Vol.2010-CSEC-51 No.4
2010/12/10

P2P ネットワークを用いた 脆弱性修正支援システムの提案

志田明生，岡本剛
神奈川工科大学大学院 工学研究科 情報工学専攻
〒243-0292 神奈川県厚木市下荻野 1030

概要：PC の一般ユーザは，コンピュータの脆弱性に対する認識が低く，ソフトウェアの脆弱性を修正していないことが多い．そこで脆弱性の修正を促すため，ユーザインタフェースにより視覚的に脆弱性を確認できる脆弱性修正支援ツールが提案されている．しかし，これらのツールはクライアント-サーバ方式により構成されているためアクセス集中時や通信障害時に機能が停止する可能性がある．そこで，本稿では上記の問題を解決するため，P2P ネットワークの可用性の高さを取り込んだシステムを提案した．提案システムでは，ファイルの改ざんやサーバのなりすましの対策として電子署名とサーバ認証を導入した．また，ユーザインタフェースの利便性を改善し，簡単に脆弱性を修正できるようにした．

A Vulnerability Remediation Support System using a Peer-to-Peer Network

Haruki Shida, Takeshi Okamoto
Department of Information Network and Communication,
Kanagawa Institute of Technology.
1030, Shimo-ogino, Atsugi, Kanagawa 243-0292, Japan

Abstract Most users have little awareness of the vulnerability of their computer and tend not to remedy the vulnerability. To address this, a vulnerability remediation support tool using a graphical user interface was proposed, Unfortunately, these tools have the drawback that in the server-client model the remediation server may become overloaded by high volume of traffic. Therefore, in this paper, we propose a vulnerability remediation support system using peer-to-peer network. In the network, all shared files are e-signed and the remediation servers are protected by a digital certificate. In addition, we simplified the graphical user interface making it easier to use.

1. はじめに

ソフトウェアが持つ脆弱性の多くは，ソフトウェアベンダーが用意した修正プログラムにより修正が可能である．しかし，ユーザの多くは特にコンピュータの初心者は脆弱性への認識が低い．コンピュータに脆弱性があることや，脆弱性の修正方法を知らないため，脆弱性は修正されないことが多い．そこで，ユーザに脆弱性があることを提示するため，脆弱性情報の提示ツールが作成された[1] [2] [3] [4]．脆弱性情報の提示ツールは，クライアント-サーバ方式により構成され，ユーザが使用しているソフトウェアが持つ脆弱性を提示する．また，クライアントのソフトウェアに脆弱性がある場合，ソフトウェアの開発元など修正方法が掲載されたウェブサイトへのリンクを提示する．このツールにより，ユーザはどのソフトウェアに脆弱性があるかを知ることができ，脆弱性を修正しやすい．しかし，これらのツールは，クライアント-サーバ方式により構成されているためアクセス集中時や通信障害時にシステムのサーバが停止することが考えられる．そこで本稿では， P2P ネットワークの特徴を活かしシステムの可用性を向上した脆弱性修正支援システムを提案する．また，インターフェースの利便性を改善し 2 つの操作で脆弱性を修正できるようにする．同時に，P2P ネットワーク上でのファイルの改ざんやサーバのなりすましの対策として電子署名やサーバ認証を導入する．本稿では，提案システムの仕様やアルゴリズムについて報告する．

2. 脆弱性情報

脆弱性の情報を掲載しているサイトとして，US-CERT[5]などが世界中に存在する．これらのサイトは，脆弱性の情報を該当するソフトウェア名やバージョン値などにより分類し，脆弱性の危険度やその概要をまとめて掲載している．また，サイトにはそれぞれ情報のフォーマットが決まっているため，OVAL[6]などの独自のプロトコルを用いて脆弱性の情報を取得できる．

日本には，JVN（Japan Vulnerbility Notes）[7]が存在し，国内で使用されているソフトウェアなどの脆弱性情報とその対策情報を提供する．JVN には，JVN iPedia という国内で利用されているソフトウェア等の製品の脆弱性対策情報を中心に収集している脆弱性対策情報データベースがある． JVN iPedia は，日本語で脆弱性の情報を掲載していることから，日本人が容易に脆弱性情報を理解できるようになっている．そこで，本システムで用いる脆弱性の詳細な情報や脆弱性対策情報には JVN iPedia の情報を利用する．JVN iPedia では，脆弱性対策情報に識別番号を割り当てて管理している．本システムでは，脆弱性対策情報が掲載された HTML を読み込み，タグを目安に以下の項目を抽出してデータベースでの処理を行いやすくするため csv 形式に変換した．

- JVN iPedia 識別番号
- 脆弱性の概要

- 深刻度
- 影響を受けるシステム
- ベンダ情報

影響を受けるシステムには，脆弱性が確認されているソフトウェア名とバージョン値が含まれる．また，ベンダ情報にはソフトウェアベンダーのウェブサイトへのリンク情報が含まれる．このウェブサイトには，脆弱性の対策情報や修正ソフトウェアの取得方法などが掲載されている．以下では，JVN iPedia から取得する情報をまとめて“脆弱性情報”とよぶ．

3. 脆弱性情報の提示ツール

脆弱性の提示ツールは，いくつか存在しいずれも同じような構成や処理をおこなっている．そこで，既存の脆弱性情報の提示ツールについての概要と問題点を述べる．

3.1 概要

脆弱性情報の提示ツールの目的は，ユーザが使用しているソフトウェアの脆弱性とベンダ情報を，ユーザにわかりやすく提示することである．

脆弱性情報の提示ツールは，ユーザインタフェースを用いてユーザが使用しているソフトウェアの脆弱性情報を提示している．また，ベンダ情報も提示するため，ユーザは脆弱性の情報と修正方法をわかりやすく知ることができる．

3.2 処理の流れ

脆弱性情報の提示ツールの処理の流れを以下に示す．

(1) **ソフトウェア情報を取得する．**
コンピュータにインストールされているソフトウェア名とバージョン値の情報をソフトウェア情報として取得する．

(2) **脆弱性情報を取得する．**
JVN iPedia などの脆弱性対策情報データベースから脆弱性情報を取得する．

(3) **ソフトウェア情報を用いて脆弱性情報をフィルタリングする．**
ソフトウェア名とバージョン値の 2 つの項目によりフィルタリングを行う．
フィルタリングした情報は，ユーザのコンピュータにインストールされているソフトウェアが持つ脆弱性の情報となる．

(4) **ユーザインタフェースによりフィルタリングした情報を提示する．**
WEB アプリケーションやユーザインタフェースを用いて，フィルタリングした脆弱性情報をわかりやすくユーザに提示する．

以下，コンピュータにインストールされているソフトウェア名とバージョン値の情報をまとめて“ソフトウェア情報”とよぶ．

3.3 ツールの問題点

既存の脆弱性情報の提示ツールは，不要な脆弱性情報をフィルタリングし，ユーザに対し必要な情報をわかりやすく提示している．既存のツールは，クライアント-サーバ方式により構成されるため，サーバに処理の負荷が集中している．このため，ユーザの増加により，サーバにかかる負荷が大きくなるとサーバの処理の限界を超え，機能が停止することが考えられる．

4. P2P ネットワークを用いた脆弱性修正支援システム

4.1 目的

本システムの目的はシステムの可用性と実用性の向上である．具体的には，P2P ネットワークによるシステムのサーバや関連するウェブサイトにかかる負荷の軽減と，操作ウインドウのボタン操作により脆弱性の修正を容易にすることである．

本システムでは，事前に P2P ネットワーク上にソフトウェアベンダーやシステム管理サーバにより脆弱性の情報や修正プログラムをアップロードする．次に，ユーザが修正したい脆弱性を選択すると，システムは自動的に P2P ネットワークを介して修正プログラムを取得し，修正プログラムを実行するため脆弱性の修正が簡単に行える．

4.2 P2Pネットワーク

本システムは，ファイルの取得に P2P ネットワークを用いることにより，システムのサーバにかかる負荷の軽減を可能にしている．ファイルを共有する方式において，P2P 方式の通信はクライアント-サーバ方式と比べてシステムの可用性が高い．クライアント-サーバ方式では，ユーザの数の増加とサーバの負荷が比例する．このため，負荷がサーバの処理の限界を超えたときシステムが停止することが考えられる．しかし，P2P 方式では，基本的にサーバを使わないためユーザの数が増加してもシステムの可用性を維持できる[8][9].

本システムでは，システムの可容性の向上のため P2P ネットワーク方式の共有するファイル共有ソフトウェアを用いた．ファイル共有ソフトウェアは，P2P ネットワークの構成の仕方によりピュア P2P 方式とハイブリッド P2P 方式に分類される．

本システムでは，BitTorrent などと同じ方式であるハイブリッド P2P 方式によるファイル共有ソフトウェアを用いる．ハイブリッド P2P 方式は，クライアント-サーバ方式と P2P 方式の両方の性質をもつ通信方式である．まず，ハイブリッド P2P 方式によるファイル共有ソフトウェアを用いたファイルの取得手順を以下に示す．

(1) シーダーはファイルを所有しているピアとして，トラッカーサイトにファイルを所有していることを伝える．

(2) ファイルを求めるピアは，トラッカーサイトに接続してファイルのインデックス

情報を問い合わせる（インデックス情報には，シーダーのアドレスが含まれる）．
(3) トラッカーサイトは，ファイルを求めるピアにインデックス情報を伝える．
(4) ファイルを求めるピアは，インデックス情報をもとにシーダーに接続して，シーダーからファイルを取得する．
(5) ファイルを取得したピアは，新たなシーダーとして他のピアへファイルの共有を行う．
トラッカーサイトには，共有されている全てのファイルの情報があるため，システム管理者が共有されているファイルの情報を管理しやすく，著作権物の流出などを監視できる．このため，本システムではハイブリッド P2P 方式によるファイル共有ソフトウェアを用いた[10][11][12].

4.3 構成

本システムは，主にサーバとクライアントに分けられる．それぞれ以下のコンポーネントから構成される．本システムの構成を図 1に示す．

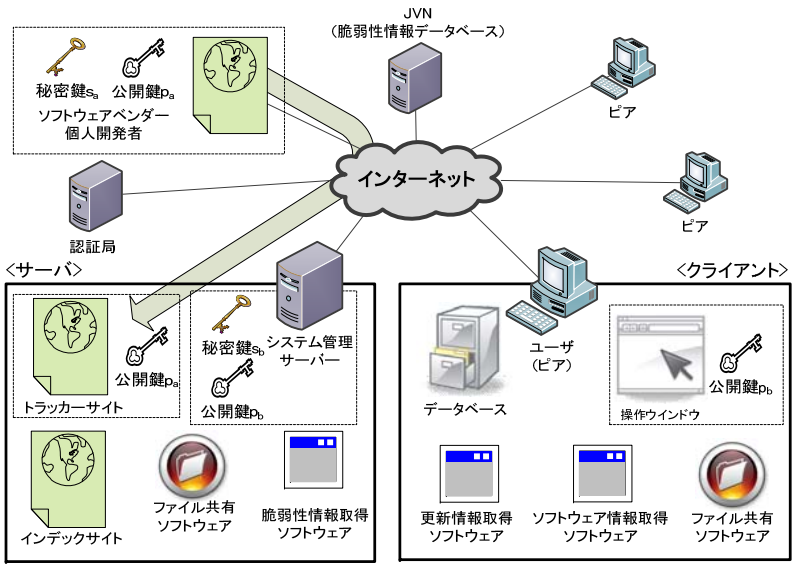

図 1　システムの構成図

4.3.1 サーバ

- 脆弱性情報取得ソフトウェア
  JVN iPedia に定期的に接続して更新があるかを確認する．更新がある場合，新しい脆弱性情報を取得する．
- P2P ネットワーク型ファイル共有ソフトウェア
  脆弱性情報や修正プログラムをピアと共有する．
- トラッカーサイト
  共有するファイルをアップロードするために必要なウェブサイトである．
- インデックスサイト
  脆弱性情報や修正プログラムのアップロード情報を記録するウェブサイトである．
- システム管理サーバの秘密鍵 s_b と公開鍵 p_b
  秘密鍵 s_b は，取得した脆弱性情報につける電子署名の署名鍵として用いる．公開鍵 p_b は，あらかじめクライアントの操作ウインドウに組み込んでおき，電子署名の検証鍵として用いる．

4.3.2 クライアント

- 更新情報取得ソフトウェア
  インデックスサイトに接続して脆弱性情報やアップロードされた修正プログラムの一覧情報の更新を確認する．更新がある場合，更新されたインデックス情報を取得する．
- ソフトウェア情報取得ソフトウェア
  ユーザのコンピュータにインストールされたソフトウェアの情報を取得する．
- P2P ネットワーク型ファイル共有ソフトウェア
  ファイル共有ソフトウェアを用いて，インデックスサイトの情報をもとに脆弱性情報や修正プログラムを取得する．
- データベース
  脆弱性情報とソフトウェア情報を格納し，ソフトウェア情報をもとに脆弱性情報をフィルタリングする．フィルタリングにより脆弱性情報からユーザが必要とする情報を取得する．
- 操作ウインドウ
  フィルタリングされた脆弱性情報をユーザに提示する．また，ユーザの操作により脆弱性の修正が行える．

4.3.3 JVN

JVN に掲載されている脆弱性の詳細な情報や脆弱性対策情報を本システムの脆弱性情報として用いる．

4.3.4 ソフトウェアベンダー

ソフトウェアベンダーが公開している脆弱性の修正プログラムをファイル共有ソフトウェアにより共有する．また，ソフトウェアベンダーは公開鍵 p_a と秘密鍵 s_a を所有し，秘密鍵 s_a は修正プログラムにつける電子署名の署名鍵として，公開鍵 p_a は電子署名の検証鍵として用いる．

4.4 処理の流れ

本システムのアルゴリズムを 3 段階に分けて述べる．本システムでは，脆弱性が公開されてからユーザにより脆弱性が修正されるまでの処理を，ソフトウェアベンダーと本システムのシステム管理サーバ，ユーザに分けて行う．これらの処理の流れを図 2に示す．

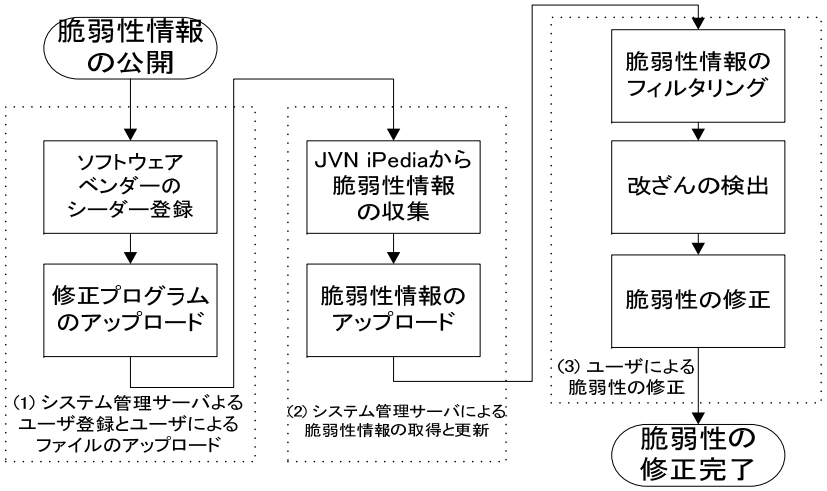

図 2　システムの処理の流れ

4.4.1 シーダーによるユーザ登録とファイルのアップロード

本システムでは，著作権物の共有などP2Pネットワークの悪用を防ぐため，ファイルのアップロードにはシーダーのユーザ登録を必要とする．このフローチャートを図 3に示す．本システムでは，システム管理サーバがソフトウェアベンダーをファイル共有ソフトウェアにおけるシーダーとして登録する．このユーザ登録時の登録者のなりすましを防止するため，PKIを用いて本人確認を行いユーザの登録をする．

また，登録者はアップロードファイルの改ざんの防止のため，共有するファイルには秘密鍵 s_a を用いて電子署名をおこない，復号化に用いる公開鍵 p_a をユーザの ID としてトラッカーサイトに登録する．次に，ソフトウェアベンダーはシーダーとして修正プログラムや更新プログラムの情報をトラッカーサイトにアップロードする．共有するファイルには，秘密鍵 s_a を用いた電子署名を必要とする．電子署名のないソフトウェアは，ユーザによる実行時にアラートを表示して廃棄する．また，システムの実用性を向上するため，ソフトウェアベンダーにより開発中のソフトウェアのアップデート情報もアップデート可能にする．アップデート情報のアップロードと同時に更新プログラムの情報をアップロードする．これにより，本システムを介してソフトウェアを利用中のユーザに更新を促すことができ，ユーザも簡単にアップデートできる．

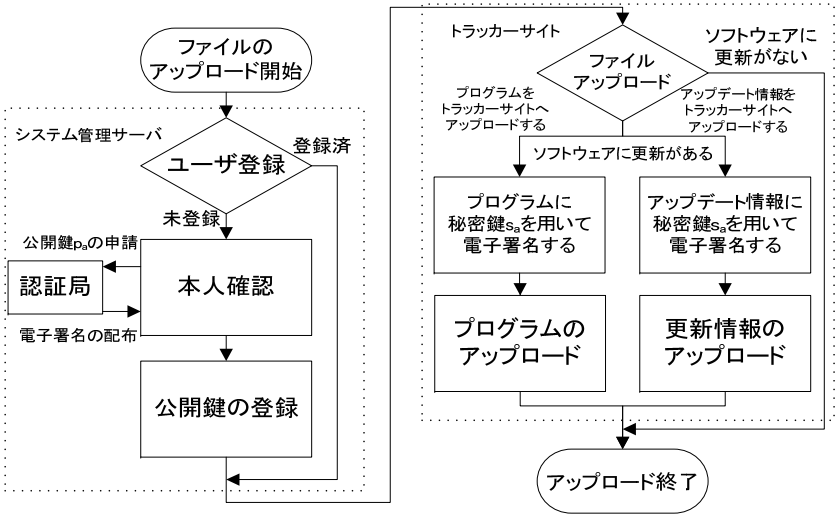

図 3　システム管理サーバとトラッカーサイトのフローチャート

4.4.2 サーバによる脆弱性情報の取得と更新

システム管理サーバは，ピアが共有する脆弱性情報の取得と配布を行う．このフローチャートを図 4に示す．

始めに，脆弱性情報取得ソフトウェアが JVN iPedia に接続し，脆弱性情報の更新があるかを確認する．更新がある場合，脆弱性情報が掲載された HTML を読み取りタグにより要素に分け 2 章で述べた脆弱性情報を csv 形式に変換する．また，この脆弱性情報のファイルにもシステム管理サーバの秘密鍵 s_b を用いて電子署名を付けて改ざんを防止する．脆弱性情報は，ファイル共有ソフトウェアを介してピアと共有する．

次に，インデックスサイトを更新する．インデックスサイトは，ファイル共有ソフトウェア上で共有されている全ての修正プログラムや，脆弱性情報をまとめた csv 形式のファイルの情報と，シーダーによりアップデートされた脆弱性情報やアップデート情報をまとめて掲載する．インデックスサイトに情報をまとめておくことにより，ユーザはインデックスサイトからファイル共有ソフトウェアにより取得可能なファイルの一覧を取得できる．

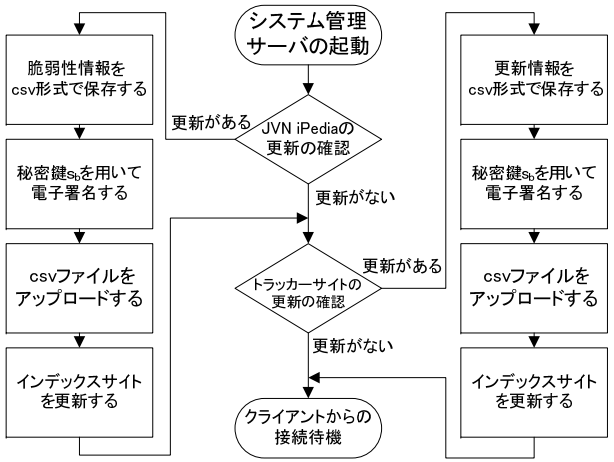

図 4　システム管理サーバのフローチャート

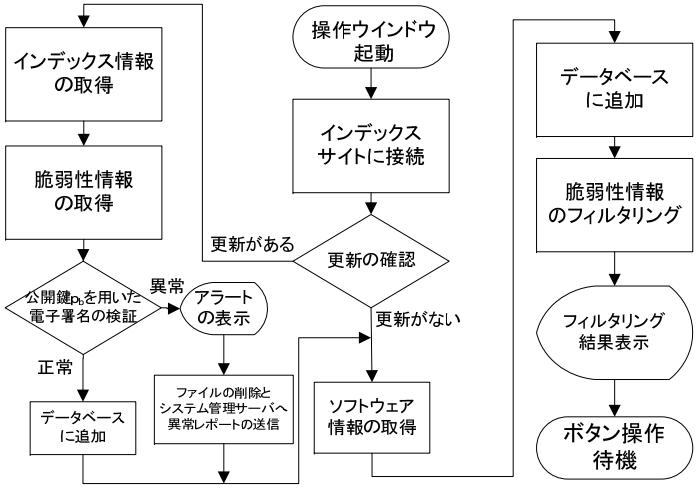

図 5　操作ウインドウ起動時のフローチャート

4.4.3 ユーザによる脆弱性の修正

操作ウインドウは，ユーザに脆弱性情報の提示や脆弱性の修正を促す．この処理の流れは,操作ウインドウの起動時と操作ウインドウの操作時に分けられる.このフローチャートを図 5に示す.

● 操作ウインドウの起動時

始めに,インデックスサイトに接続して共有ファイルの一覧に更新があるかを確認し，更新がある場合は更新情報の一覧を取得する．次に，インデックスサイトからファイル共有ソフトウェアを用いて脆弱性情報を取得する．脆弱性情報は，事前に操作ウインドウに組み込んであるシステム管理サーバの公開鍵p_bを用いて電子署名を検証することにより，脆弱性情報の改ざんの検出を行う．正当な脆弱性情報のファイルであれば，脆弱性情報としてデータベースに追加する．次に，ユーザのPCのソフトウェア情報を取得する．ソフトウェア情報は，レジストリのHKEY_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Uninstall にあるサブキーのDisplayNameキーとDisplayVersionキーの値を用いて作成する（レジストリに登録されていないアプリケーションは，本システムの対象外とする）.作成したソフトウェア情報は，脆弱性情報と同様にデータベースに追加する．最後に，インストールしているソフトウェアとデータベース内の脆弱性を持つソフトウェア情報が一致したソフトウェアのみについて，脆弱性情報を作成する．この脆弱性情報は操作ウインドウを介してユーザにリスト形式で提示される（図 6 (a)）.

● 操作ウインドウのボタン操作時

操作ウインドウには２つのボタンがある．１つは脆弱性修正ボタン（図 6 (b)）であり，もう１つは脆弱性情報更新ボタンである（図 6 (c)）．これらのフローチャートを図 7 に示す.

脆弱性修正ボタンは，ユーザが脆弱性を修正するためのボタンである.

脆弱性修正ボタンのクリック時には，インデックスサイトの情報を参照してユーザが選択した脆弱性の修正ソフトウェアが存在するかを確認する．存在する場合，ファイル共有ソフトウェアを用いて修正プログラムを取得する．次に，シーダーの公開鍵 p_a を用いて電子署名を検証することにより，修正プログラムの改ざんの検出を行う．正当なファイルである場合，プログラムを実行し脆弱性を修正する．また，修正プログラムが存在しない場合は，JVN から取得した脆弱性情報のうちベンダ情報をユーザに提示して脆弱性の修正を促す.

脆弱性情報更新ボタンは，ソフトウェアが持つ脆弱性情報を更新するためのボタンであり，ソフトウェアのバージョンアップや修正を行った後，脆弱性が修正されたかを確認できる．脆弱性情報更新ボタンのクリック時には，再度ソフトウェア情報を取得し，データベースのソフトウェア情報を更新する．次に，新しいソフトウェア情報をもとに脆弱性情報をフィルタリングして脆弱性情報をユーザに提示する.

Vol.2010-CSEC-51 No.4
2010/12/10

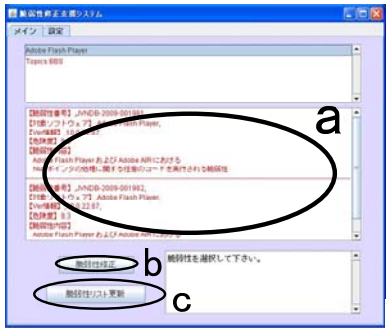

図 6　操作ウインドウのユーザインタフェース画面

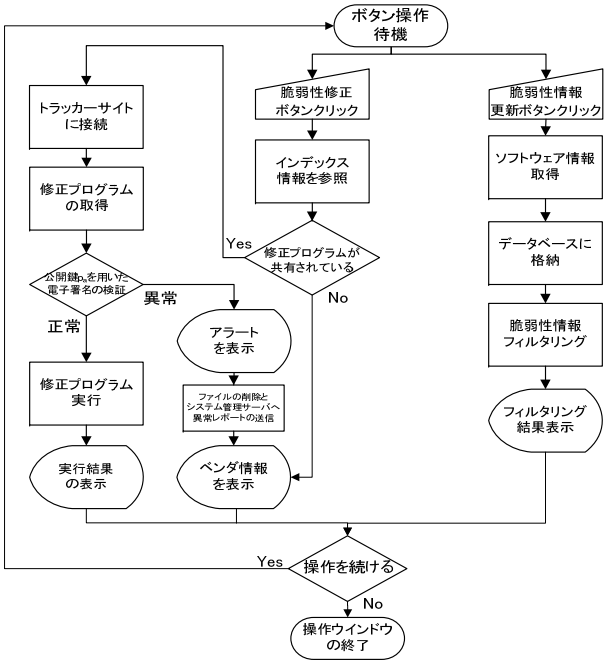

図 7　操作ウインドウ操作時のフローチャート

5. おわりに

本稿では，P2P ネットワークを用いた脆弱性の修正支援システムの提案を行った．提案システムでは，P2P ネットワークによりシステムの管理サーバにかかる負荷を軽減し，ユーザインタフェースの利便性の向上により簡単なボタン操作で脆弱性を修正できるようにした．さらに，ファイルの改ざんやサーバのなりすましの対策として電子署名やサーバ認証を導入した．今後は提案システムの実装をおこない，システムの実用性を評価する．

参考文献

1)　寺田真敏，杉山賢，山岸正，小林偉昭，土居範久: MyJVN を用いた脆弱性対策情報提供サービスの検討，情報処理学会 コンピュータセキュリティ研究報告, vlo.2009, no.20, pp283-288 (2009).
2)　藤堂洋介，朝倉康生，森井昌克: 既存脆弱性情報を利用したクライアント向け脆弱性検査システムの提案と評価，電子情報通信学会 信学技報, vlo.109, no.205, pp107-112 (2009).
3)　榊原裕之: 不正情報生成装置，不正情報生成方法，不正情報生成ソフトウェア，脆弱性検査装置，脆弱性検査方法および脆弱性検査ソフトウェア，日本国特許庁，特開 2007-259171 (2007).
4)　津田光夫: ソフトウェア更新要否判定方法，日本国特許庁，特開 2007-323349 (2007).
5)　UNITED STATES COMPUTER EMERGENCY READINESS TEAM: US-CERT, http://www.us-cert.gov/index.html, (2006).
6)　Homeland Security: OVAL, http://oval.mitre.org/, (2002).
7)　JPCERT/CC and IP A: Japan Vulnerbility Notes, http://jvn.jp/, (2003).
8)　金子勇: Winny の技術, p24, 株式会社アスキー, (2005).
9)　東森ひろこ，宇田隆哉: P2P セキュアファイル共有システムにおける共有機能の改善，情報処理学会 研究報告, vlo2006, no.129, pp51-56(2006).
10)　Allan Friedman, L Jean Camp: Peer-to-Peer security, The Internet Encyclopedia by Hossein Bidgoli, John Wiley & Sons, (2003).
11)　Jochem van Vroonhoven: Peer-to-Peer security, 4th Twente Student Conference on IT, Enschede, 30 January(2006).
12)　Stefan Saroiu, P. Krishna Gummadi, Steven D. Gribble: A Measurement Study of Peer-to-Peer File Sharing Systems, Proceedings of the SPIE/ACM Conference on Multimedia Computing and Networking (MMCN) 2002, San Jose, CA, January (2002).