SBOMに備えよ！基幹インフラ事業者が直面したツラいSBOM運用とその対策

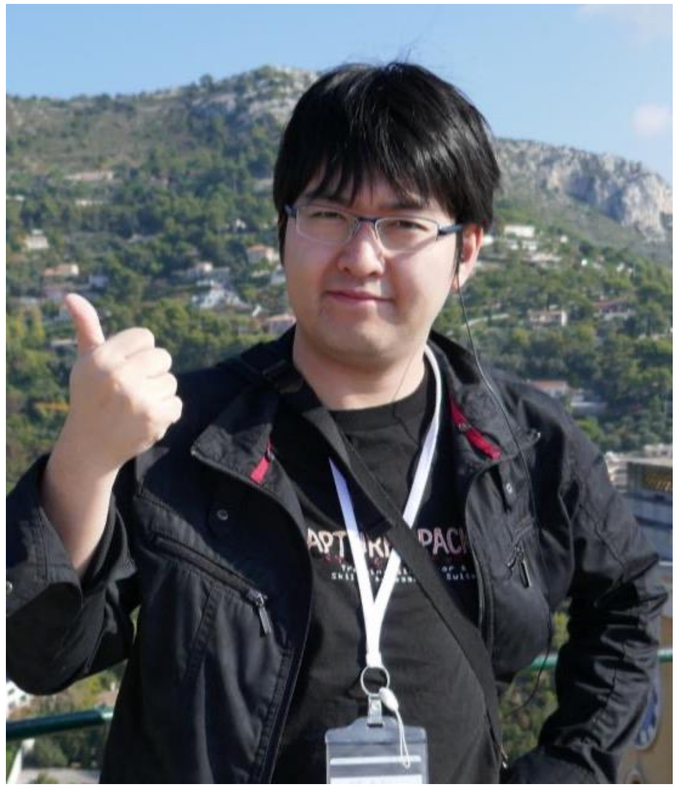
2025年2月5日

NTTコミュニケーションズ株式会社

イノベーションセンターテクノロジー部門

西野 卓也

自己紹介



名前: 西野 卓也
Cyber Threat Intelligence Operations Architect

NTTコミュニケーションズ イノベーションセンター テクノロジー部門
Metemcyber PJ 所属

主な業務
脅威インテリジェンスの収集、分析、活用、共有（売買）に関する研究開発

経歴
2015：NTTコミュニケーションズ入社
2015 – 2016：悪性サイトクローラによる脅威インテリジェンス収集システムの開発
2016 – 2017：マルウェアの自動分析と脅威インテリジェンス管理基盤の開発と運用
2018 – 2019：マルウェアサンドボックスのマルチベンダ性能比較プラットフォームの開発
2018：情報セキュリティ大学院大学へ入学
2018 – 2020:脅威インテリジェンスの共有と拡充に関する研究に従事
2020：脅威インテリジェンスの共有戦略をゲーム理論的に分析して論文化
2020：情報セキュリティ大学院大学を卒業
2020：ブロックチェーン技術を利用した脅威インテリジェンス流通基盤「Metemcyber」を開発
2020：Metemcyberプロジェクトのリーダーとして活動開始
2020：Ethereum in the Enterprise – Asia Pacific 2020でMetemcyberの事例を発表
2021：Metemcyberの実証実験を開始
2022：脅威インテリジェンスとSBOMを用いた迅速かつ効果的なサイバー脅威対策手法を発明し、社内トライアルを展開
2023：ICT-ISACオープンセミナー で「SBOMを活用した脆弱性管理の取り組み」を講演
2023：J-Auto-ISACのメンバーとしてSBOMガイドラインの執筆に参加

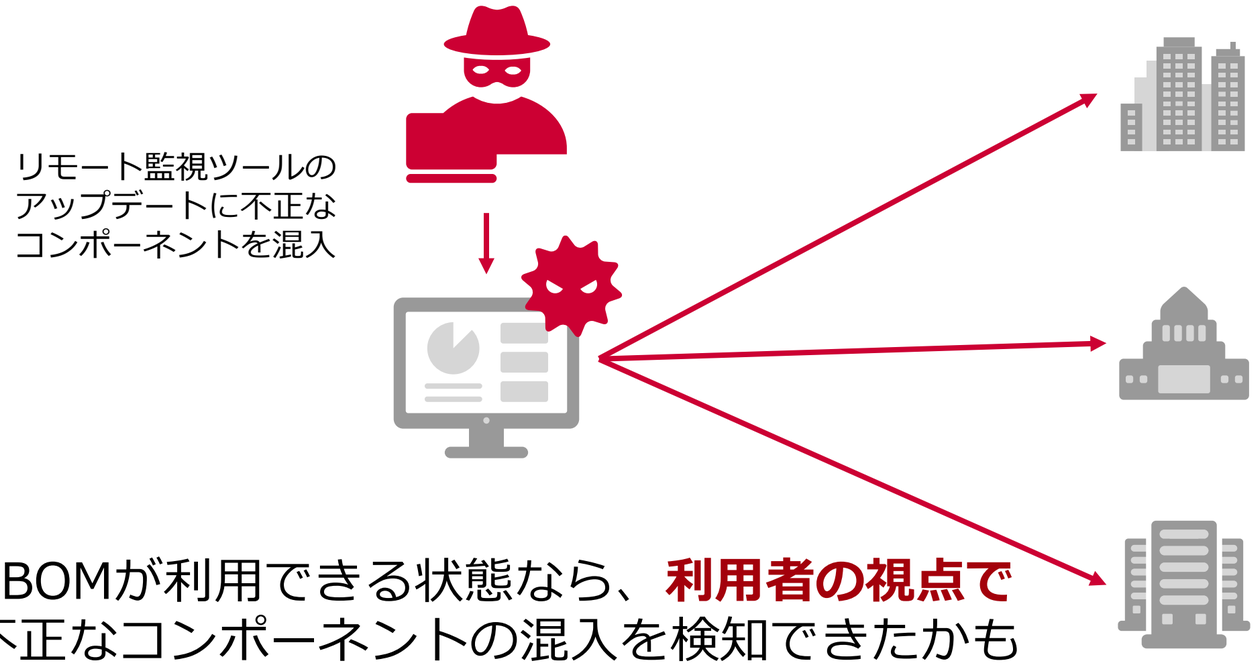
SBOM（Software Bill of Materials）とは

製品に含まれるソフトウェアの構成を可視化した一覧表のこと

SBOMのメリット

食品の成分表示のように「製品の安心・安全」を見える化が可能

SolarWindsを狙ったサイバー攻撃の事例（2020年）



SBOMが利用できる状態なら、**利用者の視点で**
不正なコンポーネントの混入を検知できたかも

SBOM（Software Bill of Materials）の利用目的

サプライチェーンセキュリティ

01 ライセンス管理

- SBOMの最初の利用目的
- SPDXフォーマットが開発された元々の理由

02 脆弱性管理

- SBOMの応用的な利用
- セキュリティ用途のCycloneDXフォーマット

03 ポリシー&コンプライアンス管理

- 調達に関する制限のチェックなど
- SBOMの新しい注目領域

法的な規制に対応するためのSBOM利用

薬機法

IMDRF（国際的なガイダンス）準拠で
医療機器を設計製造するためのSBOM

大統領令

政府調達品に関するサプライチェーン
セキュリティの具体策としてのSBOM

EU CRA

EUで流通するデジタル製品の
「自己適合宣言」としてのSBOM

利用者視点で、デジタル製品に包括的なセキュリティを

欧州サイバーレジリエンス法（EU CRA）が**2025年後半から段階適用**

デジタル要素を備えたすべての製品が対象

https://www.era.europa.eu/system/files/2022-12/01Policy-0%20-%20DG%20CNECT%20-%20Cyber%20Resilience%20Act%20and%20NIS2.pdf

製造者の義務（一部） ※2023/11/30に欧州議会で合意

市場流通する**90%程度の製品**が該当
例：スマートスピーカー、ハードドライブ、ゲーム機など（重要な製品は更に厳格）

1. 自己適合宣言または第三者認証の選択
   - **SBOMを提出するかセキュリティ認定を受けるか（実質SBOM一択）**
2. 5年間のセキュリティアップデート
3. 脆弱性の悪用やインシデント発見後、24時間以内にENISAへ報告
4. 罰金は最高1,500万ユーロまたは当該企業の全世界売上高の2.5%以内
   - **1ユーロ160円の場合、1,500万ユーロは日本円で24億円に相当**

企業におけるSBOM運用の現状

技術と法令が先行しており運用成熟度が皆無

SBOMはライセンス管理、脆弱性管理、
ポリシー＆コンプライアンス管理に
利用することができます

利用したとは
いってない

**技術
先行**

技術的には可能です

できるって言ったよね？

**法令
先行**

SBOMがあれば、サイバーセキュリティ
要件を厳格化できる。サイバー攻撃から
みんなを守れる世界をつくろう！

検証したとは
いってない

企業でのベストプラクティスが存在しないため、

**運用上の落とし穴が
現時点では多数存在**

SBOM運用の落とし穴

01 SBOMのフォーマットに関する問題

02 SBOM生成のツールに関する問題

03 SBOMの同一性に関する問題

04 SBOMの生成手法に関する問題

05 セキュリティ対応に関する問題

06 セキュリティ管理に関する問題

①SBOMのフォーマットに関する問題

標準的に使われているフォーマットが複数存在（**互換性なし**）

SPDX

- 最も人気があるSBOMフォーマット
- ISO/IEC 5962:2021
- コンプライアンスに関する情報管理に強い
- 詳細な分析情報を提供できる
- SBOM領域の全てをカバーする理想の高い規格設計

CycloneDX

- セキュリティ用途のSBOMフォーマット
- サイバーリスク管理に十分な情報が記載可能
- SPDXに比べてシンプルな構造
- 実用性を重視した実装ありきの規格設計

②SBOM生成のツールに関する問題

ツールとフォーマットの組み合わせでファイル出力が大きく変化

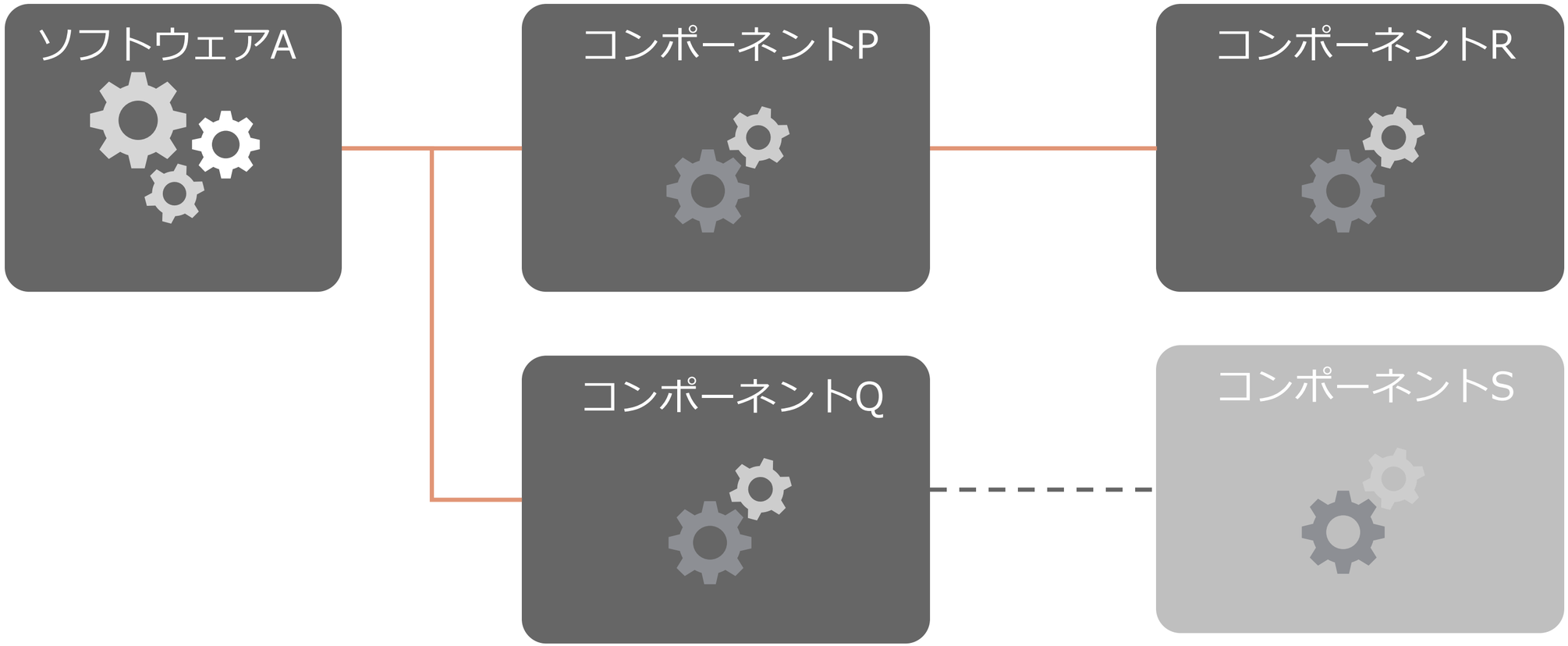
SBOM生成ツールで
コンテナイメージを
スキャンして出力

SBOMファイルの行数	Trivy	Syft
SPDX	5,327 → **33倍**	178,301
CycloneDX	9,799	16,577

主要なOSSツールTrivyとSyftでSBOMファイルの出力を比較
（2023年6月当時:公式djangoコンテナイメージを対象にSBOM生成）

③SBOMの同一性に関する問題

解析対象が同じでも、ツールによって分析結果が異なる場合がある



SBOM生成ツールによっては
検知されないコンポーネント

④SBOMの生成手法に関する問題

条件を満たす対象でしか、精度が高いSBOMファイルを作成できない

SBOM生成ツールで
対象のソフトウェア
をスキャンして出力

ソースコード

部品の検出精度を上げるには、開発時にSBOM生成ツールが対応済みのパッケージマネージャの利用が必要

バイナリ

バイナリの解析に関しては、そもそも技術的な限界によってコンポーネント検出がとても難しい場合が存在

コンテナイメージ

一般的に、ファイルシステムだけでなくOSパッケージ管理の仕組みも利用して検出するため、OS特定が必要

⑤セキュリティ対応に関する問題

SBOMファイルのデータ量とセキュリティ対応の容易さに関係がない

SBOMファイルの行数	Trivy	Syft
SPDX	5,327	178,301
CycloneDX	9,799	16,577

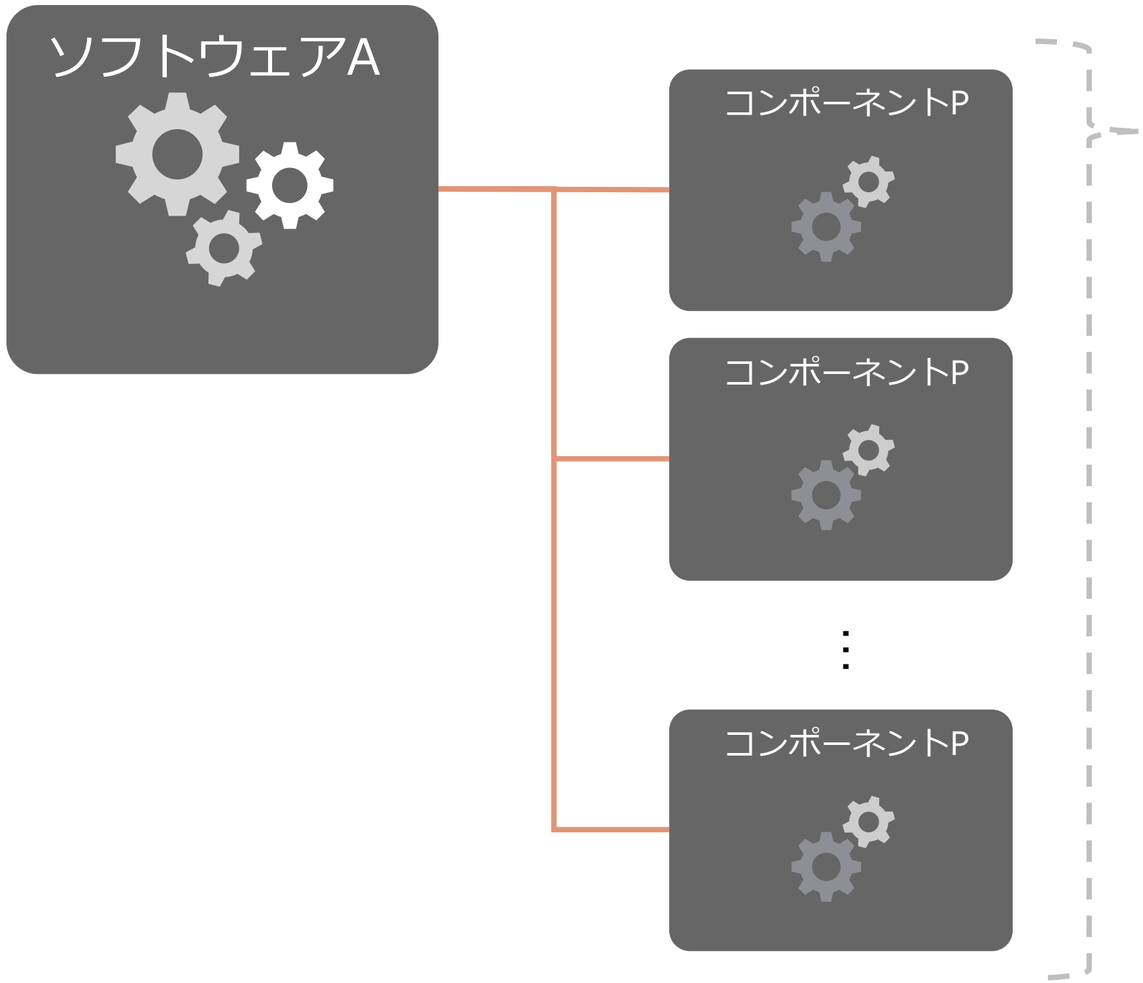
弊社で利用を検討した組み合わせ
（実運用上も問題なし）

SPDXは詳細に情報を記載できるが、
一般的な脆弱性対応にはデータが冗長
（問題発生時のデバッグがたいへん）

CycloneDXで脆弱性対応は十分可能。
しかし、ラインセンス管理と併用する
場合を考えると、SPDXフォーマット
に統合できたほうが嬉しい場面も

⑥セキュリティ管理に関する問題

SBOMを活用するためには、部品と脆弱性のマッチングシステムが必要

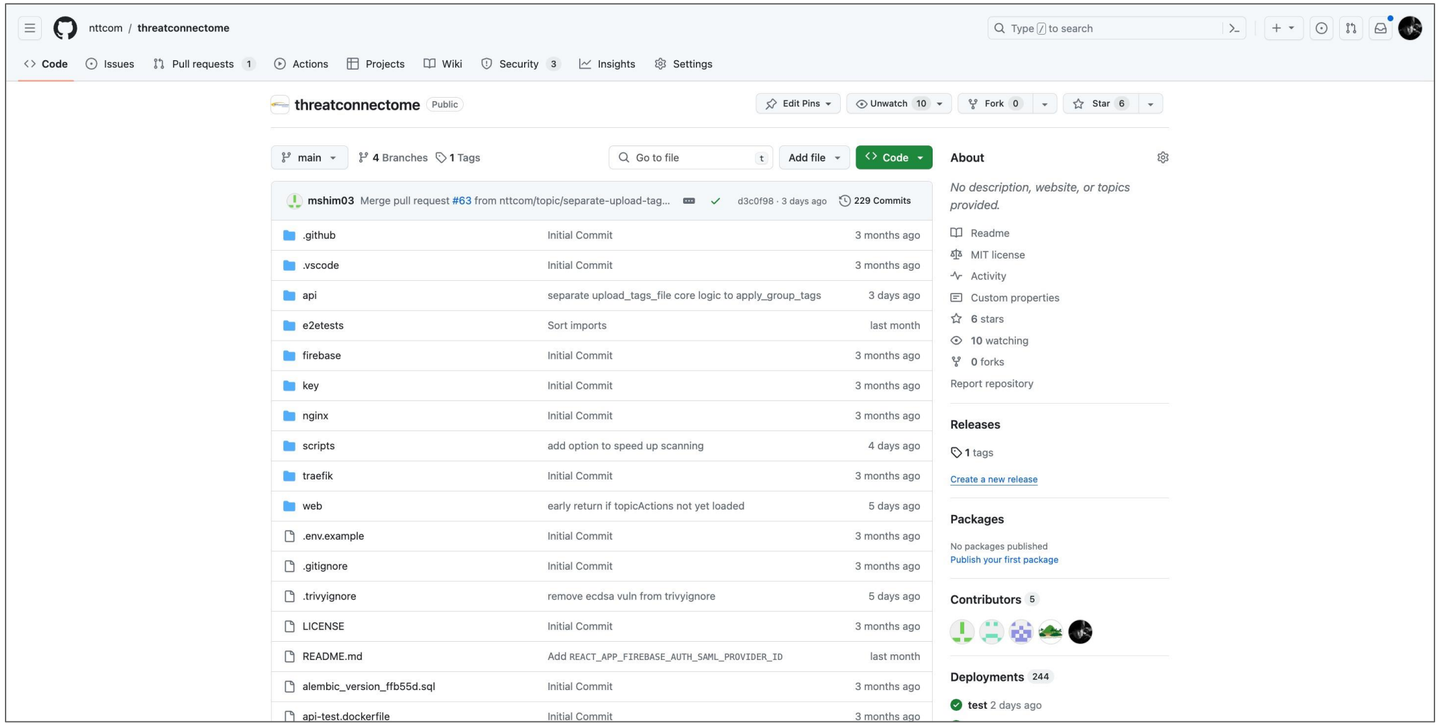


1つのソフトウェアで数百～数千のコンポーネント
大規模なソフトウェアでは万単位の数になることも

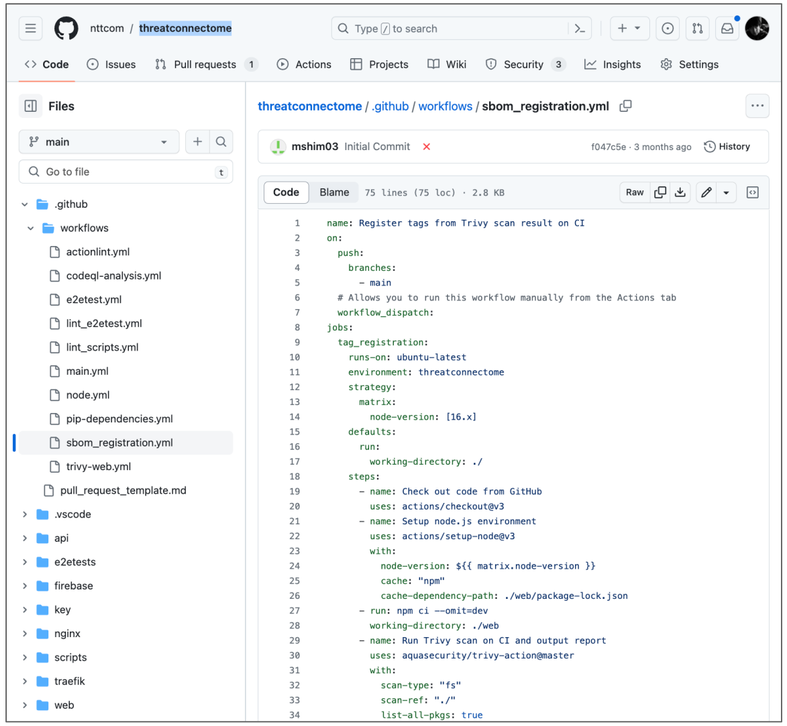
管理するコンポーネント数が多すぎて
ファイルだけ渡されても管理できない

基幹インフラ事業者から見たSBOM作成プロセス

内製や委託開発の場合はSBOM作成フローを導入



GitHub上での
ソースコード管理



GitHubアクション
によるSBOM作成

GitHub標準機能でSBOMエクス
ポートを行うことも現在は可能

開発ライフサイクルとSBOMの関係性

○ お客様がSBOMで検証できる要素

Requirements	Design	Implementation	Verification	Release	Response
セキュリティ要件の決定	脅威モデリング	PRに対する脆弱性検証	動的環境の脆弱性検証	セキュリティ検証結果	構成部品のセキュリティ継続監視
利用するライセンスの決定		新規ライブラリの健全性確認	既存ライブラリの健全性確認	ライセンスの検証結果	コンプライアンス適合の継続監視

最終ビルドのSBOM

開発関係者で共有するために、プレリリースSBOMが作成される場合も

SBOM

第三者機関やお客様への提供を想定

SBOMデータの共有方法や頻度について
外部提供するための最終的な合意を形成

提供ソフトウェアの更新があれば
新しいSBOMの提供も継続して実施

Microsoft セキュリティ開発ライフサイクル
https://learn.microsoft.com/en-us/windows/security/threat-protection/msft-security-dev-lifecycle

GitHubでのソースコード管理とSBOM作成

新機能のPR → PRを修正

脆弱性通知 → 脆弱性修正PR

脆弱性スキャンの
ワークフローを実行

NG

Vulns

脆弱性スキャンの
ワークフローを実行

OK

ソースコード単位の管理では
依存コンポーネントの抽象化が
管理する製品で大きく異なる

依存コンポーネントを
規格的に管理したい

脆弱性スキャンの
ワークフローを実行

OK

LGTM → mainへmerge

LGTM → mainへmerge

main ブランチ

SBOMの提供でOSS部品
の脆弱性検知が容易に

SBOM生成の
ワークフローを実行

SBOM生成の
ワークフローを実行

社内外のセキュリティ
アナリスト

purlを利用した管理が
好まれることが多い

SBOM

定期的な
脆弱性スキャン

Vulns

SBOM

GitHubでのソースコード管理とSBOM作成（例：Trivy）

新機能のPR → PRを修正 → 脆弱性通知 → 脆弱性修正PR

Trivyで脆弱性スキャン → NG → Vulns

Trivyで脆弱性スキャン → OK → LGTM → mainへmerge

Trivyで脆弱性スキャン → OK → LGTM → mainへmerge

main ブランチ

TrivyでCycloneDXファイルを出力

SPDXとCycloneDXの両方を生成して配布する場合が多いかも？

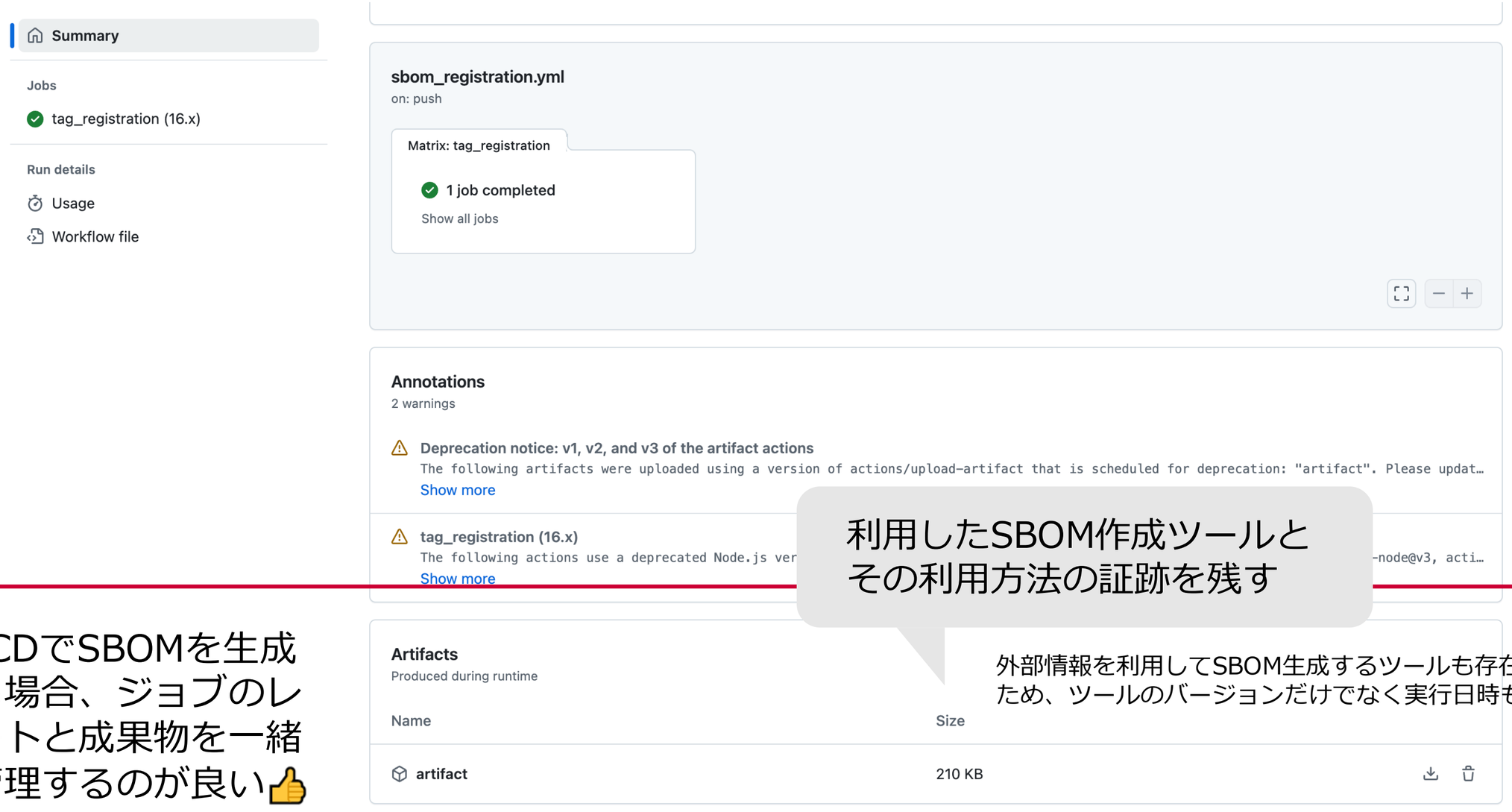
TrivyでCycloneDXファイルを出力

社内外のセキュリティアナリスト

SBOM → **定期的な脆弱性スキャン** → **Vulns**

作成されたSBOMは？

GitHubアクションのアーティファクトでSBOMを保管



利用したSBOM作成ツールと
その利用方法の証跡を残す

外部情報を利用してSBOM生成するツールも存在するため、ツールのバージョンだけでなく実行日時も重要

CI/CDでSBOMを生成する場合、ジョブのレポートと成果物を一緒に管理するのが良い👍

運用に必要なものは可能な限りSBOMを収集する

運用対象のOSSから抽出
（コード、コンテナイメージ、バイナリ等）

SBOMジェネレータ

SBOMファイル自体の提供
（コミュニティからの配布や保守等）

SBOM

コンポーネントの一元管理で
問題のあるコンポーネントが
横断的に見つけやすくなる

SBOM
管理

「製品に含まれるOSSの特定」
を最初の目的にすること

専用アプライアンスや古いソフトウェアの場合、メーカーであってもSBOM情報が提供できない場合がある。さらに言えば、専用ソフトウェアの専用コンポーネントは、脆弱性やライセンス情報の収集自体がが難しい場合も。

SBOMを利用した脆弱性管理

（パッチおよび）脆弱性管理とは

IT 脆弱性の悪用を事前に防止するために設計されたセキュリティプラクティス

NIST SP 800-40 Version 2.0 日本語版 「パッチおよび脆弱性管理プログラムの策定」 翻訳: IPA
https://www.ipa.go.jp/security/reports/oversea/nist/ug65p90000019cp4-att/000025330.pdf

NIST SP 800-40 の 最新版（2022年4月発行）では、インベントリについて以下のように記述されている。
NIST SP 800-40 Rev. 4: Guide to Enterprise Patch Management Planning: Preventive Maintenance for Technology

3.2 ソフトウェアと資産のインベントリ

（中略）
現実的な目標は、常に新しい資産を発見し、すべての資産に関する最新情報を収集する自動化に頼ることで、ほぼ包括的なインベントリを維持することです。ベンダーによっては、ソフトウェア部品表（SBOM）のような、資産のソフトウェア構成に関する機械消費可能なデータを提供し、組織のインベントリを補強するために使用することもできます。

常に更新を行わなければ、インベントリはすぐに古くなり、パッチの適用に必要な情報もますます不正確で不完全なものになってしまうでしょう。かつて資産やソフトウェアがほとんど固定的であり、静的な論理的・物理的境界内に配置されていた頃は、**脆弱性スキャンを実施することにより、月次または四半期ごとにインベントリを更新することが一般的に許容されると考えられていました。このようなモデルは、もはや使用されるべきではありません。**

- **すべての資産に関する最新情報の収集を自動化（SBOMの利用含む）**
- **常に更新（脆弱性スキャン）を実施**
- **月次または四半期ごとにインベントリを更新する運用は非推奨**

インベントリを更新する運用とは

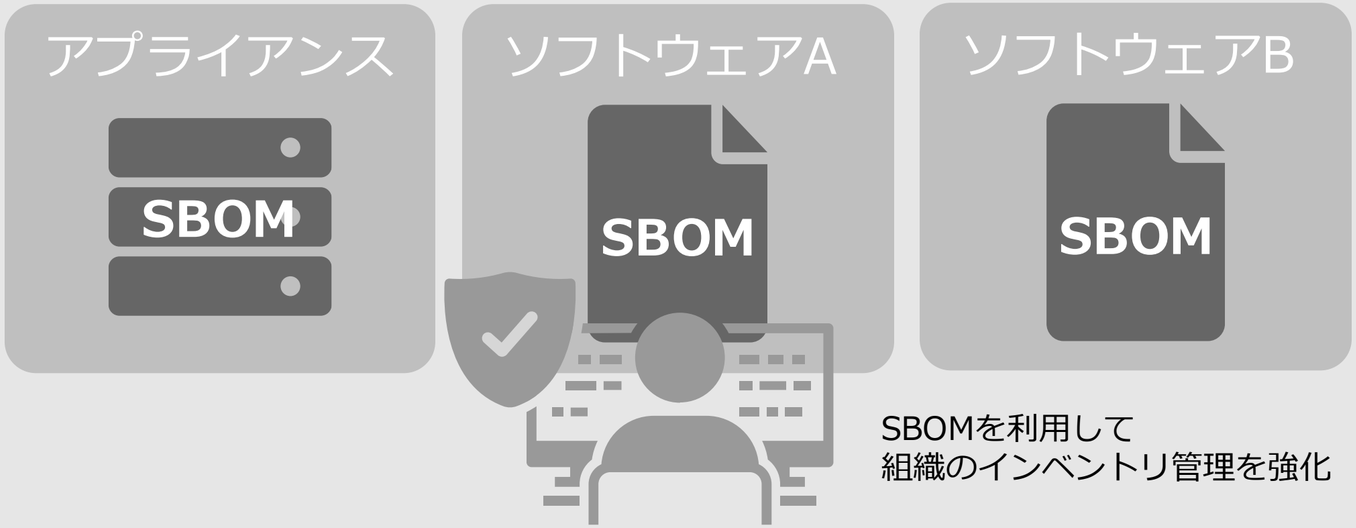
インベントリ＝自組織で管理するIT資産のこと。
（物理的なものやソフトウェア単体だけでなく、ソフトウェアライブラリ等も含まれる場合がある）

NIST SP 800-40 の最新版（2022年4月発行）では、インベントリについて以下のように記述されている。
NIST SP 800-40 Rev. 4: Guide to Enterprise Patch Management Planning: Preventive Maintenance for Technology

3.2 ソフトウェアと資産のインベントリ

組織は、OT、IoT、コンテナ資産を含む、物理および仮想のコンピューティング資産のソフトウェアインベントリの最新化を常に維持する必要がある。
この情報は、単一の企業資産インベントリに含めることもできるし、複数のリソースに分割して含めることもできる。

SBOMがベンダーから提供された場合



SBOMを利用して
組織のインベントリ管理を強化

SBOMを利用できれば脆弱性管理はラクに？

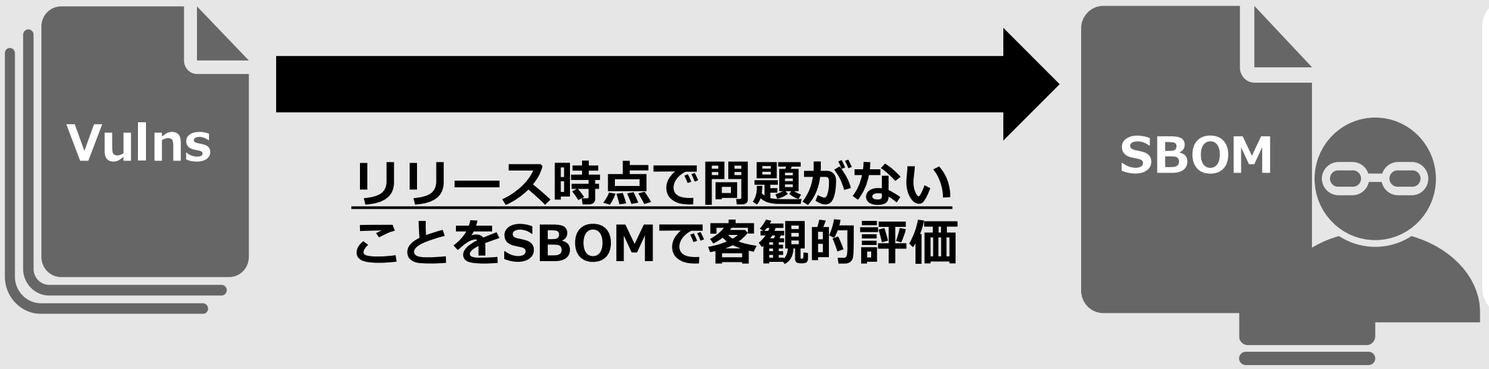
SBOMには利用ライブラリ名やバージョンが記載されているため、スマートな脆弱性管理が理論的には可能

1. **SBOMの作成**
   - SBOM作成ツールを利用
2. **脆弱性の検知**
   - SBOMで脆弱性スキャン
3. **検出された脆弱性の調査**
   - サービス利用者に与える影響は？
4. **脆弱性の対応**
   - 必要があれば脆弱性に対応

SBOM型脆弱性スキャナ
例：TrivyによるSBOM作成とスキャン

リスク評価と対応管理
コンテナ環境（K8s/Docker）であれば、
OSSの**KubeClarity**が比較的試しやすい
https://github.com/openclarity/kubeclarity

脆弱性管理



リリース時点で問題がない
ことをSBOMで客観的評価

SBOMで依存コンポーネントの確認が迅速に

新しく発見された脆弱性の自社影響を複数のプロダクトやサービス横断で常に漏れなく確認できる

コンテナ環境で使いやすい「KubeClarity」

OSS製のSBOM脆弱性管理ツール
コンテナイメージのセキュリティ管理

- SBOM作成ツールとの統合
  - Syft
  - Trivy
- SBOM型脆弱性スキャナとの統合
  - Grype
  - Trivy

https://github.com/openclarity/kubeclarity

OpenClarityへの移行

2024年10月に「KubeClarity」プロジェクトは凍結（アーカイブ化）されました。KubeClarity利用者は、「OpenClarity」プロジェクトへの移行が現在は推奨されています。

https://github.com/openclarity/openclarity

リスク評価と対応管理は製品特性で変化

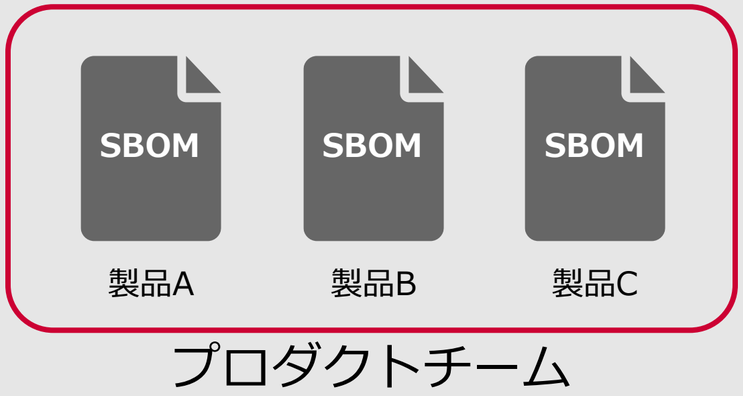
コンテナではない、アップデートが難しい環境ではCVSSベースの脆弱性通知がただのノイズに……

1. **SBOMの作成**
   - SBOM作成ツールを利用
2. **脆弱性の検知**
   - SBOMで脆弱性スキャン
3. **検出された脆弱性の調査**
   - サービス利用者に与える影響は？
4. **脆弱性の対応**
   - 必要があれば脆弱性に対応

SBOM型脆弱性スキャナ
例：TrivyによるSBOM作成とスキャン

リスク評価と対応管理
コンテナ環境（K8s/Docker）であれば、
OSSの**KubeClarity**が比較的試しやすい
https://github.com/openclarity/kubeclarity

脆弱性管理



プロダクトチーム

SBOM利用で複数製品の脆弱性スキャンは容易になったが、

ユーザー影響がほぼない脆弱性が大量に通知される場合が存在

（ソフトウェアごとに数百～数千のコンポーネントが存在するため）

サービス利用者への影響を考えた対応優先度の決定

SSVC(Stakeholder-Specific Vulnerability Categorization）の利用を検討。運用課題も明らかに

	CVSS	SSVC
特徴	**•脆弱性の深刻度を定量的に判断するための評価方法** •ベンダフリーな評価方法であり、脆弱性の評価方法として一般的によく知られている •CVE報告数の爆増により、**実組織でCVSSだけを利用した脆弱性対応はほぼない**（セキュリティ担当者が、何らかの方法でフィルタリングや場合分けをしている運用が多数）	**•ステークホルダー（パッチ適用者）のニーズに基づいた脆弱性の優先順位付け** •優先順位名と時間的猶予の関係がシンプル •Defer（放置OK） •Scheduled（定期アップデートで修正） •Out-of-cycle（緊急アップデートが必要） •Immediate（即時対応が必須） •サービス運用状況（インターネットからアクセス可能か等）を反映でき、**現場の肌感覚に近い脆弱性判断が機械的に実現可能**
最新バージョン (2024/11現在)	**4.0** (2023/11/1公開)	**v2024.3** (2024/3/9公開) ※正確には v2024.3.8が最新（2024/11/1）
主な管理団体	**FIRST** インシデント対応およびセキュリティチームの国際的に有名なフォーラム団体	**CERT/CC** カーネギーメロン大学の名義で発表されたが、現在は同大学のCSIRTで管理
既知の課題	参考にはするが、最終的には各社独自の意思決定で脆弱性対応を行う場合がほとんど	**決定木の各パラメータに必要な値を個別に設定するため、各組織での運用が面倒**

脆弱性の検知と優先度の判断を、機械的かつ現実的に

CVEのADP(Authorized Data Publishers)の取り組みで、SSVCに必要な情報がより簡単に取得可能※1に！

1. **SBOMの作成**
   - SBOM作成ツールを利用
2. **脆弱性の検知**
   - SBOMで脆弱性スキャン
3. **検出された脆弱性の調査**
   - サービス利用者に与える影響は？
4. **脆弱性の対応**
   - 必要があれば脆弱性に対応

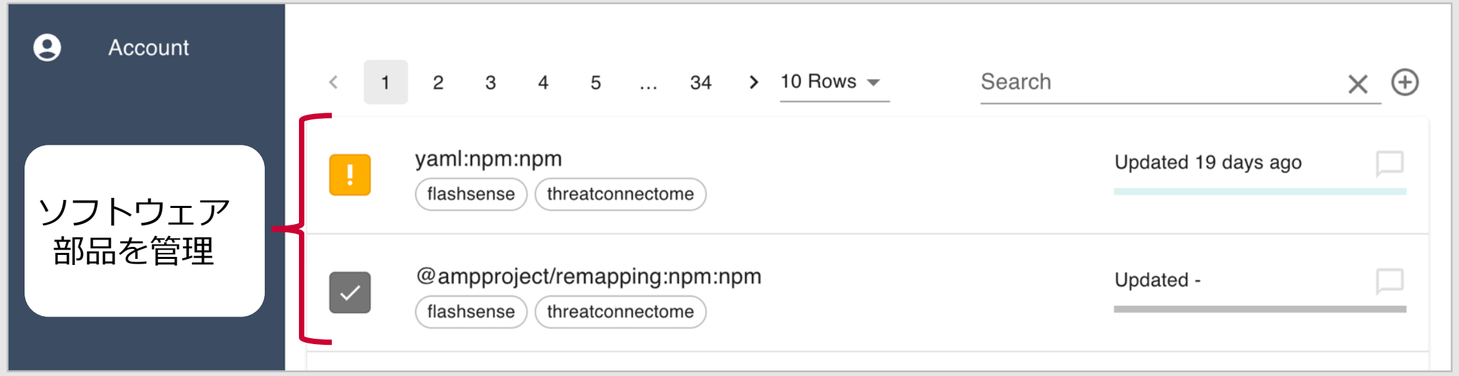
※1 CISAは、CVEのADPとしてSSVCに必要なパラメータをCVEのサイトから提供しているが、中身は同組織が提供している「Vulnrichment」と同じであり、弊社システムは現状そちらを利用

SBOM型脆弱性スキャナ

リスク評価と対応管理

SBOM＋SSVCで脆弱性管理

即時対応の必要な範囲がSSVCで明らかになるため



01 **初動対応時間の短縮**

02 **電気通信業や製造業でのアプライアンス管理への応用**

SBOMもSSVCも手動運用は難しいのでツールで簡単に

パッチアップデートが難しい機器のセキュリティ管理の社内トライアルを実施

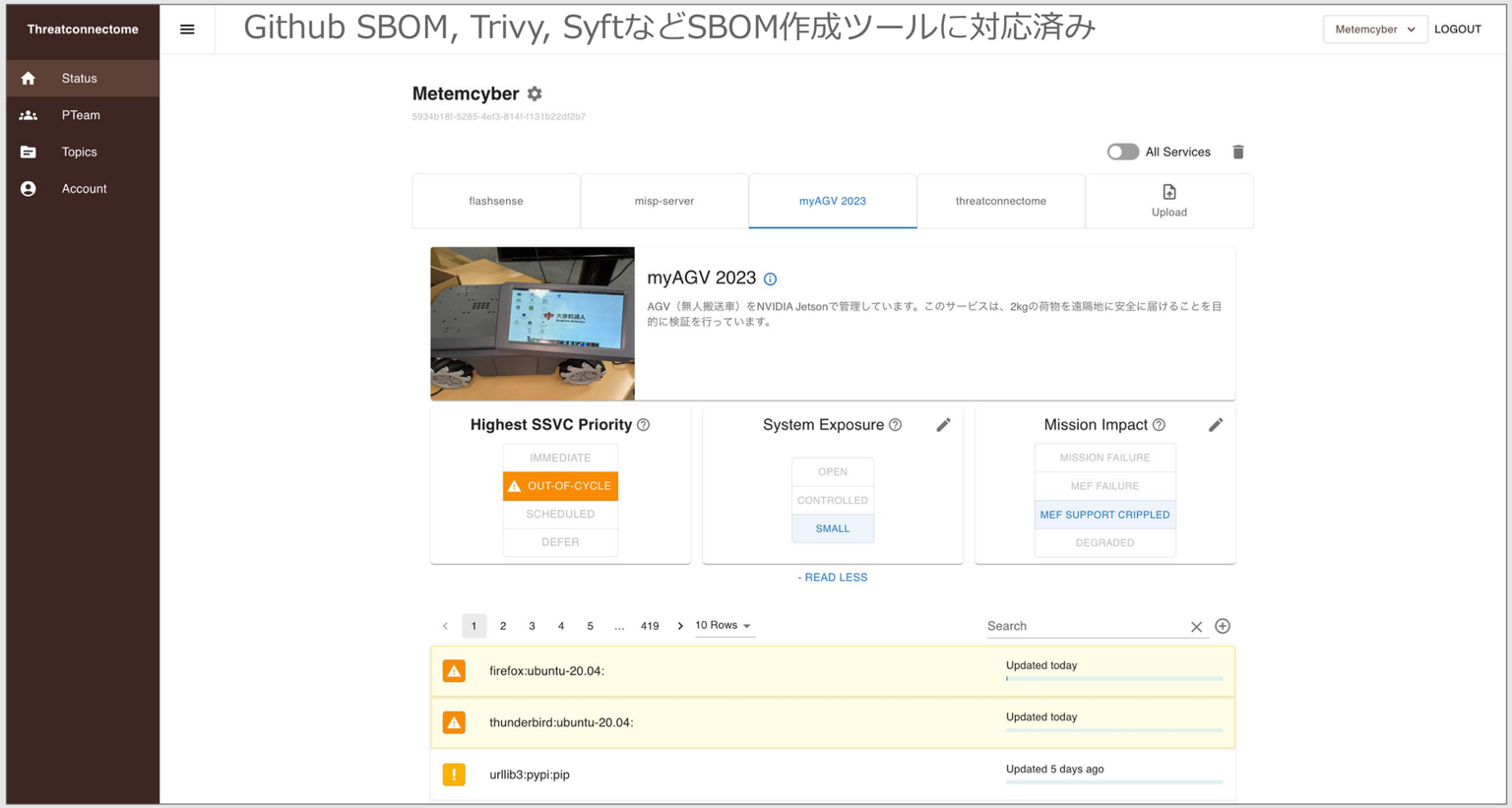
スレットコネクトーム
Threatconnectome

プロダクト開発チームがSBOMを投入し、
SSVCパラメータ設定をアナリストが検証

01 SSVCを利用したセキュリティアラート通知

02 SPDX 2.3と CycloneDX 1.6のSBOMフォーマットに対応

03 国内の自動車業界標準に合わせたSBOM管理（予定）



https://github.com/nttcom/threatconnectome

SBOM管理ソフトウェア「Threatconnectome」はOSSとして提供中

まとめ

- **「食品の成分表示」の概念をソフトウェアの世界に持ち込んだもの≒SBOM**
- サプライチェーンセキュリティの観点で、SBOMの利用が注目されており法規制が進んでいる
  - **ライセンス管理**
  - **脆弱性管理**
  - **ポリシー&コンプライアンス管理**
- 「SBOM運用のベストプラクティス」は現状まだ存在しないので、導入する場合は手探りで検証を行うことになる。自社で検証した範囲でも、解決が容易ではない問題が６つ見つかった
  - 各業界からガイドラインが出つつあるので、可能な限りそれを参照すべき。特にSBOMで必要となる概念を整理するためには、経産省が公開している「SBOM導入の手引き」は一読をオススメ
  - 具体的な運用方法が記載されているわけではないため、そこは自力or各業界のガイドラインを参考に頑張る必要あり
- 現状、SBOMに関するデータの収集は主体的な行動が必要であり、場合によってはSBOM作成ワークフロー構築を主導したり、SBOM作成ツールを利用して自分から集めに行く必要がある
  - CRAなどの法的影響で、今後メーカーからのSBOM提供が一般的になる可能性も
- **脆弱性管理が目的であれば、まずはOSSに依存する部分から。目的を明確にしたSBOM管理が重要**
- SBOMを脆弱性管理に利用する場合、脆弱性検知は正確になるが運用は全然楽にならない
  - 管理対象のソフトウェアが多くなると、脆弱性通知されるアラート数もほぼ線形に増加していくため
- **「SBOM＋SSVCの脆弱性管理」は運用の現実解になりそう。ただし、手動運用は難しく自動化は必須**