MyJVN を用いた脆弱性対策情報提供サービスの検討

寺田真敏†1 杉山賢†1 山岸正†1
小林浩昭†1 土居範久†2

†1)（独）情報処理推進機構（IPA）
〒113-6591 東京都文京区本駒込 2-28-8
†2) 中央大学大学院 理工学研究科
〒112-8551 東京都文京区春日 1-13-27

概要：MyJVN は，処理の機械化や自動化を考慮した流通基盤上に，JVN の脆弱性対策情報を利用したサービスを構築するフレームワークである。しかしながら，流通基盤上での交換される情報は文書情報であり，脆弱性の有無をチェックして対策を促すなどの脆弱性対策に関わる処理の機械化については発展途上にある。本稿では，脆弱性対策情報の流通基盤の拡張として，セキュリティ問題をチェックする手続き仕様 OVAL（Open Vulnerability Assessment Language）を用いた脆弱性対策に関わる処理の機械化を提案すると共に，開発したプロトタイプシステムについて述べる。

キーワード：セキュリティ，脆弱性，Web API，JVN，OVAL

Feasibility study of vulnerability information service by MyJVN

Masato Terada†1 Ken Sugiyama†1 Tadashi Yamagishi†1
Hideaki Kobayashi†1 Norihisa Doi†2

†1) Information-technology Promotion Agency, Japan
2-28-8 Honkomagome, Bunkyo, Tokyo, 113-6591 Japan
†2) Graduate School of Science and Engineering, Chuo University.

Abstract: MyJVN is framework for exchange of security information and automation of vulnerability countermeasure. Currently, MyJVN has a Web service API that is based on CPE (Common Platform Enumeration) as a structured naming scheme for products. And most of security information of MyJVN is deployed as human readable documents. It is necessary to improve the security information service environment for automation of vulnerability countermeasure. In this paper, firstly we will explain the specification and applications of MyJVN. Secondly, we will introduce our feasibility study of OVAL (Open Vulnerability Assessment Language) for MyJVN.

Key words: Security, Vulnerability, Web API, JVN, OVAL

1 はじめに

2004年の情報セキュリティ早期警戒パートナーシップ開始以降，国内においても，JVN（Japan Vulnerability Notes），製品発展ベンダ，コミュニティなど様々な層での脆弱性対策情報の提供が充実している。しかし，対策情報の多くは主に文書として構成されているために，脆弱性的有無をチェックして対策を促すなどの脆弱性対策に関わる処理の機械化については発展途上にある。

米国では，2002年のFISMA（Federal Information Security Management Act：連邦情報セキュリティマネジメント法）の施行以降，セキュリティ規格やガイドラインに従い，情報システムにセキュリティ要件を反映する活動が進められている。特に，セキュリティ設定に関する作業を手作業で行うと，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満たす可能性があることから，作業の自動化が試みられている。また，EUでは情報セキュリティ推進機関である ENISA（European Network and Information Security Agency：欧州ネットワーク情報セキュリティ庁）を中心としてセキュリティ関連情報の共有システムを実現するためのプロジェクトを推進している。

このような状況を踏まえ，JVNでは，情報セキュリティの脆弱性対策が国内だけでなく，国際的に
も対応可能とするグローバルなJVN実現に向け，脆弱性対策に関わる共通基準を積極的に採用すると共有する表1).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を推進している[1]。

本稿では，脆弱性対策情報の流通基盤の拡張として，セキュリティ問題をチェックする手続き仕様OVAL（Open Vulnerability Assessment Language）を用いた脆弱性対策に関わる処理の機械化を提案すると共に，開発したプロトタイプシステムについて述べる。

表1：JVNで採用している共通基準[2][3][4][5]
<table>
<thead>
<tr>
<th>共通基準</th>
<th>概要</th>
</tr>
</thead>
<tbody>
<tr>
<td>脆弱性の共通識別子CVE (Common Vulnerabilities and Exposures)</td>
<td>プログラムのセキュリティ問題に一意の番号（脆弱性識別子）を付与する仕様</td>
</tr>
<tr>
<td>共通脆弱性評価システムCVSS (Common Vulnerability Scoring System)</td>
<td>脆弱性自体の特性，バッチの提供状況，ユーザ環境での影響度などを考慮した脆弱性の影響度を評価する仕様</td>
</tr>
<tr>
<td>共通脆弱性タイブ一覧CWE (Common Weakness Enumeration)</td>
<td>設定上のセキュリティ問題（脆弱性の種類を識別するための仕様</td>
</tr>
<tr>
<td>共通プラットフォーム一覧CPE (Common Platform Enumeration)</td>
<td>情報システム，プラットフォーム，ソフトウェアパッケージに一意の名称を付与する仕様</td>
</tr>
</tbody>
</table>

表2：SCAPを構成する仕様群の概要
<table>
<thead>
<tr>
<th>共通基準</th>
<th>概要</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCE (Common Configuration Enumeration)</td>
<td>プログラムが従属するための設定上のセキュリティ問題に一意の番号を付与する仕様</td>
</tr>
<tr>
<td>XCCDF（Xensible Checklist Configuration Description Format）</td>
<td>セキュリティチェックリストやチェックマークなどの文書を記述するための仕様</td>
</tr>
<tr>
<td>OVAL（Open Vulnerability Assessment Language）</td>
<td>プログラムのセキュリティ問題を表示上のセキュリティ問題とチェックするための手続き仕様</td>
</tr>
</tbody>
</table>

(2) 共通基準

表1と表2に示した7つの共通基準の他に，米NISTでは，プログラムが従属するための設定上のセキュリティ問題に関して影響度を評価する仕様CCSS（Common Configuration Scoring System）の検討を行なっている[7]。セキュリティに関わる共通基準を整備することにより，セキュリティ対策全般に関わる処理の機械化も促進されることになることから，CWE，CCE，OVALなどの仕様開発を担当しているMITRE社では，共通基準の拡張"Making Security Measurable"を推進している。進行している代表的な共通基準は次の通りである[8][9][10]。

- 攻撃パターンの識別を共通化するCAPEC（Common Attack Pattern Enumeration and Classification）
- イベントの識別を共通化するCEE（Common Event Expression）
- 情報資源のセキュリティ評価結果を記述するCRF（Common Result Format）

3 MyJVN

本稿では，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について述べた後，解決したい課題について提示する。

3.1 JVN


- JVN：国内の情報セキュリティ早期警戒ネットワーク，および海外の調整機関に届けられた脆弱性関連情報に関わる製品開発者に調整した脆弱性対策情報とタイムリーに公開する。
- JVN iPedia：国内の製品開発者から公開された
対策情報、および海外の脆弱性対策データベースに登録された情報に基づき、国内で利用されている製品を対象にした脆弱性対策情報を網羅し蓄積する。

3.2 MyJVN

MyJVNは、処理の機械化や自動化を考慮した流通基盤上に、JVNの脆弱性対策情報を用いたサービスを構築するフレームワークである。2008年10月時点で、脆弱性対策情報の流通基盤として、共通プラットフォーム一覧CPE、WebサービスAPI(MyJVN Web API)を、次に示す3つのXMLフォーマットを規定している。

- JVN RSS (JVN RDF Site Summary)[13]
- mod_sec (Qualified Security Advisory Reference)[14]
- VULDEF (The VULnerability Data publication and Exchange Format data model)[15]

また、JVNの脆弱性対策情報を用いたサービスとして、MyJVN Web APIと組み合わせ、「製品にどのような脆弱性が存在するのか」という視点から対策情報を選別するフィルタリング型情報提供サービスを実現している。

3.3 解決したい課題

本節では、脆弱性対策情報の流通基盤と脆弱性対策情報を用いたサービスの視点から解決したい課題を示す。

- 脆弱性対策情報の流通基盤における課題
  流通基盤上で交換される情報は文書情報であり、脆弱性の有無をチェックして対策を促すなどの脆弱性対策に関する処理の機械化については発展途上にある。

- 脆弱性対策情報用いたサービスにおける課題
  フィルタリング型情報提供サービスでは、MyJVN Web APIを利用した脆弱性対策情報チェックツールにより、定規的な脆弱性対策情報をチェック支援を実現している。しかしながら、ツール初期起動時に、普及している製品等をフィルタリング条件とした既定設定か、ツールが個別に設定するカスタム設定のいずれかにベンダ名、製品名などのフィルタリング条件を設定する必要がある（図1）。

4 脆弱性対策情報の流通基盤の拡張

本章では、前述の課題を解決するため、脆弱性対策情報の流通基盤の拡張項目である、セキュリティ問題をチェックする手続き仕様OVALの導入と、OVALを利用するためのWebサービスAPIの拡張について述べる。

4.1 セキュリティ問題をチェックする手続き仕様OVALの導入

OVAL[16]は、プログラム上のセキュリティ問題や設定上のセキュリティ問題をチェックする手続き仕様である。2002年10月に開催されたSANS Network Security 2002において、MITRE社から、その仕様が開示された。XMLを用いて汎用的に作られた仕様であり、セキュリティ修正プログラムの適用状況や、稼動しているプラットフォーム/ソフトウェアパッケージの判定にも利用できる。OVAL開発を担当しているMITRE社では、チェック方法をXMLに記述する仕様だけではなく、そのXMLに従い『プログラム上のセキュリティ問題』や『設定上のセキュリティ問題』をチェックするプログラム(OVALインタプリタ)を参照実装として提供している。

MyJVNにおいても、このようなセキュリティ問題をチェックする手続き仕様を整備していくことで、脆弱性の有無をチェックして対策を促すなどの脆弱性対策に関する処理の機械化が実現できる。将来には、チェック項目が記載されたXMLファイル(以下、OVAL定義データ)を流通基盤上で交換することにより、脆弱性対策の支援や国際間での脆弱性対策情報の相互運用といった可能性が広がる。

4.2 WebサービスAPIの拡張

OVALの導入に伴い、「システムにはどのような脆弱性が存在するのか」「システムにはどのような製品をインストールされているのか」という視点からチェック項目が記載されたOVAL定義データを提供することで、脆弱性の有無チェック処理の機械化だけではなく、脆弱性対策情報チェックツールのフィルタリング条件自動設定が可能となる。また、OVAL定義データを提供するにあたり、フィルタリング条件を自動設定する機能を実装する必要がある。
フィルタリング型情報提供サービスの Web サービス API を拡張することで対応できる（表3）。

表3：MyJVN Web APIの拡張メソッド一覧

<table>
<thead>
<tr>
<th>名称</th>
<th>概要</th>
</tr>
</thead>
</table>
| OVALID    | フィルタリング型情報提供サービスの Web サービス API を拡張するためのメソッド

5 フィルタリング型情報提供サービスの拡張

本章では、フィルタリング型情報提供サービスの拡張として、開発したプロトタイプシステムについて述べる。プロトタイプシステムは、脆弱性対策情報の流通基盤の拡張で導入した OVALID 定義データを取得する機能と、OVALID 定義データを解釈する簡易インタプリタ機能を実装している。

5.1 SWF バージョンチェックツール

SWF バージョンチェックツールは、Web ブラウザ上で稼動する SWF ペースの GUI ツールである。OVALID 定義データの<file_state>に記載された Flash Player バージョン情報（図2）と、自身のバージョンとの比較を行うことで Flash Player のバージョンチェックを実施する。

バージョン情報が一致しない場合には、最新版のインストールが必要であると判断し、SWF ベースの GUI ツールからダウンロードサイトへのリンクを有効にする（図3）。

図2：Flash バージョン用 OVALID 定義データ（抜粋）

5.2 脆弱性影響有無チェックツール

脆弱性影響有無チェックツールは、Web ブラウザ上で稼動する Java ベースの GUI ツールである。OVALID 定義データに記載された Windows システムのレジストリ情報を利用して、レジストリ設定値のチェックを実施する。例えば、Windows システムにおいて、すべての種類のドライプが自動再生無効となっている場合には、レジストリキー "NoDriveTypeAutoRun" の値が 0xFF が設定されている。OVALID 定義データでは、これらの情報を<registry_object>と<registry_state>に格納する（図4）。

チェック結果は、図5に示すようにレジストリ設定値の一致を提示すると共に、Java ベースの GUI ツールから関連サイトへのリンクを有効にする。

図4：レジストリ "NoDriveTypeAutoRun" 用 OVALID 定義データ（抜粋）

図5：脆弱性影響有無チェックツールの結果表示

5.3 脆弱性対策情報チェックツール

脆弱性対策情報チェックツールは、Web ブラウザ上で稼動する Java ベースの GUI ツールである。開発したプロトタイプは、フィルタリング条件設定機能、脆弱性対策概要表示機能の2つの機能から構成している。

（1）フィルタリング条件設定機能（図8の左上部パネル）

MyJVN Web API の拡張メソッドである getOvalList と getOvalData を介して取得した OVALID 定義データを用いて、レジストリ設定値のチェックを実施する。レジストリ設定値からインストールされていると判定した場合には、インストールされている製品の一覧を表示する。
図 6：OVAL 定義一覧のレスポンス例

getOvalList では、図 6 に示すようなチェックの対象となる製品の OVAL 定義一覧を取得する。また、getOvalData では、図 7 に示すように製品インストールを判定するレジストリ設定値データを取得する。図 7 の場合、レジストリの存在を、<registry_object> を用いて確認する。

図 7：OVAL 定義データ取得のレスポンス例

(2) 脆弱性対策概要情報表示（図 8 の右パネル）
一覧から選択した製品の CPE 名をフィルタリング条件として設定し、脆弱性対策概要情報を取得した RSSS+mod_sec で記載された XML 形式の結果を表示する。脆弱性対策概要情報の一覧から項目を選択すると、ブラウザ上に当該項目の脆弱性対策詳細情報を表示する。
6 おわりに

本稿では、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 の課題について、脆弱性対策情報の流通基盤と脆弱性対策情報を用いたサービスの視点から示した。また、課題を解決するため、セキュリティ問題をチェックする手続き仕様 OVAL を用いた脆弱性対策に関する処理の機械化と、Web サービス API の拡張について提案すると共に、開発したプロトタイプシステムについて報告した。

脆弱性対策に関する処理の機械化を推進するにあたり、脆弱性対策情報の流通基盤の活用、特に、提供している Web サービス API の活用情報であると考えています。

今後は、図 9 の示すような Web サービス API 活用促進のための実装実証を示しながら、脆弱性対策に関する処理の機械化を目指すフレームワーク MyJVN の普及推進を進めていく予定である。

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、高崎仁氏ならびに、IPA の関係者各位に深く感謝致します。

参考文献
1）寺田、杉山他，“脆弱性対策情報の利活用基盤 MyJVN の提案”、CSS2008 (2008)
2）共通脆弱性識別子 CVE 概説
http://www.ipa.go.jp/security/vuln/CVE.html
3）共通脆弱性評価システム CVSS v2 概説
http://www.ipa.go.jp/security/vuln/SeverityCVSS2.html
4）共通脆弱性タイプ一覧 CWE 概説
http://www.ipa.go.jp/security/vuln/CWE.html
5）共通ブラッシュフォーム一覧 CPE 概説
http://www.ipa.go.jp/security/vuln/CPE.html
10）CRF: Common Result Format, http://makingsecuritymeasurable.mitre.org/crf/
11）寺田、高田、土居, “脆弱性対策情報データベース JVN の提案”、情報処理学会 Vol.46 No.5 (2005)
12）経済産業省、「情報セキュリティ早期警戒パートナーシップ」の運用開始について
16）OVAL: Open Vulnerability Assessment Language http://oval.mitre.org/