社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2007－DPS－130 (25)
2007－CSEC－ 36 (25)
2007／3／1

動的API検査方式によるキーロガー検知方式の提案（その2）

高見知寛† 鈴木功一† 馬場達也†† 前田秀介†† 松本隆明†† 西垣正勝†††
† 静岡大学大学院情報学研究科 †† NTTデータ技術開発本部
††† 静岡大学創造科学技術大学院

あらまし 我々は，キーボード入力を取得するというキーロガーの挙動に着目し，キーボード入力に用いられるAPIの使用を検出することでキーロガーの検知を行う方式を提案している．基礎実験により本方式が未知キーロガー検知に対して有効であることを示したが，本方式には自プロセスをユーザから隠すキーロガーを検知できないという課題を残していた．そこで本稿では，「自プロセスをユーザから隠す」という挙動を新たにキーロガーらしい挙動として追加することでキーロガーの検知精度の向上を図る．本方式の有効性を示すための実験を行い，本方式をWindows OS上に実装したときの検知率と誤検知率，オーバーヘッドについて評価する．

A keylogger detection using dynamic API inspection (part 2)

Tomohiro Takami†, Koichi Suzuki†, Tatsuya Baba††, Shusuke Maeda††,
Takaaki Matsumoto††, Masakatsu Nishigaki†††
† Graduate School of Informatics, Shizuoka University †† R&D Headquarters, NTT Data corp.
††† Graduate School of Science and Technology, Shizuoka University

Abstract We have proposed a keylogger detection scheme by monitoring APIs which are often employed by keylogger to capture user's keyboard input. The previous study has shown that the scheme is effective against unknown-keyloggers. However the scheme has a drawback that it may fail to detect keyloggers which conceal their own processes from users. Therefore, we enhance our scheme by including this behavior of "concealing itself" as another typical keylogger behaviors to be monitored. This paper carries out basic experiments to evaluate its detection rate, false detection rate and overhead.

1. はじめに

近年，スパイウェアと呼ばれる悪質なプログラムがインターネット上を横行し，その被害が増加している[1]．スパイウェアの被害が急速に拡大している原因の一つとして，スパイウェアに対するユーザの認識の低さが挙げられる．スパイウェアがインストールされている可能性のある環境（インターネットカフェ等）でのオンラインバンクやクレジットカードによるオンラインショッピングの利用により，スパイウェアの被害に遭うケースが少なくない．さらにスパイウェアは，ウイルスのように感染活動を行わず，ユーザに気付かれないように行動するという特徴を持つ．そのためユーザは自分のPCがスパイウェアに感染していることにすら気付かず，それがスパイウェアによる被害を拡大する要因となっている．

このような現状に対して，各アンチウイルスベンダはスパイウェア検知機能を搭載したウイルス対策ソフトウェアを開発している．このようなスパイウェア検知機能の多くはウイルス検知の場合と同様にパターンマッチング法を用いたものが主流となっている．しかしパターンマッチング法は既知のスパイウェアを検知するには非常に有効な方式であるが，未知のスパイウェアを検知することができないという欠点が存在する．

そこで著者らは，スパイウェアの中でも近年深刻な被害をもたらしているキーロガーを対象に，パターンマッチング法に依らない未知キーロガー検知方式を提案している[2]．文献[2]では，キーボード入力を取得するというキーロガーの挙動に着目し，キーボード入力の取得に用いられるAPIの使用を検出することでキーロガーの検知を行う方式を提案するとともに，評価実験により，本方式が未知キーロガー検知に対して有効であることを示した．しかし本方式には，自プロセスをユーザから隠すキーロガーを検知できないという課題を残していた．

そこで本稿では，文献[2]の方式に「自プロセスをユーザから隠す」という挙動を新たにキーロガーらしい挙動として追加することで，キーロガーの検知精度の向上を図る．「自プロセスをユーザから隠す」という挙

動は正規のプログラムには通常用いられない挙動であると思われるため，この挙動をキーロガーらしい挙動として追加した場合の誤検知の心配も少ないと予想される．

2. スパイウェア及びキーロガーの特徴

2.1. スパイウェアの定義

スパイウェアとは，ユーザの適切な同意なしにインストールされ，以下の事項をユーザの制御を超えて実装したものである[3]．

- ユーザの体験，プライバシー，システムセキュリティに影響を与える重要な変更
- ユーザに無断でコンピュータにプログラムをインストールさせることを含む，システムリソースの使用
- 個人情報や機密情報の収集，使用，配布

スパイウェアはウイルスとよく類似しているが，感染機能や増殖機能については備えておらず，その活動が潜在的である．スパイウェアの中には，自身が動作していることをユーザに知られないように，例えば他プロセスに寄生することによって自プロセスをユーザから隠すような挙動をとるものも存在する．そのためユーザは，自身の PC 内でスパイウェアが動作していることにさえ気付かないことが多い．

2.2. スパイウェアの種類

代表的なスパイウェアの種類のいくつかを以下に示す[1,4]．

- アドウェア：ユーザが予期しない広告や求めていない広告などをユーザに提供する．
- トラッキングクッキー：Web サイトの閲覧 URL 履歴などを収集するために利用する．
- キーロガー：キーボードからの入力を記録する．
- ブラウザハイジャッカー：スタートページや検索ページなどの Web ブラウザの設定を改ざんする．
- ダイアラー：アダルトサイトなどの有料または特定の番号に接続させる．
- リモートアクセスツール：システムの遠隔アクセスを許可，もしくは制御するように設計されたツール．

2.3. キーロガーの特徴

2.2 節で示した通り，スパイウェアは多種多様であるため，全てのスパイウェアを一括して検知する挙動を規定することは非常に困難である．そこで本稿では，スパイウェアの中でも近年オンラインバンクの不正送金事件[5]など深刻な被害を引き起こしているキーロガーに焦点を当て，「キーロガーらしい挙動」を検出することでキーロガーを検知する方式を提案する．

2.2 節で述べたように，キーロガーとはユーザのキーボード入力を取得するスパイウェアであるため，「キーボード入力を取得する」という挙動をキーロガーらしい挙動と規定する．またキーロガーはスパイウェアの一種であるため，ユーザに自分の存在を気付かれないようにすると考えられる．そこで，「自プロセスをユーザから隠す」という挙動もキーロガーらしい挙動と規定する．これらの挙動を検出することでキーロガーを検知できるのではないかと考えられる．

なお本稿では，クライアント PC の OS として広く普及しており，キーロガーの被害が拡大している Windows に焦点を当て，Windows 環境下にて動作するキーロガーを対象とする．

2.4. キーボード入力を取得する挙動の規定

Windows 環境下において，キーボード入力を取得する方法は，著者らが確認した限りでは二つ存在する．キー判定 API を用いる方法と，フックを用いる方法である．Windows 環境下で実行されるキーロガーは，このいずれかの方法を用いてキーボード入力を取得していると考えられる．以下，これらの仕組みとその検出方法について説明する．

2.4.1. キー判定 API を用いる方法

キー判定 API である GetAsyncKeyState は，呼び出し時に指定したキーが押されているか，また前回の呼び出し時以降に指定したキーが押されていたかどうかを判定する API である．GetAsyncKeyState は，単に指定したキーが押されているか否かを判定する API であるため，任意のプロセスへのキーボード入力を取得することが可能である．

キーロガーがキー判定 API を用いてキーボード入力を取得する場合，ID やパスワードなどを盗み出すために，少なくとも A～Z と 0～9 のキーボード入力を取得すると考えられる．またキーロガーは，キーロガー自身のプロセスへのキーボード入力だけでなく，他プロセスへのキーボード入力も取得する．そのため，他プロセスがキーボードフォーカスを所持している場合でも，キーロガーは GetAsyncKeyState を用いてキーボード入力を取得していると考えられる．

そこで本方式では，キー判定 API を用いたキーロガーの挙動を，「他プロセスがキーボードフォーカスを所持しているときにも，GetAsyncKeyState によって A～Z，0～9 の全てのキーボード入力を取得する」と規定する．

2.4.2. フックを用いる方法

フックとは，Windows OS が各プロセスへ送信するメッセージを SetWindowsHookEx という API を用い

て取得する方法であり，自身のプロセスへのメッセージのみを取得するローカルフックと，全プロセスへのメッセージを取得することが可能なグローバルフックが存在する[6].

例えばキーロガーがフックを用いて Web ブラウザ上でユーザにより入力される ID やパスワードを盗むためには，Web ブラウザへのキーボード入力メッセージを取得する必要がある．キーロガーから見て Web ブラウザは他プロセスであるため，キーロガーは必然的にグローバルフックを用いることになる．なお，フックには様々なタイプが存在し，その中にはキーボード入力メッセージをフックできないタイプも存在する．キーロガーは，他プロセスへのキーボード入力メッセージを取得することが目的であるため，キーボード入力メッセージをフックできるフックタイプを指定していると考えられる．

そこで本方式では，フックを用いるキーロガーの挙動を，「キーボード入力メッセージをフックできるフックタイプのいずれかを指定し，SetWindowsHookEx によるグローバルフックを用いてキーボード入力を取得する」と規定する．

2.5. 自プロセスをユーザから隠す挙動の規定

Windows において，自プロセスをユーザから隠す方法として，CreateRemoteThread を用いる方法が挙げられる[7]．CreateRemoteThread は，他プロセスのメモリ空間にスレッドを生成する API であり，この API を用いれば，例えばキーボード入力を取得するスレッドを他プロセスに生成することも可能である．他プロセスにスレッドを生成してしまえば，キーボード入力の取得はそのスレッドが代行するため，キーロガー本体が終了してもキーボード入力を取得し続けることが可能となる．キーロガー本体は終了しているため，ユーザからキーロガー本体を視認することはできない．そのため，この方法を用いれば，ユーザからキーロガー自身のプロセスを隠すことが可能となる．

そこで本方式では，自プロセスをユーザから隠すキーロガーの挙動を「CreateRemoteThread を用いて他プロセスにスレッドを生成する」と規定する．

なお，2.4.2 節で説明した SetWindowsHookEx によるグローバルフックも，任意の DLL を他プロセスにロードさせる仕組みの一つである．すなわちキーロガーは，SetWindowsHookEx を用いて，キーロギングを行う機能を持たせた不正 DLL を他プロセスにロードさせ，キーボード入力の取得をその DLL に代行させることができる．よってキーロガーは，SetWindowsHookEx を用いてキーロギングを他プロセスに代行させた上で，自プロセスを終了するという方法を用いることによっても，自分自身を隠すことが可能である．しかしこの場合は，不正 DLL にキーボード入力を取得させるために，キーロガーはキーボード入力メッセージをフックできるフックタイプを指定して SetWindowsHookEx を呼び出す必要がある．このため，SetWindowsHookEx を用いて自プロセスを隠蔽するキーロガーの挙動は，2.4.2 節で規定したキーロガーの挙動（キーボード入力メッセージをフックできるフックタイプを指定しての SetWindowsHookEx 呼び出し）と同一となる．ただし，攻撃者が巧妙な手口を用い，キーボード入力メッセージをフックするフックタイプを指定することなくキーロギングを可能にする不正 DLL を作ることが可能かもしれない．本稿では要点を絞るために，そのような巧妙な手口の可能性およびその対策の検討に関しては今後の課題とする．

3. 実験

本章では，キーロガー検知実験を行い，提案方式の実現可能性を示す．また，規定した挙動によって正規のプログラムがキーロガーとして検知されてしまう可能性もあるため，誤検知評価実験を行い，その結果についても考察する．さらに，実装した提案システムのオーバーヘッドについても評価する．

3.1. 提案システムの概要

本方式では，キーロガーらしい挙動を検出するために，2.4 節，2.5 節で示した三つの API の挙動

挙動1.　他プロセスがキーボードフォーカスを所持しているときにも，GetAsyncKeyState によって A〜Z, 0〜9 の全てのキーボード入力を取得する．

挙動2.　キーボード入力メッセージをフックできるフックタイプのいずれかを指定し，SetWindowsHookEx によるグローバルフックを用いてキーボード入力を取得する．

挙動3.　CreateRemoteThread を用いて他プロセスにスレッドを生成する．

を監視する．

Windows 環境下において，任意の API を監視するために様々な方式が提案されている．本方式では任意の API を監視する方式として，Microsoft Research の提供するライブラリである Detours[8]を用いて上記の三つの API をインターセプトし，それぞれの API においてキーロガーらしい挙動（挙動 1〜3）が検出されるか否かを監視することでキーロガーを検知する検査用 DLL を実装する．この検査用 DLL を検査対象プログラムの起動時にロードさせれば，上記の三つの挙動が監視され，対象プログラムがキーロガーか否かを

検査することが可能となる．なお，検査用 DLL の開発環境には Microsoft Visual C++ 6.0 を用いた．

3.2. キーロガー検知実験

本方式の実現可能性を示すために，提案方式を用い，実際にキーロガーの検知実験を行った．ただしキーロガーは検体の入手が非常に困難であるため，実験には商用のキーロガーやインターネット上で公開されているキーロガーを用いた．なお本実験は，本学の LAN から物理的に切り離された Windows 2000 Professional SP4 がインストール済みの PC 上で行った．本実験に使用したキーロガーとその実験結果を表 1 に示す．

表 1 キーロガー検知実験

検査対象	挙動 1	挙動 2	挙動 3
SpyAnywhere [9]	×	○	×
Perfect Keylogger Lite [10]	×	○	×
Activity Logger [11]	×	○	×
XPCSpy [12]	×	○	×
きいろがぁ[13]	○	×	×
キーロガー[14]	○	×	×
Parasite [15]	×	○	×
WingKEY [16]	×	○	×
キーのログをとる者 [17]	○	×	×
Casper [18]	×	×	○

表 1 から，今回の実験で用いた全てのキーロガーを本方式で検知可能であることが示された．このことから，本方式の有効性を示すことができた．

3.3. 誤検知評価実験

次に正規のアプリケーションを用いて，本方式による誤検知の評価実験を行った．本実験に用いるアプリケーションには，Microsoft Office 製品や Web ブラウザ，メーラ等のアプリケーションに加え，誤検知の可能性のあるアプリケーションを重点的に選択した．しかしこれらのアプリケーションは多機能であり，全ての機能を網羅して実験を行うことは非常に困難である．そこで本実験では，基礎実験として各アプリケーションの比較的使用頻度の高い機能を実行した場合に対してのみの挙動監視となっている．なお本実験は，本学の LAN から物理的に切り離された Windows 2000 Professional SP4 がインストール済みの PC 上で行った．本実験に使用したアプリケーションとその結果を表 2 に示す．

表 2 から，比較的使用頻度の高い機能を使用する限りの検査においては，一部のアプリケーションを除き正規のアプリケーションを本方式によって誤検知するようなことはないことが示された．誤検知してしまったアプリケーションについて，以下でその理由を示す．

xkeymacs と AltIME は，グローバルフックを用いてキーボード入力を取得しキーバインドを置換するアプリケーションである．そのため本方式ではキーロガーとして誤検知されてしまった．また Orchis はランチャ（プログラムのショートカットの管理ツール）であり，特定のキーを押下した場合にランチャがアクティブ化する機能が実装されている．この機能を実現するためにグローバルフックを用いてキーボード入力を取得していたため，本方式ではキーロガーとして誤検知されてしまった．しかしこのようなアプリケーションは実際にキーロガーとしての機能を果たし得る可能性を有しているわけであるので，ユーザにその旨を通知したほうが良いという考え方もできるかもしれない．

表 2 誤検知評価実験

検査対象	種類	挙動 1	挙動 2	挙動 3
Mozilla Firefox 2.0	WEB ブラウザ	×	×	×
Microsoft Word 2000	ワードプロセッサ	×	×	×
Microsoft Excel 2000	表計算ソフト	×	×	×
Microsoft PowerPoint 2000	プレゼンテーションツール	×	×	×
Internet Explorer 6	WEB ブラウザ	×	×	×
Outlook Express 6	メーラ	×	×	×
Mozilla Thunderbird 1.5.0.7	メーラ	×	×	×
Orchis [19]	ランチャツール	×	○	×
xkeymacs [20]	キーバインドツール	×	○	×
AltIME [21]	キーバインドツール	×	○	×

3.4. オーバーヘッド

本方式を実装することによって，通常操作に遅延等の悪影響を及ぼしてはならないため，オーバーヘッドを測定した．オーバーヘッドの測定に用いた実験環境は，Windowsをインストールした直後のPCではなく，ある程度の期間，ユーザが実際に使用していたWindows PCを用いた．これは，測定環境をより実環境に近づけるためである．測定に用いたWindows PCのスペックは，OS：Windows 2000 Professional SP4，CPU：Athlon 900MHz，メモリ：128MBである．

ここでは特に，監視対象プロセスにおいて挙動1～3を検査するために要するオーバーヘッドを測定する．オーバーヘッドは，GetAsyncKeyState，SetWindowsHookEx，CreateRemoteThreadの各APIに対して，オリジナルのAPIを呼び出した際の処理時間と検査用DLL付きAPIを呼び出した際の処理時間の差として求めた．各APIに対して，それぞれを単独で1万回呼び出したときの処理時間の平均とそのオーバーヘッドを表3に示す．

表3 キーロガーらしい挙動の検出に要するオーバーヘッド

	オリジナル [μs]	検査用DLL付き [μs]	オーバーヘッド[μs]
挙動1の検出	2.9	15.8	12.9
挙動2の検出	2.8	3489.4	3486.6
挙動3の検出	178.1	191.2	13.1

表3から，最大でも約3.5ミリ秒のオーバーヘッドしか生じていないことが示された．このことから，これらのAPIがプログラム中で使用されていても，キーロガーらしい挙動の検出を行うタスクは，プログラムの処理に大きな影響を与えることはないと考えられる．

4. まとめ

本稿では，文献[2]の提案方式に「自プロセスをユーザから隠す」という挙動を新たにキーロガーらしい挙動として規定することで，キーロガーの検知精度の向上を達成した．キーロガー検知実験や誤検知評価実験の結果から，本方式が自プロセスをユーザから隠すキーロガーの検知にも有効であることを示した．またオーバーヘッドの測定実験から，本方式がユーザにストレスを与えるほどのオーバーヘッドを生じないことを示した．

本方式はパターンマッチングに依らないキーロガー検知方式であるため，たとえ特定の相手を狙うようにカスタマイズされたキーロガーであっても，これを検知することが可能であると期待できる．さらに本方式では，キーロガーらしい挙動として「自プロセスをユーザから隠す」挙動を新たに規定したが，このような挙動はキーロガーだけでなく他のスパイウェアでも行い得る挙動であると考えられるため，本稿にて規定した「自プロセスをユーザから隠す」挙動を，キーロガーという特定のスパイウェアだけでなく，各種のスパイウェア全体の振る舞いを規定する「スパイウェアらしい挙動」として利用することも可能であると期待される．

今後は，本方式の誤検知の低減を検証していきたい．本方式を用いての誤検知評価実験では，xkeymacs，AltIME，そしてOrchisというアプリケーションが誤検知されてしまった．このようなアプリケーションをキーロガーと切り分けるためには，さらに詳細な「キーロガーらしい挙動」の規定が不可欠であると考えられる．例えばxkeymacsやAltIMEなどは，SetWindowsHookExによりユーザのキーボード入力を取得しているが，その情報を外部へ送信するといった挙動は行わない．このため，取得したキーボード入力を「外部に送信する」という動作をキーロガーらしい挙動として規定すれば，誤検知を低減できるのではないかと考えられる．

また本方式では，ユーザのキーボード入力を取得するキーロガーのみを対象としているが，一定時間ごと，あるいはユーザのマウスクリックをトリガとして画面のスクリーンショットを撮るタイプのキーロガーも存在する．このようなタイプのキーロガーに対処するためには，「スクリーンショットを撮る挙動」を新たに規定し，それを検出するルーチンを検査用DLLに実装する必要があるだろう．

さらに近年は，Windowsのカーネルモードで動作するタイプのキーロガーの存在も報告されている[22]．本方式はユーザモードで動作するキーロガーを検知する方式となっており，現時点ではカーネルモードで動作するキーロガーには対応していない．しかし，本方式の「キーロガーらしい挙動を規定し，それを検出することでキーロガーを検知する」というコンセプトは，カーネルモードで動作するキーロガーに対しても適用できるのではないかと考えられる．

これらのキーロガーに対する挙動の規定とその検出に関しては，今後速やかに検討を行っていく予定である．

参考文献

[1] 与那原亨・大谷尚通・馬場達也・稲田勉，トラフィック解析によるスパイウェア検知の一考察，情報処理

学会研究報告, Vol.2005, No.70, 2005-CSEC-30, pp. 23-29, 2005年7月
[2] 高見知寛・鈴木功一・馬場達也・前田秀介・松本隆明・西垣正勝, 動的API検査方式によるキーロガー検知方式の提案, 情報処理学会研究報告, Vol.2006, No.26, 2006-CSEC-32, pp.209-214, 2006年3月
[3] Anti-Spyware Coalition, "Anti-Spyware Coalition Definitions and Supporting Documents", http://www.antispywarecoalition.org/documents/definitions.htm
[4] Anti-Spyware Coalition, "Glossary", http://www.antispywarecoalition.org/documents/glossary.htm
[5] ITmedia, "スパイウェアによる不正送金被害が拡大, みずほ銀行やジャパンネット銀行でも", http://www.itmedia.co.jp/enterprise/articles/0507/06/news024.html, 2005/07/06
[6] Microsoft, "SetWindowsHookEx", http://msdn.microsoft.com/library/default.asp?url=/library/en-us/winui/winui/windowsuserinterface/windowing/hooks/hookreference/hookfunctions/setwindowshookex.asp
[7] Jefferev Richter (著), 長尾高弘, 株式会社ロングテール (翻訳),"Advanced Windows 改訂第4版",アスキー出版局, 2001/05
[8] Microsoft Research, Detours, http://research.microsoft.com/sn/detours/
[9] Symantec Security Response,Remacc.SpyAnywhere, http://www.symantec.com/region/jp/avcenter/venc/data/jp-remacc.spyanywhere.html
[10] Symantec Security Response, Spyware.Perfect, http://www.symantec.com/region/jp/avcenter/venc/data/jp-spyware.perfect.html
[11] Symantec Security Response,Spyware.ActivityLog, http://www.symantec.com/region/jp/avcenter/venc/data/spyware.activitylog.html
[12] Symantec Security Response, Spyware.XpcSpy, http://www.symantec.com/region/jp/avcenter/venc/data/jp-spyware.xpcspy.html
[13] Vector, きいろがぁ, http://rd.vector.co.jp/soft/win95/util/se322072.html
[14] Vector, キーロガー, http://www.vector.co.jp/soft/win95/util/se334334.html
[15] Vector, Parasite, http://www.vector.co.jp/soft/winnt/util/se327656.html
[16] Vector, WingKEY, http://www.vector.co.jp/soft/winnt/util/se263226.html
[17] Vector, キーのログをとる者, http://www.vector.co.jp/soft/win95/util/se369025.html
[18] S-CENTER, Casper, http://www.s-center.net/index.php
[19] Vector, Orchis, http://www.vector.co.jp/soft/win95/util/se127007.html
[20] xkeymacs, http://www.cam.hi-ho.ne.jp/oishi/
[21] AltIME, http://www.chombo.com/
[22] Symantec.com, Spyware.InvisibleKey.B, http://www.symantec.com/en/uk/small_business/security_response/writeup.jsp?docid=2004-110609-2102-99