TDU
TOKYO DENKI UNIVERSITY

Information Security Lab
Collaborate together
to make our Internet secure.

Network Security Forum 2025 (NSF2025)

SBOMと脆弱性ハンドリング

2025年01月24日

寺田真敏
https://blog.isl.im.dendai.ac.jp/

自己紹介

セキュリティ分野での経歴

1994	1996	1998	2000	2002	2004	2006	2008	2010	2012	2014	2016	2018	2020	2022	2024
情報処理振興事業協会（IPA）非常勤研究員（1994～1997年）															

1998年、情報処理学会コンピュータセキュリティ研究会（CSEC）の発足に携わる。

1998年、日立製作所において、国内初の企業シーサート（HIRT：Hitachi Incident Response Team）を立ち上げ、シーサートとしての活動を開始する。

2002年、慶応義塾大学社会人学生として、JVN（JPCERT/CC Vendor status Notes）試行サイトの立ち上げる。

JPCERTコーディネーションセンター専門委員、情報処理推進機構（IPA）セキュリティセンター研究員として、脆弱性対策情報JVN（Japan Vulnerability Notes）、脆弱性対策データベースJVN iPediaとMyJVNを立ち上げて開発を推進する。

2007年、国内5つのCSIRTと共に日本シーサート協議会を設立し、2015年より運営委員長として、2021年より理事としてシーサート活動の普及を推進する。

2008年より中央大学大学院客員講師（/客員教授）として、マルウェアとサイバー攻撃対策研究人材育成ワークショップ（MWS）を立ち上げる。

2015年よりICT-ISAC Japan運営委員としてサイバーセキュリティのための情報共有基盤の整備を推進する。

2019年より東京電機大学教授として次世代のセキュリティ人材育成に取り組んでいる。

寺田真敏（てらだまさと）

TDU
TOKYO DENKI UNIVERSITY

Information Security Lab
Collaborate together
to make our Internet secure.

本講演では、SBOMを活用していくにあたり、総務省「サイバー攻撃への集団防御に向けた情報共有基盤に関する実証事業（2016年～2017年）」総務省「ソフトウェア脆弱性を狙ったサイバー攻撃の防御に向けた情報共有基盤に関する実証実験（2019年～2021年）」の活動で得た学びを紹介します。

1. はじめに
2. 品質の見える化
3. NISTの取組み
4. IT資産管理
5. 製品識別における課題
6. 製品識別における課題解決のアプローチ
7. 製品識別子
8. 管理された製品識別子を提供するために、、、
9. おわりに

せっかくの機会なので、
製品識別の課題についてお話します。

はじめに

SBOM(ソフトウェア部品表)が果たす役割

- Q:日本で実装されるのか その意義は？
- A:次の課題を解決するための「きっかけ」になることを期待したい。

【 課題 】

国内で利用されているソフトウェアや装置を対象とした脆弱性情報ならびに対策情報は流通しているのか？	セキュリティ設定に関する作業を手作業で行なうと、設定ミスや設定者のセキュリティ知識の程度や判断の相違などによりセキュリティ要件を損なう可能性大

【 解決策 】

製品識別による脆弱性への対処	作業の機械化(自動化)による対処
脆弱性対策情報DBへの集約	共通技術仕様による(自動化)の普及

品質の見える化

Software Component Transparency

- SBOM(Software Bill of Materials、ソフトウェア部品表)
  - 米国商務省電気通信情報局(National Telecommunications and Information Administration)が主体となって推進し、2018年7月に、初回打合せが開催された。
  - ソフトウェアを構成するコンポーネントの透明性(Software Component Transparency)をあげることを目的とする。
  - よく使用されている例: 食品の成分表のようなもの

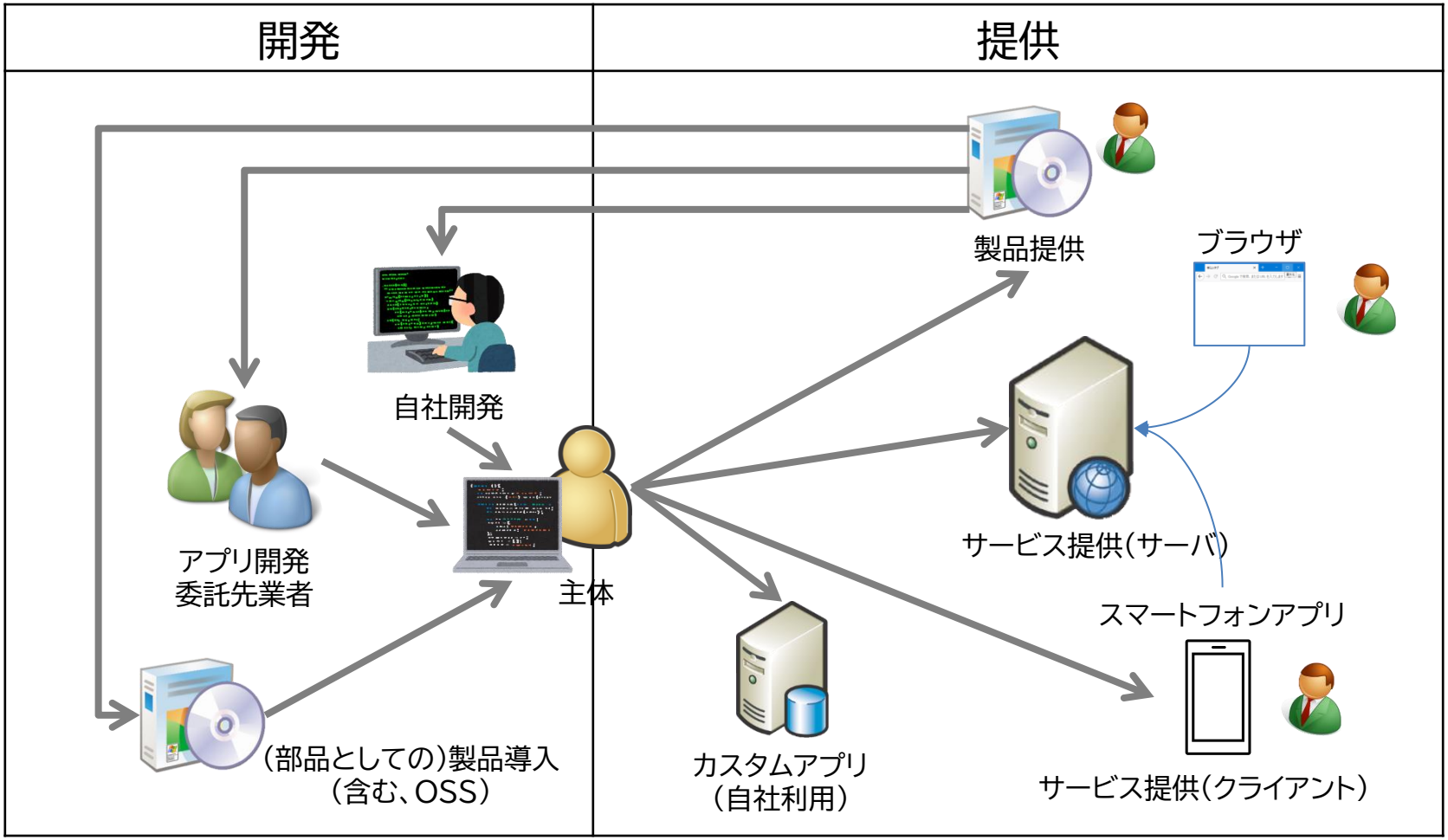
- BOM(Bill of Materials)
  製造業では、製品を製造する際に必要な部品や原材料などの構成情報を部品管理している。部品構成管理はBOM管理と呼ばれたり、プロセス製造業ではレシピ管理と呼ばれたりしている。

[出典] NTIA Software Component Transparency
https://www.ntia.doc.gov/SoftwareTransparency

品質の見える化

Software Component Transparency

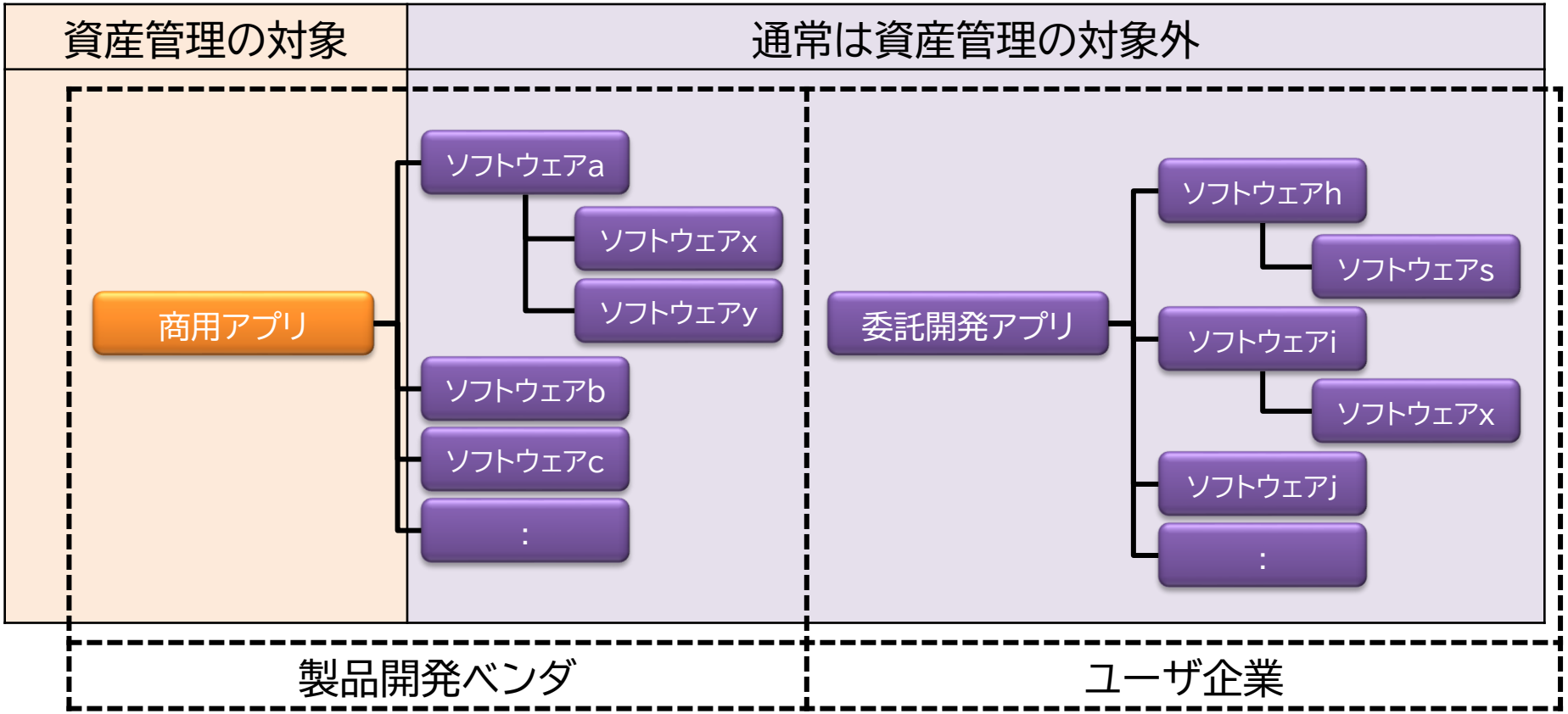
- ソフトウェア開発ならびに提供形態の多様化



品質の見える化

Software Component Transparency

- ソフトウェアの依存関係とIT資産管理



品質の見える化

SBOM(ソフトウェア部品表)

- 加工食品の表示の具体例
  - 表示の方式等
    表示は、容器包装を開かないでも容易に見ることができるように当該容器包装の見やすい箇所に表示します。など
  - 原材料名及び添加物
    使用した原材料は、添加物以外の原材料と添加物を明確に区分し、それぞれに占める重量の割合の多いものから順に記載します。など

名称	マカロニサラダ
原材料名	マカロニ（小麦を含む、イタリア製造）、マヨネーズ（卵を含む）、きゅうり、にんじん、玉ねぎ、ハム（豚肉を含む）、食塩
添加物	調味料（アミノ酸）、リン酸塩（Na）、酸化防止剤（V.C）、カゼインNa（乳・大豆由来）、発色剤（亜硝酸Na）
内容量	２００g
消費期限	令和２年〇月〇日
保存方法	要冷蔵（１０℃以下で保存）
製造者	〇〇食品株式会社 新潟市中央区紫竹山３－３－１１

［出典］　食品表示法による表示について(新法に基づく表記)
https://www.city.niigata.lg.jp/smph/iryo/shoku/shokueigyo/hyoji/shyouzihou20200330.html

品質の見える化

SBOM(ソフトウェア部品表)

- ソフトウェアにおける具体例
  - ソフトウェアを構成するコンポーネントを明示することで、ソフトウェアの透明性(Software Component Transparency)を確保できる。

    ⇒品質の見える化の一手段として有用

名称	MyJVN API
ソフトウェア部品表 (使用している 外部コンポーネントや 前提プログラム)	● オラクル JRE ● OpenSSL Project OpenSSL ● OpenBSD OpenSSH ● NTP Project NTP ● Apache Software Foundation APR-util (Apache Portable Runtime Utility library) ● pcre.org PCRE (Perl Compatible Regular Expression library) ● Apache Software Foundation Apache HTTP Server ● Apache Software Foundation Apache Tomcat ● MySQL AB MySQL
開発業者	IPAシステムインテグレーション開発

品質の見える化

SBOM(ソフトウェア部品表)

- 2021年7月12日、NTIA(米国商務省電気通信情報局)より公開
  SBOM「最小要素」には、コンポーネントを一覧化した部品表に含まれる情報だけでなく、SBOMの利活用者が実施すべき事項も規定されている。

区分	概要	具体的な定義
データフィールド	各コンポーネントに関する基本情報を明確化すること	次の情報をSBOMに含めること。 ● サプライヤー名 ● コンポーネント名 ● コンポーネントのバージョン ● その他の一意な識別子 ● 依存関係 ● SBOMの作成者 ● タイムスタンプ
自動化サポート	SBOMの自動生成や可読性などの自動化をサポートすること	SBOMデータは機械判読可能かつ相互運用可能なフォーマットを用いて作成され、共有されること。現状では、国際的な議論を通じて策定された、SPDX、CycloneDX、SWIDタグを用いること。
プラクティスとプロセス	SBOMの要求、生成、利用に関する運用方法を定義すること	SBOMを利活用する組織は、次の項目に関する運用方法を定めること。 ● SBOMの作成頻度 ● SBOMの深さ ● 既知である未知なこと ● SBOMの共有 ● アクセス管理 ● 誤りの許容

[出典] The Minimum Elements For a Software Bill of Materials (SBOM)
https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf

品質の見える化

SBOM(ソフトウェア部品表)

- セキュリティ・トランスペアレンシー・コンソーシアムの活動報告によると、、、

品質上の課題	内容
最小要素不足	SBOMの実データを調査したところ、米国NTIAが定めるSBOMの最小要素のうち、作成者や作成時刻の記載は非常に少ない状況である。 ＊NTTによる約1,800件の調査
要素の表現が不適切・表記揺れ	SBOMの要素にはソフトウェアの開発元を示すベンダ項目があるが、開発元が示されない場合がある。また略称等を用いる場合もあり、正式名称とのマッチングがとれない。
コンポーネントの網羅性	ソースコードのコピーペーストや一部を改変した再利用によって発生するパッケージ情報には記されないソフトウェアの依存関係があり、脆弱性に気づけない。

要素	記載割合
SBOMの作成ツールやコンポーネントに関する名称や識別子	9割程度
ベンダやバージョン	6割程度
SBOMの作成者	3割程度
SBOMの作成時刻	1割程度

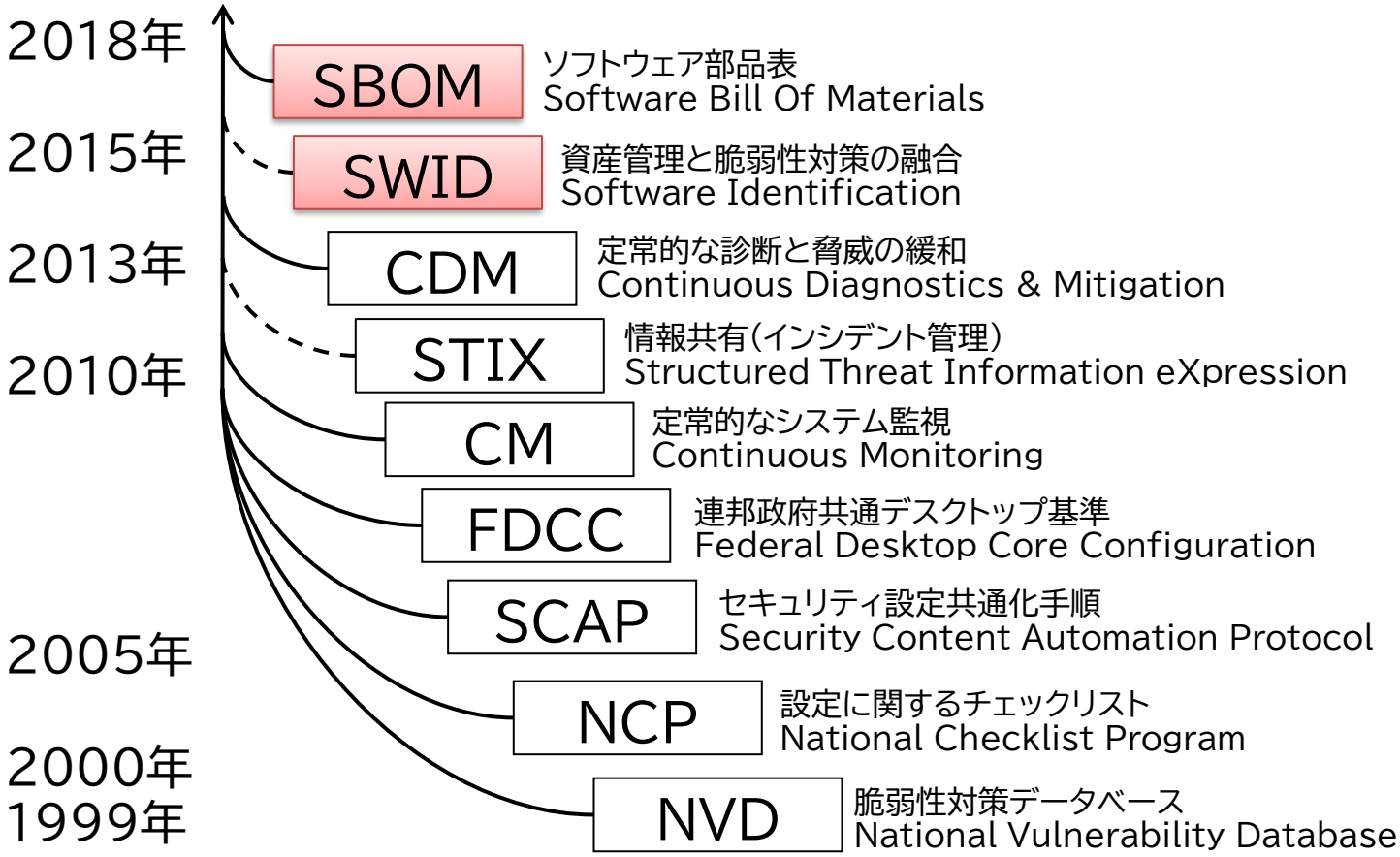
［出典］　サプライチェーンのサイバーレジリエンス　― セキュリティ透明性の向上と活用―
https://www.st-consortium.org/?download=1297&tmstv=1734422883

ISO/IEC 19770-2:2015
Software identification tag
SWID(ソフトウェア識別子)に
肩入れしているのでは？と思う方もいるので、
誤解を解くところから始めます。

NISTの取組み

機械処理可能な情報活用基盤の整備

- 米国でのSecurity Automation(機械/コンピュータ処理可能な基盤)は、拡がり始めた。



NISTの取組み

資産管理と脆弱性対策の融合のきっかけ

- ソフトウェアライセンス契約に準拠すること。

2014年01月09日時点
http://tagvault.org/government/
米国では、大統領執行命令第13103号（1998年9月30日）により、すべての政府機関が正確なソフトウェアインベントリを保持し、ソフトウェアライセンス契約に準拠することが義務付けられている。

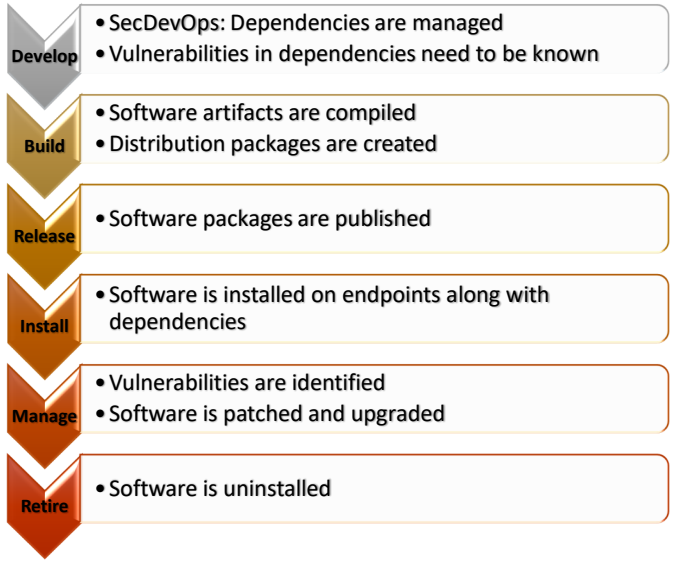
NISTの取組み

資産管理と脆弱性対策の融合のきっかけ

- NVDが抱えている悩みを解決すること。

CPE Myth: CPEs are produced by software suppliers

- ~5% of CPE Names are provided by software providers around the point of software “release”
- ~95% of CPE Names are created by NVD analysts during the “manage” phase
  - Produced during the vulnerability analysis process
  - **Software is identified after a known vulnerability is found**
- Identifying software after a vulnerability is discovered is way too late!



9

［出典］ SCAPv2 April Developer Days Face to Face (2019/04/30～05/02)
https://csrc.nist.gov/Projects/security-content-automation-protocol-v2/Material-Archives

NISTの取組み

資産管理と脆弱性対策の融合のきっかけ

- NVDが抱えている悩みを解決すること。
  - 製品識別子(CPE:Common Platform Enumeration)とは、情報システムを構成するハードウェア、ソフトウェアの名称をプログラムで(機械)処理しやすい形式で記述する仕様のこと。
  - 製品識別子(CPE)付与の95%は、脆弱性が発見されてから、、、
    ≒ リファレンス(製品辞書)に登録されるのは、脆弱性が発見されてから、、、

IPAが提供するMyJVN

IPAが提供するマイ・ジェイ・ブイ・エヌ

アイ・ピー・エーが
提供するMyJVN

情報処理推進機構が
提供するMyJVN

情報処理推進機構が
提供するマイ・ジェイ・ブイ・エヌ

CPE
common platform enumeration

cpe:/a:ipa:myjvn

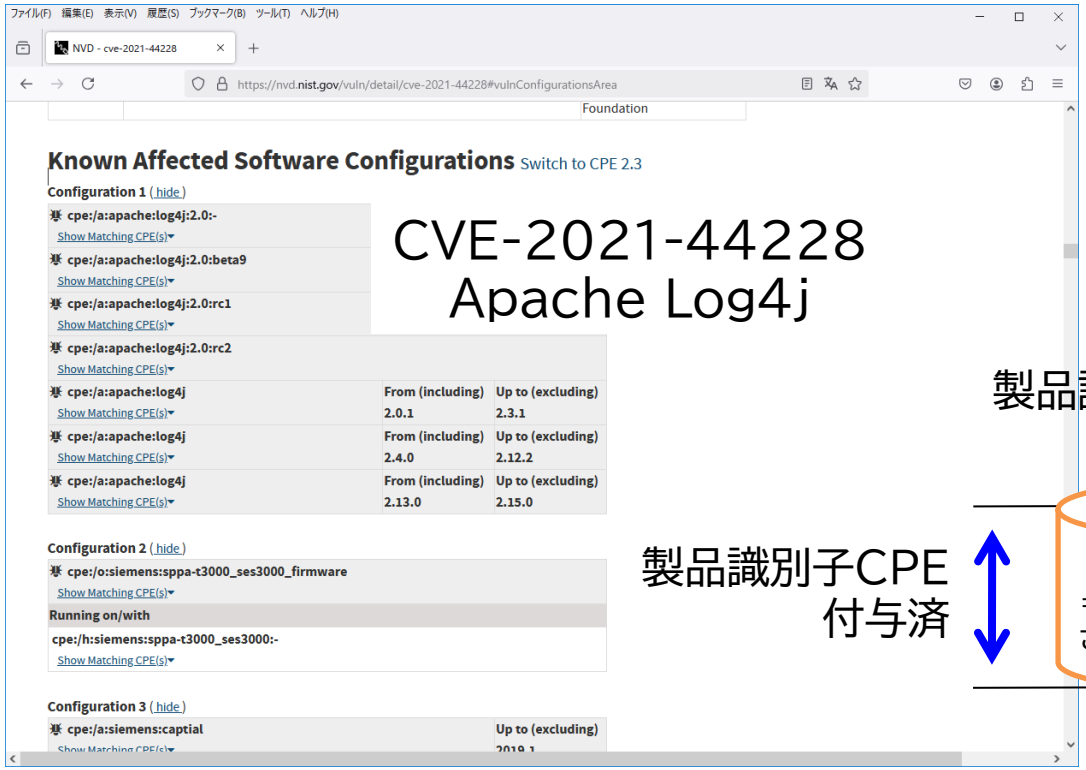
cpe:/{種別}:{ベンダ}:{製品}:{バージョン}
:{アップデート}:{エディション}:{言語}

種別:h=ハードウェア、o=OS、a=アプリケーション

NISTの取組み

資産管理と脆弱性対策の融合のきっかけ

- NVDが抱えている悩みを解決すること。
  - この課題を解決するために、NVDは製品識別子としてSWIDの採用推進
    - CPE：NVD主体で製品識別子を付与
    - SWID：製品ベンダ主体で製品識別子を付与（インストール時に同梱）



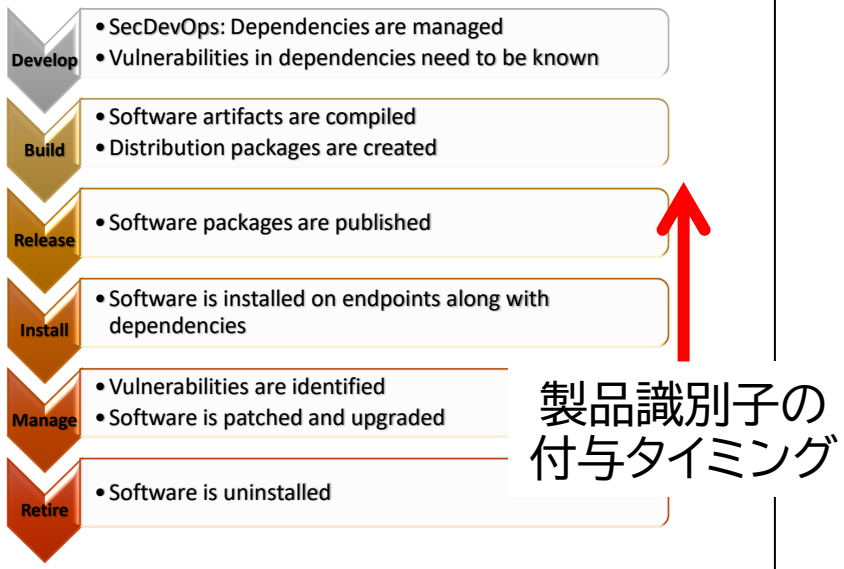
NISTの取組み

資産管理と脆弱性対策の融合のきっかけ

- NVDが抱えている悩みを解決すること。

CPE Myth: CPEs are produced by software suppliers

- ~5% of CPE Names are provided by software providers around the point of software “release”
- ~95% of CPE Names are created by NVD analysts during the “manage” phase
  - Produced during the vulnerability analysis process
  - **Software is identified after a known vulnerability is found**
- Identifying software after a vulnerability is discovered is way too late!



9

［出典］ SCAPv2 April Developer Days Face to Face (2019/04/30～05/02)
https://csrc.nist.gov/Projects/security-content-automation-protocol-v2/Material-Archives

NISTの取組み

資産管理と脆弱性対策の融合のきっかけ

- NVDが抱えている悩みを解決すること。

■ コマンドプロンプト □□□

```
C:¥ProgramData>dir reg*
C:¥ProgramData のディレクトリ

2021/12/24  01:04    <DIR>          regid.1986-12.com.adobe
2024/12/18  08:45    <DIR>          regid.1991-06.com.microsoft

C:¥ProgramData>dir  regid.1986-12.com.adobe
2021/12/24  01:04    <DIR>          .
2021/12/24  01:04    <DIR>          ..
2021/02/15  02:59             1,700 regid.1986-12.com.adobe_AcrobatPro-AS1-Win-GM-MUL.swidtag
2019/06/28  16:30             1,698 regid.1986-12.com.adobe_V7{}AcrobatCont-12-Win-GM.swidtag

C:¥ProgramData>dir  regid.1991-06.com.microsoft
2021/06/28  17:45             1,096 regid.1991-06.com.microsoft Office 16 Click-to-Run Extensibility
Component 64-bit Registration.swidtag
2021/07/20  03:35             1,076 regid.1991-06.com.microsoft Office 16 Click-to-Run Extensibility
Component.swidtag
2021/07/20  03:36             1,075 regid.1991-06.com.microsoft Office 16 Click-to-Run Localization
Component.swidtag
2024/12/18  09:02               999 regid.1991-06.com.microsoft_Windows-10-Pro.swidtag
```

NISTの取組み

資産管理と脆弱性対策の融合のきっかけ

- NVDが抱えている悩みを解決すること。

■ コマンドプロンプト □□□

```
C:¥ProgramData¥regid.1991-06.com.microsoft>type regid.1991-06.com.microsoft_Windows-10-Pro.swidtag
･ﾘ<?xml version="1.0" encoding="utf-8"?>
<software_identification_tag xmlns="http://standards.iso.org/iso/19770/-2/2009/schema.xsd">
        <entitlement_required_indicator>true</entitlement_required_indicator>
        <product_title>Windows 10 Pro</product_title>
        <product_version>
                <name>10.0.19041.2546</name>
                <numeric>
                        <major>10</major>
                        <minor>0</minor>
                        <build>19041</build>
                        <review>2546</review>
                </numeric>
        </product_version>
        <software_creator>
                <name>Microsoft Corporation</name>
                <regid>regid.1991-06.com.microsoft</regid>
        </software_creator>

中略

</software_identification_tag>
```

NISTがどのくらい力を入れていたのでしょうか？

SWIDは、どんな仕様なのでしょうか？

IT資産管理

ISO/IEC 19770 ITAM(IT資産管理)

規格名	概要	状況
ISO/IEC 19770-1:2017 Requirements 要求事項	IT資産管理(ITAM)のためのマネジメントシステムの確立，実施，維持及び改善の要求事項について規定している。	JIS X 0164-1:2019
ISO/IEC 19770-2:2009 Software identification tag SWID(ソフトウェア識別子)	ソフトウェアの導入状況を把握するために、導入されたソフトウェアを識別するためのタグの規格である。	JIS X 0164-2:2018 (ISO/IEC 19770-2:2015)
ISO/IEC 19770-2:2015 Software identification tag SWID(ソフトウェア識別子)	ISO/IEC 19770-2:2009のタグ必須の多くが任意となった。また、パッチ対応属性(delta)、ハッシュ値属性(sha1，sha256など) がサポートされるようになった。	
RFC 9393 (2023年6月28日) Concise Software Identification Tags (CoSWID)	ISO/IEC 19770-2:2015の記述軽量版で制約を備えたデバイスを想定した仕様である。	2016年3月から 検討開始
ISO/IEC 19770-3:2016 Entitlement schema 権利スキーマ	導入されているソフトウェアのライセンス情報を記述するためのタグの規格である。	JISX 0164-3:2019
ISO/IEC 19770-4:2017 Resource Utilization Measurement 資源利用測定	IT資産の使用に伴うリソースの消費に関する規格である。	JISX 0164-4:2019
ISO/IEC 19770-5:2015 Overview and vocabulary 概要及び用語	IT資産管理のコンセプトや原理を概説し、ISO/IEC 19770シリーズで用いられる用語の定義や、各規格の関連について説明した規格である。	JISX 0164-5:2019
NISTIR 8060 Guidelines for the Creation of Interoperable Software Identification (SWID) Tags	ISO/IEC 19770-2「ソフトウェア識別タグ」の生成に関するガイドライン	2016年4月発行
NISTIR 8085 Forming Common Platform Enumeration (CPE) Names from Software Identification (SWID) Tags.	ISO/IEC 19770-2「ソフトウェア識別タグ」から X.1528「製品識別子」の生成に関する指針	2015年12月 ドラフト発行

IT資産管理

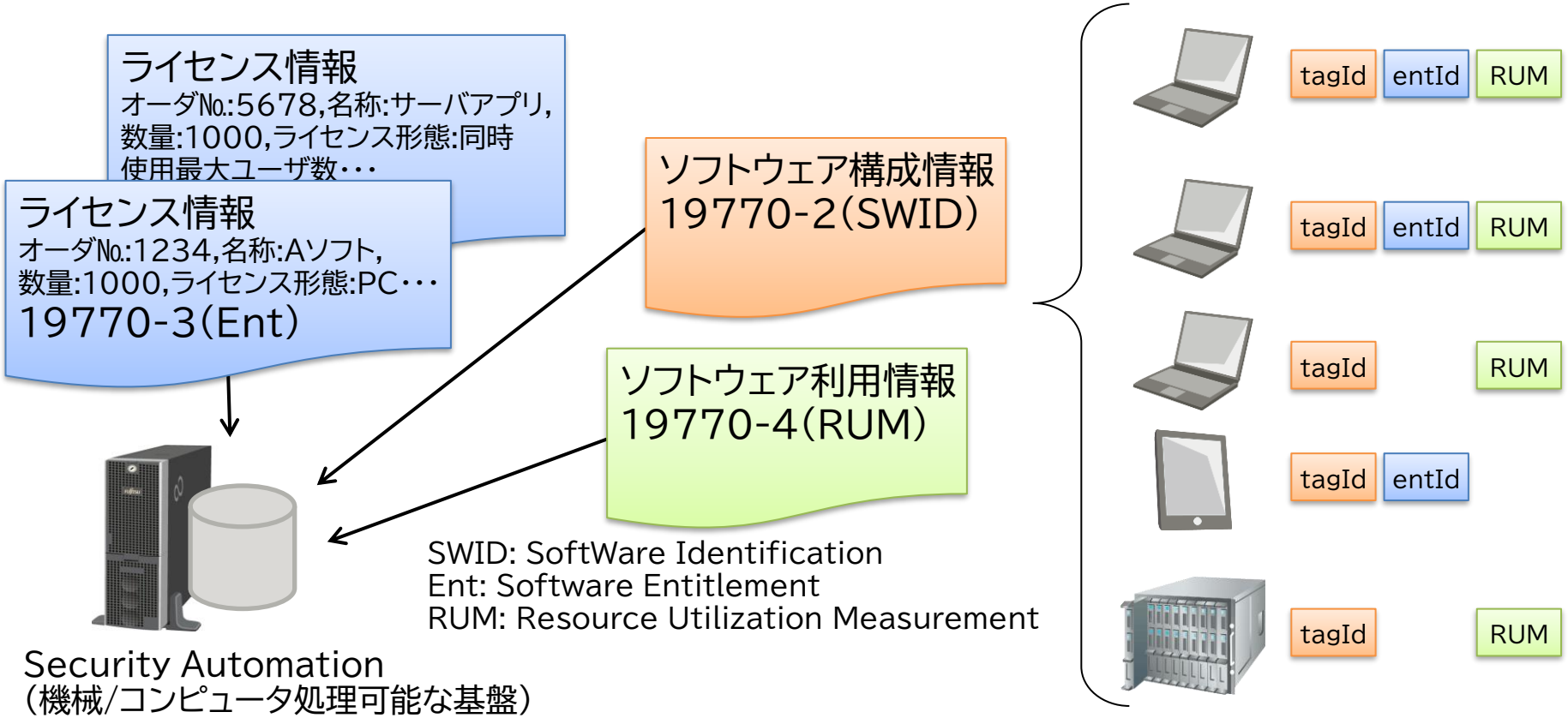
ISO/IEC 19770　ITAM(IT資産管理)

規格名	概要	状況
ISO/IEC 19770-6:2024 Hardware identification tag	HWIDタグとしてのハードウェア識別(HWID)データのカプセル化の規格である。	
ISO/IEC 19770-8:2020 Guidelines for mapping of industry practices to/from the ISO/IEC 19770 family of standards	業界の慣行をISO/IEC 19770シリーズにマッピングする方法を定義するマッピングドキュメントを作成するときに使用する要件、ガイドライン、形式、およびアプローチを定義する。	
ISO/IEC DTS 19770-10 Guidance for implementing ITAM	ITAMシステムを実装し、現在の市場の需要(調査とその後の検証活動によって決定)を満たすためのガイダンスを提供する。	
ISO/IEC 19770-11:2021 Requirements for bodies providing audit and certification of IT asset management systems	ISO/IEC 19770-1に従ってITAMSの監査および認証を提供する認証機関の要件を指定し、ガイダンスを提供する。	

IT資産管理

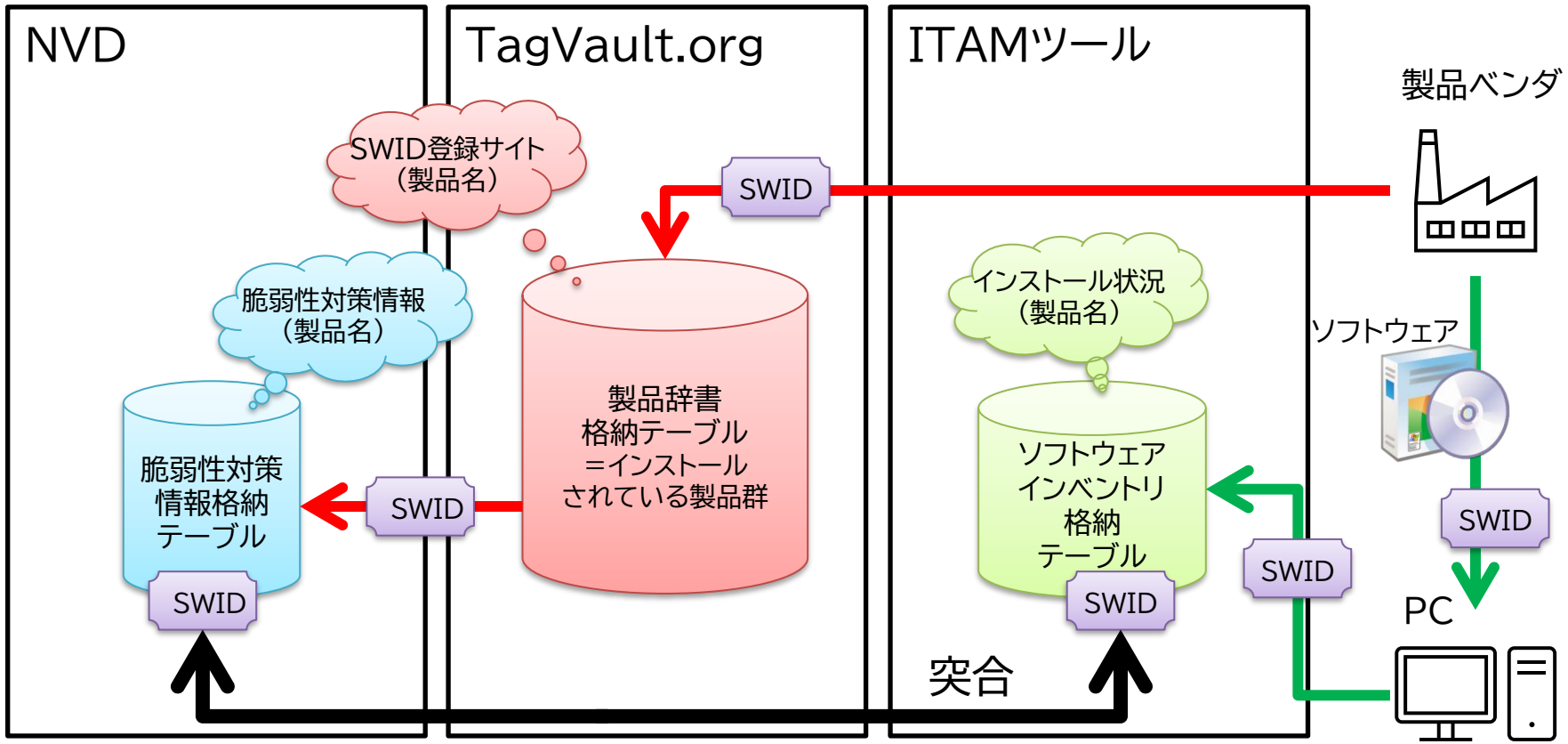
ISO/IEC 19770　ITAM(IT資産管理)

- 各種情報を集約し、自動チェックするなどの基盤整備に利用可能
- ソフトウェア構成情報(SWID)は19770-2、ライセンス情報(Ent)は19770-3、リソース利用情報(RUM)は19770-4で情報構造が標準化される。



ISO/IEC 19770-2 製品識別子(SWID)の仕組み

- ISO/IEC 19770-2の仕様で明記していることは、製品識別子(SWID)のXMLファイルをソフトウェアインストール時に格納すること。



なぜ、うまくいかなかったのでしょう？
おそらく

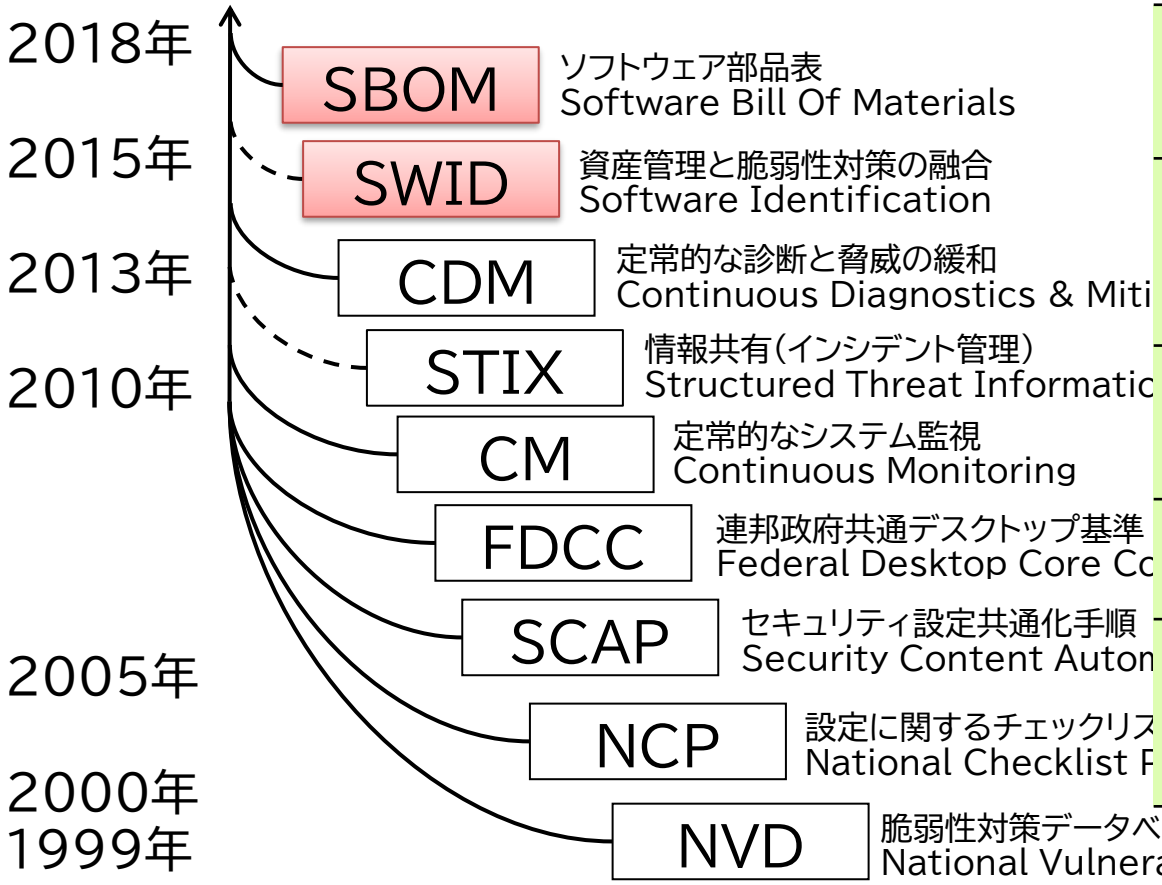
- TagVault.orgに加盟する必要がある。
- SWIDタグ登録には費用が掛かる。

その頃、どのような活動をしていたのか
振り返ります。

MyJVN(JVN脆弱性対策機械処理基盤)

国内脆弱性対策情報データベースのあゆみ

- グローバルな利活用基盤整備にあたってはNVDとの連携を考慮



第5期 利活用基盤普及Ⅱ 2019年～
MyJVN脆弱性と資産管理連携への対応
総務省実証実験参加(脆弱性/資産管理連携)

第4期 利活用基盤普及Ⅰ 2013年～2018年
総務省実証実験(脅威情報)(2016)
ソフトウェア識別(SWID)タグ活用導入(2013)
脆弱性と資産管理連携検討開始(2013)

第3期 利活用基盤導入 2008年～2012年
MyJVN開始(2008)
CPE導入(2008)

第2期 対策情報充実 2007年～2008年
CVE、CVSS、CWE導入(2007～2008)

第1期 立上げ 2002年～2007年
JVNiPedia開始(2007)
JVN開始(2004)
試行サイト開始(2002)

MyJVN(JVN脆弱性対策機械処理基盤)

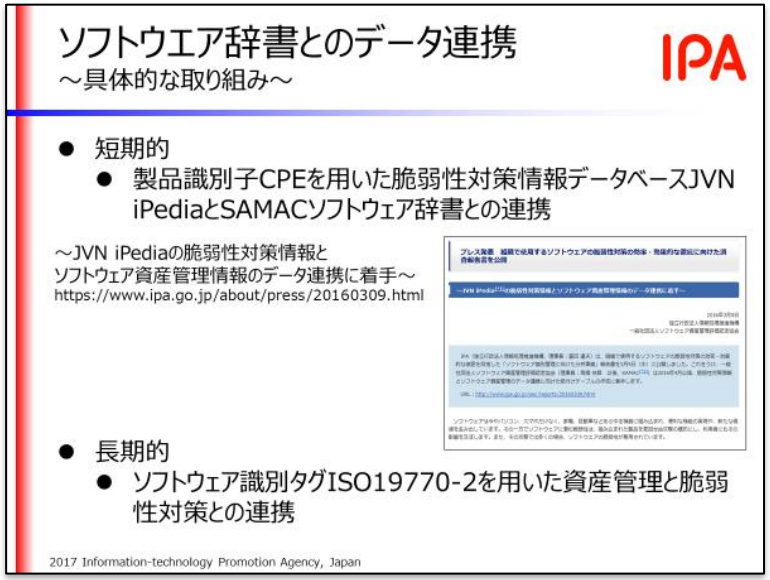
資産情報と脆弱性対策情報との連携

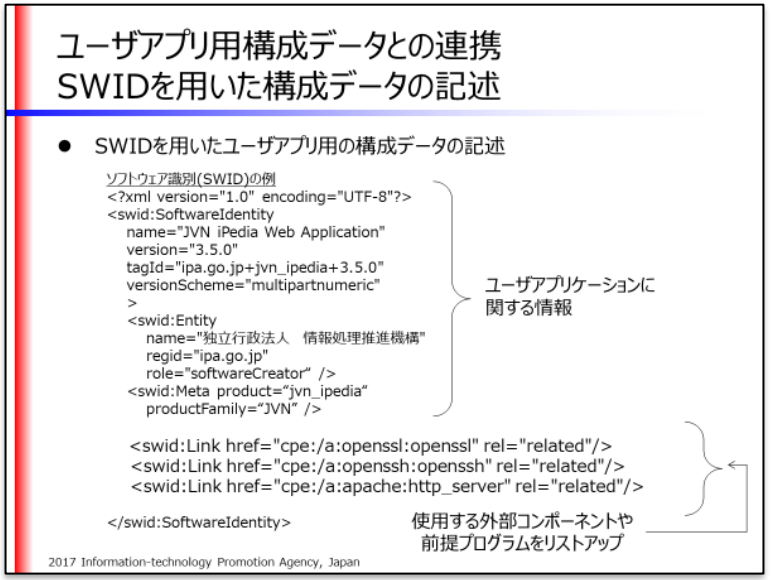
2013年	4月	SWID(ソフトウェア識別子)活用の検討開始
2014年	11月	MyJVN脆弱性対策情報収集ツールV3のリリース ● CPE(共通プラットフォーム一覧) ● SWID(ソフトウェア識別子)
2016年	2月	ICT-ISACの一般社団法人化にあわせて情報活用基盤の検討を開始 一般社団法人ICT-ISACにおいて、総務省:サイバー攻撃の防御に向けた情報共有基盤に関する実証事業を通じたSecurity Automation(機械/コンピュータ処理可能な基盤)の普及啓発を開始
	3月	一般社団法人 IT資産管理評価認定協会(SAMAC)を通じた資産管理と脆弱性管理連携の普及啓発を開始 ● JVN iPediaの脆弱性対策情報とソフトウェア資産管理情報のデータ連携に着手 https://www.ipa.go.jp/about/press/20160309.html ● 「ソフトウェア識別管理に向けた分析事業」の報告書の公開 ● 第72回CSEC研究発表会にて「製品識別子を用いた脆弱性対策情報データベースと資産管理との連携に関する検討 」を発表
2019年	4月	一般社団法人ICT-ISACにおいて、総務省:ソフトウェア脆弱性を狙ったサイバー攻撃の防御に向けた情報共有基盤に関する実証事業を通じた資産情報と、脅威情報、脆弱性対策情報との連携の普及啓発を開始

MyJVN(JVN脆弱性対策機械処理基盤)

資産情報と脆弱性対策情報との連携

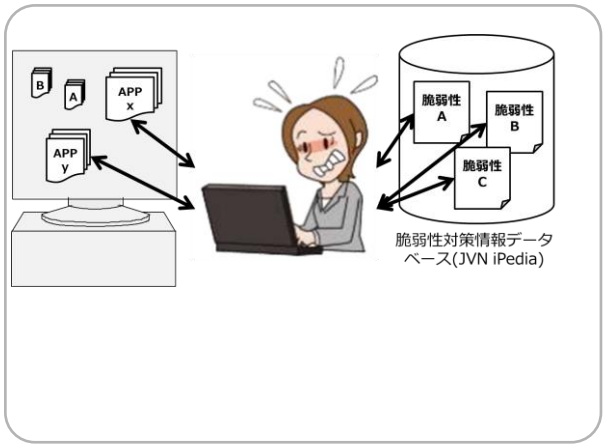
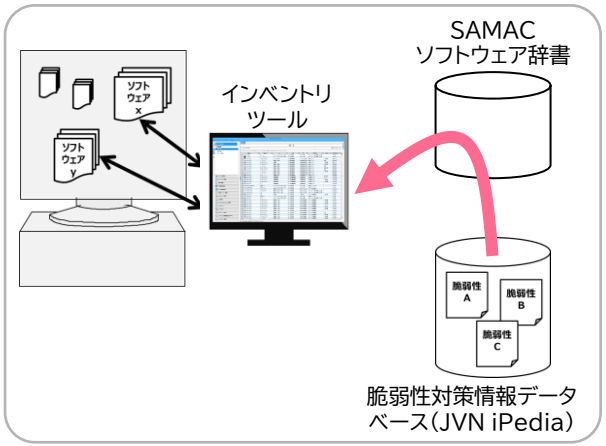
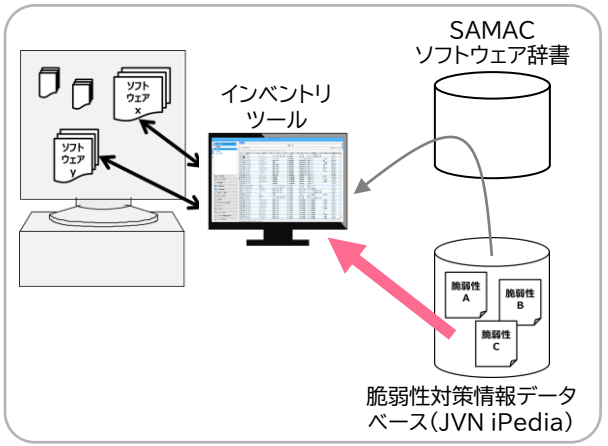
- 資産情報と脆弱性対策情報との連携
  - 短期的:製品識別子CPEを用いた脆弱性対策情報データベースJVN iPediaとSAMACソフトウェア辞書との連携
  - 長期的:ISO/IEC 19770-2 SWID(ソフトウェア識別子)を用いた資産管理と脆弱性対策との連携
- 資産管理と脆弱性管理との連携拡大(SBOMに近い概念を提案)
  - カスタムアプリケーションと、カスタムアプリケーションで使用している外部コンポーネントや前提プログラムとを把握し管理





MyJVN(JVN脆弱性対策機械処理基盤)

資産情報と脆弱性対策情報との連携

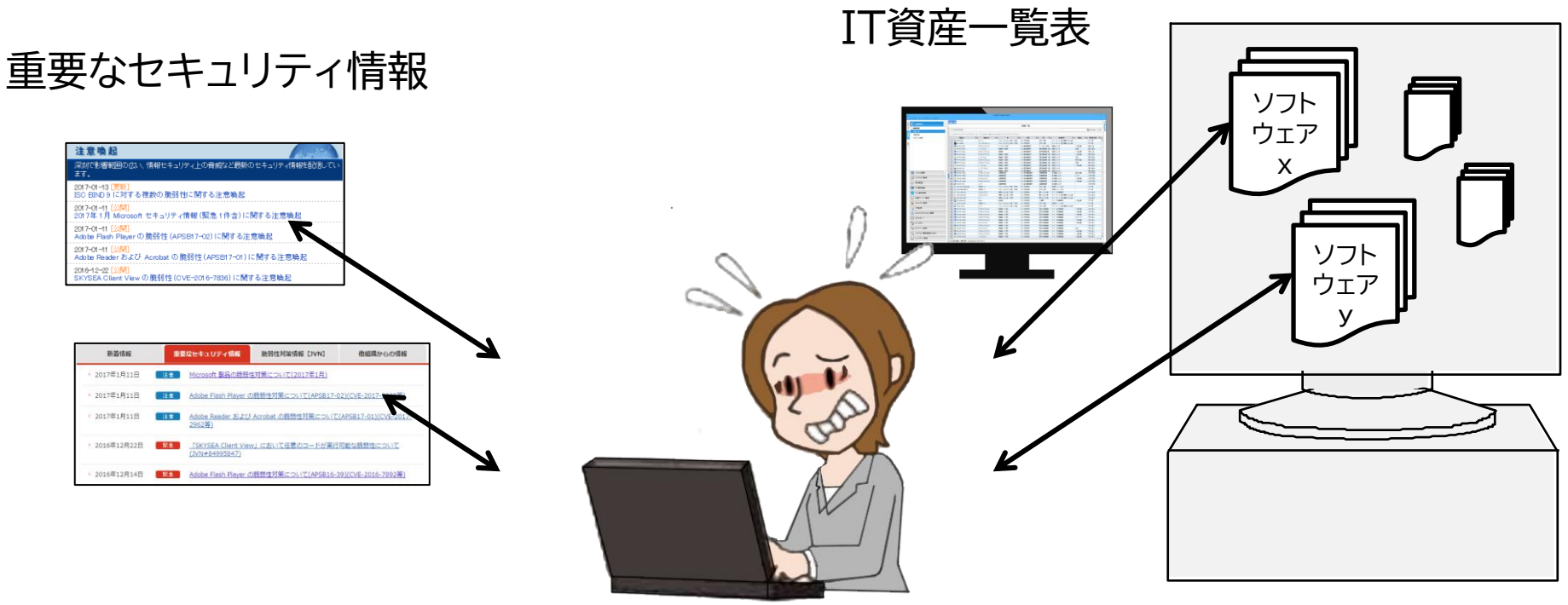
現在	① 現在の取組み 『JVN iPediaの活用』	② 将来的な連携 『JVN iPedia SWID連携』
		
人が脆弱性対策情報を検索し、各ソフトウェアを確認する。	ソフトウェア辞書を介して脆弱性対策情報と連携する。	SWIDを利用すると直接インベントリツールと連携できる。
『全部なんて調べきれない！』	『辞書登録にあるものは、ツールで調べられる！』	『辞書登録に関わらず、SWIDがあればツールで調べられる！』
『いつの間にか脆弱性のあるソフトが入ってた！』	『でも直近の新しい情報は反映が遅くなってしまう･･･。』	『かなりリアルタイムに情報が反映される！』

製品識別における課題を振り返ります。

製品識別における課題

インストール状況と脆弱性との紐付けは人手で実施

- 重要なセキュリティ情報が発信されるたびに、手作業でIT資産一覧表を検索して、影響範囲の調査をしていませんか？
  - 多くの場合、インストール状況と脆弱性との紐付けを人手で実施している（資産管理と脆弱性対策とが連携できているわけではない）。



製品識別における課題

カスタムアプリ管理は整備途上

- 重要なセキュリティ情報が発信されたときに、カスタムアプリ（SIで開発した業務アプリケーションなど）の影響範囲を調査していますか？
  - 多くの場合、カスタムアプリ（SIで開発したアプリケーションなど）は、資産管理や脆弱性管理の対象に含まれていない。

重要なセキュリティ情報

?

カスタムアプリ
Ex. 在庫管理アプリ

製品識別における課題

望ましい環境

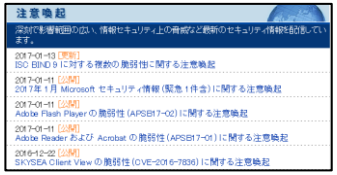
- 製品やカスタムアプリ(SIで開発した業務アプリケーションなど)に関わらず、情報連携により、人が介在しなくても良い脆弱性対策環境(脆弱性の影響有無の判定など)が実現できる。

製品識別における課題

ソフトウェア名が同じものであるという前提

- 【Question】情報連携により、人が介在しなくても良い脆弱性対策環境（脆弱性の影響有無の判定など）を実現するには、重要なセキュリティ情報で使用するソフトウェア名と、IT資産一覧表で使用するソフトウェア名が同じものである（or 製品識別子が同一）という前提で運用できること。

重要なセキュリティ情報
（MyJVN）



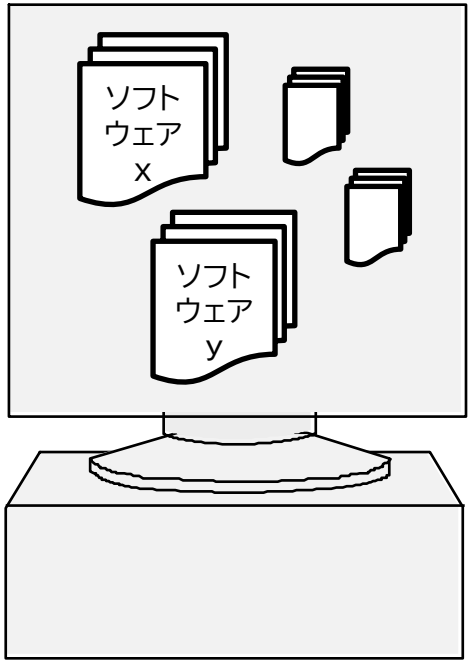
重要なセキュリティ情報に
記載されているソフトウェア名
- 製品ABC

同じ？

同じ

IT資産一覧表
（IT資産管理ツール）

IT資産一覧表に登録
されているソフトウェア名
- 製品ABC



製品識別における課題

インベントリに登録されているデータの表記揺れ

- 脆弱性対策情報データベースとインベントリに登録されているデータの表記揺れがあり、紐付けが難しいと言われている。
  - 名前(ベンダ名、製品名など)の表記揺れ
  - 製品識別子の表記揺れ
  - その他(買収による名前の変更、コードの独自改修による名前の変更など)

- 名前(ベンダ名、製品名など)の表記揺れ
  - 同一ベンダ・同一製品なものの、異なる基準により情報登録がされた状態
    - ベンダ名
      - 株式会社の表記の揺れ
      - 記号の表記の揺れ
      - 日本語、英語の揺れ
    - 製品名
      - セパレータの表記の揺れ
      - 日本語、英語の揺れ
      - ベンダ名、製品名、バージョン情報カラムの揺れ

製品識別における課題

インベントリに登録されているデータの表記揺れ

表記揺れ
- 企業形態によるもの
- 言語によるもの
- 記号によるもの

企業形態	言語	記号
Dendai	Dendai	Dendai inc
Dendai 株式会社	電大	Dendai inc.
Dendai Company, Ltd.	(株)Dendai	Dendai,Inc
株式会社 電大	Dendai 株式会社	Dendai,inc.

カラム揺れ
- 製品名で多く見られる
- 製品名カラムに別のカラム情報が入っている

カラム揺れ
Learn Japan
Learn Japan 19.1.1
Learn Japan 10.1(x64 edition)
Learn Japan 19.1.1(X64)

製品識別における課題

調査:インベントリに登録されているデータの表記揺れ

- 調査方法
  - インベントリデータ283, 326件(収集ツールは1種類)を対象に、
  - ベンダ名、製品名に対して変換処理した後で一意性を調査
    - 変更: 大文字を小文字に変更
    - 置換: 特定の文字を置換(ex. “ “ → “_” など)
    - 削除: 特定の情報を削除(ex. Inc、corp、株式会社、x64など)

- 変換例(変更 + 置換 + 削除)

変換前	
ベンダ名	製品名
Dendai	Learn Japan
Dendai Company, Ltd.	Learn Japan 19.1.1
株式会社 電大	Learn Japan19.1.1(X64)

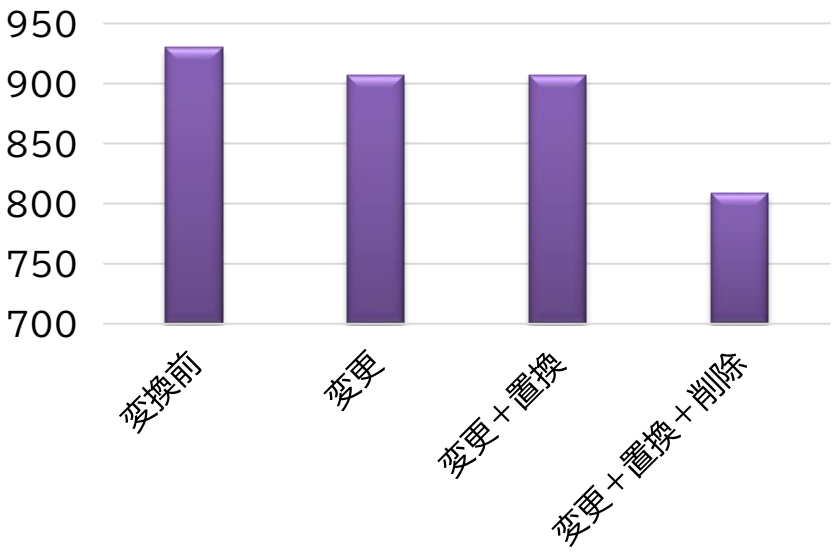
変換後	
ベンダ名	製品名
dendai	learn_Japan
dendai	learn_japan
電大	learn_japan

製品識別における課題

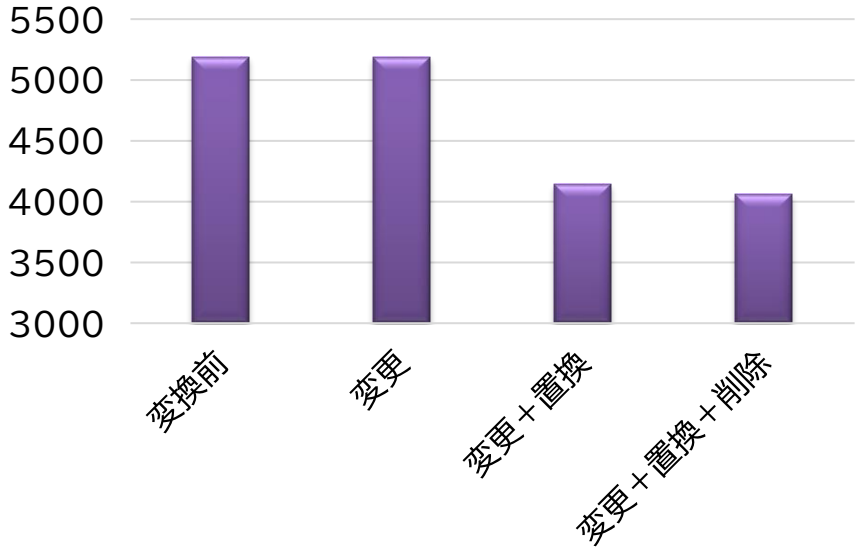
調査結果:インベントリに登録されているデータの表記揺れ

- ベンダ名では、変更、変更＋置換により約30件、削除を追加することで約100件のユニークベンダ数の減少
- 製品名では、変更だけでは変化しないものの、置換を追加することで約1,000件、削除を追加することで約1,100件のユニーク製品数の減少

ベンダ名



製品名



製品識別における課題

調査：脆弱性対策情報データベースでの製品検索のヒット率

- 調査方法
  - インベントリデータ283，586件（収集ツールは2種類）と、
  - 脆弱性対策情報データベースJVN iPediaに登録されているベンダ名（27，783件）、製品名（72，075件）を対象に検索ヒット率を調査

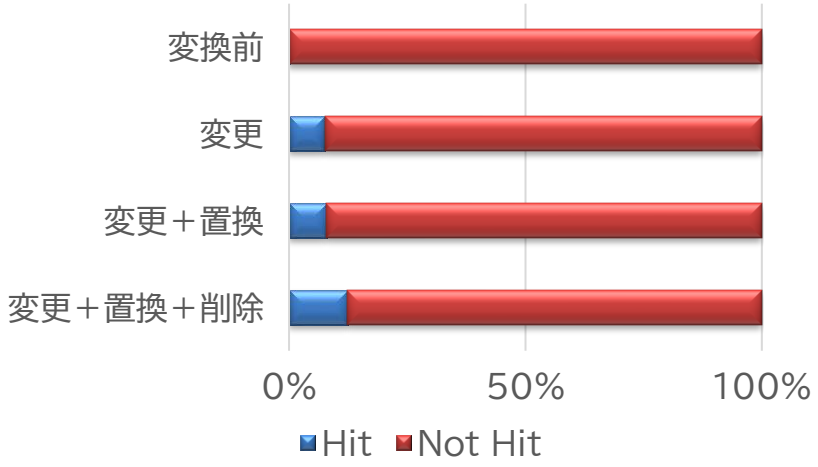
- 検索ヒット率
  - インベントリに登録されているデータ（ベンダ名、製品名）に対し、変換（変更＋置換＋削除）処理した後、（仮の）製品識別子CPEを生成して脆弱性対策情報データベースに登録されているデータ（ベンダ名、製品名）を検索
  - 検索エンジンにはElasticsearchを使用
  - 検索ヒット率には“完全一致” “あいまい検索（Fuzzyを使用）”を使用

製品識別における課題

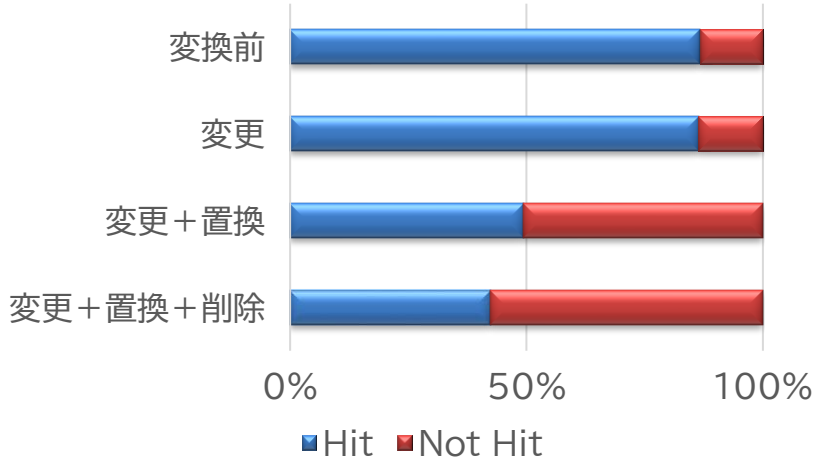
調査結果：脆弱性対策情報データベースでの製品検索のヒット率

- ベンダ名
  - 完全一致による検索では、取得した情報をそのまま検索してもほとんどヒットしないが、変換を加えることでヒット率が上昇
  - あいまい検索では、変換を加えることで、ヒット率が低下
    - “株式会社”“inc”等の共通する部分で、誤った情報が多くヒットしていると推測

完全一致



あいまい検索

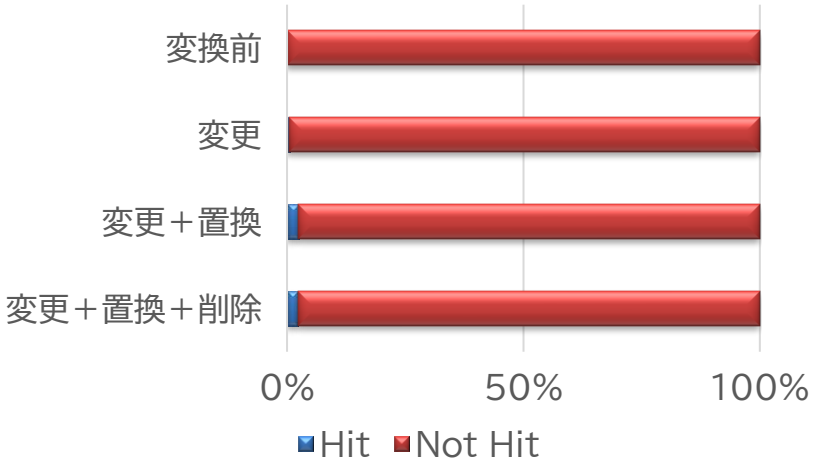


製品識別における課題

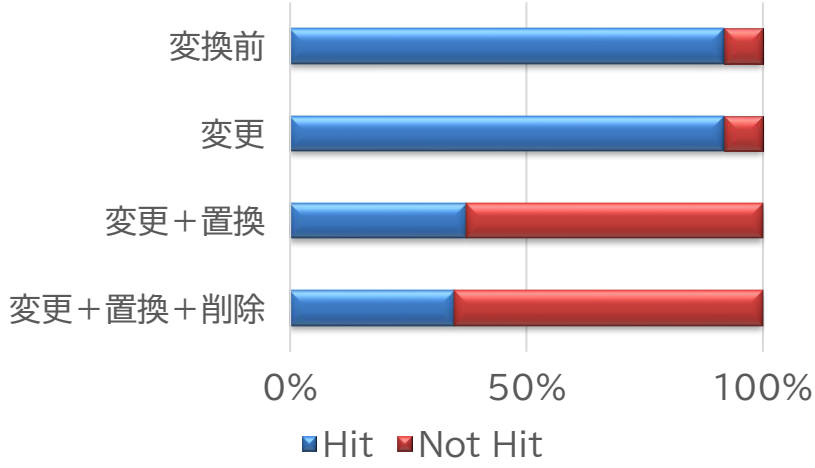
調査結果：脆弱性対策情報データベースでの製品検索のヒット率

- 製品名
  - 完全一致による検索では、ベンダ名と同様に、取得した情報をそのまま検索してもほとんどマッチングしないが、変換を加えることでヒット率が上昇
  - あいまい検索でも、変換を加えることでヒット率が低下
    - Edition情報やメジャーバージョンなどの共通する部分で、誤った情報が多くヒットしていると推測

完全一致



あいまい検索



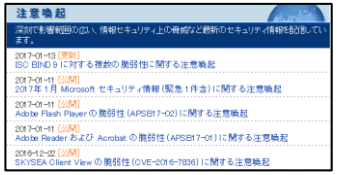
インベントリデータをそのまま使うのは
難しいということが、なかなか伝わらない。

製品識別における課題

ソフトウェア名が同じものであるという前提

- 【Question】情報連携により、人が介在しなくても良い脆弱性対策環境（脆弱性の影響有無の判定など）を実現するには、重要なセキュリティ情報で使用するソフトウェア名と、IT資産一覧表で使用するソフトウェア名が同じものである（or 製品識別子が同一）という前提で運用できること。

重要なセキュリティ情報
（MyJVN）



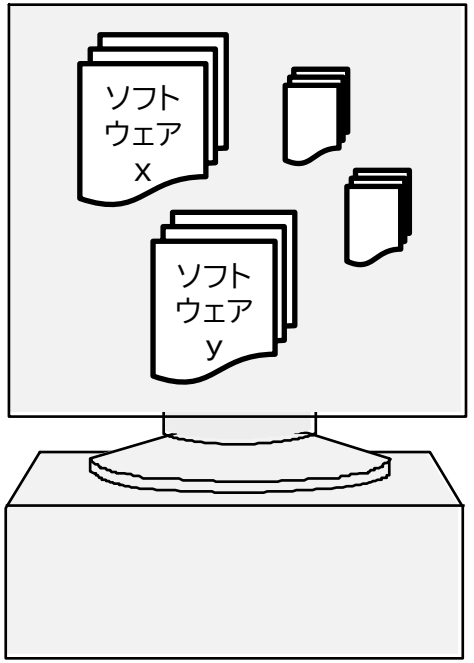
重要なセキュリティ情報に
記載されているソフトウェア名
- 製品ABC

同じ？

同じ

IT資産一覧表に登録
されているソフトウェア名
- 製品ABC

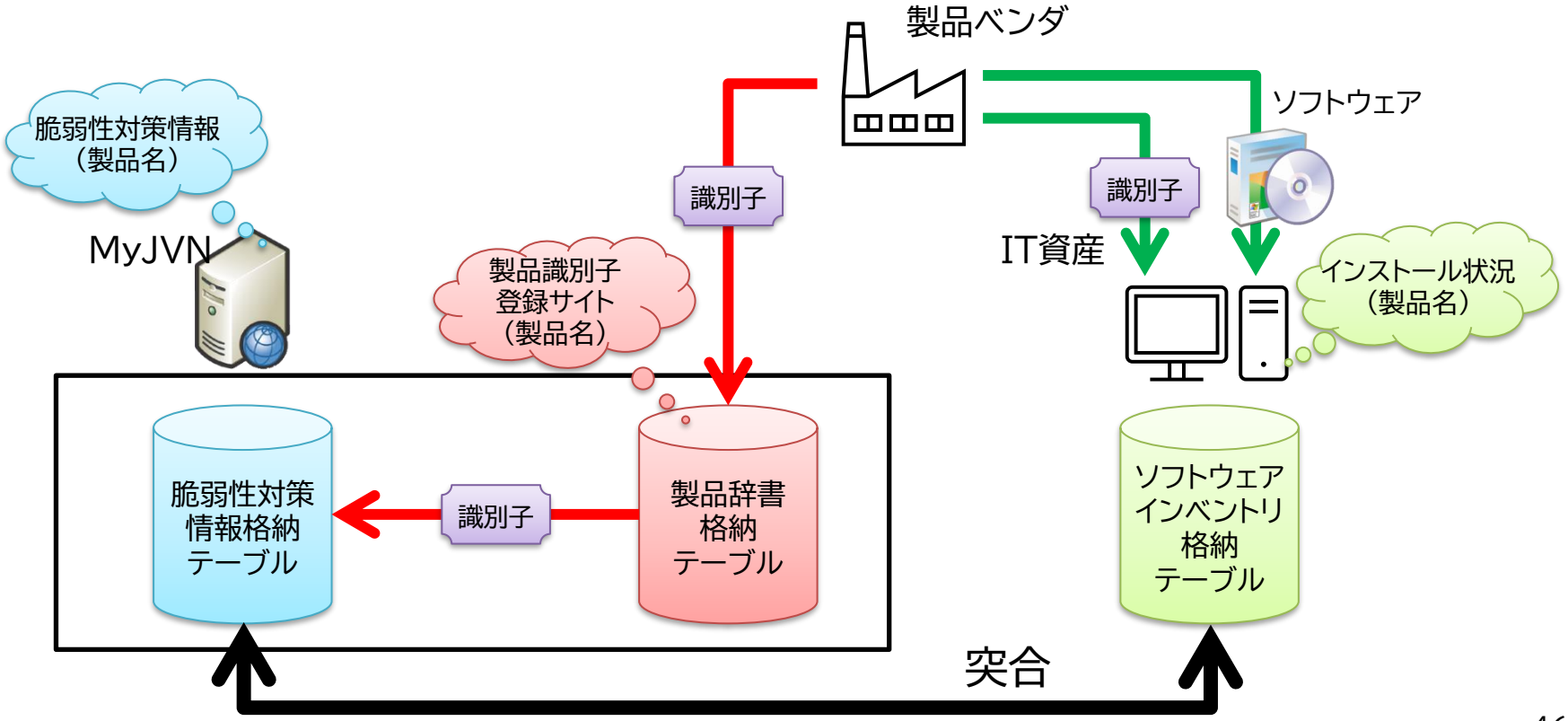
IT資産一覧表
（IT資産管理ツール）



製品識別における課題解決のアプローチ

より望ましい環境(理想):製品識別子を用いた連携

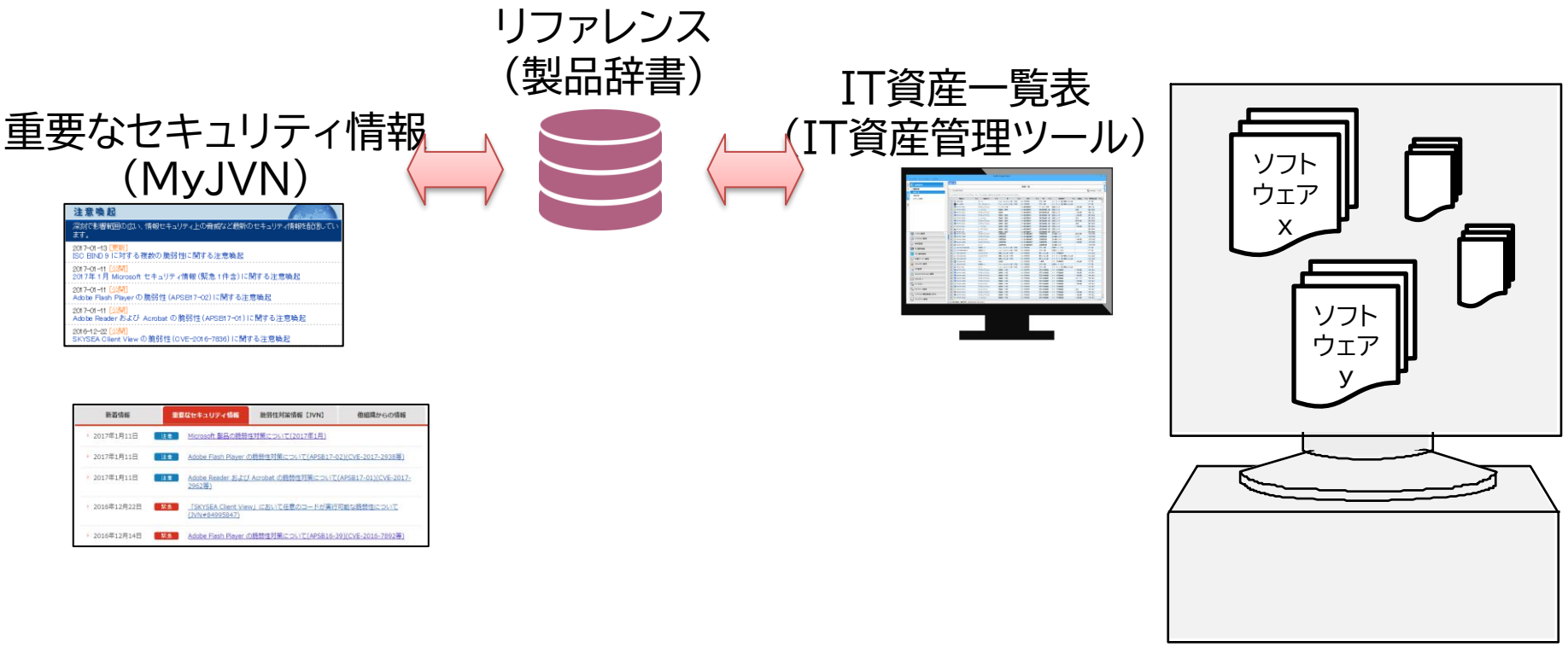
- 表記揺れのない管理された製品識別子を利用できれば、
  - インストール状況(製品名)と製品識別子登録サイト(製品名)が一致し、
  - インストール状況(製品名)と脆弱性対策情報(製品名)のデータも一致するので、名寄せなどの変換が不要となるはず。



製品識別における課題解決のアプローチ

ソフトウェア名が同じものであるという前提

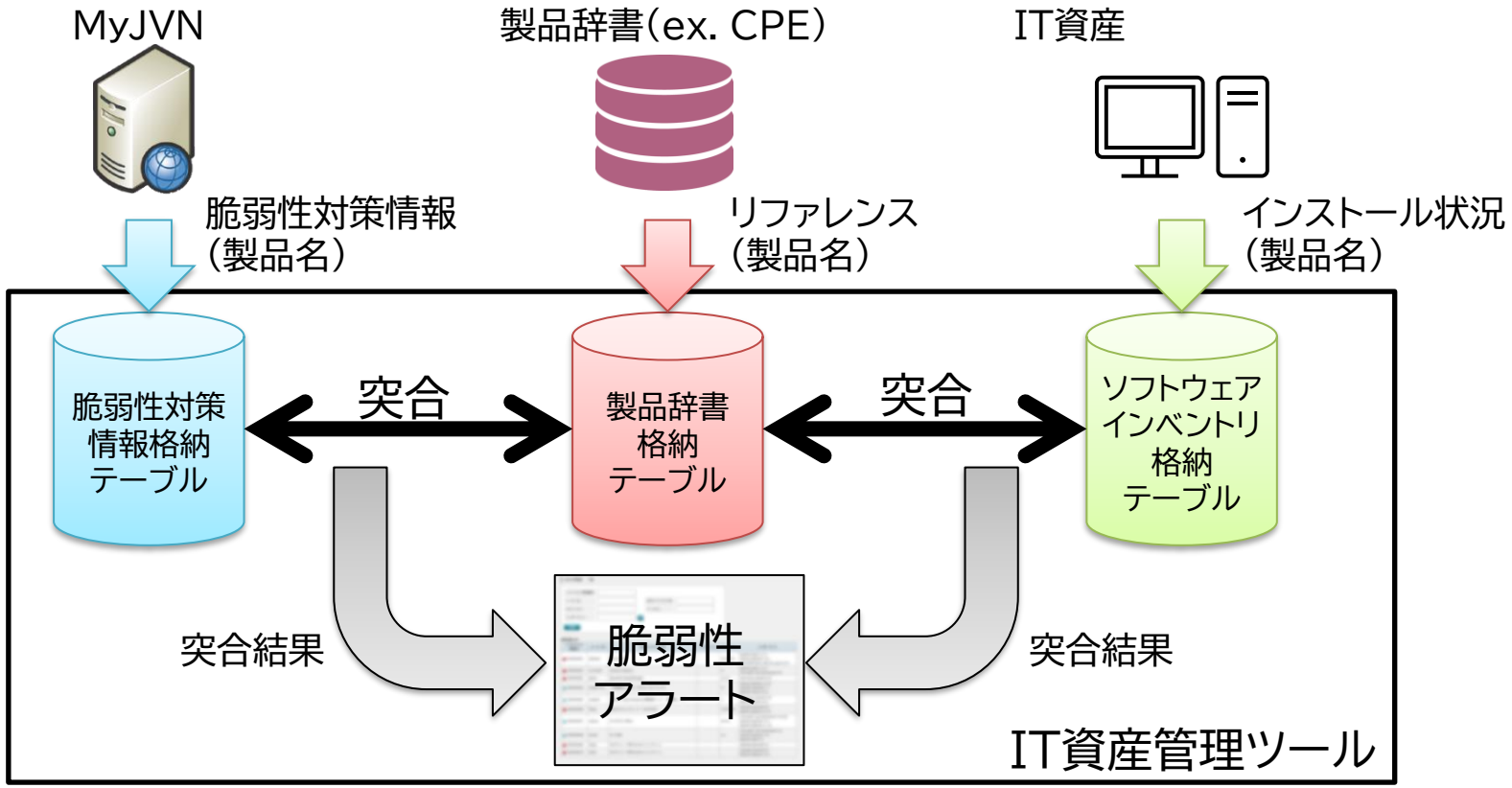
- 【Answer】リファレンス（製品辞書）を利用することで、重要なセキュリティ情報とIT資産一覧表との連携させ、ソフトウェア名が同じものである（or 製品識別子が同一）という前提を確保する。



製品識別における課題解決のアプローチ

製品識別子(ex. CPE)を用いた連携

- 脆弱性対策情報(製品名)とインストール状況(製品名)との対応表である製品辞書(ex. CPE)を用いた突合



製品識別における課題解決のアプローチ

製品識別子(ex. CPE)を用いた連携

- 製品辞書(CPEの場合)を用いた突合

インストール状況(製品名)

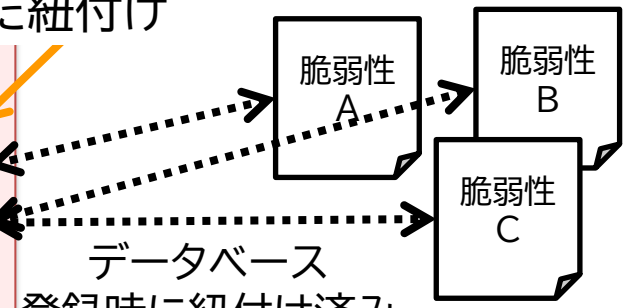
脆弱性対策情報(製品名)

製品辞書(CPEの場合)

ソフトウェアインベントリ項目				連携用項目
sw id	sw vendor	sw name	その他項目	CPE
・・・	・・・	Adobe Acrobat 8.2.0 Professional	・・・	cpe:/a:adobe:acrobat
・・・	・・・	Adobe Acrobat 8.2.0 Standard	・・・	cpe:/a:adobe:acrobat
・・・	・・・	Adobe Acrobat 8.2.1 - CPSID_50570	・・・	cpe:/a:adobe:acrobat
・・・	・・・	Adobe Acrobat 8.2.1 Professional	・・・	cpe:/a:adobe:acrobat
・・・	・・・	Adobe Acrobat 8.2.1 Standard	・・・	cpe:/a:adobe:acrobat
・・・	・・・	Adobe Acrobat 9.3.0 - CPSID_52073	・・・	cpe:/a:adobe:acrobat

製品識別子CPEを用いた紐付け

ソフトウェア名	CPE
Adobe Acrobat	cpe:/a:adobe:acrobat
製品Y	cpe:/a:y:yyy



では、SBOMにおいて、
製品識別子はどのようなものが使えるのでしょうか？

製品識別子

製品識別子使用にあたっての課題

どの製品識別子を使えば良いの？

- CPE v2.3
  cpe:2.3:{種別}:{ベンダ}:{製品}:{バージョン}:{アップデート}:{エディション}:{言語}:{sw_edition}:{target_sw}:{target_hw}:{その他}
- purl
  pkg:{タイプ}/{名前空間}/{パッケージ名}@{バージョン}?{修飾子}#{パス}
- SWID
  グローバルなユニークIDで、16バイトのUUID、またはタグ生成者によって定義された値

識別子 / 形式	CPE	Package-Manager	SWHIDs	Hash	SWID	
					UUID	ユニークID
SPDX	CPEフィールド使用(CPE 2.2, 2.3)	Package-Managerフィールド使用(purl)	swhフィールド使用	Otherフィールド使用		
CycloneDX		purlフィールド使用	SWIDフィールド使用	Hashフィールド使用	SWIDフィールド使用	
SWID	tagidフィールド使用					

製品識別子

製品識別子使用にあたっての課題

どの製品識別子を使えば良いの？

- CPE v2.3
  cpe:2.3:{種別}:{ベンダ}:{製品}:{バージョン}:{アップデート}:{エディション}:{言語}:{sw_edition}:{target_sw}:{target_hw}:{その他}
- purl
  pkg:{タイプ}/{名前空間}/{パッケージ名}@{バージョン}?{修飾子}#{パス}
- SWID
  グローバルなユニークIDで、16バイトのUUID、またはタグ生成者によって定義された値

代替案
Transparency Exchange Identifier
OWASPが提案する複数の製品識別子を統合する仕様

urn:tei:{TEI type}:{DNS domain}:{製品識別子}
urn:tei:uuid:products.example.com:d4d9f54a-abcf-11ee-･･･

でも、CPEの場合、どうやって記述するのだろう？
urn:tei:cpe:products.example.com:cpe:/a:ipa:myjvn

製品識別子

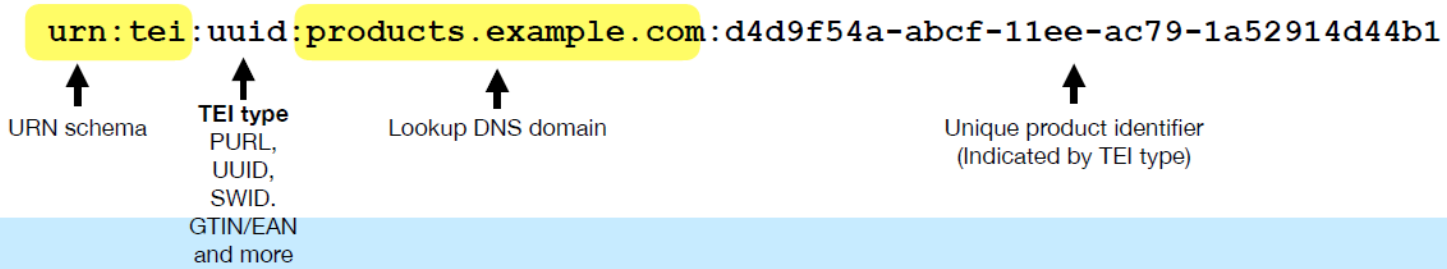
製品識別子使用にあたっての課題

No single identifier.

There is no single identifier for software. Any solution has to support as many existing identifiers as possible.

Introducing our proposal: the TEI URN.

TEI is the **Transparency Exchange Identifier**. A unique identifier created by the manufacturer for a specific product regardless of software version.



製品識別子

SBOMの書き方:SPDX

- SPDX(Software Package Data Exchange)
  ISO/IEC 5962:2021として国際標準化された規格

SPDXVersion: SPDX-2.2
DataLicense: CC0-1.0
DocumentNamespace: http://www.spdx.org/spdxdocs/8f141b09-1138-4fc5-aefb-fc10d9ac1eed ・・・一意な識別子
DocumentName: SBOM Example
SPDXID: SPDXRef-DOCUMENT
Creator: Organization: Company A ・・・SBOMの作成者
Created: 2022-05-09T13:00:00Z ・・・タイムスタンプ
Relationship: SPDXRef-DOCUMENT DESCRIBES SPDXRef-Application-v1.1 ・・・依存関係
PackageName: Application ・・・コンポーネント名
SPDXID: SPDXRef-Application-v1.1 ・・・一意な識別子
PackageVersion: 1.1 ・・・コンポーネントのバージョン
PackageSupplier: Organization: Company A ・・・サプライヤー名
PackageDownloadLocation: NOASSERTION
FilesAnalyzed: false
PackageChecksum: SHA1: 75068c26abbed3ad3980685bae21d7202d288317
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION
ExternalRef: SECURITY cpe23Type cpe:2.3:a:company_a:application:1.1:*:*:*:*:*:*:* ・・・一意な識別子
Relationship: SPDXRef-Application-v1.1 CONTAINS SPDXRef-Browser-v2.1 ・・・依存関係
Relationship: SPDXRef-Application-v1.1 CONTAINS SPDXRef-Protocol-v2.2 ・・・依存関係
(以下省略)

製品識別子

SBOMの書き方：CycloneDX

- CycloneDX
  - OWASPで開発されたSBOM用の仕様

```
<?xml version=”1.0” encoding=”utf-8”?>
<bomxmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
 xmlns:xsd=”http://www.w3.org/2001/XMLSchema”
 serialNumber=”urn:uuid:3e671687-395b-41f5-a30f-a58921a69b71” version=”1”  ･･･一意な識別子

 xmlns=”http://cyclonedx.org/schema/bom/1.3”><metadata>
<timestamp>2022-05-09T13:00:00Z</timestamp>                    ･･･タイムスタンプ
<authors><author><name>Company A</name></author></authors>  ･･･SBOMの作成者
<component type=”application”>
<name>Application</name>                                       ･･･コンポーネント名  ･･･依存関係
<version>1.1</version>                                         ･･･コンポーネントのバージョン
<hashes><hash alg=”SHA-
1”>75068c26abbed3ad3980685bae21d7202d288317</hash></hashes>
<cpe>cpe:2.3:a:company_a:application:1.1:*:*:*:*:*:*:*</cpe>   ･･･一意な識別子
<externalReferences/><components /></component>
<manufacture><name>Company A</name></manufacture>
<supplier><name>Company A</name></supplier></metadata>･･･サプライヤー名
(中略)
<dependencies>
<dependency ref=“pkg:maven/org.company_b/browser@2.1”>         ･･･依存関係
<dependency ref=”pkg:maven/org.c/CompressionEng@3.1” /></dependency>
<dependency ref=”pkg:maven/org.community_p/protocol@2.2” />  ･･･依存関係
</dependencies>
(以下省略)
```

SBOMの書き方：SWID

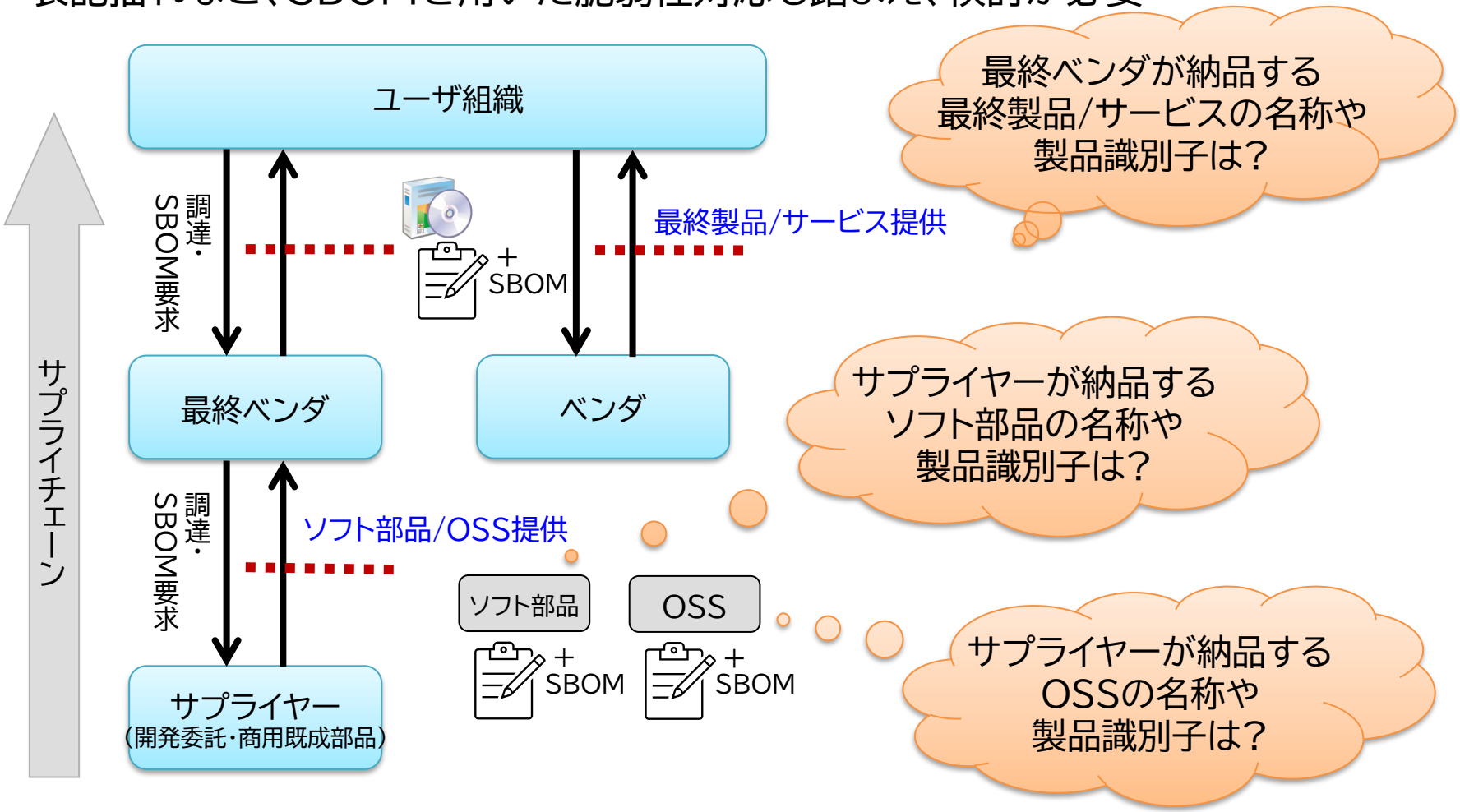
- SWID(Software identification tag、ソフトウェア識別子)
  - ISO/IEC 19770-2:2015として国際標準化された規格

```
<SoftwareIdentity
 xmlns="http://standards.iso.org/iso/19770/-2/2015/schema.xsd"
 xmlns:sha512="http://www.w3.org/2001/04/xmlenc#sha512"
 name="application"                                          ・・・コンポーネント名
 tagId="Company A/application@1.1"                           ・・・一意な識別子
 version="1.1">                                              ・・・コンポーネントのバージョン
 <Entity name="Company A" role="tagCreatorsoftwareCreator" /> ・・・サプライヤー名  ・・・SBOMの作成者
 <Meta title="Company AApplication v1.1" timestamp="2022-05-09T13:00:00Z" />・・・タイムスタンプ
 <Link href="swid:CompanyB/browser@2.1" rel="component" /> ・・・依存関係
 <Link href="swid:CommunityP/ptotocol@2.2" rel="component" /> ・・・依存関係
 <Payload >
 <File name="Company-A-application-1.1.exe"

sha512:hash="BC55DEF84538898754536AE47CC907387B8F61D9ACD7D3FB8B8A624199682C
8FBE6D163108
 8AE6A322CDDC4252D3564655CB234D3818962B0B75C35504D55689"/>
 </Payload>
 </SoftwareIdentity>
(以下省略)
```

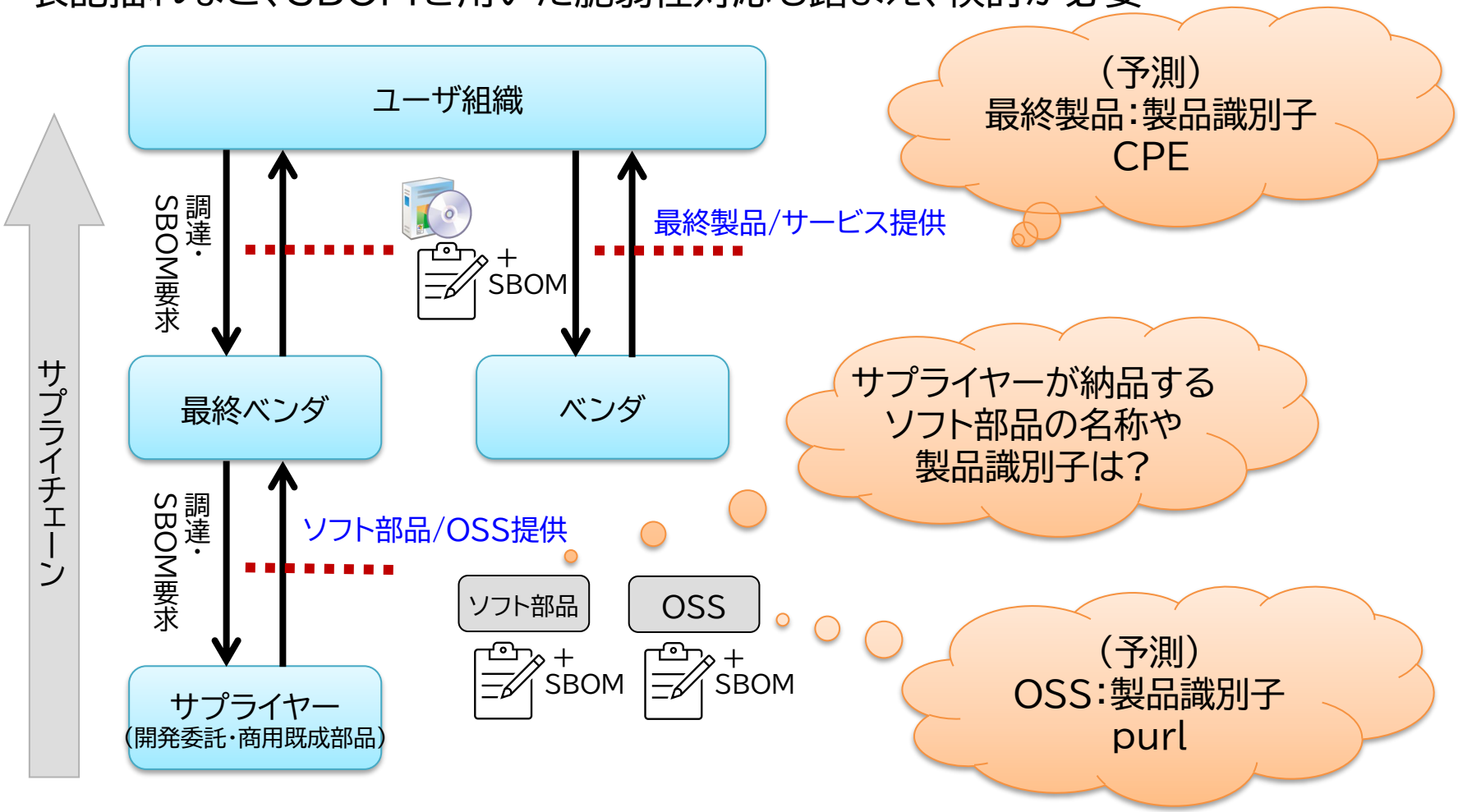
SBOMにおけるソフトウェア名称などの記載に係る課題

- 表記揺れなど、SBOMを用いた脆弱性対応も踏まえ、検討が必要



SBOMにおけるソフトウェア名称などの記載に係る課題

- 表記揺れなど、SBOMを用いた脆弱性対応も踏まえ、検討が必要



なぜならば、、、

管理された製品識別子を提供するために、、、

Secure by Design Pledge（自主宣言）

- 2024年5月のRSAカンファレンスで発表されたCISAの施策
- Secure by Designを宣言させ、7つの具体的な目標に対し、署名から1年以内に取り組みや達成状況を測定可能な形で公表すること

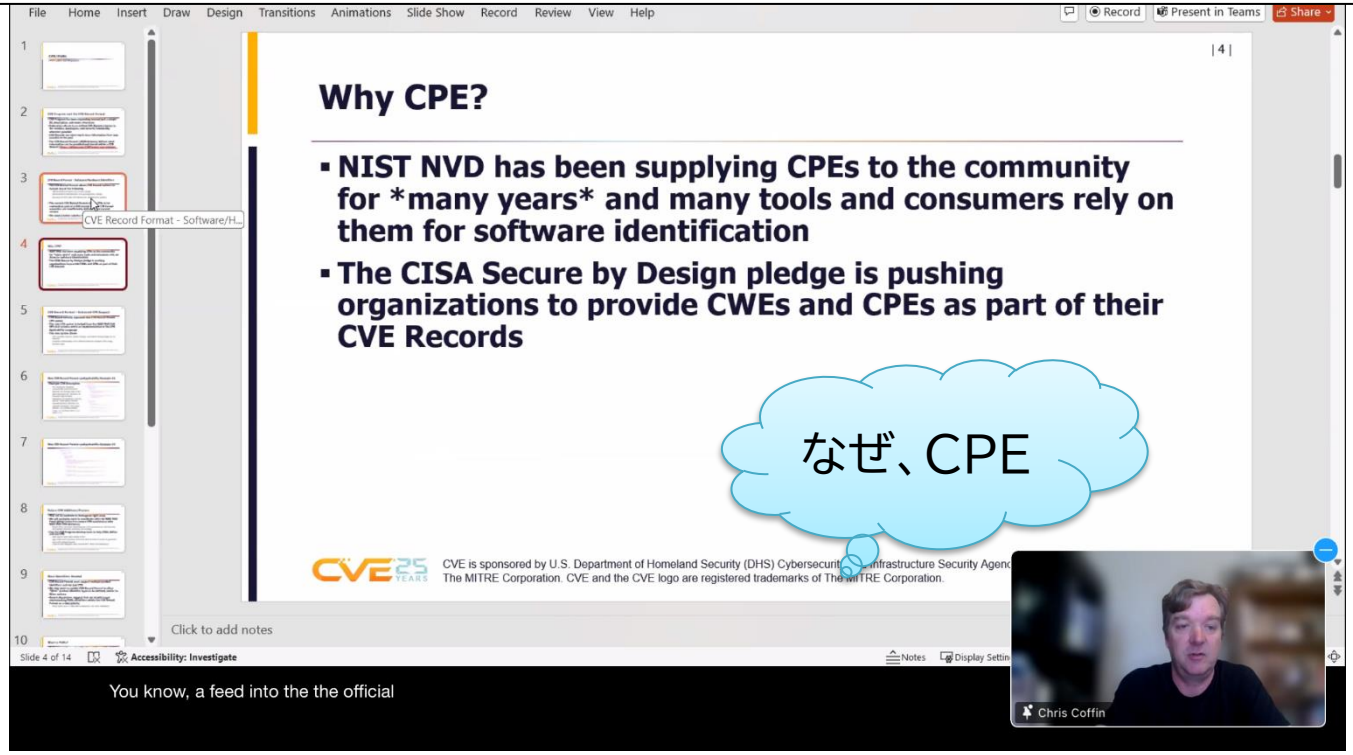
事項	ゴール
多要素認証（MFA）	1年以内に、製品全体で多要素認証の使用を増加させるために講じた措置を示すこと。
デフォルトパスワード	1年以内に、製品全体でデフォルトパスワードを削減に向けた進捗を示すこと。
脆弱性のクラス全体を削減	1年以内に、製品全体で1つ以上の脆弱性クラスを削減させるために講じた措置を示すこと。
セキュリティパッチ	1年以内に、顧客によるセキュリティパッチのインストールを増加させるために講じた措置を示すこと。
脆弱性開示ポリシー（VDP）	1年以内に、脆弱性開示ポリシーを公開すること。
共通脆弱性識別子（CVE）	1年以内に、脆弱性報告の透明性を示すこと。 共通脆弱性タイプ一覧（CWE）、共通プラットフォーム一覧（CPE）など脆弱性対策に必要となる項目を記載した情報を発信すること。
侵入の証拠	1年以内に、製品に影響を及ぼす侵害の証拠を顧客が収集できる状況を向上させること。

管理された製品識別子を提供するために、、、

Secure by Design Pledge（自主宣言）

- CVE Program CNA(CVE Numbering Authority、CVE採番機関) Workshopにおいて、CVEエントリの拡充が話題に

https://youtu.be/7s6zMejFY84?si=CTs9m-FV1N2FZs6-
8:43

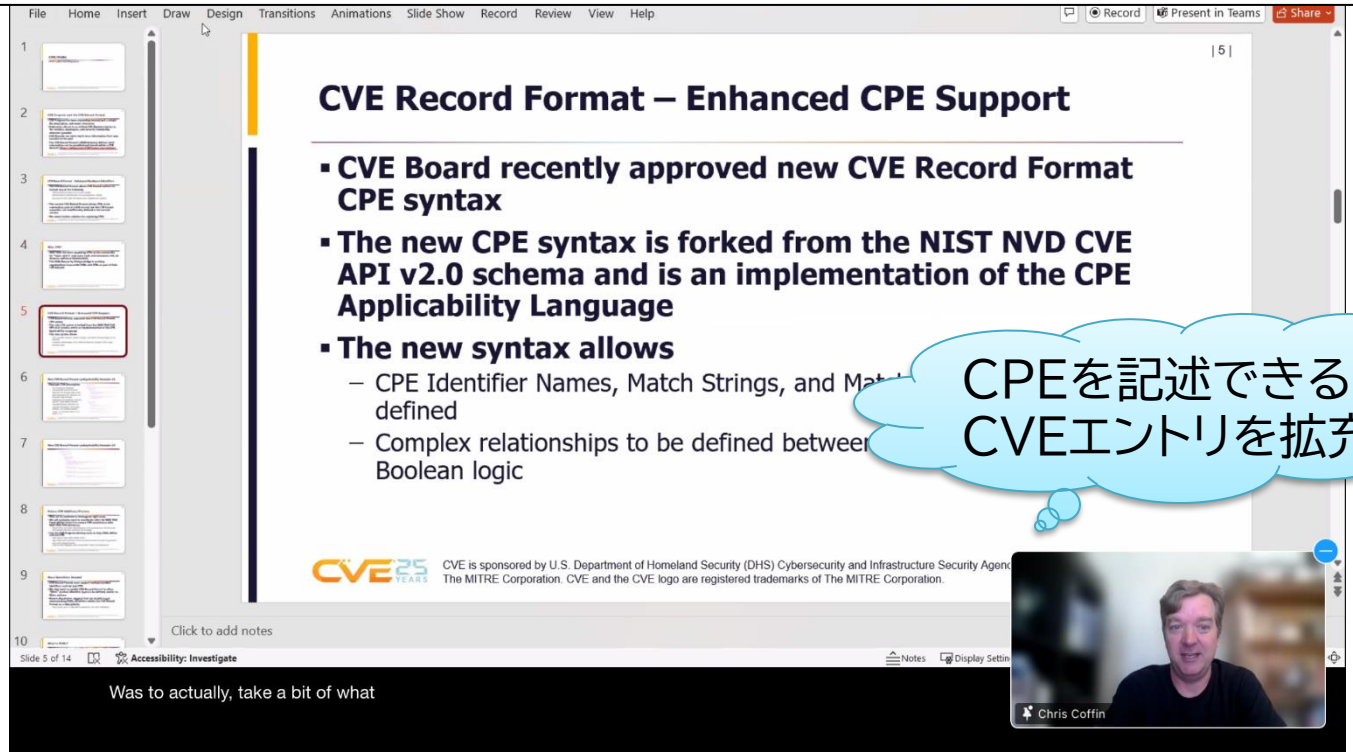


管理された製品識別子を提供するために、、、

Secure by Design Pledge（自主宣言）

- CVE Program CNA(CVE Numbering Authority、CVE採番機関) Workshopにおいて、CVEエントリの拡充が話題に

https://youtu.be/7s6zMejFY84?si=CTs9m-FV1N2FZs6-
11:29



管理された製品識別子を提供するために、、、

Secure by Design Pledge（自主宣言）

- CVE Program CNA(CVE Numbering Authority、CVE採番機関) Workshopにおいて、CVEエントリの拡充が話題に

https://youtu.be/7s6zMejFY84?si=CTs9m-FV1N2FZs6-
14:18

```
{
  "operator": "OR",
  "negate": false,
  "cpeMatch": [
    {
      "vulnerable": true,
      "criteria": "cpe:2.3:a:exmaple_org:examle_prod:*:*:*:*:*:*:*:*",
      "versionStartIncluding": "1.0.0",
      "versionEndExcluding": "1.0.6"
    },
    {
      "vulnerable": true,
      "criteria": "cpe:2.3:a:exmaple_org:examle_prod:*:*:*:*:*:*:*:*",
      "versionStartIncluding": "2.1.5",
      "versionEndExcluding": "2.1.9"
    }
  ]
}
```

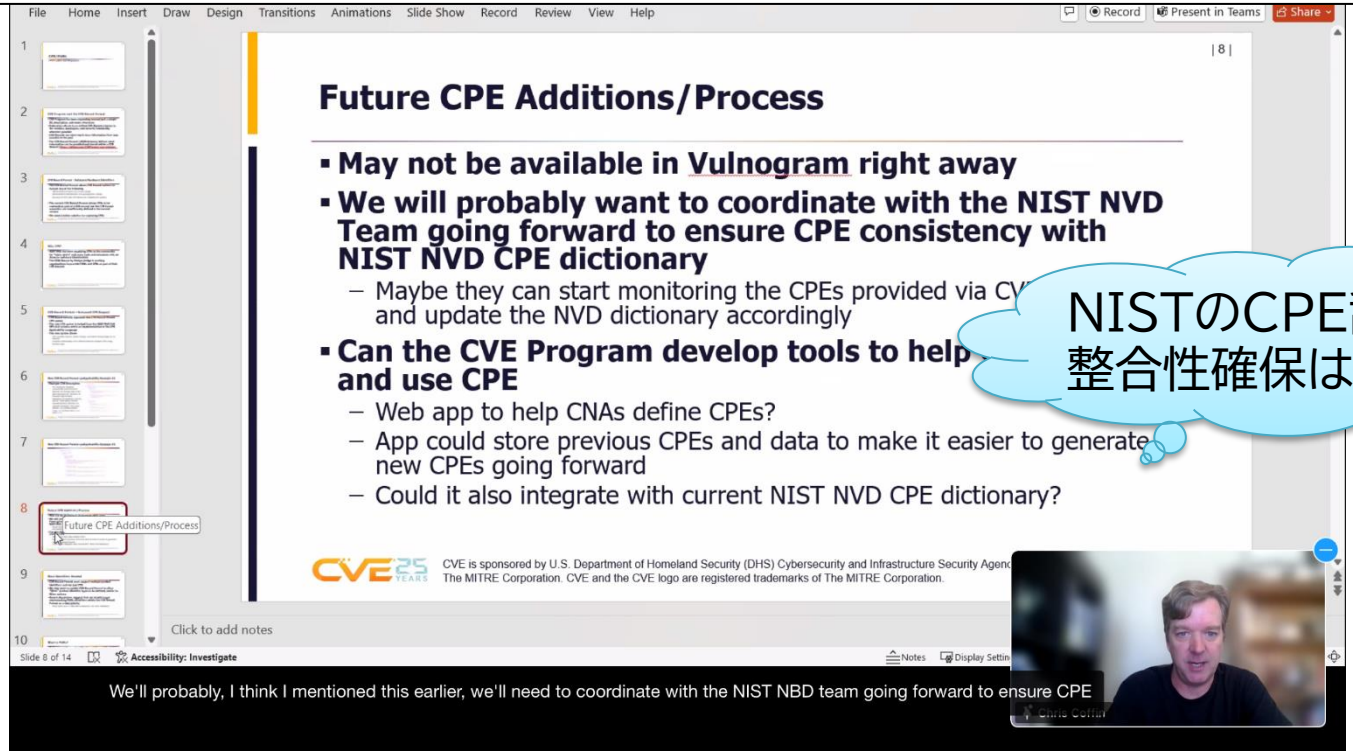
CPEの記述例

管理された製品識別子を提供するために、、、

Secure by Design Pledge（自主宣言）

- CVE Program CNA(CVE Numbering Authority、CVE採番機関) Workshopにおいて、CVEエントリの拡充が話題に

https://youtu.be/7s6zMejFY84?si=CTs9m-FV1N2FZs6-
16:55



それならば、
製品識別の課題を認識してもらうためにも、
このような実証実験をするのも良いのでは？

管理された製品識別子を提供するために、、、

製品識別子(CPE)登録ステップバイステップ

- Secure by Design Pledge(自主宣言)により、CVEエントリの拡充が求められている。

ステップ	内容
ステップ1	
ステップ2	
ステップ3 (予測)	CNA(CVE Numbering Authority)の役割を持つ製品ベンダは自身でCPEを付与管理することができるようになるであろう。

管理された製品識別子を提供するために、、、

製品識別子(CPE)登録ステップバイステップ

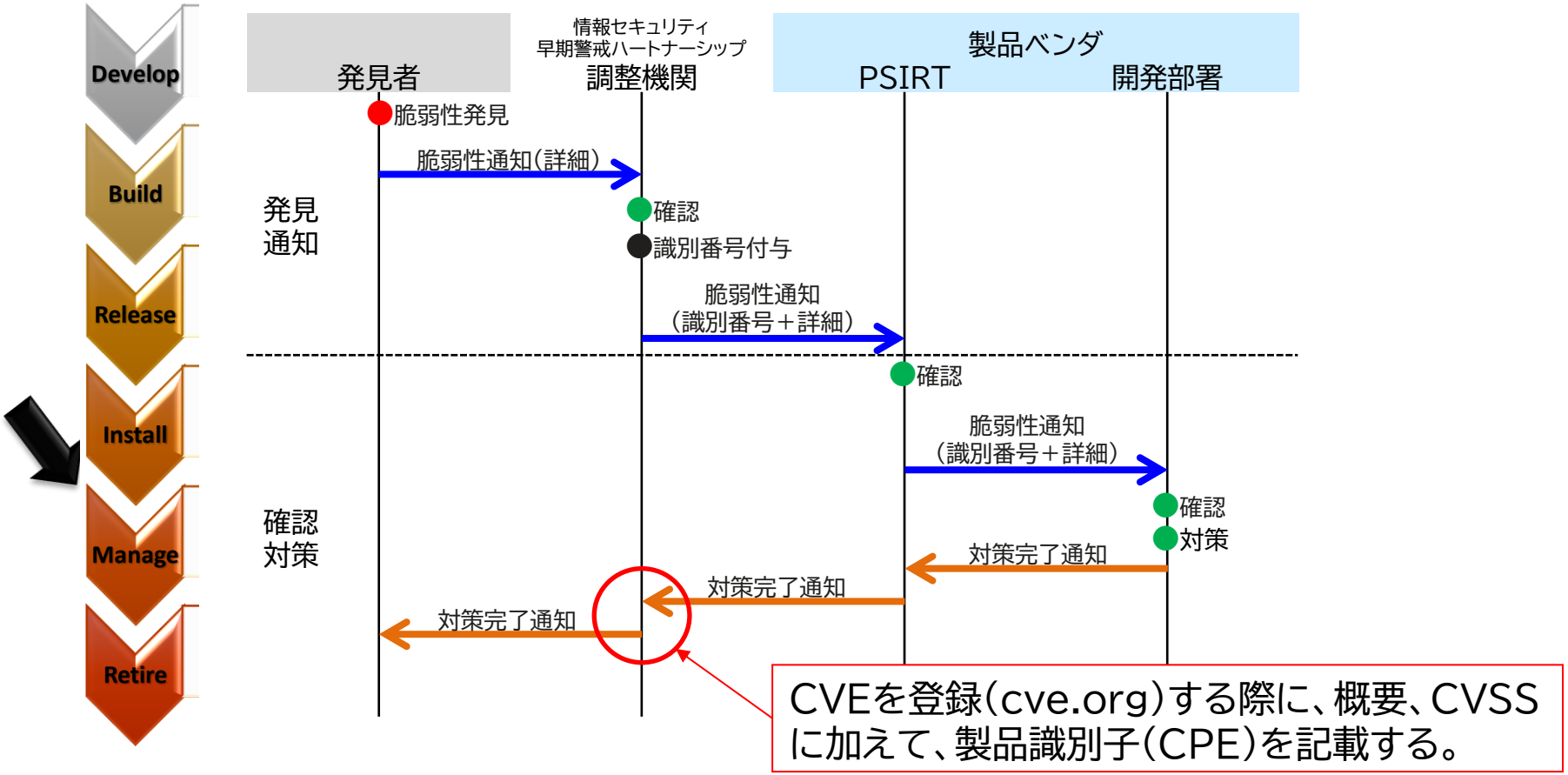
- 表記揺れのない管理された製品識別子を提供するために、、、
  次のような実証実験をしてみるのも良いのではないか？

ステップ	内容
ステップ1	脆弱性が報告されたときに製品識別子(CPE)が登録できる仕組みを検証する。 ● 情報セキュリティ早期警戒パートナーシップにおいて、調整機関と製品ベンダとの協力により、脆弱性が報告されたときに製品識別子(CPE)を登録する際の課題(CPE辞書への反映、言語依存、商標登録など)を明らかにする。
ステップ2	脆弱性が報告される前に製品識別子(CPE)が登録できる仕組みを検証する。 ● 公的DB(JVN/JVN iPedia)と製品ベンダとの協力により、脆弱性が報告される前に製品識別子(CPE)を登録する際の課題(CPE辞書への反映、言語依存、商標登録など)を明らかにする。
ステップ3 (予測)	CNA(CVE Numbering Authority)の役割を持つ製品ベンダは自身でCPEを付与管理することができるようになるであろう。

管理された製品識別子を提供するために、、、

製品識別子(CPE)登録ステップバイステップ［ステップ1］

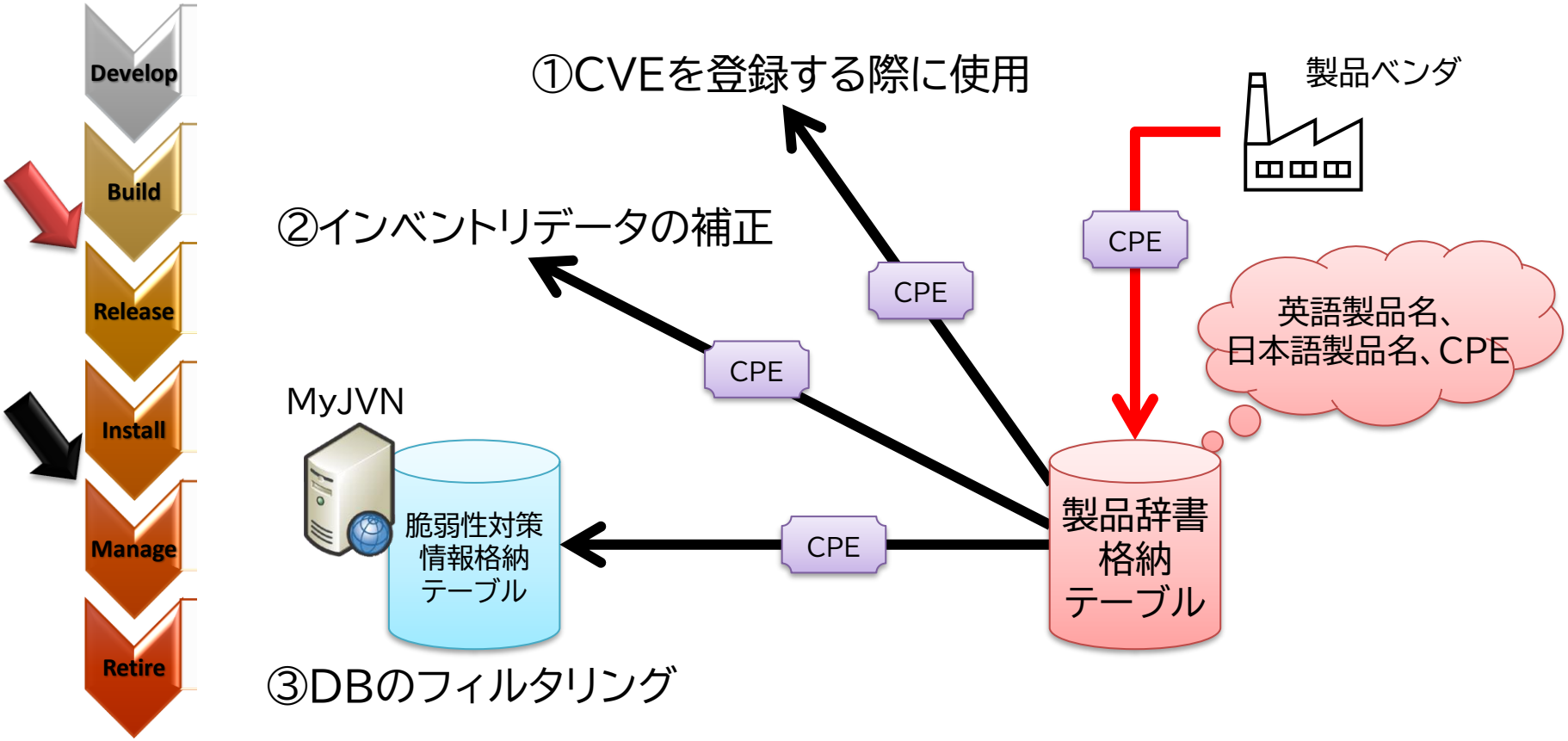
- 脆弱性が報告されたときに製品識別子(CPE)が登録できる仕組みを検証する。



管理された製品識別子を提供するために、、、

製品識別子(CPE)登録ステップバイステップ［ステップ2］

- 脆弱性が報告される前に製品識別子(CPE)が登録できる仕組みを検証する。



おわりに

2023年末頃から各国が署名するガイドが発行

年月日	イベント
2024年10月29日	2025-2026年度 CISA 国際戦略計画
2024年4月30日	バイデン・ハリス政権が重要インフラに関する新たな国家安全保障覚書を発表
	NSM-22 重要インフラのセキュリティとレジリエンスに関する国家安全保障覚書
2023年10月16日	セキュア・バイ・デザインの原則とアプローチに関するガイダンス
2023年10月	2023-2025年度 CISA ステークホルダー関与戦略計画
2023年8月4日	2024-2026年度 CISA サイバーセキュリティ戦略計画
2022年9月12日	CISA 戦略計画 2023-2025
2021年11月12日	21会計年度国防権限法(NDAA)
2013年2月12日	EO13636 重要インフラのサイバーセキュリティの向上
	PPD21 重要インフラのセキュリティとレジリエンス

おわりに

2025～2026年度 CISA 国際戦略計画

目標1	米国が依存する外国インフラのレジリエンスの強化
目標2	統合サイバー防御の強化 ① パートナーと協力してサイバー防御を実現し、集団的リスクを軽減する。 ② サイバーの安全性を高めるために、標準とセキュリティを大規模に推進する。 ③ 主要パートナーのサイバーおよび物理的なレジリエンス能力を強化する。
目標3	国際活動の機関間調整の統一

ポイント

- CISAの国際的な焦点は、ソフトウェア部品表、安全なAIシステム、オープンソースセキュリティ、脆弱性の協調的な開示などのSecure by Designプラクティスの広範な採用を促進することである。

おわりに

SBOMが果たす役割

- Q:日本で実装されるのか その意義は？
- A:次の課題を解決するための「きっかけ」になることを期待したい。

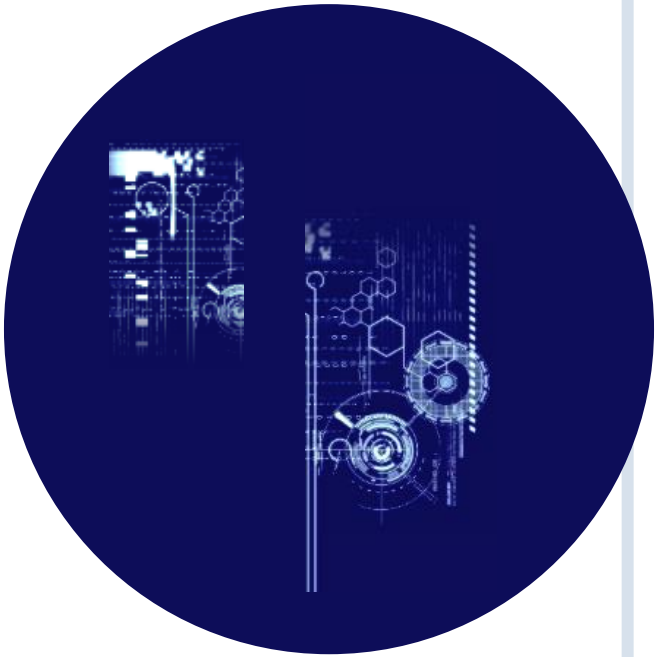
【 課題 】

国内で利用されているソフトウェアや装置を対象とした脆弱性情報ならびに対策情報は流通しているのか？	セキュリティ設定に関する作業を手作業で行なうと、設定ミスや設定者のセキュリティ知識の程度や判断の相違などによりセキュリティ要件を損なう可能性大

【 解決策 】

製品識別による脆弱性への対処	作業の機械化（自動化）による対処
脆弱性対策情報DBへの集約	共通技術仕様による（自動化）の普及

TDU
TOKYO DENKI UNIVERSITY



Collaborate
together
to make our
Internet
secure.