脆弱性情報 Web サービスを利用した Linux 脆弱性管理ツールの開発*

中村章人† 野辺良一‡ 松田勝己§ 松下浩之** 戸村 哲†

産業技術総合研究所† SRA‡ リソナル§ 創夢**

1 はじめに

セキュリティ対策の重要な方策の一つが脆弱性の管理である。しかし、膨大な脆弱性情報を収集・分析するコストは大きく、システムの状態に合わせて適切な対策を行うのは難しい。

本稿では、脆弱性の検出から解決までの一連のプロセスを省力化するツールについて述べる。我々が開発した脆弱性管理システムは、どの脆弱性をどう解決するか、すなわち解決方策とその適用のタイミングは人間の判断に委ねるが、それ以外の処理は自動化した。また、脆弱性の深刻さや管理ドメインの状態を定量的に評価することで、対策プロセスの効率化や効果の把握を容易にした。

2 脆弱性の解決プロセスと定量化

一般に、個々の脆弱性に対して、ユーザ（システム管理者）は以下のプロセスによりその解決を図る。

① 情報収集
② 検出(使用しているソフトウェアへの影響の有無を判定する)
③ 解決策の決定(ソフトウェアの更新または削除、サービスの停止など)
④ 解決策の適用

③と④について、「常に最新版のソフトウェアを使用する」という方策がよく用いられる。ソフトウェア自動更新機能を持つ OS もある。しかし、最新版だからといって脆弱性が排除されているとは限らず、新たな脆弱性があるかもしれない。また、システムやサービスをすぐに停止できない、新版の動作確認ができていないといった理由ですぐに更新できない場合もある。セキュリティ管理上重要なのは、版が最新かどうかではなく、使用しているソフトウェア（とその版）および影響する脆弱性を漏れなく把握すること（②）である。

脆弱性対策のコストは、脆弱性の数、ソフトウェアの種類、ノード数に影響を受ける。限られた時間と人員で効果的な対策を実施するには、脆弱性の定量化という手法が有効である[2]。脆弱性の深刻度を定量化する標準的手法として CVSS (Common Vulnerability Scoring System) [3]がある。

3 脆弱性管理システム

本システムは、上記の脆弱性解決プロセス全体に亘ってシステム管理者の作業を支援する。また、CVSS を基にして、種々の定量的評価結果を提示する。

3.1 システム構成

本システムは、以下のサブシステムから構成される（図 1）。

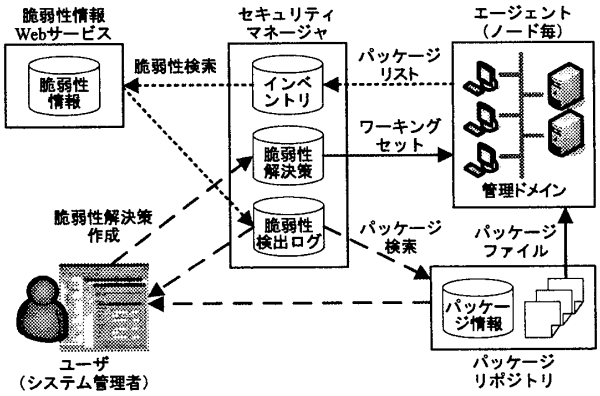

図 1：脆弱性管理システムの構成

- セキュリティマネージャ（SM）： 脆弱性解決プロセスを統轄するサーバ。脆弱性の検出状況など各種データの表示と、解決策の作成および適用に関する種々の操作を行う GUI を持つ。
- エージェント（AGT）： 各ノードで動作し、情報収集および脆弱性解決策の実行を行う。現状では、RPM を使用する Linux システムのみをサポートする。

---

* "A Vulnerability Management Tool for Linux using the Vulnerability Information Web-Service", Akihito NAKAMURA, Yoshikazu NOBE, Katsumi MATSUDA, Hiroyuki MATSUSHITA, Satoru TOMURA

† National Institute of Advanced Industrial Science and Technology (AIST)

‡ Software Research Associates, Inc. (SRA)

§ LISONAL, Ltd.

** SOUM Corporation

- 脆弱性情報 Web サービス（VWS）： 脆弱性に関する検索サービスを提供するサーバ。脆弱性情報は NVD[4]および OSVDB[5]を用いている。データを最新に保つために、データソースと一定間隔で同期する機能を持つ。
- パッケージリポジトリ（PR）： 複数ディストリビューションの Linux パッケージに関する検索サービスを提供するサーバ。パッケージファイルサーバの役割も持つ。現状では、RPM パッケージのみをサポートする。

各サブシステムの実装には Java 言語を用いた。ただし、GUI には JavaScript を用い、Ajax 技法を駆使して利便性を高めた。

動作環境や実装言語の多様化に備えて、サブシステム間の通信には XML-RPC を用いた。一方または双方の認証や通信内容の秘匿が必要な場合、SSL を使用できる。

3.2 脆弱性管理機能

本システムを用いた脆弱性解決プロセスを示す。

3.2.1 脆弱性の検出

AGT はノードにインストールされているソフトウェア（パッケージ）のリストを SM に通知する。このデータを保持するインベントリは「ノードーパッケージ」関係を表す。SM は、各パッケージに影響する脆弱性を VWS で調べ、結果を脆弱性検出ログに記録する。この時点で「ノードーパッケージー脆弱性」関係が検出される。

この処理はシステムが定期的に自動実行するが、ユーザが GUI を通じて起動することもできる。

3.2.2 脆弱性解決策の作成

ユーザは、解決する脆弱性を選択し、その解決策を決定する。脆弱性の選択は、ドメイン全体の脆弱性リストまたはノード毎の脆弱性リストから選択する。多数からの選択を容易にするために、深刻度（CVSS スコア）や公開日などの属性でソートや絞込みを可能にした。

次に、解決策としてパッケージの更新または削除を選択する。更新では、PR を検索して特定の版を指定するか、「最新版」を指定する。このように「脆弱性ーパッケージ」関係のインスタンス毎に作成した解決策をワーキングセット（WS）と呼ぶ。

3.2.3 脆弱性解決策の適用

作成済みの WS をどのノードにいつ適用するかを決定する。システムは、WS を適用可能なノード、すなわちそのパッケージがインストールされているノードのリストを提示する。ユーザは、この中から適用するノードを選択する（複数選択可）。最後に、適用する時間帯を日時のペアで指定する。これで WS は実行可能状態になる。

AGT は、自分宛の実行可能な WS を定期的に検索し、見つかればそれを取得する。指定された時間帯内で WS を実行し、その結果を SM に送信する。指定の時間帯に実行できなかった場合はエラーである。更新 WS の場合、そのバージョンのパッケージファイルを PR から取得する。

4 性能評価

脆弱性検出の性能を図 2 に示す。ここで応答時間は、SM が脆弱性検索要求を VWS に送信してからその結果を受信し終えるまでの時間である。Intel Pentium 4 3GHz / DDR400 2GB の PC で Windows XP SP2、JDK 5.0、MySQL 5.0 を使用した。

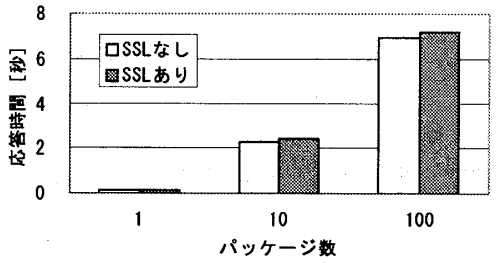

図 2： 脆弱性検出の応答時間

5 おわりに

我々が開発したシステムは、現在多くの部分を人手に頼っている脆弱性管理作業を自動化し、IT システムのセキュリティ向上に貢献する。管理ドメイン内の脆弱性の所在をすばやく検出し、その対処を正確かつ確実に行える。

今後の課題は、スケーラビリティの検証、より多様なシステム（deb 系 Linux、Solaris, Windows など）への適合などである。

参考文献

[1] 中村, 戸村: 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, 2003 年 3 月.
[2] 日本ネットワークセキュリティ協会: 脆弱性定量化に向けての検討報告書, 2007 年 3 月.
[3] http://www.first.org/cvss/
[4] http://nvd.nist.gov/
[5] http://www.osvdb.org/