Vol.2016-DPS-166 No.3
Vol.2016-CSEC-72 No.3
2016/3/3

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

寺田真敏† 斉藤良彰† 亀山友彦† 金野千里†

概要：JVN で整備を進めている脆弱性対策に関わる処理の機械化を目指すフレームワーク MyJVN(JVN 脆弱性対策機械処理基盤)では，脆弱性対策の処理の機械化や自動化を考慮した流通基盤の整備を進めている．本稿では，脆弱性対策に関わる処理の機械化の推進と共に，脆弱性対策の裾野を広げるために，CPE や SWID などの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ついて検討した結果を報告する．

キーワード：製品識別子，脆弱性対策情報データベース，資産管理

Feasibility study of the collaboration possibility between vulnerability database and asset management by Product Identifier

MASATO TERADA† YOSHIAKI SAITO†
TOMOHIKO KAMEYAMA† CHISATO KONNO†

Abstract: MyJVN (JVN Security Automation Framework) promotes the development of the machine-readable vulnerability handling mechanism and platform in Japan. In this paper, we report the feasibility study of the collaboration possibility between Vulnerability Database and Asset Management by Product Identifier such as CPE and SWID to improve the machine-readable vulnerability handling platform against cyber attacks.

Keywords: Product Identifier, Vulnerability Database, Asset Management

1. はじめに

国内においても，JVN や JVN iPedia など脆弱性対策情報の提供環境は充実してきており，脆弱性の傾向などを把握しやすくなってきている．しかし，対策情報の多くは主に文書として構成されており，脆弱性の有無をチェックして対策を促すなど脆弱性対策に関わる処理の機械化については未だ発展途上にある．2009 年に流布した Web 誘導型マルウェアである Gumblar 以降，クライアントアプリケーションの脆弱性を悪用したマルウェア感染への対策は急務であり，クライアントアプリケーションを常に最新バージョンに維持することを促した．また，2014 年に報告された Apache Struts，OpenSSL(Heartbleed)，GNU Bash(Shellshock)などのオープンソースの脆弱性は，サーバにおいて影響を受けるコンポーネントを使用しているかどうか判断しにくいという新たな課題を投げかけた．

JVN で整備を進めている脆弱性対策に関わる処理の機械化を目指すフレームワーク MyJVN(JVN 脆弱性対策機械処理基盤)では，このような問題解決に向けて，脆弱性対策の処理の機械化や自動化を考慮した流通基盤の整備を進めてきている．本稿では，脆弱性対策に関わる処理の機械化の推進と共に，脆弱性対策の裾野を広げるために，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ついて検討した結果を報告する．

2. 関連研究

本章では，脆弱性対策に関連する製品識別子，資産管理の状況について述べる．

2.1.1 製品識別子

(1) 共通プラットフォーム一覧(CPE)

共通プラットフォーム一覧(CPE: Common Platform Enumuration)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ための仕様である．米 MITRE によって開発され，2007 年に v1.0，2011 年に v2.3 が公開された．また，2012 年には，v2.3 が ITU-T : X.1528[1]として標準化されている．CPE v2.3 の記述形式は，次の通りである．

cpe:2.3:{種別}:{製品ベンダ名}:{製品名}:{バージョン}:{アップデート}:{エディション}:{言語}:…

種別は，h=ハードウェア，o=OS，a=アプリケーションとなっている．

(2) ソフトウェア識別(SWID)タグ

ソフトウェア識別(SWID: Software Identification)タグは，導入されたソフトウェアを識別するための国際標準規格 ISO/IEC 19770-2 である．2009 年に初版が公開された．改訂版では，パッチ対応属性やハッシュ値属性のサポートなど脆弱性対策に関連する項目が盛り込まれ，2015 年 10 月に国際標準として発行された[2]．記述形式は，ソフトウェ

† (独)情報処理推進機構
Information-technology Promotion Agency, Japan.

ア名，バージョン，製品ベンダ名などを XML フォーマットに表記する(図 1)．しかし，該当製品の脆弱性に関わる情報との連携については充分とはなっておらず，利用面で環境整備が必要な状況にある．

(3) **CPE と SWID タグとの関連付け**

CPE と SWID タグとの関連付けについては，相互運用可能な SWID タグの導入と作成のガイドラインである NIST IR 8060[3]において検討されている．この Draft3 のガイドラインでは，関連付けの手法として，SWID タグから CPE v2.3 を生成する手順について示している．

```
<?xml version="1.0" encoding="UTF-8"?>
<swid:SoftwareIdentity
  xmlns:swid="http://standards.iso.org/iso/19770/-2/2014-DIS/schema.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:MD5="http://www.w3.org/2001/04/xmldsig-more#md5"
  xmlns:SHA1="http://www.w3.org/2000/09/xmldsig#sha1"
  xmlns:SHA256="http://www.w3.org/2001/04/xmlenc#sha256"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xsi:schemaLocation="http://standards.iso.org/iso/19770/-2/2015/schema.xsd"
  name="MyJVN API"
  version="1.0.0"
  tagId="ipa.go.jp+myjvn_api+1.0.0"
  versionScheme="multipartnumeric"
  >
  <swid:Entity
    name="独立行政法人 情報処理推進機構"
    regid="ipa.go.jp"
    role="softwareCreator"
    xml:lang="ja"/>
  <swid:Payload>
    <swid:File name="myjvn_api.jar"
      MD5:hash="583DE6974FCC12CCF6C7F189A8A51808"
      SHA1:hash="D1FA84BE3561E46A5C787DAD5FD1556F240B4863"
      SHA256:hash="3A91AC16AC90F354989314B65F2C73E0365D800
                   87760B7D833D8F6BD7813F01F"/>
  </swid:Payload>
</swid:SoftwareIdentity>
```

図 1：SWID タグの例

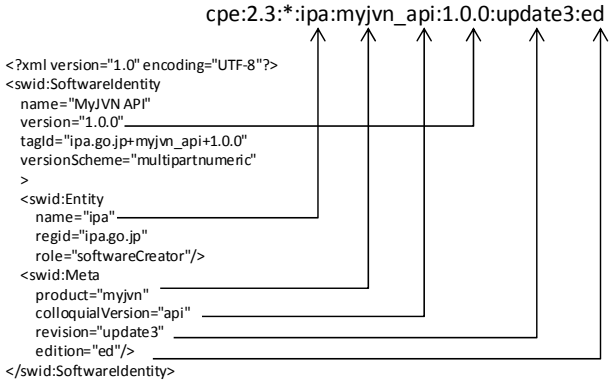


図 2：SWID タグから CPE v2.3 の生成手順

2.1.2 資産管理

(1) **ISO/IEC 19770**

ISO/IEC 19770 は，ソフトウェア資産管理(SAM: Software Asset Management)に関する国際標準規格であり，表 1 に示すパートでの規格化が進められている．

(2) **ソフトウェア辞書**

一般社団法人ソフトウェア資産管理評価認定協会(SAMAC: association of SAM Assessment & Certification)では，実際に利活用されているソフトウェアの識別情報をソフトウェア辞書として蓄積している．このソフトウェア辞書には，インベントリー収集ツールで収集可能な[プログラムの追加と削除]に表示されるインストール名称，メーカー名，ソフトウェア名，エディション，バージョン，ソフトウェア種別などの情報が格納されている[4].

表 1：ISO/IEC 19770

パート	概要
ISO/IEC 19770-1:2006 (JIS X 0164-1：ソフトウェア資産管理-第 1 部：プロセス)	IT サービスマネジメント全体の有効な支援となるのに充分な規格を基準にソフトウェア資産管理(SAM)を実行していることを証明できるようにするものとして開発された仕様
ISO/IEC 19770-1:2012 Processes and tiered assessment of conformance 情報技術―ソフトウェア資産管理 第 1 部：プロセス及び段階的適合性評価	ソフトウェア資産管理(SAM)のための統合されたプロセス群のベースラインを定める国際規格．2012 年の改訂で，付加的な導入，アセスメント及び認識を可能にする「段階」という考え方が取り入れられている．また，ISO/IEC 20000 と緊密な整合がとられており，ISO/IEC 20000 の IT サービスマネジメントを支援することを意図している．(2006 年 5 月に発行され，2012 年 6 月に改訂版が発行された).
ISO/IEC 19770-2:2009 Software identification tag	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
ISO/IEC 19770-2:2015	● ISO/IEC 19770-2:2009 のタグ必須の多くが，任意となった． ● SoftwareIdentity タグでパッチ対応属性(delta)サポート ● File タグでハッシュ値属性(sha1,sha256 など)サポート
ISO/IEC 19770-3 Software entitlement tag	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タグの標準化
ISO/IEC 19770-5:2015 Overview and vocabulary	ISO/IEC 19770 で用いられる用語の定義など

3. JVN 脆弱性対策機械処理基盤整備の課題

本章では，JVN 脆弱性対策機械処理基盤整備にあたり検討を進めてきた 2 つの課題について述べる．

- **課題 1：脆弱性対策をより一層推進するためには，どうしたら良いのか？**

  脆弱性対策の裾野を広げるためのアプローチについての課題である．本課題については，資産管理とのデータ連携などによる多様な脆弱性対策の推進が解決策のひとつになると考えている．そこで，本研究では，製品識別子を用いて資産管理と脆弱性対策との連携の可能性を探ることとした．

- **課題 2：国内において，脆弱性対策の対象となりえる製品の件数規模は，どのくらいあるのだろうか？**

  資産管理と脆弱性対策との連携で使用する製品識別子の件数規模の見積りについての課題である．一般的に，JVN iPedia，NVD などの脆弱性対策情報データベースに登録されている製品は，脆弱性が報告されて初めて該当製品としてデータベースに登録される．一方，資産管理では，脆弱性が報告されていない製品も

管理対象となっている．このため，双方に登録されている製品と，どちらか一方にしか登録されていない製品が出てくることになる．そこで，本研究では，連携のために片方にしか登録されていない製品情報の規模を探ることとした．

以降の各章では，課題解決のために検討した結果について報告する．

4. ソフトウェア辞書とのデータ連携

本章では，SAMAC ソフトウェア辞書とのデータ連携について述べる．

4.1 目的

2014 年に報告された Apache Struts, OpenSSL(Heartbleed), GNU Bash(Shellshock)などへの対処を通して，脆弱性対策には，資産管理との連携が必要であることが再認識された．しかし，多くの場合，ソフトウェアのインストール状況と脆弱性との紐付けは人手で行われている(図 3)．すなわち，新たに報告された脆弱性の影響を受けるソフトウェアの有無，影響を受けるシステムの特定，影響を受けるホストの特定のために，資産管理と脆弱性管理が連携するに至っていない．

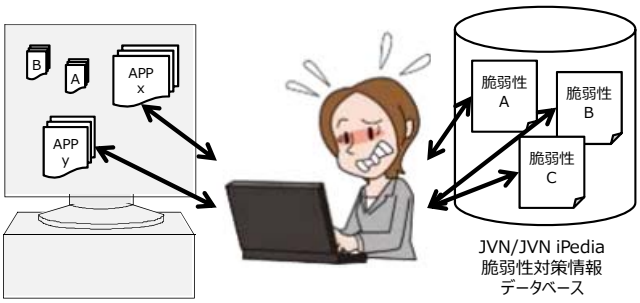


図 3：人手による紐付け

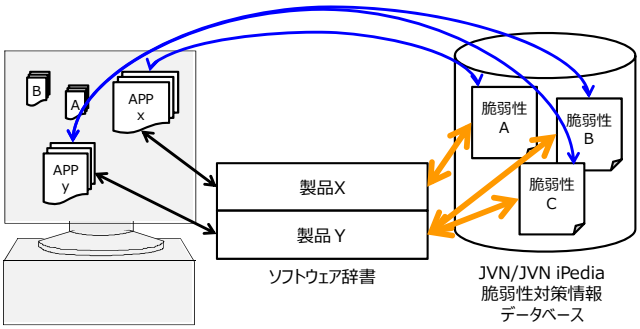


図 4：データ連携による紐付け

SAMAC ソフトウェア辞書のようなソフトウェアの識別情報と JVN/JVN iPedia 脆弱性対策情報データベースとの紐付けができると(図 4 の橙線)，資産管理を利用してインストール状況と脆弱性との紐付けが可能となる(図 4 の青線)．すなわち，インストール状況に合わせた脆弱性対策の推進につながるだけではなく，資産管理ツールから脆弱性管理への歩み寄りが容易になる．

そこで，SAMAC ソフトウェア辞書と JVN 製品データベースのデータ調査を通して，次の課題について検討した．

- データ連携の実現方法と，連携のための実現上の課題を明らかにすること
- 今後，JVN 製品データベースに登録される可能性のある製品件数を把握すること
- JVN 脆弱性対策機械処理基盤の製品データベースの改善事項を明らかにすること

4.2 調査結果

本節では，2014 年 11 月 10 日時点での調査結果について述べる．

(1) SAMAC ソフトウェア辞書

SAMAC ソフトウェア辞書に登録されている辞書エントリ数は 86,103 件で，製品ベンダ数は表 2 の通りであった．

(2) JVN 製品データベース

JVN 脆弱性対策機械処理基盤では，脆弱性対策情報に関連する製品情報を JVN 製品データベースに格納している．この製品データベースに登録されている製品数は 22,027 件，製品ベンダ数 10,014 件であった．

(3) ソフトウェア辞書と JVN 製品データベースの紐付け

製品の紐付けについては，SAMAC ソフトウェア辞書の各エントリに，JVN 製品データベースで使用している CPE v2.2 を追記する形式で手作業での紐付けを実施した(図 5)．なお，JVN 製品データベースに登録されていない製品ベンダ，製品については，手作業で一意の製品ベンダ，製品を識別すると共に，仮の CPE を付与して試算した．

表 2 辞書に登録されている製品ベンダ数

項目	件数
製品ベンダ不明の辞書エントリ数	11,440 件
製品ベンダ記載の辞書エントリ数	74,663 件
製品ベンダ数	一意の製品ベンダ名を機械的に識別すると，製品ベンダ数は 7,688 件

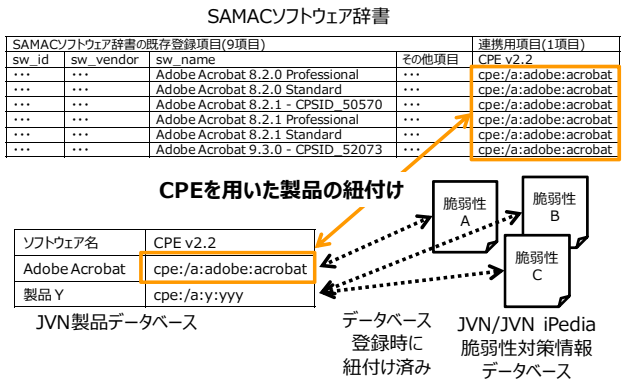


図 5：CPE を用いた製品の紐付け

辞書エントリ約 86,000 件を対象に製品ベンダの紐付けをした結果，約半分の辞書エントリ(43,348 件)に対して紐付けが実施でき，104 件の登録済み製品ベンダ，49 件の未登録の製品ベンダとなった(表 3). また，製品については，サンプリングした辞書エントリ 6,753 件を対象に製品の紐付けをした結果，約半分の辞書エントリ(3,243 件)に対して紐付けが実施でき，66 件の登録済み製品，112 件の未登録の製品となった(表 4).

表 3：製品ベンダの紐付け(辞書エントリ約 86,000 件)

項目		件数
製品ベンダの紐付け不可であった辞書エントリ数		42,755 件
	製品ベンダ不明の製品数(辞書エントリ数)	11,440 件
	上記以外(辞書エントリ数)	31,315 件
製品ベンダの紐付け可であった辞書エントリ数		43,348 件
	うち，JVN 製品データベースに登録済みの製品ベンダ数	104 件
	うち，JVN 製品データベースに未登録の製品ベンダ数	49 件

表 4：製品の紐付け(辞書エントリ 6,753 件)

項目		辞書エントリ数	CPE 付与数
紐付け不可であった辞書エントリ数		3,510 件	－
紐付け可であった辞書エントリ数		3,243 件	－
	うち，JVN 製品データベースに登録済みの辞書エントリ数	2,070 件	66 製品
	うち，JVN 製品データベースに未登録の辞書エントリ数	1,173 件	112 製品

(4) 考察

- 紐付け不可であった製品ベンダ/製品について

  紐付け不可であった製品ベンダのうち，もともと製品ベンダ情報が格納されていない辞書エントリが 26%(11,440 件)ある．残り 74%(31,315 件)は，本調査期間中に，SAMAC ソフトウェア辞書の登録項目だけでは特定できないと判断したものである．今後，データ連携を具体化していく中で，紐付け不可の比率は多少なりとも下がられると考えている．

- 紐付け可であった製品ベンダ/製品のうち，JVN 製品データベースに登録済みの製品ベンダ/製品について

  本調査では，調査対象範囲を SAMAC ソフトウェア辞書に登録されている辞書エントリ 6,753 件に絞ってはいるが，辞書エントリの 2,070 件，製品数にすると 66 製品に対して，JVN 製品データベースで使用している CPE を追記することができた．すなわち，この 66 製品については，SAMAC ソフトウェア辞書を介して資産管理ツールから MyJVN API 経由で脆弱性対策情報の検索が可能となる．なお，脆弱性対策情報検索までの流れは，(i)資産管理ツールでインストール情報を収集し，(ii)ソフトウェア名をキーとして SAMAC ソフトウェア辞書から CPE を取得した後，(iii)CPE をキーとして MyJVN API でアクセスするというものである(図 6).

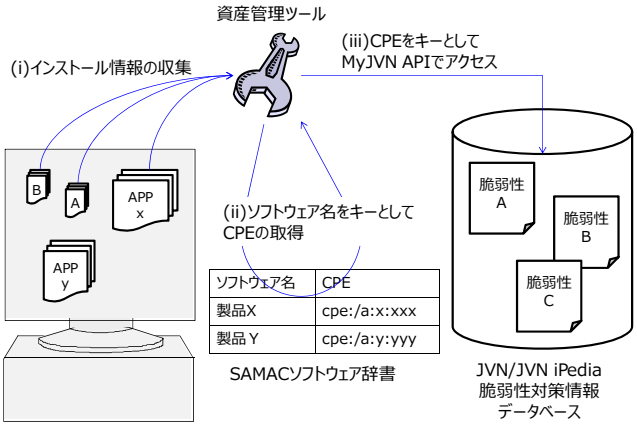


図 6：連携による脆弱性対策情報の参照

- 紐付け可であった製品ベンダ/製品のうち，JVN 製品データベースに未登録の製品ベンダ/製品について

  本調査の結果，辞書エントリの 1,173 件，製品数にすると 112 製品は，JVN 製品データベースに登録されていない製品である．すなわち，この 112 製品については，今後，JVN 製品データベースに登録される可能性のある製品となる．また，JVN 製品データベースで使用している CPE がないため，SAMAC ソフトウェア辞書と JVN/JVN iPedia 脆弱性対策情報データベース連携の対象外になってしまっていることになる．これら未登録製品に対しては，表 5 に示す CPE 連携方式，SWID タグ連携方式で解決できると考えている．ただし，いずれの方式でも，国内での SWID タグの管理，グローバルでの SWID タグの連携について検討していく必要がある．

表 5：未登録の製品ベンダ/製品への対応

連携方式	概要
CPE 連携方式	脆弱性が報告されていない製品に対しても，事前作成した CPE(CPE v2.2/v2.3)を JVN 製品データベースに登録する．なお，CPE 製品名の事前作成については，NVD などの脆弱性対策情報データベースとのグローバルな運用連携を踏まえ，SWID タグから CPE を生成する手法を採用する必要がある．
SWID タグ連携方式	SAMAC ソフトウェア辞書ならびに，JVN 製品データベースに SWID タグを取り込む．JVN 製品データベースで SWID タグを取り込む際には，脆弱性が報告されている製品，脆弱性が報告されていない製品の双方を登録対象とする．MyJVN API で SWID タグに格納されている項目，例えば，tagId をキーとして脆弱性対策情報を取得できるよう拡張する．

5. JVN 脆弱性対策機械処理基盤での SWID タグ付与

5.1 目的

JVN/JVN iPedia 脆弱性対策情報データベースと資産管理とのデータ連携を進めるには，紐付けのための製品識別子が必要となる．特に，JVN 製品データベースに登録されていない製品の対応については，SWID タグから CPE を生成

やSWIDタグに格納されている項目を利用したMyJVN API拡張を検討する必要がある．また，NIST IR 8060 Draft3では，相互運用可能なSWIDタグの導入と作成について言及しているものの，英語圏以外での利用，すなわち，英語表記以外の製品ベンダ名，製品名への対応や既存 CPE(CPEに記載されている製品ベンダ名や製品名)との整合性確保など，運用上の課題が残っている．

そこで，JVN脆弱性対策機械処理基盤のツールを対象にSWID タグ付与の試行を通して，次の課題について検討した．

- 既存 CPE との整合性を踏まえた日本語表記の製品ベンダ名，製品名への対応方法を明らかにすること
- SWIDタグを展開する上での課題を明らかにすること

5.2 SWID タグ付与の試行

SWIDタグ付与の試行は，3つの異なる環境で動作する4つのツールを対象に実施した(表 6)．本節では，この中から，多言語版 CVSS v3 計算機(以降，ScoreCalc3)について述べる．

表 6：SWID タグ付与の試行

環境	SWIDタグを付与したツール
Adobe AIR	● MyJVN 脆弱性対策情報フィルタリング収集ツール(略称：mjcheck3)
Adobe Flash Player	● MyJVN 脆弱性対策情報収集ツール(略称：mjcheck) ● 多言語版 CVSS v3 計算機(略称：ScoreCalc3)
.NET Framework	● MyJVN バージョンチェッカ for .NET

(1) 多言語版 CVSS v3 計算機(以降，ScoreCalc3)

ScoreCalc3は，Adobe Flash Player版として提供している多言語版 CVSS v3 計算機である．作成したSWIDタグファイルを付図1に，特徴を表 7に示す．

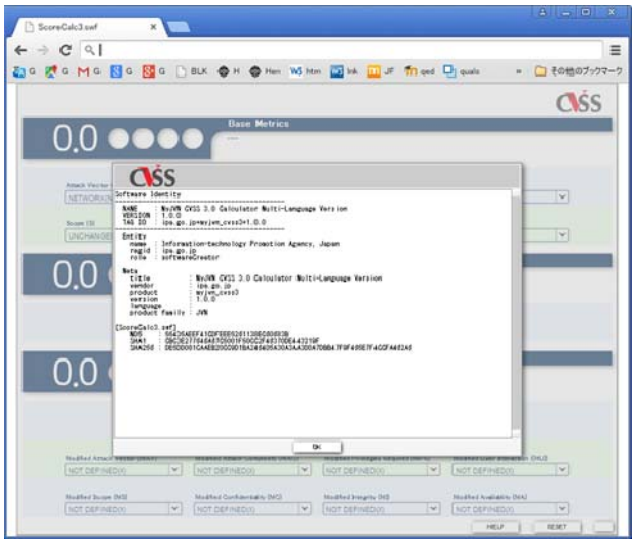


図 7：ipa.go.jp+myjvn_cvss3.swidtag ファイルを用いたバージョン情報表示

表 7：ipa.go.jp+myjvn_cvss3.swidtag ファイルの特徴

項目	特徴
SoftwareIdentity タグ	tagIdは，16バイトのGUIDが推奨されている．今回，tagIdは，GUIDの代替案であるregid＋製品＋バージョン情報を連結したテキスト形式＝"ipa.go.jp+myjvn_cvss3+1.0.0"で構成した．
Meta タグ	既存 CPE との整合性を確保しつつ，SWIDタグから CPE を生成するため，Meta タグを CPE v2.3 生成用に拡張した(図 8)．なお，SWIDタグから CPE v2.3 を生成する手順は，NIST IR 8060 Draft3 で示されている手順とは異なる．
Payload タグ	ScoreCalc3.swf のハッシュ値として MD5，SHA1，SHA256 を記述した．
Link タグ	ScoreCalc3.swf と ipa.go.jp+myjvn_cvss3.swidtag ファイルの URL を記述した．
その他	Adobe Flash Player 版は，Web サイトからオンラインでダウンロードされ，ブラウザ上で実行されるため，ScoreCalc3.swf 自身は，クライアント側にインストールフォルダを持たない．このため，Web サイト上の ScoreCalc3.swf と同一フォルダに，ipa.go.jp+myjvn_cvss3.swidtag ファイルを配置し，http://jvndb.jvn.jp/cvss/ipa.go.jp+myjvn_cvss3.swidtag でアクセス可能とした．また，ScoreCalc3.swf の CVSS ボタンをクリックした際には，この URL の ipa.go.jp+myjvn_cvss3.swidtag ファイルをダイアログ表示することとした(図 7)．

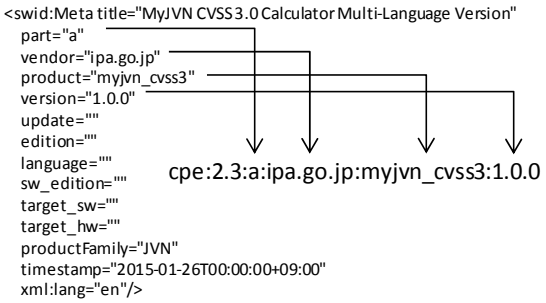


図 8　CPE v2.3 記述のための Meta タグ拡張

(2) 考察

SWID タグ付与の試行を通した課題解決のため，SWIDタグ付与の試行と 2015 年 8 月にリリースされた NIST IR 8060 Draft3 を踏まえて，下記3つの要件を満たす日本語表記の SWID タグの作成について検討した．

(a) 日本語と英語の併記への対応
(b) NIST IR 8060 Draft3 との整合性の確保
(c) 既存 CPE(CPE に記載されている製品ベンダ名や製品名)との整合性の確保

検討の結果，図 9に示す通り，SWIDから CPE を生成するための Entity/Meta タグ，日本語表記用の Entity/Meta タグ，英語表記用の Entity/Meta タグを用意することで，3つの要件を満たす日本語表記の SWID タグを実現できる可能性がある．ただし，NIST IR 8060 Draft3 では，Entity タグ，Meta タグが複数記載されていることを想定していないことから，SWIDタグから CPE v2.3 生成を実現するためには，どの Entity タグ，Meta タグを使用するのかを明確にしなければならないという課題もあることがわかった．

```
<?xml version="1.0" encoding="UTF-8"?>
<swid:SoftwareIdentity
  xmlns:swid="http://jvndb.jvn.jp/schema/swid.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:MD5="http://www.w3.org/2001/04/xmldsig-more#md5"
  xmlns:SHA1="http://www.w3.org/2000/09/xmldsig#sha1"
  xmlns:SHA256="http://www.w3.org/2001/04/xmlenc#sha256"
  xsi:schemaLocation="http://jvndb.jvn.jp/schema/swid.xsd
            http://jvndb.jvn.jp/schema/swid.xsd"
  name="MyJVN Filtered Vulnerability Countermeasure Information Tool 3"
  version="3.0.0"
  tagId="559c90d5-68ba-4d6e-80ab-fba3f59bbcef"
  versionScheme="multipartnumeric"
  >
  <swid:Entity                                   (b)NIST IR 8060 Draft3との整合性の確保
    name="ipa"                                   (c)既存CPEとの整合性の確保
    regid="ipa.go.jp"
    role="softwareCreator"/>
  <swid:Entity
    name="Information-technology Promotion Agency, Japan"
    regid="ipa.go.jp"
    role="softwareCreator"
    xml:lang="en"/>                              (a)日本語と英語の
  <swid:Entity                                     併記への対応
    name="独立行政法人 情報処理推進機構"
    regid="ipa.go.jp"
    role="softwareCreator"
    xml:lang="ja"/>
  <swid:Meta                                     (b)NIST IR 8060 Draft3との整合性の確保
    product="myjvn"                              (c)既存CPEとの整合性の確保
    colloquialVersion="api"
    revision="update3"
    edition="ed"/>
  <swid:Meta
    product="MyJVN Filtered Vulnerability Countermeasure Information Tool 3"
    colloquialVersion="Application Program Interface"
    revision="Uppdate Version 3"
    edition="Edition"
    xml:lang="en"/>                              (a)日本語と英語の
  <swid:Meta                                       併記への対応
    product="MyJVN 脆弱性対策情報フィルタリング収集ツール"
    colloquialVersion="アプリケーションプログラムインタフェース"
    revision="アップデートバージョン3"
    edition="エディション"
    xml:lang="ja"/>
  <swid:Payload>
    :
  </swid:Payload>
  <swid:Link href="cpe:/a:openssl:openssl" rel="related"/>
    :
</swid:SoftwareIdentity>
```

図 9：3 つの要件を満たす日本語表記の SWID タグ

6. 資産管理と脆弱性管理の連携

本章では, SWID タグファイルの活用範囲の拡大として, SWID タグファイルを用いたユーザアプリケーション関連の資産管理と脆弱性管理の連携について述べる.

6.1 目的

ユーザアプリケーション開発では，Apache Struts のようなフレームワークや OpenSSL などのライブラリが利用されることも多い．その一方で，納品物であるユーザアプリケーションに，どのような外部コンポーネントが組み込まれているか，前提プログラムが何であるのかを管理する仕組みが整備されているわけではない．このため，2014 年に報告された Apache Struts，OpenSSL などの脆弱性が報告された場合，影響を受けるコンポーネントを使用しているかどうかは，開発業者に問い合わせない限り判断できないという状況につながっている.

ユーザアプリケーションの納品において，ユーザアプリケーションだけではなく，使用している外部コンポーネントや前提プログラムを把握できるユーザアプリケーション用 SWID タグファイルを納品してもらうことができれば，ユーザアプリケーション自身についても資産管理だけではなく，脆弱性管理も可能となる.

そこで，ユーザアプリケーション用 SWID タグの試作を通して，ユーザアプリケーション開発における次の課題について検討した.

- ユーザアプリケーション用 SWID タグに必要となる項目を明らかにすること
- ユーザアプリケーションを対象とする資産管理と脆弱性管理の連携の実現方法を明らかにすること

6.2 ユーザアプリケーション用 SWID タグの試作

(1) ユーザアプリケーション用 SWID タグ

ユーザアプリケーションとして開発している MyJVN を対象に作成したユーザアプリケーション用 SWID タグファイルを付図 2 に，特徴を表 8 に示す.

表 8：ipa.go.jp+myjvn.swidtag ファイルの特徴

項目	特徴
SoftwareIdentity タグ	tagId は, 16 バイトの GUID が推奨されている. 今回，tagId は，GUID の代替案である regid＋製品＋バージョン情報を連結したテキスト形式＝" ipa.go.jp+myjvn_api+3.2.0"で構成した.
Meta タグ	既存 CPE との整合性を確保しつつ, SWID タグから CPE を生成するため，Meta タグを CPE v2.3 生成用に拡張した(図 8). なお，SWID タグから CPE v2.3 を生成する手順は，NIST IR 8060 Draft3 で示されている手順とは異なる.
Payload タグ	ユーザアプリケーションの主要モジュールである MyJVN.class のハッシュ値として MD5, SHA1，SHA256 を記述した.
Link タグ	使用している外部コンポーネントや前提プログラムの CPE v2.2 を記述した(図 10).

```
<swid:Link href="cpe:/a:oracle:jre" rel="related"/>
<swid:Link href="cpe:/a:openssl:openssl" rel="related"/>
<swid:Link href="cpe:/a:openssh:openssh" rel="related"/>
<swid:Link href="cpe:/a:ntp:ntp" rel="related"/>
<swid:Link href="cpe:/a:apache:apr-util" rel="related"/>
<swid:Link href="cpe:/a:pcre:pcre" rel="related"/>
<swid:Link href="cpe:/a:apache:http_server" rel="related"/>
<swid:Link href="cpe:/a:apache:tomcat" rel="related"/>
<swid:Link href="cpe:/a:mysql:mysql" rel="related"/>
```

図 10：外部コンポーネントや前提プログラムの CPE 記述

Link タグに記載した使用している外部コンポーネントや前提プログラムの CPE を用いた脆弱性対策情報検索の流れは，(i)納品物として納品されたユーザアプリケーション用 SWID タグを資産管理ツールに取り込み, (ii)Link タグに記載されている CPE を抽出した後, 脆弱性管理ツールに通知する．(iii)脆弱性管理ツールは，CPE をキーとして MyJVN API でアクセスするというものである(図 11).

(2) mjcheck3 の機能拡張プロトタイプの試作

既存版 mjcheck3 は，CPE を手入力することで，製品と JVN/JVN iPedia 脆弱性対策情報データベースとの紐付けを行っている．この紐付けで，JVN/JVN iPedia 脆弱性対策情報データベースから該当する製品の脆弱性対策情報を取得している．機能拡張版プロトタイプ mjcheck3 では，SWID タグファイルをインポートすることにより，製品と

JVN/JVN iPedia 脆弱性対策情報データベースとの紐付けを行う(図 12). これにより，発注者側での資産管理や脆弱性管理ツールへの手入力工数をなくすことができ，さらに，Link タグに記載されている外部コンポーネントや前提プログラムの CPE を用いた脆弱性対策情報検索が可能となる．

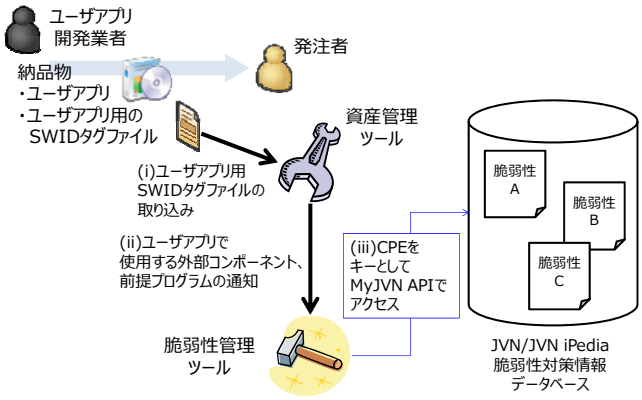


図 11：ユーザアプリケーション用 SWID タグファイルを利用した脆弱性対策情報検索

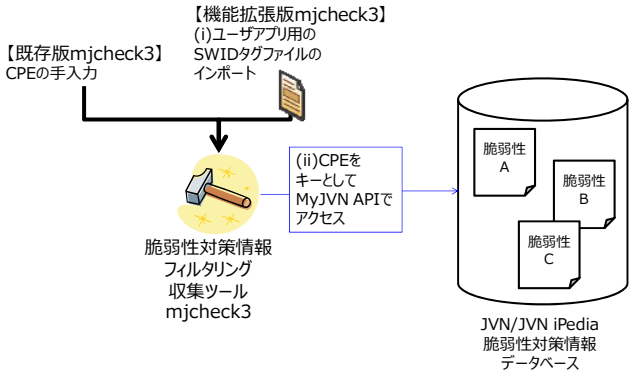


図 12：機能拡張プロトタイプによる脆弱性管理の流れ

(3) 考察

本章では，ユーザアプリケーション用 SWID タグファイルを利用した，ユーザアプリケーション関連の資産管理と脆弱性管理の連携について検討した．検討の結果，ユーザアプリケーション用 SWID タグファイルを活用することで，ユーザアプリケーションを資産管理の一部として管理できる可能性を示した．また，Link タグに記載した使用している外部コンポーネントや前提プログラムの CPE を利用することで，ツールを用いた脆弱性管理実現の可能性を示すことができたと考えている．

7. おわりに

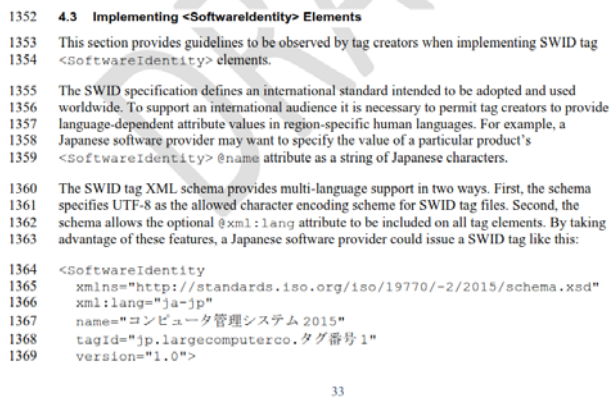
本稿では，脆弱性対策に関わる処理の機械化の推進と共に，脆弱性対策の裾野を広げるために，製品識別子を用いた脆弱性対策情報データベースと資産管理との連携について検討した結果を報告した．

4 章の「ソフトウェア辞書とのデータ連携」では，SAMAC ソフトウェア辞書と JVN/JVN iPedia 脆弱性対策情報データベースとの紐付けは可能であり，資産管理ツールを用いた脆弱性管理実現の可能性を示した．

5 章の「JVN 脆弱性対策機械処理基盤での SWID タグ付与」では，日本語表記の SWID タグ作成にあたっては，日本語と英語の併記，NIST IR 8060 ならびに既存 CPE との整合性の確保の点で，解決すべき課題が残っていることを示した．ただし，2015 年 12 月に公開された NIST IR 8060 Draft4 に，英語以外での SWID タグの表記について記載されており，検討結果の一部が役立てられている．

6 章の「資産管理と脆弱性管理の連携」では，ユーザアプリケーション用 SWID タグファイルを活用することで，ユーザアプリケーションを資産管理の一部として管理できる可能性を示した．

今後，これら検討結果を踏まえ，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を進めていく予定である．



1352 **4.3 Implementing <SoftwareIdentity> Elements**

1353 This section provides guidelines to be observed by tag creators when implementing SWID tag
1354 <SoftwareIdentity> elements.

1355 The SWID specification defines an international standard intended to be adopted and used
1356 worldwide. To support an international audience it is necessary to permit tag creators to provide
1357 language-dependent attribute values in region-specific human languages. For example, a
1358 Japanese software provider may want to specify the value of a particular product's
1359 <SoftwareIdentity> @name attribute as a string of Japanese characters.

1360 The SWID tag XML schema provides multi-language support in two ways. First, the schema
1361 specifies UTF-8 as the allowed character encoding scheme for SWID tag files. Second, the
1362 schema allows the optional @xml:lang attribute to be included on all tag elements. By taking
1363 advantage of these features, a Japanese software provider could issue a SWID tag like this:

```
1364 <SoftwareIdentity
1365   xmlns="http://standards.iso.org/iso/19770/-2/2015/schema.xsd"
1366   xml:lang="ja-jp"
1367   name="コンピュータ管理システム 2015"
1368   tagId="jp.largecomputerco.タグ番号 1"
1369   version="1.0">
```

33

図 13：英語以外での SWID タグの表記
出典：NIST IR 8060 Draft4

謝辞

本研究を進めるにあたって有益な助言と協力を頂いた一般社団法人ソフトウェア資産管理評価認定協会の関係各位に深く感謝申し上げる．

参考文献

[1] ITU-T X.1528 : Common platform enumeration.
https://www.itu.int/rec/T-REC-X.1528/
[2] ISO/IEC 19770-2:2015: Information technology -- Software asset management -- Part 2: Software identification tag.
http://www.iso.org/iso/catalogue_detail.htm?csnumber=65666
[3] NIST IR 8060: DRAFT (Fourth & Final Draft) Guidelines for the Creation of Interoperable Software Identification (SWID) Tags.
http://csrc.nist.gov/publications/PubsDrafts.html#NIST-IR-8060
[4] SAMAC. ソフトウェア辞書について.
http://www.samac.or.jp/software_dictionary.html

付録

付録では，5 章の「JVN 脆弱性対策機械処理基盤での SWID タグ付与」で作成した多言語版 CVSS v3 計算機用の SWID タグファイル ipa.go.jp+myjvn_cvss3.swidtag(付図 1), 6 章の「資産管理と脆弱性管理の連携」で作成したユーザアプリケーション MyJVN 用の SWID タグファイル ipa.go.jp+myjvn.swidtag(付図 2)を示す.

```
<?xml version="1.0" encoding="UTF-8"?>
<swid:SoftwareIdentity
  xmlns:swid="http://jvndb.jvn.jp/schema/swid.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:MD5="http://www.w3.org/2001/04/xmldsig-more#md5"
  xmlns:SHA1="http://www.w3.org/2000/09/xmldsig#sha1"
  xmlns:SHA256="http://www.w3.org/2001/04/xmlenc#sha256"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xsi:schemaLocation="http://jvndb.jvn.jp/schema/swid.xsd
                      http://jvndb.jvn.jp/schema/swid.xsd"
  name="MyJVN CVSS 3.0 Calculator Multi-Language Version"
  version="1.0.0"
  tagId="ipa.go.jp+myjvn_cvss3+1.0.0"
  versionScheme="multipartnumeric"
  >
  <swid:Entity
    name="Information-technology Promotion Agency, Japan"
    regid="ipa.go.jp"
    role="softwareCreator"
    xml:lang="en"/>
  <swid:Entity
    name="独立行政法人 情報処理推進機構"
    regid="ipa.go.jp"
    role="softwareCreator"
    xml:lang="ja"/>
  <swid:Meta title="MyJVN CVSS 3.0 Calculator Multi-Language Version"
    part="a"
    vendor="ipa.go.jp"
    product="myjvn_cvss3"
    version="1.0.0"
    update=""
    edition=""
    language=""
    sw_edition=""
    target_sw=""
    target_hw=""
    productFamily="JVN"
    timestamp="2015-01-26T00:00:00+09:00"
    xml:lang="en"/>
  <swid:Meta title="MyJVN CVSS 計算ソフトウェア 多国語版"
    part="a"
    vendor="ipa.go.jp"
    product="myjvn_cvss3"
    version="1.0.0"
    update=""
    edition=""
    language=""
    sw_edition=""
    target_sw=""
    target_hw=""
    productFamily="JVN"
    timestamp="2015-01-26T00:00:00+09:00"
    xml:lang="ja"/>
  <swid:Payload>
    <swid:File name="ScoreCalc3.swf"
      MD5:hash="55435AEEF41CDFEEE5261138EC60683B"
      SHA1:hash="CBC3E277646A67C5001F50CC2F46370DE443219F"
      SHA256:hash="DE5DD081CAAEB20CC9D1BA246405A30A
                   3AA3D0A7DBB47F9F465E7F4CCFA462A6"/>
  </swid:Payload>
  <swid:Link href="http://jvndb.jvn.jp/cvss/ScoreCalc3.swf" rel="component"
            type="application/vnd.adobe.flash-movie" />
  <swid:Link href="http://jvndb.jvn.jp/cvss/ipa.go.jp+myjvn_cvss3.swidtag"
            rel="swidtag" type="application/xml" />
</swid:SoftwareIdentity>
```

付図 1：http://jvndb.jvn.jp/cvss/ipa.go.jp+myjvn_cvss3.swidtag

```
<?xml version="1.0" encoding="UTF-8"?>
<swid:SoftwareIdentity
  xmlns:swid="http://jvndb.jvn.jp/schema/swid.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:MD5="http://www.w3.org/2001/04/xmldsig-more#md5"
  xmlns:SHA1="http://www.w3.org/2000/09/xmldsig#sha1"
  xmlns:SHA256="http://www.w3.org/2001/04/xmlenc#sha256"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xsi:schemaLocation="http://jvndb.jvn.jp/schema/swid.xsd
                      http://jvndb.jvn.jp/schema/swid.xsd"
  name="MyJVN API"
  version="3.2.0"
  tagId="ipa.go.jp+myjvn_api+3.2.0"
  versionScheme="multipartnumeric"
  >
  <swid:Entity
    name="Information-technology Promotion Agency, Japan"
    regid="ipa.go.jp"
    role="softwareCreator"
    xml:lang="en"/>
  <swid:Entity
    name="独立行政法人 情報処理推進機構"
    regid="ipa.go.jp"
    role="softwareCreator"
    xml:lang="ja"/>
  <swid:Meta title="MyJVN API"
    part="a"
    vendor="ipa.go.jp"
    product="myjvn_api"
    version="3.2.0"
    update=""
    edition=""
    language=""
    sw_edition=""
    target_sw=""
    target_hw=""
    productFamily="JVN"
    timestamp="2015-02-06T00:00:00+09:00"
    xml:lang="en"/>
  <swid:Meta title="MyJVN API"
    part="a"
    vendor="ipa.go.jp"
    product="myjvn_api"
    version="3.2.0"
    update=""
    edition=""
    language=""
    sw_edition=""
    target_sw=""
    target_hw=""
    productFamily="JVN"
    timestamp="2015-02-06T00:00:00+09:00"
    xml:lang="ja"/>
  <swid:Payload>
    <swid:File name="MyJVN.class"
      MD5:hash="21c4dd282ce97ea67da7e08f31311ba2"
      SHA1:hash="750e9e7dfd57b60a0cb9114cc5f88a9ba82f14cc"
      SHA256:hash="86ce76595fa4ff0e487079206178ddc9
                   a2e37c58b5232d4a4af0442805cb37c2"/>
  </swid:Payload>
  <swid:Link href="cpe:/a:oracle:jre" rel="related"/>
  <swid:Link href="cpe:/a:openssl:openssl" rel="related"/>
  <swid:Link href="cpe:/a:openssh:openssh" rel="related"/>
  <swid:Link href="cpe:/a:ntp:ntp" rel="related"/>
  <swid:Link href="cpe:/a:apache:apr-util" rel="related"/>
  <swid:Link href="cpe:/a:pcre:pcre" rel="related"/>
  <swid:Link href="cpe:/a:apache:http_server" rel="related"/>
  <swid:Link href="cpe:/a:apache:tomcat" rel="related"/>
  <swid:Link href="cpe:/a:mysql:mysql" rel="related"/>
</swid:SoftwareIdentity>
```

付図 2：ipa.go.jp+myjvn.swidtag