Windows API の監視による 不正インストール検出手法の検討

040427180　三根 健司

急速なインターネットの普及によりコンピュータウイルスに感染し個人情報が流出するなどの事件が頻繁に起き，大きな社会問題となっている．ウイルスは多くの場合不正なプログラムをインストールした後，Windows API を介してレジストリを書き換える．そこで本稿では，この挙動に着目し，レジストリ関連の Windows API を監視することにより不正なインストールを検出する手法の検討を行った．本手法は監視プログラムがレジストリの状態を保存しておき，不正なインストールが生じた場合に復元する機能を持ち，ユーザに対してポップアップで危険を表示することにより，ユーザの意図に反していないかどうか判断を促す．また，正常なプログラムと不正なプログラムのリストをデータベースとして蓄積しておき，不正なインストールの再発を防止する．

Researches on Illegal Install Detection Method based on Windows API Monitoring

040427180　Kenji Mine

It is infected with the computer virus by the spread of the rapid Internet, the event of the outflow of individual information etc. happens frequently, and big social issues. After installing an illegal program in many cases, the virus rewrites the registry through Windows API. Then, the technique for detecting an illegal installation by paying attention to this behavior, and observing Windows API related to the registry was examined in this text. This technique preserves the state of the registry by the supervisor, and presses the judgment to be whether contradiction to the user's intention by having the function to restore when an illegal installation is caused, and displaying danger to the user by pop up. Moreover, the list of a normal program and an illegal program is accumulated as a data base, and an illegal installation is prevented from relapsing.

1. はじめに

急速なインターネットの普及によりコンピュータウイルス（以下，ウイルス）による被害の増加が大きな社会問題となっている．これらのウイルスを検出・駆除するためには，ウイルス対策ソフト（以下，アンチウイルス）の利用が一般的である．アンチウイルスはウイルスの特徴を収めたウイルス定義ファイルと，対象となるファイルを比較することでウイルスを検出している（パターンマッチング方式）．そのため，ウイルス定義ファイルに情報が存在しないウイルスは，アンチウイルスでは未知のものであり検出することは不可能である．しかし，近年では新種の未知ウイルスが 1 日に約 30 種類発生[1]しており，アンチウイルスメーカはウイルス定義ファイルの生成に平均 10 時間を費やしているのが現状である．また，ウイルス自体も難読化，複雑化してきており，ウイルス定義ファイルに含まれない未知のウイルスを検出するための研究が行われている[2]~[6]．しかし，完璧な検出手法というものは存在せず，未知のウイルスに感染するユーザをなくすことはできない．

そこで，ウイルスは多くの場合，不正なプログラムをインストールし，コンピュータ起動時に自動実行するためにレジストリを追加したり，変更することに着目し，Windows API の監視による不正インストール検出手法の検討を行った．本手法では監

視プログラムがレジストリ関連の Windows API を監視し，レジストリの追加や変更があった場合に正常な状態に復元する機能を持つ．また，レジストリの追加や変更を行ったプログラムのインストールをポップアップでユーザに通知し，インストールがユーザの意思に反しているかどうかの判断を促す．このとき，ユーザによって，プログラムは正常なプログラムと不正なプログラムに分けられ，リスト化してデータベースとして蓄積される．このようにして不正インストールの再発を防止する．

以下，2 章ではウイルスについて述べ，3 章では従来のウイルス検出技術とその課題について述べる．また，4 章では Windows API の監視による不正インストール検出手法の概要について述べる．5 章では実装の詳細について述べ，6 章でまとめる．

2. コンピュータウイルス

経済産業省の定義によれば，コンピュータウイルスとは第三者のプログラムやデータベースに対して意図的に何らかの被害を及ぼすように作られたプログラムであり，次の機能を 1 つ以上有するものである．

(1) 自己伝染機能

自らの機能によって他のプログラムに自らをコピーし又はシステム機能を利用して自らを他のシステムにコピーすることにより，他のシステムに伝染する機能.

(2) 潜伏機能

発病するための特定時刻，一定時間，処理回数等の条件を記憶させて，発病するまで症状を出さない機能.

(3) 発病機能

プログラム，データ等のファイルの破壊を行ったり，設計者の意図しない動作をする等の機能.

初期のコンピュータウイルスは，これら 3 つの機能をすべて有し，感染，潜伏，発病というサイクルを繰り返すものが主であった．しかし，現在では狭義のウイルス定義では含まれなかったワームやトロイの木馬といったものも含めた，不利益をもたらす不正プログラム全体をウイルスと呼んでいる．ワームとは自己増殖を繰り返しながら破壊活動を行なうプログラムで，トロイの木馬とは正体を偽ってコンピュータへ侵入し，データ消去やファイルの外部流出，他のコンピュータの攻撃などの破壊活動を行なうプログラムである．ただし，実際のウイルスのほとんどが Microsoft Windows OS（以下，Windows）を対象としていることを考慮して，本稿でも Windows 上で動作する実行ファイル形式の不正プログラムのみを対象とする．

3. 従来のウイルス検出技術とその課題

3.1 ウイルス検出技術

ウイルス検出技術にはパターンマッチング，ヒューリスティックスキャン，ビヘイビアブロッキングなどがある．

(1) パターンマッチング

あらかじめウイルスの特徴（パターン）を記述したファイル（ウイルス定義ファイル）をウイルス検出ソフト内に持っておき，この情報と検査対象ファイルを比較する手法である．

(2) ヒューリスティックスキャン

動作前のプログラムの内容をチェックし，システム領域や DLL の書き換えなど，通常のプログラムが実行しないようなウイルス特有の挙動をしていないか予測して検知する手法である．

(3) ビヘイビアブロッキング

プログラムが発行するシステムコールなどの動作を監視して，レジストリの内容の変更やディスクへの書き込みなどの動作とあらかじめ定義された「ウイルスらしいふるまい」と比較して悪質なプログラムかどうか判断する手法である．

3.2 ウイルス検出技術の課題

パターンマッチングは常にウイルス定義ファイルを最新の状態にしておかなければ新種のウイルスに対応できない．また，ウイルス定義ファイルを作成するために数時間を要するため，ウイルスの拡散が速い場合に更新が追いつかないという課題がある．

ヒューリスティックスキャン，ビヘイビアブロッキングは未知のウイルスを検出できるが，誤って正常なプログラムをウイルスと判断してしまう可能性がある．また，ウイルスによる不正な指令なのか正常なプログラムの指令なのかを判断するためのルールを定義することが難しいという課題がある．

4. 不正インストール検出手法の提案

4.1 提案手法の目的

従来のウイルス検出技術だけでは未知のウイルスの検出が困難である．また，ウイルスは多くの場合，不正なプログラムをインストールし，コンピュータ起動時に自動実行するためにレジストリを追加したり，変更したりする．このウイルスの挙動に着目し，不正なインストールを防止すれば，ウイルスによる感染を防ぐことができると考えた．

4.2 レジストリ

レジストリとはシステムやアプリケーションソフトの設定データが記録されているデータベースのことである．ウイルスは多くの場合，コンピュータ起動時に実行されるように自分自身をインストールする．Windows において，コンピュータ起動時にプログラムを実行する方法は，スタートアップのフォルダにプログラムのショートカットを作成する方法と，レジストリに直接登録する方法がある．多くのウイルスはユーザに気づかれにくくするため，不正なプログラムをレジストリに直接登録し，コンピュータ実行時に自分自身を実行させ，感染させる．本提案はレジストリに直接登録するプログラムを対象とする．

レジストリに直接登録するためには，Windows では API を用いる必要がある．API（Application Programming Interfaces）とは OS やシステムが提供する機能にアクセスするためのインターフェイスのことである（図 1）．したがって，レジストリに関連した Windows API を監視することにより不正なインストールを防ぐことができる．

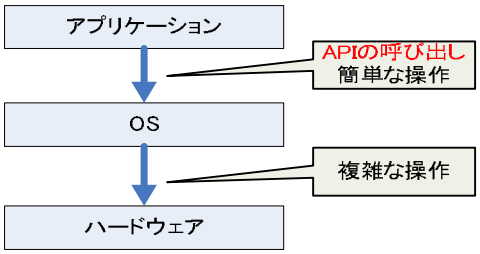

図 1 API が呼び出される部分

4.3 API フック

Windows API を監視する方法として API フックがある．API フックとは Windows API 関数を横取りすることである．

本提案では全プロセスの Windows API をフックすることにより，レジストリに関連した Windows API を監視することにより，不正なインストールを防止する．

4.4 提案手法の構成

提案手法は監視プログラムとデータベースで構成される（図 2）．それぞれの機能を次に示す．

(1) 監視プログラム

・Windows API の監視する．
・レジストリを比較してユーザに対してポップアップを表示する．
・レジストリの状態の保存と復元を行う．
・データベースの情報を参照する．

(2) データベース

信頼済プログラムと不正なプログラムのリストを蓄積する．

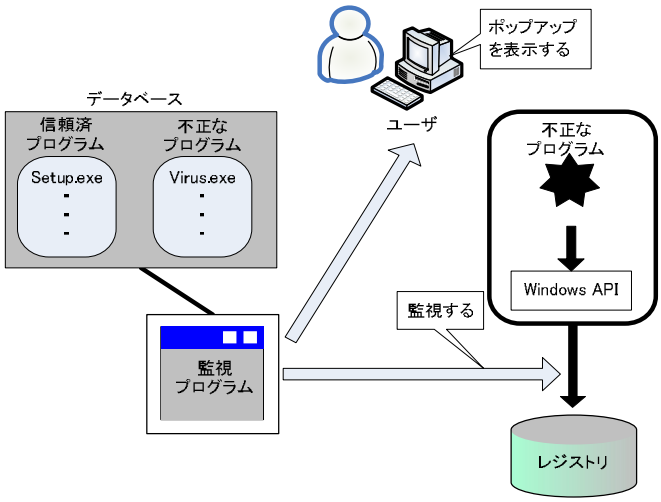

図 2　システムの構成

4.5　監視プログラムの動作

監視プログラムは現時点でのレジストリの状態を保存しておき，あるプログラムがWindows API を介してレジストリの変更を行った場合，ユーザに対してレジストリの変更を行ったプログラムをポップアップで通知する．ユーザはそのプログラムが行ったレジストリの変更を許可するか拒否するかを選択し，許可されればレジストリは変更された状態で新たに保存され，プログラムは正常なプログラムとしてデータベースに追加される．もしユーザが拒否した場合は，レジストリを以前に保存した状態に戻し，レジストリの変更を行ったプログラムは不正なプログラムとしてデータベースに追加される（図3）．

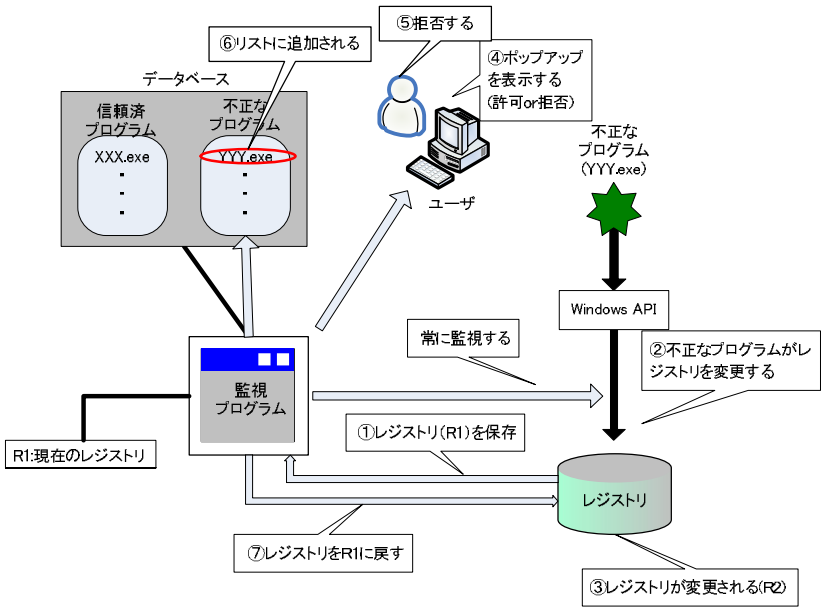

図 3 監視プログラムの動作（不正なプログラムの場合）

5. 実装

本提案はレジストリに関連したWindows API を監視することにより不正インストールを防止するものである．この API の監視には API フックを用いる[7]．API フックの方法には次のようなものがある．

(1) **DLL を仲介にする**

フック対象となる API をエクスポートしている DLL（以下，本物 DLL と呼ぶ）とまったく同じエクスポート関数を持つ DLL（以下，偽 DLL と呼ぶ）を生成する．偽 DLL を適当なプログラムに本物 DLL の代わりに呼ばせることで API フックを行う方法である．この偽 DLL は，仲介 DLL 又はラッパーDLL と呼ばれている．

(2) **インポートセクションを書き換える**

プログラム内のインポートセクションをスキャンし，フック対象となる API を見つける．インポートセクションにはそのモジュールが実行するために必要な DLL や, その DLL からインポートしている関数のアドレスが保存されている．そして見つけた API のアドレスをフック用関数のアドレスに書き換える．Windows の場合，API 呼び出しのジャンプ先をインポートセクションから探し出すので，インポートセクションに記述された各 API のアドレスを書き換えれば，プログラムを騙して別関数にジャンプさせることが可能となる．ただし GetProcAddress 関数を用いて API のアドレスを取得される場合があるので，GetProcAddress 関数も別途フックして，フック対象 API のアドレスが取得されそうになった場合にフック用関数のアドレスを返すように実装しておく必要がある．

(3) **フック対象となる API の先頭数バイトを JMP 命令に置き換える**

フック対象となる API の先頭数バイトを JMP 命令に置き換える．ジャンプ先としてフック用関数のアドレスを指定することにより，API 呼び出しをフックすることが可能となる．実装するにはアセンブラの知識が必要であり難易度が高い．ただ Microsoft Research の提供するライブラリ Detours を使えば比較的容易に実装することが可能である．

(4) **SSDT(System Service Descriptor Table)を書き換える**

SSDT というネイティブ API とそのアドレスを保持しているテーブルのようなものを書き換えることにより API フックを行う方法．アプローチは「インポートセクションを書き換える」方法と似ている．フック対象となるネイティブ API のアドレスをフック用関数のアドレスに書き換えてやることで，ネイティブ API が呼び出された時にフック用関数が呼ばれる．

本提案では，(2)のインポートセクションを書き換える方法を用いて全プロセスの API をフックする．

実装には参考文献[7]における API フックのサンプルプログラムを用いる．このサンプルプログラムは全プロセスの API をフックするものであり，メッセージボックス関数をフックする．このサンプルプログラムに変更を加え，フックの対象をレジストリに関連した API 関数にすることで API の監視を実現する．レジストリに関連した API はいくつか存在するが，具体的にどの関数をフックするかについては現在検討中である．また，このプログラムにレジストリの保存と復元機能とポップアップを表示する機能，データベースの作成機能を追加することにより本提案の実装を実現する．

6. まとめ

本稿では，未知ウイルスを検出する前の段階として Windows API の監視による不正インストール検出手法の検討を行った．監視プログラムが Windows API を監視することによって不正なインストールかどうかユーザに判断を促す．今後はプログラムの実装と動作検証を行う．

参考文献

1) シマンテックインターネットセキュリティ脅威レポート
（http://www.symantec.com/content/ja/jp/enterprise/white_papers/wp_istr08_2005.pdf）
2) 三宅崇之，白石善明，森井昌克：仮想サーバを使った未知ウイルス検知システムの提案，情報処理学会研究報告, Vol.2002, No.68, pp.45-52 (2002)
3) 松浦佐江子，加藤道子，小島量：未知のコンピュータ・ウイルス検出プログラムの開発，情報処理学会研究報告, Vol.2003, No.18, pp.89-94 (2003)
4) 市川幸宏，神薗雅紀，白石善明，森井昌克：ウイルス解析を目的としたメモリ上の不正コード検出システムの構築，電子情報通信学会技術研究報告, Vol.104, No.424, pp.57-62 (2004)
5) 伊沢亮一，市川幸宏，白石善明，毛利公美，森井昌克：静的/動的解析によるウイルス自動解析システム，暗号と情報セキュリティシンポジウム (2006)
6) 小池竜一，中谷直司，厚井祐司：未知コンピュータウイルスを駆除する USB フラッシュメモリの開発，情報処理学会論文誌, Vol48, No.4, pp.1595-1605 (2007)
7) Windows API Hooking Tutorial
（http://ruffnex.oc.to/kenji/text/api_hook/）

謝辞

本研究を遂行するにあたり，多大なるご指導，ご鞭撻を賜りました渡邊晃教授に心より感謝します．また有益なご助言，ご検討いただきました渡邊研究室の学部生，学院生の皆様に深く感謝いたします．