれぞれの効果測定結果と、算出に使用された情報は下表のとおりである。今回、手動による SBOM 生成、及び SBOM なしの場合の工数測定は行っていないため、机上での想定数値である。

表 4-177　効果測定結果

分類	効果測定			
	項目	結果		
				（前提条件）
リスク低減	脆弱性残留リスクの低減	0	個多くの OSS につき脆弱性リスクの管理が可能	
	脆弱性対応期間の低減	0	日間短縮可能	
	ライセンス違反リスクの低減	0	個多くの OSS につきライセンス違反リスクの管理が可能	
コスト低減	脆弱性管理などにかかるコストの低減	0	人時間の工数削減効果を確認	
				本実証の範囲では、脆弱性管理に SBOM を活用するか否かは自動化とは関係がなく、コストの低減に影響しない。

表 4-178　効果の算出用情報一覧

分類	効果測定	効果算出用情報						
	項目	情報項目	ツール利用時		手動生成時		SBOM なしの場合	
			値	単位	値	単位	値	単位
リスク低減	脆弱性残留リスクの低減	OSS 特定数	861	個	861	個	—	個
	脆弱性対応期間の低減	脆弱性発見から対応完了までの必要期間	2	日	2	日	2	日
		（小計：ケース別の、SBOM なしの場合からの短縮期間）	0	日	0	日	—	
	ライセンス違反リスクの低減	OSS 特定数	0	個	0	個	—	
コスト低減	脆弱性管理などにかかるコストの低減	初期導入工数	10.5	人時間	80	人時間	10.5	人時間
		1OSS 当たりの運用工数	0.01451	人時間/OSS	27	人時間/OSS	0.01451	人時間/OSS
		一般的な SW での平均 OSS 利用数	700	個	700	個	700	個
		（小計：ケース別の管理工数）	20.657	人時間	18980	人時間	20.657	人時間
		（小計：SBOM なしの場合からの削減効果）	0	人時間	-18959.343	人時間	—	

なお、脆弱性管理などにかかるコストの低減効果の算出において、小計として算出した「ケース別の管理工数」は以下の計算式で算出している。

$$\text{初期導入工数} + 1\text{OSS当たりの運用工数} \times \text{一般的なSWでの平均OSS数}$$

基幹ネットワーク管理者を交えた検討

ネットワーク機器に関してはSBOMが提供されなかったため、この点については効果を評価できない。一方で管理サーバーについては導入パッケージの脆弱性スキャンを自動で行うことができ、その証跡としてのSBOMが自動生成されること、また継続的な実行についても自動化が容易であることは、運用コストの低減に有効であると考える。

また普段の運用としてSBOMを作成・管理することが、脆弱性に起因する攻撃などを受けたことで悪影響が発生した場合に、原因追求の一助となり得る。また、顧客やセキュリティ監査に対し脆弱性管理について説明する際にもSBOMを作成・管理していることがセキュリティに関する取組の健全性を示すことにつながる可能性がある。

4.9.6.5. SBOM 適用範囲（SBOM 活用モデル）の改訂案

以下は、本実証を踏まえた SBOM 適用範囲(SBOM 活用モデル)の改定案である。

まず、(a1)及び(a2)に関しては、適用が可能と思われる部分も基本的にはケースバイケースであり、(c2)を除いては「部分適用(事例依存)」としている。これは適用主体・作成範囲の如何に関わらず、対象となり得るシステム環境、開発環境、使用言語が多様であり、一概にツールを用いれば適用可能ということが難しいためである[88]。

(a3)に関しては、取引契約のないサプライヤーに対して SBOM の提供を義務付けたり、内容や形式について指定したりすることは、サプライチェーン全体に強制力のある行政上の、あるいはデファクトの制度などがない限りは困難であるため、「適用困難」とした。

凡例	
	適用可能
	部分適用（事例依存）
	適用困難
	不要又は対象外

表 4-179　本実証を踏まえた SBOM 適用範囲(SBOM 活用モデル)の改定案

適用主体	作成範囲	検出・生成・検査手段	生成項目	活用法
(a1)自社	(b1)直接利用部品	(c1)手動で特定（構成管理情報利用）・ツールで生成	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記の一部のみ	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価

[88] 別の言い方をすれば、一概に「適用困難」であるとも言えない。

適用主体	作成範囲	検出・生成・検査手段	生成項目	活用法
		(c2)ツールで特定・生成・誤検知精査なし	(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
	(b2)間接利用部品	(c1)手動で特定（構成管理情報利用）・ツールで生成	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価

適用主体	作成範囲	検出・生成・検査手段	生成項目	活用法
(a2)サプライヤー（開発委託先）取引契約あり	(b1)直接利用部品	(c1)手動で特定（構成管理情報利用）・ツールで生成	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
	(b2)間接利用部品	(c1)手動で特定（構成管理情報利用）・ツールで生成	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価

適用主体	作成範囲	検出・生成・検査手段	生成項目	活用法
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
(a3)サプライヤー（サードパーティ）取引契約なし	(b1)開発者自身	(c1)手動で特定（構成管理情報利用）・ツールで生成	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
	(b2)開発者以外（調達	(c1)手動で特定（構成管理情報利	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価

適用主体	作成範囲	検出・生成・検査手段	生成項目	活用法
	者、利用者）	用）・ツールで生成	(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット（SPDX、SPDXLite 等）	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定
				(e3)(e4)悪用可能性・深刻度の評価

4.9.6.5.1. 大口顧客とのコミュニケーションコスト削減

当社が SBOM を提供するケースとして考えられるのは、当社サービスの大口顧客に対してである。IaaS/PaaS/SaaS といったプラットフォームの障害や不具合が、その上でシステムを稼働させるユーザーの事業や風評を毀損することは現実に起きている。システムがプラットフォームに依存する度合いが高いユーザーにとっては、SBOM を用いたソフトウェア構成の精査や脆弱性の確認を行う動機が存在する。当社のようなプラットフォームの提供側からすれば、平常時より正しく SBOM を作成していれば、そのような顧客のニーズを素早く満たすことができ、かつそのコストも少なくて済むという利点がある。

4.9.6.6. 取引契約における要求条項等

4.9.6.6.1. SBOM の悪用に関する懸念と期待

SBOM は提供する場合とされる場合、いずれにおいても悪用が懸念される。前者では悪意ある取引相手が当社の提供した SBOM を用いて利用 OSS とその脆弱性を特定し攻撃を行う、後者ではサプライヤーが SBOM の内容を偽って脆弱性や悪意のあるソフトウェアを機材に混入させるといった行為が想定される。

企業・組織が SBOM をやり取りする際には、相手方との契約において SBOM の目的外利用の禁止、SBOM の正確性担保の責任、不正確な SBOM によって生じた損害に対する責任といった条項を設けるなどして、SBOM のサプライチェーンが機能するよう配慮する必要がある。しかしながら民間企業単独の努力では悪用を十分に防止できるとは断言できず、被害に対して泣き寝入りとならないためにも、重い刑事罰をもって対応するなどの法や制度が求められるのではないか。一方で、厳格な法や制度は SBOM の普及を妨げる要因にもなりかねないため、バランスをとる必要もある。今後の進展を注視していきたい。

4.9.6.6.2. フォーマットに関する期待

SBOM を提供される側としての将来像として、SBOM が普及し、あらゆる市販の機器について SBOM が取得可能となれば望ましいが、一方で

その精査・継続的な監視に人的コストが増大し続けることは得策ではない。これを機械的・自動的に行えるようにするため、どのような形式であれ使用されるフォーマットの種類は少ない方が望ましく、そのためには弊社独自のフォーマットを策定しそれに沿った SBOM を要求するといったことはせず、デファクトとなるフォーマットが定まり、それを採用することで円滑に SBOM が流通することを期待したい。

4.9.6.6.3. メーカーの対応が消極的だったことについて

今回の実証では、米国のネットワーク機器メーカー2 社に対して、国内販社を通じてネットワーク機器に関する SBOM の提供可否を問い合わせたが、いずれも現時点で提供は行っておらず、提供の具体的な予定もないとのことだった。SBOM への対応については米国企業の方がより進展しているものと期待していたため、この結果は遺憾であった。市販品を購入する際にはメーカーが SBOM の提供に対応している点を選定条件とすることには一考の価値があると考える。一方で価格や仕様面で折り合う機材が SBOM の提供に対応していないメーカーの製品だった場合、どのような対応が可能かについての検討が必要である。

4.10. 実証成果と課題の整理

実証分野の事例ごとの結果に基づき、実証分野・事例の横断的な分析と整理を行う。

4.10.1. 検討すべき課題

実証で抽出された課題に基づき、解決ノウハウの検討や今後の取組施策を整理すると以下のとおりである。

表 4-180　実証で抽出された課題に対する解決ノウハウ・今後の取組施策

区分	実証から抽出した課題	解決ノウハウ（導入ガイダンスに反映予定）	今後の課題（国、民間）	医療機器	自動車	ソフト
技術	検出した脆弱性の対応要否、優先度の判断が困難	医療機器分野の脆弱性対応フローなどを参考にアドバイザリー、脅威情報を活用し、対応の要否、優先度を判断。	脆弱性管理の高度化、脅威情報の普及促進	●		
	SBOM ツールの使い分け・変更による負担増大	機能ニーズを洗い出し、ツール比較情報をもとに選択	—	●	●	●
	CI/CD など継続的なアップデートへの対応負担	ツールによる自動化可能な範囲で管理	CI/CD に対応した自動管理			●
	SBOM 初期導入、ツール等のコスト負担が大きい	ツールの効率的な導入方法、OSS ツールの選択活用	OSS ベースのツール整備	●	●	●
管理	SBOM に要求される精査のレベルが不明確	SBOM 対応モデルの選択肢や SBOM ツールの機能に応じて精査の要否を判断する。	—	●	●	
	SBOM 生成の対象範囲が不明確	OS,MW を含めて対象全体の上位構成を事前に明確化	—	●	●	●
	ツールの環境構築、SBOM 共有のコストが大きい	SaaS 型 SBOM ツールで初期導入と共有の工数を低減	サプライチェーンを通じた脆弱性管理	●	●	●
	ユーザー組織による SBOM の活用・管理が困難	SBOM ツール導入、ベンダー支援の活用	—	●	●	●
	部品の脆弱性残存期間に応じたリスク評価	SBOM の履歴管理により脆弱性残存期間を特定	脆弱性の履歴評価			●

区分	実証から抽出した課題	解決ノウハウ（導入ガイダンスに反映予定）	今後の課題（国、民間）	医療機器	自動車	ソフト
	開発部署、PSIRT など部署ごとの脆弱性管理が非効率	社内で SBOM を一元管理することで、脆弱性管理を効率化	SBOM による脆弱性の社内一元管理		●	●
	サプライヤーごとの部品粒度のバラつき	取得した全ての粒度をツールで自動管理	脆弱性マッチングの高度化		●	
	サプライヤーのサポート切れなどのリスク対応	部品の EOL 等に基づくサポート計画・管理を実施	—	●		
取引	サードパーティからの SBOM 取得が困難、バイナリ納品物の脆弱性の監視・修正が負担	ソースコード取得と SBOM ツールの適用、（バイナリ納品の場合）SBOM 提供と脆弱性修正を契約で要件化	—	●	●	●
	サプライヤー部品の精査コストが大きい	SBOM の提供と信頼性に関する責任を契約で規定	—	●	●	

4.10.2. SBOM 対応モデル（案）の比較

SBOM 対応項目選択肢について、実証を通じて評価検討を行い改訂した部分及び未評価の仮説を含めて色分けにより整理したものが下図である。

医療機器分野

適用主体	作成範囲	生成方法	生成項目	活用法
製販業者	直接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記の一部のみ	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
	間接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
委託開発先	直接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記の一部のみ	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
	間接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			q	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
サードパーティ	開発者自身	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
	開発者以外(調達者、利用者)	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価

自動車分野

適用主体	作成範囲	生成方法	生成項目	活用法
ECUベンダ(ティア１)	直接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記の一部のみ	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
	間接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
開発委託先(ティア２等)	直接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
	間接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
サードパーティ	開発者自身	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
	(b2)開発者以外(調達者、利用者)	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)悪用可能性・深刻度の評価

ソフトウェア分野(トレンドマイクロ)

適用主体	作成範囲	生成方法	生成項目	活用法
最終ベンダー	直接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記の一部のみ	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
	間接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
開発委託先	直接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
	間接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
サードパーティ	開発者自身	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
	開発者以外(調達者、利用者)	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価

ソフトウェア分野(コラボスタイル)

適用主体	作成範囲	生成方法	生成項目	活用法
最終ベンダー	直接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記の一部のみ	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
	間接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
開発委託先	直接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
	間接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
サードパーティ	開発者自身	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
	開発者以外(調達者、利用者)	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価

ソフトウェア分野(さくらインターネット)

適用主体	作成範囲	生成方法	生成項目	活用法
最終ベンダー	直接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記の一部のみ	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
	間接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
開発委託先	直接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
	間接利用部品	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
サードパーティ	開発者自身	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
	開発者以外(調達者、利用者)	(c1)手動で特定(構成管理情報利用)・ツールで生成	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c2)ツールで特定・生成・誤検知精査なし	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
		(c3)ツールで特定・生成・誤検知精査あり	(d1)標準フォーマット(SPDX、SPDXLite等)	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d2)大統領令最小要素を含む	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価
			(d3)上記を満たさない要素	(e1)(e2)脆弱性・ライセンスの特定 (e3)(e4)事用可能性・深刻度の評価

いずれの分野も OSS を含むサードパーティからの SBOM 取得は困難であるため、委託開発企業又はツールでの対応が考えられる。

医療機器、自動車分野は、委託開発先も含めて SBOM 生成・共有し、ソフトウェア分野については、委託開発先は部分対応が考えられる。

医療機器分野は法的に構成管理を要件化されているため、間接利用部品を含めて説明責任を果たすことが考えられる。

OSS の間接利用部品の完全な特定は技術的な課題があるため、ソフトウェア分野については部分的な対応としている。

4.10.3. SBOM 取引モデル

4.10.3.1. SBOM 取引モデルの意義と位置付け（全体像）

SBOM のメリットは、サプライチェーンを通じて標準化された部品情報の共有と自動処理による効率化が挙げられる。特に SBOM を受け取る委託元の便益は大きいが、委託先はそのために追加負担を強いられることもあり、受発注者間で得られる便益が異なる。

そのようなことから、サプライチェーンを通じた SBOM の普及のためには、受発注者間で得られる便益に応じた対価負担の取決めが必要であり、委託契約において、SBOM の対応範囲とそれに対する対価負担、責任の明確化が必要である。

SBOM 取引モデルは、そのような SBOM に対する要求事項と派生する対価負担、責任関係について取り決めること項を示すものである。SBOM 取引モデルは各社ごとの契約書作成において参考となり、SBOM の効果的な利活用に資するものである。

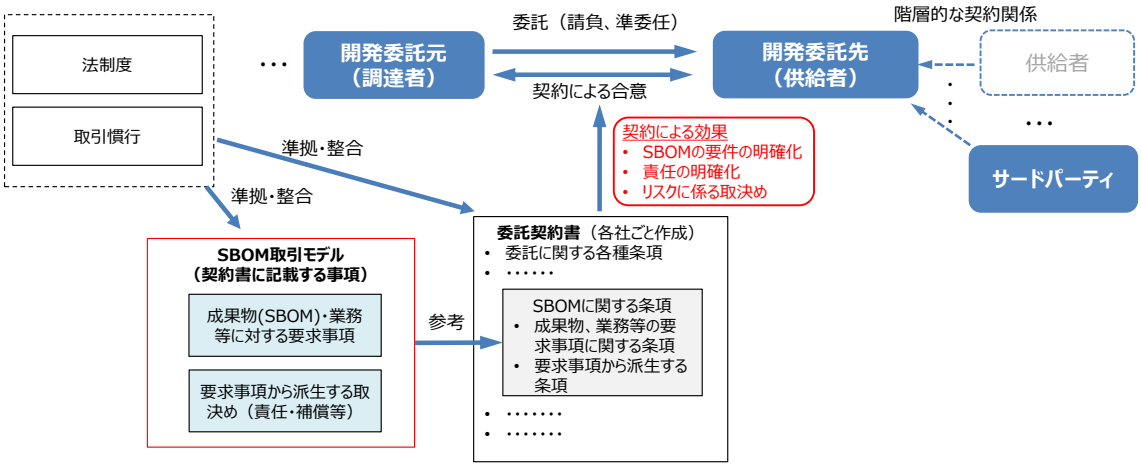


図 4-30　サプライチェーンにおける委託契約に基づく SBOM の普及促進

4.10.3.2. SBOM 取引モデル　～取引契約に規定すべき事項（案）～

分野ごとの前提条件（法制度、取引慣行、開発環境）に基づき取引契約で規定すべき事項を整理したものが下表である。

表 4-181　取引契約で規定すべき事項

要求レベル	SBOM に対する要求事項 (成果物・業務等)	要求事項から派生する取決め(責任、補償等)	想定分野例
基本事項 (最小限)	•利用するサードパーティ部品（OSS、商用部品）に関する受発注者間の事前合意と申告 •SBOM による部品情報の納品（対ユーザー、対サプライヤー） •SBOM のフォーマット、受け渡し方法の規定（ファイル、SaaS 等）	•ライセンス違反に関する責任･損害賠償責任 •最小限の SBOM 作成･活用の見積と対価負担の規定 •SBOM の知財権の帰属	全分野 （基本事項のみは、安全性や決済に関係のないソフトウェア等）
追加事項 (ハイレベル)	•悪用可能性、アドバイザリー等の情報(VEX)の要否 •納品・市販後の脆弱性情報監視・通知とその頻度 •納品・市販後の脆弱性の修正対応の要否、トリアージ •SBOM のカバー範囲と信頼性等の SBOM 対応レベルに関する要求と説明 •ソフトウェアのアップデートに伴う SBOM の更新とその頻度に関する規定 •サードパーティ部品の EOL、EOS や予定外の変更に対するサポート計画の要否	•納品した部品情報又は SBOM の瑕疵責任の規定 •納品した SBOM の瑕疵に伴う損害賠償の責任主体とその範囲 •SBOM 作成･活用の範囲に応じたコストの見積（受託開発者及びサードパーティ部品）	医療機器、自動車制御系、重要インフラ

現状で取引契約に記載される事例が存在するものは赤文字、分野固有で考慮すべき事項・条件は備考に記載した。

5. ソフトウェアの利活用に係るセキュリティリスク、課題及び対応策の検討

5.1. ソフトウェア利活用に係るセキュリティリスク・課題

ソフトウェアを利活用する際におけるセキュリティリスクについて、公開情報、ニュース記事、裁判例データベース等の調査により実施した。調査結果を基にソフトウェアの利活用における法的リスクに関し、セキュリティ及び著作権について専門性のある弁護士二者に対して有識者ヒアリングを実施した。なお、ヒアリング項目及び有識者の選定について、経済産業省との相談の上で決定した。ソフトウェアの利活用に係るセキュリティリスク、主として OSS における GPL ライセンスの OSS 利用による法的リスクに対応する取組動向、及び課題抽出を行った。

5.1.1. ソフトウェア利活用にかかる法的リスク

ソフトウェア利活用における法的なリスクとして著作権等の知的財産に関する違反リスクが挙げられる。近年、オープンソースソフトウェアは幅広い分野で活用されている。その一方でライセンスに起因するリスクが挙げられる。OSS ライセンスの中で GPL（General Public License）には、頒布時にソースコードの公開義務が課せられるものもある。主なライセンスの種類については、「平成 30 年 3 月一般財団法人ソフトウェア情報センター IoT 時代における OSS の利用と法的諸問題 Q&A 集」より抜粋し、以下表に示す。

表 5-1 主な OSS ライセンスの種類

OSS ライセンスの種類（例）	主要なライセンス条件					
	著作権表示義務	ライセンスの承継（同一条件で配布）	OSS（改変部分を含む）のソースコード提供義務	結合する他コードへの伝搬性（注1）	特許権の行使制限等（注2）	保証免責責任制限
GPLv3	有	有	有	有	必須特許許諾。差別的ライセンス禁止	無保証で配布する
GPLv2	有	有			明文なし	同上
LGPLv2.01（ライブラリ用）	有	有	有	無（注3）	明文なし	同上
Mozilla Public Licensev.1.1, v.2.0	有	有	有	無	ソース開示で権利不行使約束	同上

OSS ライセンスの種類（例）	主要なライセンス条件					
	著作権表示義務	ライセンスの承継（同一条件で配布）	OSS（改変部分を含む）のソースコード提供義務	結合する他コードへの伝搬性（注1）	特許権の行使制限等（注2）	保証免責責任制限
Apache Ver.2.0	有	有	無	無	必須特許許諾。特許ライセンス終了条件。	同上
修正版 BSD	有	有	無	無	明文なし	同上
MIT License	有	有	無	無	明文なし	同上

（注 1）「伝搬性」とは、ある OSS と一体化したソフトウェア全体（OSS の派生物）に対して、ソースコードの公開等、OSS ライセンス（許諾条件）が適用されることです。[89]（注 2）「特許権の行使制限」には、①OSS の貢献者（作者）や配布者が OSS を配布する相手 方（下流のユーザー）に対する「特許ライセンス義務」と、②OSS のライセンサーやコン トリビュータに対して特許権を行使すると、自己に対する当該 OSS に実施されている特 許ライセンスが自動的に消滅し又は解除される「特許ライセンス終了条件」があります。（注 3）LGPLv2.1 は、リンクによる自社開発プログラムへの伝搬性はありませんが、ユーザー自身の利用のための著作物の改変を許可し、その改変をデバッグするためのリバースエンジニアリングを許可しなければなりません（LGPLv2.1 第 6 条）。数字のパラメタやデータ 構造のレイアウト、アクセス機構又は小さなマクロや小さなインライン関数(長さが 10 行かそれ以下)のみ利用するならば、そのオブジェクトファイルの利用は制限されないと されています（LGPLv2.1 第 5 条）。

出所　平成 30 年 3 月一般財団法人ソフトウェア情報センター **IoT** 時代における **OSS** の利用と法的諸問題 **Q&A** 集 2．OSS 利用ポリシー（2）より表を抜粋引用

OSS ライセンスについては、主に GPL ラインセンスに関する問題が大きく、以下の 4 点が挙げられる。

- GPL 系における改変ソースコードの公表条件があること。
- 非 GPL 系コードにも、GPL 系と一緒に用いることによって、GPL が「伝搬」してしまい、当該ソースコードまで公表対象となるおそれがあること。
- 非 GPL 系コードと GPL 系コードとを、一緒に使えないケースがあること。
- オープンソースライセンスの場合、いわゆる免責条項が一般に付けられていること。

また、オープンソースライセンスの法的研究者がほとんどいないため、上記のようなラインセンスリスクについて正確に把握できていないため、「問題の所在」それ自体が理解されていないという問題もある。

企業が受ける損害として、ライセンス違反による損害賠償のほか、ソースコード開示義務

[89] 伝搬性について、独立行政法人情報処理推進機構『ビジネスユースにおけるオープンソースソフトウェアの法的リスクに関する調査』（平成 17 年 2 月）23 頁及び本書「D-3-1」以下参照。

による意図しない知的財産及び技術の流出、また訴訟による企業ブランドイメージの低下等が挙げられる。ライセンス侵害訴訟が起きた場合には、製品販売の停止、該当製品の回収等のおそれもあり、事業継続に支障が出る可能性がある。ソフトウェア開発においてはライセンス管理の不徹底による意図しない OSS の混入もあり得る。意図しない OSS の混入事例には、委託先開発した成果に OSS が含まれており、申告がなく OSS 混入に気づかないまま製品化し、後に問題が発覚する、Mergers and Acquisitions（M&A）後に被買収企業がライセンスに違反していたことが判明する等がある[90]。実際に、商流の中でプログラムの著作権の流れについて、複雑な構造を持つライセンスの場合は確認することが難しく、ライセンスの帰属先が明確になっていないと訴訟リスクとなり得る。ソフトウェア利活用においては、OSS を含むライセンス管理徹底のため、組織におけるコンプライアンス体制を整備し、ライセンス訴訟を未然に防止できるようリスクアセスメントし、事業展開を行う必要がある。効果的な OSS コンプライアンスのプロセスについて、OpenChain プロジェクトが OpenChain 仕様について検討を重ね、国際標準の規格化を進めてきた。「OpenChain 2.1」が 2020 年 12 月 16 日にオープンソースコンプライアンスの国際規格「ISO/IEC 5230:2020」が承認された[91]。既に日本企業においても既に正式認証を受けた企業もある[92]。こうした規格に準拠することにより、適切なビジネスプロセスを確立する一助となる。

5.1.1.1. ライセンス訴訟事例

ソフトウェア利用における OSS ラインセンス違反訴訟については、Software Freedom Law Center (SFLC)[93]、Software Freedom Conservancy（SFC）[94]等オープンソース活動を支援し、OSS の権利を守る非営利団体があり法定代理人となる訴訟事例がある。2007 年頃から SFLC は活発に訴訟を提起している[95]。SFLC が Software Freedom Conservancy（SFC）の代理として電子機器企業 14 社を相手取り、ソフトウェア「BusyBox」を含む製品において BusyBox の GPLv2 を侵害しているとして提訴した事例[96]が挙げられる。14 社の内一部の企業は、GPL 遵守（ソースコード開示）、責任者の任命、和解金の支払い等による条件で和解した。Westinghouse については、証拠開示要求に応じなかったため欠席判決となり敗訴した。SFC は 2010 年 7 月 27 日に Westinghouse に対する米ニューヨーク南地区地方裁判所の判決を公開している[97]。Westinghouse は、製品の販売停止命令、損害賠償金 9 万ドル及び訴訟費用 4 万 7 千ドル支払い命令受けた[98]。SFC によればこのデフォルト判決命令は、米国の裁判所が GPL 違反者に、準拠しない GPL ソフトウェアの配布を永久に停止することを命じる差止命令を出した初めての事例とされている。

日本企業においては Panasonic の関連会社で機内エンターテインメント（IFE）ソフトウェアを開発する Panasonic Avionics が 1 億ドル規模の訴訟に直面した事例がある[99]。2017 年 3

---

90 上山浩（2018）知財管理 Vol. 68　No. 5「OSS ライセンス遵守のための基礎知識」
https://www.hibiyapark.net/app/wp-content/uploads/2018/05/c484e7e1251d2ed577a08fc47179d1e4.pdf
91 https://www.openchainproject.org/featured/2020/12/15/openchain-2-1-is-iso5230
92 https://thinkit.co.jp/article/18061
93 https://softwarefreedom.org/
94 https://sfconservancy.org/
95 https://atmarkit.itmedia.co.jp/ait/articles/0812/08/news122.html
96 https://softwarefreedom.org/news/2009/dec/14/busybox-gpl-lawsuit/
97 https://sfconservancy.org/docs/2010-07-27_dj-opinion.pdf
98 https://mag.osdn.jp/10/08/05/1045202
99 https://runwaygirlnetwork.com/2017/03/press-release-ife-software-firm-cokinetic-sues-panasonic-avionics/

月、米国の CoKinetic Systems（以下、「CoKinetic」という。）は、Panasonic Avionics を相手取り、GPL 違反で 1 億ドルの賠償を求める訴訟を提起した。GPL 違反は、OSS 利用企業によるライセンス条件、規約等の確認が不徹底だったために起こる場合が多いが、利用企業の過失だけではなくライセンス違反における動機を追及される場合もある。例えば、この事例では CoKinetic がニューヨーク南部地区連邦地方裁判所に提出した訴状において「Panasonic Avionics は、競合他社が類似のソフトウェア開発を妨害する目的で故意に Linux ベースのソースコード開示を拒否した[100]」と見解を示し、裁判を求めた[101]。2018 年 1 月に両社は和解したが、損害賠償額等の和解条件の詳細は公開されていない。フリーソフトウェアファウンデーションのような組織が GPL ライセンスを分析してどこまでライセンスリスクが及ぶかというのが企業にとって論点になる。この論点について詳細を分析して啓発していく必要がある。

日本においても訴訟には至ってはいないものの GPL ライセンス違反について指摘を受けた事例があり[102]、以下表に示す。

表 5-2　日本におけるライセンス違反事例

違反した時期	企業	ライセンス違反概要
2002 年 12 月	株式会社 東芝	MP3 プレーヤーのファームウェアに GPL が適用されている Linux Kernel が含まれていると指摘され、GPL 違反が発覚した。その後、同社は該当部分（改変した Linux カーネル）のソースコードを公開した。
2002 年 9 月	エプソンコーワ（現エプソンアヴァシス）株式会社	スキャナプリンタ用の Linux 用のドライバーで、多言語対応のために gettext パッケージのソースコードの一部（libintl）を使用しており、このドライバーを無償配布していた[103]。エプソンコーワは指摘を受けた後、LGPL で提供される gettext のバージョンを使用するように変更し、LGPL に準拠するライセンスで 2002 年 10 月 18 日より配布を再開した[104]。
2007 年 11 月	ソニー株式会社	2001 年に発表されたソニーPS2 用のゲームソフト「ICO」が、GNU GPL(LGPL ではない)の下でライセンスされている libarc からコードを流用していたことがユーザーによって発見された[105]。指摘を受けた後、2008 年同ゲームソフトウェアの生産、販売を中止した。

また、OSS ライセンスに類似するクリエイティブコモンズ（CC）ライセンスに関する著

[100] https://wptavern.com/cokinetic-systems-pursues-100-million-gpl-license-violation-case-against-panasonic-avionics
[101] https://www.pacermonitor.com/public/case/20747763/CoKinetic_Systems,_Corp_v_Panasonic_Avionics_Corporation
[102] https://www.ossnews.jp/closeup/articles/?aid=201605-00005
[103] https://srad.jp/story/01/10/23/116247/
[104] http://www.cafebabe.jp/piracywiki/?%E4%BA%8B%E4%BE%8B%2F%E3%82%A8%E3%83%97%E3%82%BD%E3%83%B3%E3%82%B3%E3%83%BC%E3%83%AF%E3%81%AEGPL%E9%81%95%E5%8F%8D
[105] https://srad.jp/story/07/12/02/1340209/

作権侵害の事案が 2 件ある。令和 3 年（ワ）第 5285 号　著作権侵害差止等請求事件[106]、令和 3 年(ワ)第 21405 号 著作権侵害差止等請求事件[107]がある。OSS ライセンスと同じ構造をもっており、ライセンス放棄された範囲内で使用すれば問題ないが、それを超えた場合には著作権侵害になることに留意が必要となる。著作物について自由な利用、改変、再配布を認め、またそこから派生した著作物についても、同様に行為を制限してはならないコピーレフト[108]の考え方を適用したものと解釈できる。

5.1.1.2. ソフトウェア利活用におけるライセンスリスクの課題

セキュリティ対策においては、リバースエンジニアリングの手法が用いられる場合があるが、その際の著作権法の解釈、また SBOM 作成、利用においても知的財産の取り扱いについて、現行法令における解釈について有識者ヒアリングを通じて課題を調査した。

サイバーセキュリティ関係法令 Q&A ハンドブック Ver1.0（令和 2 年 3 月 2 日）[109]では、Q46 ソフトウェアのリバースエンジニアリングでの回答では、「サイバーセキュリティ対策の目的やプログラムの調査解析の目的で行われる当該プログラムの複製や改変等、いわゆるリバースエンジニアリングについては、同法第 30 条の 4 に規定される権利制限の対象として、著作権侵害 にはならないと考えられる」という見解が示されている。

著作権法第 30 条の 4 について社会におけるデジタル化・ネットワーク化の進展等に伴う著作物の利用環境の変化等に対応し、平成 30 年に著作権法が一部改正[110]された。この一部改正の内容の 1 つとして柔軟な権利制限規定が新設され、「デジタル化・ネットワーク化の進展に対応した柔軟な権利制限規定に関する基本的な考え方」[111]が公表された。問 3 の回答の中で、「人工知能（AI）開発のための深層学習、サイバーセキュリティ確保のためのソフトウェアの調査解析、所在検索サービス、情報解析サービス等，通常権利者に不利益を及ぼさないもの，又は権利者に及ぼし得る不利益が軽微なものに留まる 形で著作物の利用行為が行われる様々なサービス等の実施について，権利者の許諾なく行うことが可能となり、イノベーションの創出等が促進されることが期待される」という解釈が示された。日本における行政解釈は、実際に訴訟が起きた場合にも重要視されると考えられる。

一方で、日本以外の他国で訴訟が起きた場合については、どこの国の法律が適用されるか、これは準拠法も問題となる。多くの場合、オープンソースの OSS のライセンスでは準拠法の指定がない。どこの裁判所に訴えられるのか、また、訴えが認められるのかというのも、非常に重要な論点となる。SBOM の取り扱いについても今後、国際調和を進めていく必要もあると考えられる。

SBOM 作成自体に行政解釈上においては問題ないが、オープンソースライセンスの中には、GPL v3 のように特許が紛れ込んだ場合、それも無償でその自由に使用してよいと認めたことと見なすというような内容が含まれている[112]。SBOM 作成、共有する際に付随して発生する問題として、SBOM の中に GPL 系のライセンスの OSS が混入し、特許が紛れ混ん

---

[106] 東京地方裁判所　令和 3 年（ワ）第 5285 号　著作権侵害差止等請求事件
https://www.courts.go.jp/app/files/hanrei_jp/092/091092_hanrei.pdf
[107] 東京地方裁判所　令和 3 年(ワ)第 214０5 号 著作権侵害差止等請求事件
https://www.courts.go.jp/app/files/hanrei_jp/322/091322_hanrei.pdf
[108] https://www.gnu.org/licenses/copyleft.ja.html
[109] law_handbook.pdf (nisc.go.jp)
[110] https://www.bunka.go.jp/seisaku/chosakuken/hokaisei/h30_hokaisei/
[111] https://www.bunka.go.jp/seisaku/chosakuken/1422075.html
[112] https://www.gnu.org/licenses/quick-guide-gplv3.ja.html

でいることが発覚した場合、問題が生じる。現行の著作権法については、SBOM を導入し、関係者間で共有することを長期的観点で見た場合、SBOM 利活用を想定した内容にはなっていないことに留意する必要がある。SBOM を社会的にどのような形で位置づけていくかは今後の課題である。

6. ソフトウェアタスクフォースの運営

6.1. タスクフォースの実施概要

有識者（学識経験者やサイバーセキュリティ・ソフトウェア関連団体等を含む）やソフトウェア事業者等の委員によって構成されたソフトウェアタスクフォースは、産業サイバーセキュリティ研究会 WG1 の基に設置され、経済産業省によって開催されており、本事業ではその運営を行った。ソフトウェアタスクフォースでは、我が国のソフトウェア分野におけるサイバーセキュリティ向上策について検討が行われ、2022 年度にて検討した主な議題は以下の 3 点である。

- 2022 年度の実証について
- SBOM と脆弱性情報等との関連について
- SBOM 導入ガイダンス（初級者向けの SBOM 導入の手引、SBOM 対応モデル、SBOM 取引モデル）について

6.1.1. 第 7 回ソフトウェアタスクフォース

第 7 回ソフトウェアタスクフォースでは、昨年度のタスクフォースにおける議論の振り返り、ソフトウェアの管理手法等に関する国外の状況、令和 4 年度の SBOM 実証イメージ、ノウハウ集・活用モデル・取引モデルのイメージの 4 点について事務局より報告された。自由討議では、今年度の実証のイメージについて、ノウハウ集・活用モデル・取引モデルのイメージについて議論された。

6.1.2. 第 8 回ソフトウェアタスクフォース

第 8 回ソフトウェアタスクフォースでは、最近のインシデント事例、ソフトウェアの管理手法等に関する国外の状況、SBOM 実証の状況・課題の 3 点について事務局より報告された。また、自由討議では、SBOM と脆弱性情報等との関連について、VEX 等について、ノウハウ集等について、SBOM のインセンティブ・コスト等について議論された。

6.1.3. 第 9 回ソフトウェアタスクフォース

第 9 回ソフトウェアタスクフォースでは、ソフトウェアの管理手法等に関する国外の状況、SBOM 実証の結果等（全体概要）、実証結果等を踏まえた検討、今後の取組課題等の 4 点について事務局より報告された。自由討議では、SBOM の導入手引案について、対応モデル・取引モデルの要点について議論された。

6.2. 実施スケジュール

2022 年度のタスクフォースは、以下のスケジュールで実施された。また、各タスクフォースでは、以下の議事次第と資料にて議論された。

- 第 7 回ソフトウェアタスクフォース
  - 日時：2022 年 7 月 26 日（火） 14:00~15:50
  - 開催方式：Web 開催

- 議事次第
  1. 開会
  2. 事務局資料説明
  3. 自由討議
  4. 閉会
- 配布資料
  1. 資料 1 議事次第・配布資料一覧
  2. 資料 2 委員名簿
  3. 資料 3 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の検討の方向性

- 第 8 回ソフトウェアタスクフォース
  - 日時：2022 年 11 月 28 日（月） 13:00~15:00
  - 開催方式：Web 開催
  - 議事次第
    1. 開会
    2. 事務局資料説明
    3. 渡辺委員、寺田委員から脆弱性情報等と SBOM に関する取組等の紹介
    4. 自由討議
    5. 閉会
  - 配布資料
    1. 資料 1 議事次第・配布資料一覧
    2. 資料 2 委員名簿
    3. 資料 3 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の検討の方向性
    4. 資料 4 JVNiPedia_MyJVN 機能拡張について
    5. 資料 5 製品識別子を用いた脆弱性対策情報データベースと資産管理との連携に関する検討
- 第 9 回ソフトウェアタスクフォース
  - 日時：2023 年 2 月 28 日（火） 14:00~16:00
  - 開催方式：Web 開催
  - 議事次第
    1. 開会
    2. 事務局資料説明
    3. 自由討議
    4. 閉会
  - 配布資料
    1. 資料 1 議事次第・配布資料一覧
    2. 資料 2 委員名簿
    3. 資料 3 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の検討の方向性
    4. 参考資料 SBOM（Software Bill of Materials）の導入に関する手引（案）

6.3. タスクフォースの実施結果の要点整理

各タスクフォースで有識者から得られた意見の概要について以下でまとめた。

6.3.1. 第 7 回ソフトウェアタスクフォースの意見概要

今年度の実証に関する意見が多くあった。特に、SBOM の精査方法の検討や SBOM のフォーマットやツール利用に関するドキュメントの必要性について述べられた。また、SBOM を用いた脆弱性管理における課題を確認する必要性についても確認できた。ノウハウ集・活用モデル・取引モデルに関する意見として、SBOM を導入する企業が参考になる文書や業界の取引形態による責任関係を実例・サンプルを基に示す文書のニーズについて具体的に述べられた。

6.3.2. 第 8 回ソフトウェアタスクフォースの意見概要

SBOM と脆弱性情報等との関連については、製品識別子の課題に関する意見が多くあり、特に SBOM の利活用における製品識別子の重要性が述べられた。VEX 等についての意見として、SBOM における脆弱性管理における VEX の重要性が確認された。ノウハウ集などについては、多様な企業にとってわかりやすい文書であることや読者の立場によって SBOM のメリットをまとめることなどが求められた。SBOM のインセンティブ、コスト等についての意見として、今後 SBOM を促進する上でのインセンティブを示すことの重要性やコストを抑える上での取組の必要性について述べられた。

6.3.3. 第 9 回ソフトウェアタスクフォースの意見概要

SBOM の導入手引案に関する意見が多くあった。特に、経営層に対するメッセージや SBOM のメリット対する意見が多く、SBOM の普及における重要な要素について述べられた。また、対応モデル・取引モデルの要点について、事業者間での SBOM の負担を平衡化させるモデルや契約書のサンプルの必要性について述べられた。

7. 今後の課題と取組案のまとめ

本年度の調査・実証を通じて整理した課題に対して、導入ガイダンスに反映できる項目と、今後の取組が必要な課題を整理したものを下記に再掲する。

区分	実証から抽出した課題	解決ノウハウ（導入ガイダンスに反映予定）	今後の課題（国、民間）	医療機器	自動車	ソフト
技術	検出した脆弱性の対応要否、優先度の判断が困難	医療機器分野の脆弱性対応フローなどを参考にアドバイザリ、脅威情報を活用し、対応の要否、優先度を判断。	脆弱性管理の高度化、脅威情報の普及促進	●		
	SBOMツールの使い分け・変更による負担増大	機能ニーズを洗い出し、ツール比較情報をもとに選択	—	●	●	●
	CI/CDなど継続的なアップデートへの対応負担	ツールによる自動化可能な範囲で管理	CI/CDに対応した自動管理			●
	SBOM初期導入、ツール等のコスト負担が大きい	ツールの効率的な導入方法、OSSツールの選択活用	OSSベースのツール整備	●	●	●
管理	SBOMに要求される精査のレベルが不明確	SBOM対応モデルの選択肢やSBOMツールの機能に応じて精査の要否を判断する。	—	●	●	
	SBOM生成の対象範囲が不明確	OS,MWを含めて対象全体の上位構成を事前に明確化	—	●	●	●
	ツールの環境構築、SBOM共有のコストが大きい	SaaS型SBOMツールで初期導入と共有の工数を低減	サプライチェーンを通じた脆弱性管理	●	●	●
	ユーザ組織によるSBOMの活用・管理が困難	SBOMツール導入、ベンダ支援の活用	—	●	●	●
	部品の脆弱性残存期間に応じたリスク評価	SBOMの履歴管理により脆弱性残存期間を特定	脆弱性の履歴評価			●
	開発部署、PSIRTなど部署ごとの脆弱性管理が非効率	社内でSBOMを一元管理することで、脆弱性管理を効率化	SBOMによる脆弱性の社内一元管理		●	●
	サプライヤごとの部品粒度のバラつき	取得したすべての粒度をツールで自動管理	脆弱性マッチングの高度化		●	
	サプライヤのサポート切れなどのリスク対応	部品のEOL等に基づくサポート計画・管理を実施	—	●		
取引	サードパーティからのSBOM取得が困難、バイナリ納品物の脆弱性の監視・修正が負担	ソースコード取得とSBOMツールの適用、（バイナリ納品の場合）SBOM提供と脆弱性修正を契約で要件化	今後の課題として脆弱性管理・マッチングに係るものが多い	●	●	●
	サプライヤ部品の精査コストが大きい	SBOMの提供と信頼性に関する責任を契約で規定		●	●	

図 7-1　実証・調査を通じて抽出された主な課題と対応の方向性（再掲）

これらを通じて、今後の取組が期待される課題の中核となるものとして、脆弱性管理・マッチングに係るものが多いことが確認できる。このようなことから本年度、主に検討したSBOM 作成と活用の間をつなぐ SBOM と脆弱性の突合に係る課題について整理する。

サプライチェーンを通じてソフトウェアを安全に利活用する上で、構成管理、脆弱性管理の基盤として SBOM は有効である。その効果を高める上で、本年度実証では、SBOM 作成と SBOM 活用のフィージビリティスタディと効率化の検討を行ったところ、本年度の調査から、SBOM の効果を高める上で、SBOM 作成と SBOM 活用の間をつなぐ、脆弱性マッチングとそのための共通 ID 管理が課題であることが判明した。

来年度は、SBOM 作成と SBOM 活用の間をつなぐ、脆弱性マッチングとそのための共通 ID 管理についての高度化、効率化について実証に基づき検討を進めることが期待される。

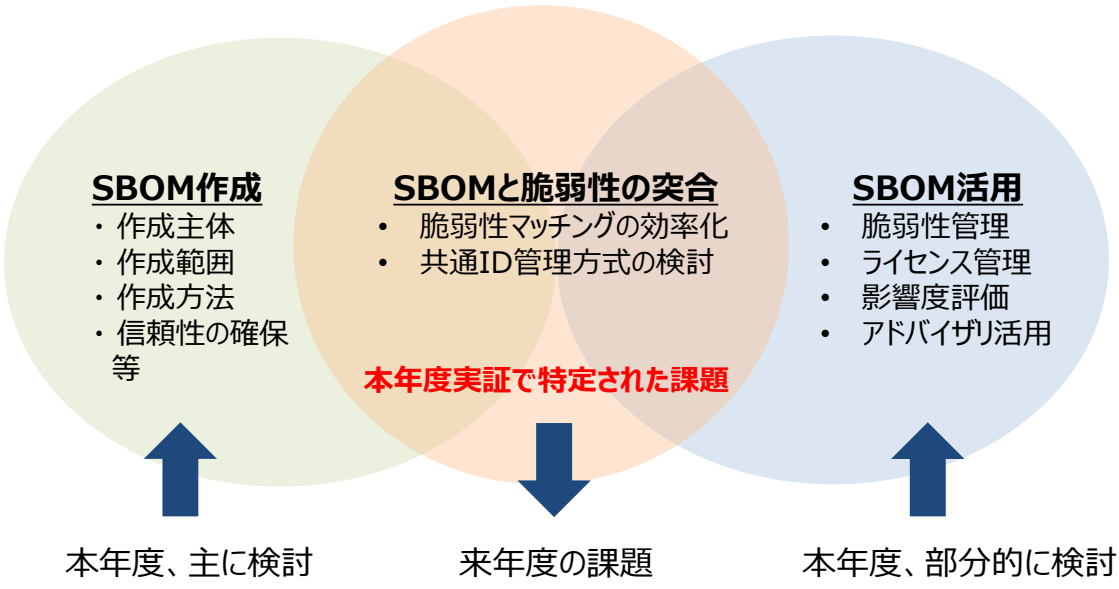


図 7-2　SBOM 管理・脆弱性管理に関する課題の分類

本年度抽出された課題を踏まえ、来年度は SBOM を活用した脆弱性管理の効率化・高度化の方式について実証し、主なステークホルダーと連携し、共通管理方式、脆弱性マッチング、脅威情報活用策、国際連携について検討することが期待される。

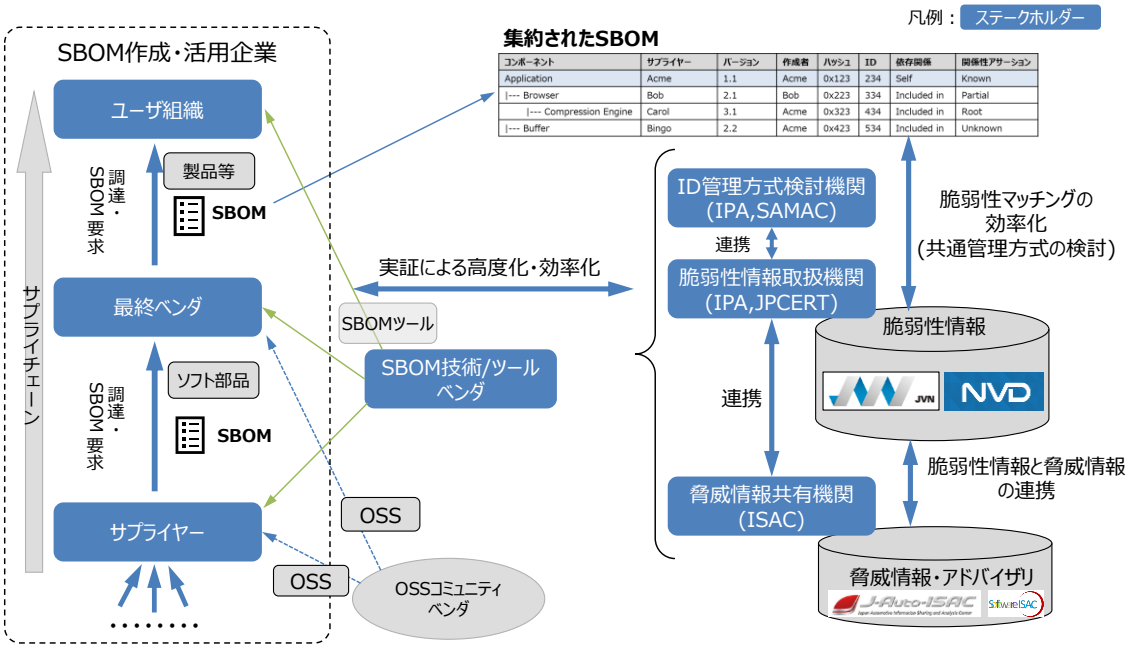


図 7-3　脆弱性の共通管理方式、脆弱性マッチング、脅威情報活用策のイメージ

令和 4 年度 サイバー・フィジカル・セキュリティ対策促進事業
SBOM を導入・活用するサプライチェーンモデルの構築に向けた調査・実証事業

2023 年 3 月

株式会社 三菱総合研究所
サイバーセキュリティ戦略グループ
Tel: 03-6858-3578

（様式２）

二次利用未承諾リスト

令和4年度 サイバー・フィジカル・セキュリティ対策促進事業
SBOMを導入・活用するサプライチェーンモデルの構築に向けた調査・実証事業
報告書

令和４年度サプライチェーン・サイバーセキュリティ対策促進事業（ＳＢＯＭを導入・活用するサプライチェーンモデルの構築に向けた調査・実証事業）

株式会社三菱総合研究所

頁	図表番号	タイトル
46	図2-11	MyJVNバージョンチェッカのイメージ図
47	図2-12	OpenSCAPの概要
47	図2-13	OpenSCAPに用いられているXCCDFの記述例
82	図4-11	システム構成図
116	図4-12	脆弱性マネジメントフロー
132	図4-13	製品ライフサイクルにおけるレガシー医療機器の概念フレームワーク
158	図4-14	システム構成1：オープン系非リアルタイムOSベースシステム
159	図4-15	システム構成2：制御系システム
180	図4-17	SBOM精査まとめ
210	図4-18	TMSLの基本機能
210	図4-19	TMSLの構成とSBOM作成範囲
215	図4-20	チームによる脆弱性調査フローの違い
233	図4-23	angularjs のコンポーネント構成と検出結果
235	図4-24	SCAツール上のプロジェクト階層構造
236	図4-25	プロジェクト構成変更後の SBOM（抜粋）
238	図4-26	SBOM に出力されるSCAツール上の設定情報例
259	図4-28	対象システムの構成とSBOM生成範囲
315	図4-29	SBOM実証 対象システム構成(さくらインターネット株式会社)