2004－CSEC－25　(8)
2004／5／21

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

菊地 大輔 [†1]　寺田 真敏 [†2,†3]　千葉 雄司 [†3]　矢田 健一 [†3]　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] 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

[†3] 中央大学 研究開発機構　〒112-8551 東京都文京区春日 1-13-27

E-mail: †1 dkikuchi@doi-lab.ise.chuo-u.ac.jp

あらまし　あらゆるソフトウェアを対象として, 脆弱性を持つソフトウェアの有無を検査するシステムの開発をおこなっている. このシステムは，検査対象のコンピュータにインストールする検査ソフトウェアと，脆弱性に関する情報を保持するデータベースからなる. 検査ソフトウェアは，コンピュータ内にインストールされたソフトウェアを検出し，検出したソフトウェアに固有な情報を検索鍵としてデータベースに送信し，脆弱性に関する情報を求める. このような汎用的なシステムを構築するにあたって問題となるのは, 検索鍵として, 何を利用するかである. 検索鍵は, 検査対象のソフトウェアから取得可能である必要があり，なおかつ，データベースにおいて脆弱性に関する情報を正しく検索可能にする情報でなければならない. そこで, 検索鍵として, ソフトウェア名とバージョン情報を利用することを考えた場合に, 検査対象のソフトウェアからどの程度取得可能か, 脆弱性の判定にどの程度利用できるか, 定量的に評価した結果を示す.

A study on detecting software with vulnerability

Daisuke Kikuchi[†1]　Masato Terada[†2,†3]　Yuji Chiba[†3]　Kenichi Yada[†3]　Norihisa Doi[†1]

[†1] Graduate School of Science and Engineering, Chuo University.
1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-88551, Japan

[†2] Faculty School of Sciencd and Technology, Keio University.
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan

[†3] Research and Development Initiative Chuo University.
1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-88551, Japan

E-mail: †1 dkikuchi@doi-lab.ise.chuo-u.ac.jp

Abstract　This paper shows a study on a system that detects software with vulnerability. The vulnerability detection system has two components: (1) a database that contains information on various vulnerabilities for various software, and (2) a detection software that checks for information on software installed on a target machine and uses that information to search the database for possible vulnerabilities. This information, i.e., search key, is important as it will affect the search for vulnerabilities. The search key should be information obtainable from the target software, and include the target software's status, such as version number and applied patches. We have conducted an experiment using the target software name and its version number as the search key. We analysed the results based on whether or not the version number information is automatically obtainable from the software and the correctness of the search result.

1　はじめに

近年，コンピュータへの不正アクセスやコンピュータウィルスによる被害が頻発しており，対策が求められている．不正アクセスやコンピュータウィルスの被害を受ける原因のひとつには, バグなどに起因するソフトウェアの脆弱性がある．ユーザが, ソフトウェアの脆弱性によって引き起こされる問題に対して, コンピュータシステムのセキュアな環境を維持するためには, ソフトウェアの脆弱性に対して, 適切な対処を取らなければならない.

ソフトウェアの脆弱性への対処は, 一般的につぎの手順でおこなわれる.

1. コンピュータ内にインストールされたソフトウェアについての情報を収集する.
2. 脆弱性を持つソフトウェアがある場合には, 収集した情報をもとにパッチの適用やサービスの停止などを行う.

ソフトウェアの脆弱性に関する情報は, 各ソフトウェアベンダの Web サイトや CERT/CC [1] ,

Bugtraq-jp [2] などで提供されている．ユーザは，これらの膨大な情報が蓄積された Web サイトから，検査対象となるソフトウェアに関する情報を抽出する．ユーザが脆弱性を持つソフトウェアに関する情報に基づいて，対処する具体的な手段には，次の 2 つがある．

手動で行う方法：ユーザ自身が，コンピュータ内のソフトウェアについて，バージョン情報や，パッチの適用状況に関する情報を収集する．そして，収集した情報をソフトウェアを特定するための情報 (以降，検索鍵とする) として利用することによって，ソフトウェアベンダの Web サイトなどから，脆弱性を持つソフトウェアの対策方法を調べ対処する．
手動で行う方法には，つぎの課題がある．

- ユーザが，コンピュータ内にどのようなソフトウェアがインストールされているのか，把握しなければならない．把握する情報は，ソフトウェアの名称だけでなく，ソフトウェアのバージョン情報，適用したパッチなど，多岐に渡る．
- ソフトウェアの脆弱性に関する情報には，専門的な知識が必要とされる場合がある．

このことから，手動で行う方法は，ユーザにかかる負担が大きい．

ツールを利用する方法：一部のソフトウェアベンダは，手動による脆弱性検査方法の負担を減少させるために，コンピュータ内のソフトウェアに関する情報を自動的に収集し，ソフトウェアへの脆弱性への対処を支援するツールを提供している．
脆弱性を持つソフトウェアを検出するツールは，脆弱性を持つソフトウェアをつぎの手順で検出する．ただし，ツールごとに多少の差異がある．

1. インストールされているソフトウェアに関する情報を取得する．
2. 検索鍵として用いるソフトウェアから取得した情報と，ツールを提供するベンダが持つ脆弱性に関する情報を比較することによって，脆弱性を持つソフトウェアの検出を行う．
3. 脆弱性検査の結果をユーザに示す．ツールによっては，必要となる修正ファイルを選別しダウンロードを行う．

ツールを利用する方法には，つぎの課題がある．

- ツールごとに，脆弱性検査の対象となるソフトウェアが限定される．なぜなら，脆弱性を持つソフトウェアを識別ときに検索鍵として利用する情報や，脆弱性検出の方法が，ベンダが提供するツールによって異なるからである．

この課題を解決するために，我々は特定のベンダに依存しないで，脆弱性を持つソフトウェアを検出するシステム (図 1) の実装を試みている．我々が，実装を試みているシステムの構成要素を，つぎに示す．

- 脆弱性データベース
  - ベンダに依存せず，さまざまなソフトウェアを対象として，ソフトウェアの脆弱性に関する情報を蓄積，検索できるデータベース．
- 検査ソフトウェア
  - 脆弱性検査の対象となるコンピュータにインストールし，脆弱性を持つソフトウェアの検出に利用するソフトウェア．コンピュータ内にインストールされたソフトウェアから検索鍵を収集し，収集した情報をもちいて，脆弱性データベースの検索をおこない，脆弱性を持つソフトウェアの有無の判定を行う．

システムの実装を試みるにあたって問題となることの 1 つは，検索鍵に何を利用するかということである．この検索鍵は，つぎの性質をもつ必要がある．

- 検査対象のソフトウェアから抽出できる．
  - 本システムの目標は，ソフトウェアを作成したベンダに依存しないで，脆弱性検査を実施可能にすることである．そのため，汎用的にソフトウェアから抽出可能な情報を利用する必要がある．
- 脆弱性データベースにおいて，脆弱性の有無を正しく判定できる．
  - たとえば，脆弱性データベースに Outlook Express 6.0 というソフトウェアに関する脆弱性情報があるとき，Outlook Express という情報だけでは脆弱性の有無を正確に判定できない．正確に判定するためには，バージョン情報なども必要になる．

検索鍵の候補のひとつとして，ソフトウェア名とバージョン情報が考えられる．その理由としては，つぎのことが挙げられる．

- ソフトウェア名とバージョン情報は，ソフトウェアの作成時に多くのソフトウェアにつけられている．
- ソフトウェアの脆弱性に関する情報がソフトウェア名とバージョン情報によって区別されていることが多い．

しかし，ソフトウェア名とバージョン情報が，ソフトウェアの作成段階でつけられているとしても，ソフトウェア自体から情報を取得できないことなどもあ

り得る．そのため, 検索鍵として, どの程度の妥当性があるかについて評価してみる必要がある．

そこで, 本稿では, 脆弱性を持つソフトウェアを検出するときの検索鍵として, ソフトウェアの名称とバージョン情報の組を利用することが妥当か否か, 評価した結果を示す．具体的な評価対象は次の 2 点である．

1. クライアント側において, どの程度の数のソフトウェアから, ソフトウェアの名称とバージョン情報を取得できるか
2. サーバ側において, ソフトウェアの名称とバージョン情報のペアから正しく脆弱性の有無を判定できるか

本稿の構成は次のとおりである．2 章で既存のツールが, 利用している検索鍵について述べ, 3 章でソフトウェア名とバージョン情報の妥当性を評価するために, 実装を試みたシステムについて述べる．4 章でソフトウェア名とバージョン情報の妥当性を, 実装したシステムの結果をもとに評価する．5 章でまとめを行う．

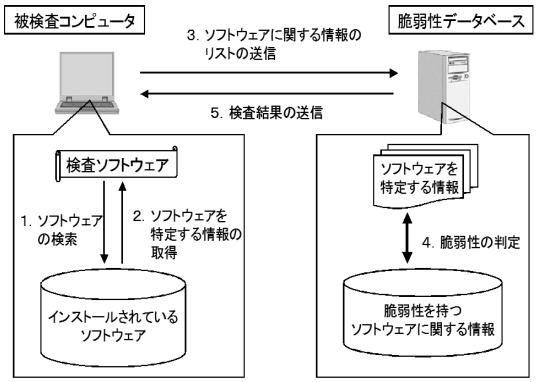

図 1　本システムの構成

2　既存のソフトウェアのツールが利用する検索鍵

代表的な “ソフトウェア更新ツール” を, 次に述べるが, ツールが利用する検索鍵は統一されていない．

■Windows Update

Windows Update [3] とは, Microsoft 社が提供するソフトウェアと一部のドライバの状態をチェックし, ソフトウェアを最新の状態にするのに必要なファイルのダウンロードや更新を支援するアプリケーションである．

Windows Update では検索鍵として, レジストリから パッチの適用情報などを取得している．

■RedHat Network

RedHat Network [4] とは,RedHat 社が提供する RedHat Linux システムの管理を支援するツールである．

RedHat Network では, RPM (RedHat Package Manager) を利用している．ユーザは, この RPM を利用することによって, ソフトウェアパッケージの構築, インストール, 照会, 検証, 更新, およびアンインストールすることができる．

RPM が検索鍵として, ソフトウェアの管理に利用するために持つ情報の例を 図 2 に示す．

```
-bash-2.05$ rpm -qi zsh
Name        : zsh                          Relocations: (not relocateable)
Version     : 4.0.6                             Vendor: Red Hat, Inc.
Release     : 5                            Build Date: 2003年02月10日 23時12分15秒
Install Date: 2003年08月08日 00時11分45秒      Build Host: stripples.devel.redhat.com
Group       : システム環境/シェル           Source RPM: zsh-4.0.6-5.src.rpm
Size        : 2710880                          License: BSD
Signature   : DSA/SHA1, 2003年02月24日 14時41分02秒, Key ID 219180cddb42a60e
Packager    : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
URL         : http://www.zsh.org/
Summary     : ksh 類似の改良型シェル
Description :
zsh シェルは対話的なログインシェルとして、またシェルスクリプトコマンド
プロセッサとして使用できるコマンドインタープリタです。zsh は ksh シェル
(Korn シェル) に似ていますがさまざまな拡張が行われています。zsh では
コマンドライン編集、組込みスペル訂正機能、カスタマイズ可能なコマンド
補完機構、シェル関数 (自動ロード機能付き)、履歴機構などがサポートされて
います。
```

図 2　RPM パッケージが持つ情報

■Patch Manager　Patch Manager [5] は Sun Microsystems 社の OS である Solaris 用のパッチ管理ツール群である．ユーザは, Patch Manager を用いることによって, コンピュータに必要なパッチの解析やダウンロードをおこなえる．

Patch Manager が検索鍵として, ソフトウェアの管理に利用する情報の例を, 図 3 に示す．

```
bash-2.05# pkginfo -l SUNWzsh
   PKGINST:  SUNWzsh
      NAME:  Z shell (zsh)
  CATEGORY:  system
      ARCH:  sparc
   VERSION:  11.9.0,REV=2002.03.02.00.35
   BASEDIR:  /
    VENDOR:  Sun Microsystems, Inc.
      DESC:  Z shell (zsh)
    PSTAMP:  sfw8120020302004057
  INSTDATE:  May 10 2003 03:37
   HOTLINE:  Please contact your local service provider
    STATUS:  completely installed
     FILES:        3 installed pathnames
                   2 shared pathnames
                   2 directories
                   1 executables
                 928 blocks used (approx)
```

図 3　Solaris のパッケージが持つ情報

3　ソフトウェア名とバージョン情報をもちいた脆弱性ソフトウェア検査システムの実装

ソフトウェア名とバージョン情報を利用することによって, 脆弱性を持つソフトウェアを検出するシ

ステムを実装した．実装したシステムにおいて，つぎの 2 点について述べる．

- 検査ソフトウェアの動作手順
  - 検査ソフトウェアは，Windows と Unix をサポートする．個々の OS において，ソフトウェア名とバージョン番号の取得方法は異なる．それぞれの OS におけるソフトウェア名とバージョン情報の取得手順を次に示す．
- 脆弱性データベースの動作手順
  - ソフトウェアから取得した情報をもとに，どのように脆弱性データベースで脆弱性の判定を行うかについて述べる．

3.1 検査ソフトウェアの動作手順

3.1.1 Windows におけるソフトウェア名とバージョン情報の取得

検査ソフトウェアが，Windows が稼動しているコンピュータ内にインストールされたソフトウェア群を抽出し，ソフトウェアの名称とバージョン情報を取得する手順を次に示す．

1. ソフトウェアが，デフォルトでインストールされるフォルダを Win32 API を利用して取得する．
2. 取得したフォルダ内を検索し，拡張子が EXE 形式であるソフトウェアの絶対パス名を調べる．
3. 調べた絶対パス名から Win32 API をもちいて，ソフトウェアに関する情報の取得を行う．
4. 取得した情報から，ソフトウェア名，バージョン情報を抜き出す．

3.1.2 Unix におけるソフトウェア名とバージョン情報の取得

検査ソフトウェアが，Unix が稼働しているコンピュータ内にインストールされたソフトウェア群を検出し，ソフトウェアの名称とバージョン情報を取得する手順を次に示す．

1. 環境変数 PATH から，ソフトウェアが保存されているディレクトリを取得する．なお，環境変数とは，システムの情報を記録している文字列変数であり，環境変数 PATH とは，ソフトウェアが保存されているディレクトリを記録した文字列変数である．
2. 取得したディレクトリから，そのディレクトリ内にあるファイルの一覧を取得する．
3. 取得したファイルのうち，ファイル末尾に実行ファイル記号 (*) がついたファイル (以降，コマンド) を取得する．
4. 取得したコマンドから，ソフトウェアの名称とバージョン情報を取得する．
   - ソフトウェアの名称としては，取得したコマンドのファイル名を利用する．
   - バージョン情報は，取得したコマンドに，コマンドオプション `-v` あるいは `--version` をつけて実行することによって取得する．

■バージョン情報の抜き出し

コマンドオプション `-v` あるいは `--version` をつけて，実行したときの出力結果は，そのソフトウェアごとに異なる (図 4)．

```
$ asetroot -v
asetroot version 1.8.8

$ netscape -v
Mozilla/5.0 (X11; U; SunOS sun4u; en-US; rv:1.0.1) Gecko/20020921
Netscape/7.0, build 2002092117

$ gphoto2 --version
gPhoto (Ver2.0) - Cross-platform ディジタルカメラ・ライブラリ。
Copyright (C) 2000-2002 Scott Fritzinger
Library GNU 公有使用許諾契約書 (LGPL) 下の ライセンス了承済み

$ Magick++-config --version
5.4.7
```

図 4　コマンドオプション付与時の出力結果

そのため，コマンドオプションによって得た出力結果から，ソフトウェア名とバージョン情報を抜き出す必要がある．そこで，コマンドオプションの出力結果に特定のパターンが存在しないか調べた．その結果，コマンドオプションの出力結果には，バージョン情報が次のパターンで出力することが判った．

1. ソフトウェア名 “区切り文字” “バージョン識別子” “区切り文字”“バージョン情報候補”
2. ソフトウェア名 “区切り文字” “バージョン情報候補”
3. “バージョン識別子” “区切り文字” “バージョン情報候補”
4. “バージョン情報候補”

このなかで，“区切り文字” とはスペースやカンマなどの記号文字を表す．“バージョン情報候補” とは，数字もしくはドットを先頭文字とし，数字とドットを必ず含み，“区切り文字” で区切られた文字列を表す．“バージョン識別子” は，“version” や “Ver” などの，バージョン情報の前につく可能性が高い文字列を表している．

“バージョン情報候補” がそのソフトウェアのバージョン情報である可能性は，ソフトウェア名や “バージョン識別子” が “バージョン情報候補” の前にあるほど，高いと推測できる．

そこで，本実装では，前述のパターン分類において付与した番号が小さいほど，その “バージョン情報候補” が，ソフトウェアのバージョン情報である可能性が高いとして，バージョン情報の抜き出しを行った．

表 1　ソフトウェア名とバージョン情報が取得できたソフトウェア数

	必要な検索鍵が取得できたソフトウェアの数	インストールされていたソフトウェアの数
Windows コンピュータ 1	275 (約 80.1%)	343
Windows コンピュータ 2	184 (約 67.1%)	274
Windows コンピュータ 3	191 (約 60.9%)	310
Solaris コンピュータ 1	466 (約 36.1%)	1290
RedHat Linux コンピュータ 1	1302 (約 36.3%)	3583
RedHat Linux コンピュータ 2	1144 (約 34.8%)	3586

3.2 脆弱性データベースの動作手順

脆弱性データベースでは, 脆弱性を持つソフトウェアの検出を, つぎの手順でおこなった.

1. 検査ソフトウェアから, ソフトウェア名とバージョン情報のリストを受け取る.
2. 受け取ったリストからソフトウェア名とバージョン情報の組を 1 組ずつ取り出し, 脆弱性データベースに登録された情報と比較を行う.
3. その結果を検査ソフトウェアに送信する.

4 評価

ソフトウェア名とバージョン情報を利用して, 実装をおこなったシステムについて, 評価を行う.

4.1 ソフトウェア名とバージョン情報の取得

コンピュータ内にインストールされたソフトウェアから，どれだけソフトウェア名とバージョン情報の組を取得できるか評価した結果を表 1 に示す．表 1 にあるように，検査対象のコンピュータにインストールされている OS は，3 台が Windows XP，2 台が Red Linux9.0，1 台が Solaris9.0 である．これらのコンピュータは大学において実際に研究用途に使用されているものであり，インストールされているアプリケーションソフトウェアはまちまちである．なお, この表 1 のうち, “必要な検索鍵が取得できたソフトウェア” とはソフトウェア名とバージョン情報が取得できたソフトウェアを示す.

4.1.1 Windows におけるソフトウェア名とバージョン情報の取得

表 1 の Windows コンピュータの結果より, Win32 API を利用することで, 平均約 70% のソフトウェアから, ソフトウェア名とバージョン情報が取得できたことが判る.

ソフトウェア名かバージョン情報のどちらか, もしくは, 両方が取得できないソフトウェアには, Windows 以外の OS から移植されたソフトウェアや個人作成のソフトウェアが多いという傾向にあった．このことから, ソフトウェア名かバージョン情報が取得できないソフトウェアは, Windows で定められているフォーマットに従って作成されていない, と考えられる.

4.1.2 Unix におけるソフトウェア名とバージョン情報の取得

表 1 の Solaris コンピュータ, RedHat Linux コンピュータの結果より, 一部のコマンドオプションを付与したときの実行結果を利用することで, 平均約 30% のソフトウェアから, ソフトウェア名とバージョン情報の取得が取得できたことが判る.

ソフトウェア名とバージョン情報が取得できたソフトウェアは, 特定のソフトウェアベンダに限らず取得することができた．ソフトウェア名とバージョン番号が取得できたソフトウェアにばらつきがあったのは, ソフトウェアに関する情報を出力するオプションが必須機能でなかったためと考えられる.

4.2 脆弱性データベースにおける脆弱性の判定

ソフトウェア名とバージョン情報から, 脆弱性を持つソフトウェアを検出できるかを調査した結果を表 3 に示す．なお, 脆弱性データベースには, 表 2 の情報を用意した．表 3 のうち, “検査結果” とは実装したシステムを利用した場合の判定結果を示している．また, “脆弱性判定の正確さ” は, 実装したシステムの判断結果が正確であったかどうかを示している.

表 3 より, 既にパッチが適応されたソフトウェアについては, 脆弱性の判定が正しくおこないことが判った.

5 まとめ

5.1 結論

本稿では, ソフトウェア名とバージョン情報が, 脆弱性を持つソフトウェアを検出するときの検索鍵として利用できるか評価をおこなった．具体的には, つぎの 2 点について, 評価をおこなった.

- どの程度の数のソフトウェアから, ソフトウェア名とバージョン情報を取得できるか
- ソフトウェア名とバージョン情報から, 脆弱性を持つソフトウェアの有無を正しく判定できるか

表 2　評価にもちいた脆弱性データベースが持つ情報

ソフトウェア名	脆弱性を持つバージョン情報	脆弱性情報
Microsoft Internet Explorer	6.0	Microsoft Internet Explorer に複数の脆弱性 (828750) (MS03-040)
Symantec Norton AntiVirus	2002	Symantec Norton AntiVirus Device Driver Memory
	2003	
Sun Solaris	9.0.0	Sun Solaris SAdmin クライアントのクレデンシャルの
	8.0.0	脆弱性により, 管理者権限でリモートからアクセスされる.
Opera	7.0 win32	Opera にクロスドメインスクリプティングの脆弱性
unzip	5.50	unzip にセキュリティホール
XFree	4.30	XFree86 パッケージにセキュリティホール
Yahoo! Messenger	5.6	Yahoo! Messenger File Transfer Buffer Overrun Vulnerability

表 3　ソフトウェア名とバージョン情報を利用した脆弱性検査システムの実装結果

検査ソフトウェア	ソフトウェアのバージョン番号	検査結果	脆弱性判定の正確さ
Yahoo! Messenger	5.6	脆弱性あり	○
Opera	7.22	脆弱性なし	○
unzip	5.50	脆弱性あり	○
unzip	5.32	脆弱性なし	○
Internet Explorer	6.0.0	脆弱性あり	× (パッチで対応済み)

評価の結果, ソフトウェア名とバージョン情報は, Windows では約 7 割, Unix では約 3 割の数のソフトウェアから取得できることが判った.

脆弱性を持つソフトウェアの有無の判定は, パッチが既に適用されたソフトウェアについては正しく判定できないことが判った.

5.2　今後の課題

今後の課題としては, つぎのことがある.

- レジストリ情報などから, パッチの適用状態といったソフトウェアに関する情報を取得し, ソフトウェア名やバージョン情報と組み合わせることによって, 脆弱性検査の精度を改善できる.
- 他のツールが利用する情報 (RPM が持つ情報など) を組み合わせることで, 検出対象とするソフトウェアの幅を広くすることが考えられる.

参考文献

[1] CERT/CC : http://www.cert.org/

[2] Bugtraq-jp : http://www.securityfocus.com/archive/79

[3] Windows Update : http://windowsupdate.microsoft.com

[4] Red Hat Network : https://rhn.redhat.com

[5] Patch Manager : http://sunsolve.sun.com /pub-cgi/show.pl?target=patchpage

[6] 寺田真敏, 土居範久: RDF Site Summary を用いたセキュリティ情報流通に関する検討, 情報処理学会研究報告, pp273-278, 2003.

[7] 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, pp.275-280, 2002.

[8] 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.195-196, 2003.

[9] 土居範久, 佐々木良一, 内田勝也, 岡本栄治, 菊池浩明, 寺田真敏, 村山優子 : 情報セキュリティ事典, 共立出版, 2003.