情報処理学会 コンピュータセキュリティ シンポジウム 2005, pp.667-672 (Oct.26-28, 2005)

マルチベンダ環境の情報システムを対象とした脆弱性管理システムの検討

菊地 大輔 †1　　寺田 真敏 †2†3†4　　福澤 淳二 †2　　土居 範久 †1

†1 中央大学大学院 理工学研究科 情報工学専攻　〒 112-8551 東京都文京区春日 1-13-27

†2 独立行政法人 情報処理推進機構　〒 113-6591 東京都文京区本駒込二丁目 28 番 8 号
文京グリーンコートセンターオフィス 16 階

†3 慶應義塾大学 大学院 理工学研究科　〒 223-8522 神奈川県横浜市港北区日吉 3-14-1

あらまし　情報システムを構成するソフトウェア環境は，提供形態ならびに複数のバージョンの混在という視点から見て，多様化が進んでいる．このため，脆弱性対策にあたり，文書という対策情報提供の形態だけで脆弱性の影響有無を判定するという手法では，検査の抜け漏れを完全に防ぐことは難しい．そこで，本稿では，現行の情報システムの脆弱性を管理するために求められる要件を提示すると共に，これら要件を満たすための脆弱性管理システムを提案する．脆弱性管理システムは，パターンファイルによる脆弱性の影響を受けるか否かを判定する機能をベースとし，パターンファイルの配布ならびに，脆弱性による深刻度算出を含んだ機能を提供する．

Study of vulnerability management system on multi-vendor environment

Daisuke Kikuchi†1　　Masato Terada†2†3†4　　Junji Fukuzawa†2　　Norihisa Doi†1

†1 Graduate School of Science and Engineering, Chuo University.
1-13-27 Kasuga, Bunkyo-ku Tokyo 112-8551, Japan

†2 INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN Center Office,
16th Floor, Bunkyo Green Court, 2-28-8, Hon-Komagome, Bunkyo-ku, Tokyo, Japan 113-6591

†3 Graduate School of Science and Technology, Keio University.
3-14-1 Hiyoshi, Kohoku-ku, Yokohama Kanagawa 223-8522, Japan

Abstract　Judging from point of sight that offering style of software and mixture of a several version, multiplicity of software environment increases in an information system. Therefore a method judging influence existence of vulnerability will generate oversight only measures report by a document. This paper shows requirements to manage vulnerability of an existing information system. And it suggests vulnerability management system which satisfy requirements. This vulnerability management system has a function to judge influence existence of vulnerability by a pattern file. In addition, it provides distribution of a pattern file and severity calculation of vulnerability.

1　はじめに

コンピュータウイルスや不正アクセスなどの攻撃活動を誘発する要因として，ソフトウェア開発時のバグや設定ミスによる脆弱性がある．情報システムのセキュリティ担当者は，セキュアな環境を維持するために脆弱性に関する情報を CERT/CC [1], JVN [2] などから収集し，適切な対応とる必要がある．しかし，情報システムを構成するコンピュータの環境は，クライアント，サーバ，ルータなど多岐にわたる．また，構成要素となる OS やアプリケーションは，それぞれのコンピュータごとに異なる．この結果，コンピュータ毎に脆弱性の確認方法や影響範囲

†4)(株) 日立製作所 システム開発研究所
セキュリティシステム研究部
〒 212-8567 神奈川県川崎市幸区鹿島田 890

も異なり，脆弱性の管理に掛かる負担は大きくなっている．

ソフトウェア製品の開発ベンダによっては，脆弱性の対応に掛かる負担を低減させるために，ツールを用意している場合もある．例えば，Microsoft 社では Microsoft Update [3] を提供している．Microsoft Update では，製品開発ベンダの Web サイトと連動することによって，脆弱性の有無チェックから修正プログラム適用までの一連の対策を実施する．しかし，その適用範囲は，製品開発ベンダの取り扱い製品に限られる．

そこで，本稿ではマルチベンダ環境のソフトウェアで構成された情報システムの脆弱性を一元的に管理するシステムについての考察を述べる．また，脆弱性の有無の検査方法を記述する言語・フレームワークである OVAL(Open Vulnerability Assessment Language) [4] を拡張することで実装を行った脆弱性管理システムについて述べる．

本稿の構成は次のとおりである．2 章で脆弱性管理に関する課題について述べる．3 章でマルチベンダ環境の情報システムを対象とした脆弱性管理システムに求められる要件を挙げる．そして，4 章で検討した要件を満たすプロトタイプシステムについて述べ，5 章でまとめを行う．

2　脆弱性管理に関する課題

本章では，脆弱性の状態管理について，脆弱性の有無の判定，脆弱性の対応・状況管理の視点から整理し，課題について述べる．

2.1　脆弱性の有無の判定に関する課題

現在，脆弱性の性質ならびに対策に関する情報 (以下，脆弱性対策情報) の提供は，Web ページやメーリングリストなどによって行われている．しかし，このような文章による脆弱性対策情報をもちいて脆弱性有無の判定を行う方法には，次のような課題がある．

- 対策の必要な脆弱性を見逃したり，脆弱性対策情報に対する誤認識が発生する可能性がある．
- 脆弱性対策情報の内容や検査方法の理解にスキルを必要とするため，ユーザの負担を軽減できない．

製品開発ベンダによっては，この脆弱性の判定に掛かる負担を軽減するために，脆弱性の状態を管理するためのツール (以下，脆弱性管理ツール) を提供している．脆弱性管理ツールを用いることにより，文章による脆弱性対策情報に基づく検査の課題であるユーザのスキルや負担を軽減できる．しかし，脆弱性管理ツールを用いる方法にも，管理対象となるソフトウェアが限定される (他社製のソフトウェアについては脆弱性判定できない) などの課題が残っている．この課題は，マルチベンダ環境の情報システムを構成するソフトウェアの多様性を考慮すると，非常に大きいものである．

2.2　脆弱性の対応・状態管理の課題

脆弱性ありと判定された場合には，脆弱性の除去，他の手段による脆弱性の回避などの対応が必要となる．また，マルチベンダ環境の情報システムにおいては，多様な機器が存在するため，脆弱性への対応優先度を決定することを求められる．ただし，脆弱性の対応優先度を決定するためには，脆弱性の深刻度評価に必要な情報を，常に収集しなければならない．しかし，情報システムのセキュリティ担当者が，常に脆弱性の攻略に関する様々な情報 (攻略コードの公開，修正プログラムの提供状態など) といった脆弱性に関連する状況変化を監視し続けることは難しい．

また，脆弱性対応は，次々に新しい脆弱性が発見されるため継続して実施する必要がある．そのため，脆弱性の有無の判定結果などの対応履歴を管理する機能も求められる．

3　脆弱性対応管理システム

本章では，マルチベンダ環境の情報システムを対象とした脆弱性管理システムに求められる要件と提案する脆弱性管理システムの機能について述べる．

3.1 脆弱性有無の判定機能

第 2 章で提示した課題を解決し，マルチベンダ環境において脆弱性の有無を判定するには，以下のような要件を満たす機能が必要となる．

要件 1 ユーザスキルなどに依存しない判定機構を持つこと．

要件 2 特定のソフトウェア環境に依存しない判定機構であること．

要件 3 判定機構の仕様などを製品開発ベンダに公開できること．

そこで，本稿では，特定のソフトウェア環境になるべく依存しない判定手法として，ソフトウェアのバージョン番号やファイルサイズ，ファイルのハッシュ値などといったローレベルの情報を用いて脆弱性有無の判定を行う機能を脆弱性管理システムに取り込む (要件 2)．さらに，ローレベルの情報による脆弱性の判定方法を記述可能なパターンファイル，ならびにパターンファイル流通のためのフレームワーク (図 1) を提案する (要件 1，3)．

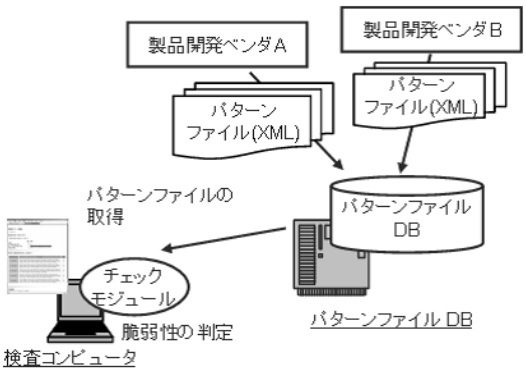

図 1: パターンファイル提供のフレームワーク

図 1 に示すフレームワークでは，パターンファイルを製品開発ベンダが作成し，一元的にパターンファイルを管理する DB(以下，パターンファイル DB) に格納する．脆弱性の有無を判定したいユーザは，あらかじめ，パターンファイルを用いて判定を行うチェックモジュールを情報システム内に導入しておく．そして，パターンファイル DB から必要なパターンファイルを取得して，脆弱性の有無を判定する．

3.2 脆弱性の対応・状態管理機能

脆弱性への対応優先度の決定ならびに，対応の履歴を管理するには，以下のような要件を満たす機能を必要とする．

要件 4 脆弱性の深刻度に関する情報を取得する機構を持つこと．

要件 5 対応履歴をレポートとして出力できること．

本稿では，脆弱性の深刻度の評価情報を，図 2 に示すように，脆弱性の深刻度を評価するサービスを提供するサーバ (以下，脆弱性深刻度評価サーバ) と連携することによって，脆弱性管理システムに取り込むことを提案する (要件 4)．

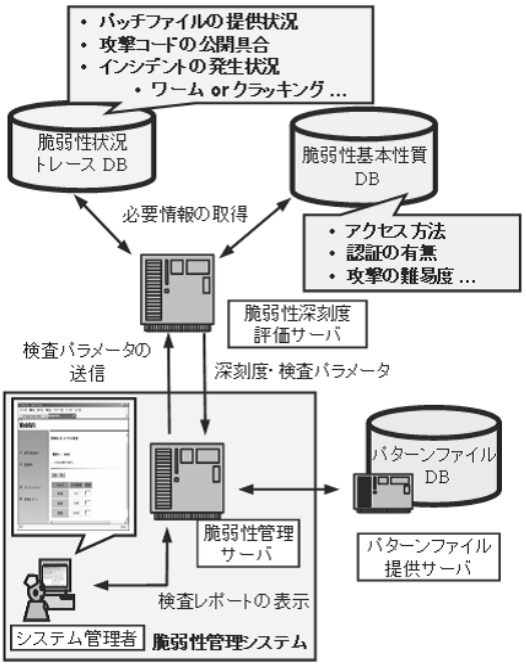

図 2: 脆弱性深刻度評価システムとの連携

深刻度の評価を外部の別のサーバに委託することによって，深刻度の計算方法の変更や脆弱性を取り巻く環境の変動などに柔軟に対応できることになる．

また，脆弱性の判定結果などの対応履歴を管理するために，検査履歴・対応状況の一覧表示，検査状況の統計情報を状態レポートとして出力する (要件 5).

4 プロトタイプシステムの実装

マルチベンダ環境を対象とした脆弱性管理システムに求められる要件を満たすプロトタイプシステムを実装した．本章では，プロトタイプシステムの構成と実装方法について述べる．

4.1 システム構成

脆弱性管理システムでは，OVAL を拡張したパターンファイルを用いて，マルチベンダ環境の情報システムの脆弱性を管理する (図 3)．さらに，脆弱性の判定結果と外部の脆弱性深刻度評価サーバから得た情報から，情報システム内の脆弱性の状態レポートを生成する．

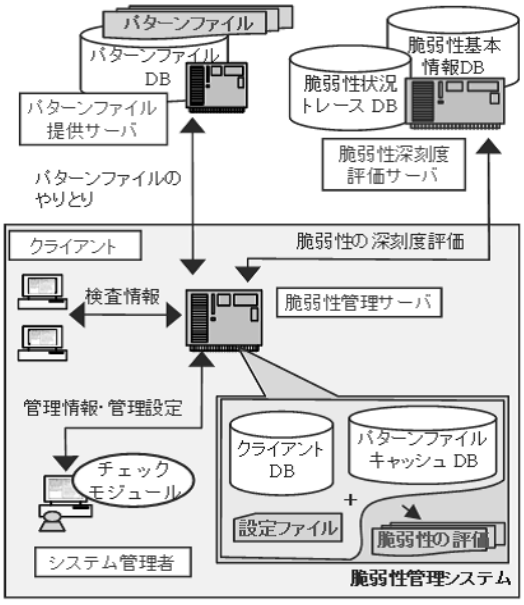

図 3: 提案する脆弱性管理システムの構成

4.2 パターンファイル構成

プロトタイプシステムにおけるパターンファイルを図 4 に示す．図 4 のパターンファイルは，OVAL で提供されている検査定義ファイルを拡張したものである．具体的には，製品開発ベンダが個別にパターンファイルを生成できるようタグ項目の変更や脆弱性の深刻度を計算するためのパラメータの追加などを行っている．

パターンファイルの構成としては，脆弱性に関する内容を説明する部分と脆弱性有無を判定するための検査項目部分の 2 つからなる．脆弱性に関する内容を説明する部分には，脆弱性の深刻度を評価するためのパラメータのうち，脆弱性の基本的な性質 (攻撃方法，攻撃前の認証有無など) を記載できるよう拡張を行なっている．

```
<?xml version="1.0" encoding="UTF-8"?>
<oval xmlns="http://oval.mitre.org/XMLSchema/oval" ...>
 <definitions>
   <!-- 脆弱性に関する情報を記載 -->
   <definition id="IPA****" class="vulnerability">
     <affected family="windows">
       <windows:platform>Microsoft Windows XP</windows:platform>
       <product> … </product> </affected>
     <contributors>
        <submitter organization="..."> ... </submitter>
     </contributors>
     <cveid status="CAN">2004-0843</cveid>
       <dates> ... </dates>
     <description> ... </description>
     <status>ACCEPTED</status>  <version>1</version>
     <notes>
       <!-- 脆弱性深刻度評価パラメータ -->
       <note author="CVSS" date="2005-05-30">
         AccessVector=remote, AccessComplexity=low,
         Authentication=required,ConfidentialityImpact=partial,
         IntegrityImpact=partial,AvailabilityImpact=complete,
         ImpactBias=confidentiality </note> </notes>
     <criteria>
       <software operation="AND">
         <criterion test_ref="wft-101" comment="…" />
       </software> </criteria> </definition>
 </definitions>

 <!-- 脆弱性の有無を判定する方法を記載 -->
 <tests>
   <file_test id="wft-101" comment="…" xmlns="...">
     <path datatype="component">
       <component type="registry_value">
         HKEY_LOCAL_MACHINE\SOFTWARE\... </component>
       <component type="literal"> …</component> </path>
   </file_test> </tests>
</oval>
```

図 4: OVAL パターンファイルの拡張 (一部省略)

4.3 脆弱性判定チェックモジュール

脆弱性有無の判定には，検査対象となるコンピュータ側に導入するチェックモジュールと，Web アプリケーションとして実装した脆弱性管理サーバとの連携によって行う．

検査対象側のチェックモジュールは，Web ブラウザを用いて実行する ActiveX タイプと，コマンドラインで実行するタイプの 2 つを用意した．ActiveX タイプのチェックモジュールは，検査対象となるコンピュータを利用しているユーザ自身によって脆弱性有無を確認するための利用形態を想定している．そのために，ユーザにとって抵抗の少ないインタフェースである Web ブラウザで実行可能な ActiveX を用いている (図 5)．また，コマンドラインタイプのチェックモジュールは，バックグランドで定期的に実行し，脆弱性有無を確認することで，対応履歴を管理していくという利用形態を想定している．

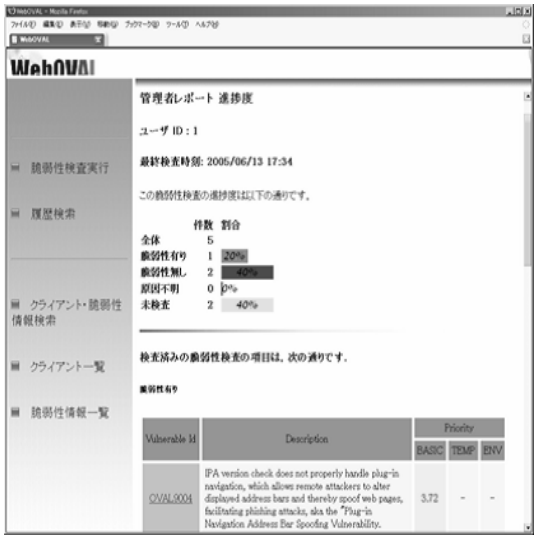

図 5: ActiveX 型チェックモジュールの結果表示

4.4 脆弱性の深刻度情報の取得機能

脆弱性の深刻度評価は，検査対象となるコンピュータの脆弱性検査履歴などから算出したパラメータを脆弱性深刻度評価サーバにリクエストすることによって行う．脆弱性深刻度評価サーバでは，パラメータの追加や補完を行い，脆弱性の深刻度を算出して結果を返す．なお，プロトタイプでは，CVSS(Common Vulnerability Scoring System) [6] を深刻度評価サーバの評価算出機構として実装した．

4.5 脆弱性検査状況レポート機能

脆弱性検査状況レポート機能は，コンピュータの検査結果やパターンファイルの情報をもとに，セキュリティ管理者に情報システムの状態の報告を行う．

このレポートでは，図 6 のようにレポート取得時点における検査状況を分析し，統計情報などを表示する．また，脆弱性の検査履歴や対応状況をもとに，検査対象となるコンピュータの評価を行いレポートする．これによって，情報システムのセキュリティ管理者は，一定期間内の情報システムの検査状況を知ることができる．

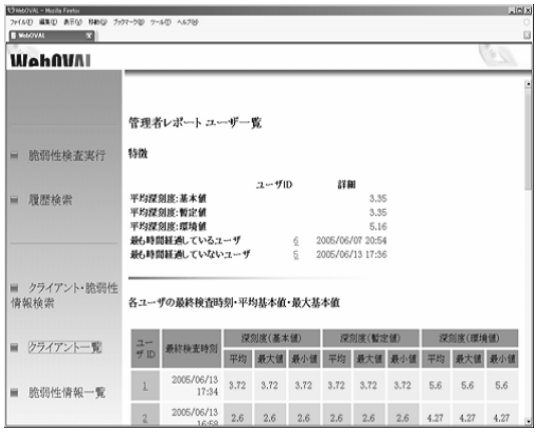

図 6: 検査状況レポートの表示画面

5 おわりに

本稿では，マルチベンダ環境のソフトウェアで構成された情報システムの脆弱性を一元的に管理するシステムを考察し，求められる要件を検討した．また，検討内容をもとに，OVAL で提供されている検査定義ファイルに，タグ項目の変更や脆弱性の深刻度を計算するためのパラメータの追加を行うと共に，チェックモジュールと Web アプリケーション連携による脆弱性

有無の判定，脆弱性の深刻度評価，脆弱性検査状況レポート機能を備えた脆弱性管理システムのプロトタイプを実装した．

今後は，脆弱性管理システムのプロトタイプ評価を行なうと共に，JVNRSS [7] のような脆弱性対策情報の配信サービスとの連携による脆弱性検査状況レポートの充実や，JVN TRnotes [8] のような脆弱性の状態情報を提供するサービスとの連携による深刻度評価機構の拡張を検討している．

謝辞

本研究は 21 世紀 COE プログラム「電子社会の信頼性向上と情報セキュリティ」および科学技術振興調整費 (人材育成，科学技術総合研究) の支援を受け，IPA と協力して実施している．本研究を進めるにあたって有益な助言と協力を頂いた，貫井千鶴氏，COE，科学技術振興調整費の関係者各位ならびに IPA の関係者各位に深く感謝致します．

参考文献

[1] CERT/CC(Computer Emergerncy Response Team/Coordination Center) http://www.cert.org/

[2] JVN(JP Vendor Status Notes) http://jvn.jp/nav/jvn.html

[3] Microsoft Update http://update.microsoft.com/microsoftupdate/v6/

[4] OVAL(Open Vulnerability Assessment Language) http://oval.mitre.org/

[5] IPA(Information-Technology Promotion Agency, Japan) http://www.ipa.go.jp/

[6] CVSS(Common Vulnerability Scoring System) http://www.packetfactory.net/papers/

[7] JVNRSS http://jvn.jp/rss/

[8] TRnotes http://jvn.jp/tr/index.html

[9] 寺田真敏，土居範久 他: “Status Tracking Notes; 時系列イベント情報の共有”，情報学会研究報告 2004-CSEC-25(7)

[10] 菊地大輔，寺田真敏，土居範久 他: “バージョン情報を用いた脆弱性ソフトウェア検査システムの検討”，情報学会研究報告 2004-CSEC-25(8)

[11] 中村章人，戸村哲: “XML と SOAP によるセキュリティ関連情報 Web サービス”，情報処理学会第 65 回全国大会講演論文集,3,pp.195-1962004.

[12] 中村章人，戸村哲: “XML によるセキュリティ関連情報 Web サービス”，マルチメディア通信と分散処理ワークショップ論文集, pp275-280, 2002.