2007－CSEC－39（10）
2007／12／14

キーロガー無効化手法の提案と開発

シェン シウェイ　安井 浩之　松山 実
武蔵工業大学

あらまし 本研究では，キーボードからの入力を監視，記録するソフトウェアであるキーロガーを，入力情報の監視，記録の観点から，メッセージ横取り型，メモリ監視型，組み込み型の三種類に分類する．この分類の中の，メッセージ横取り型を中心に，キーロガーの動作を無効化させる手法を提案する．具体的には，コンピュータの中でも低レベルで動作するフィルタドライバを用いて，入力情報をメッセージ横取り型キーロガーが動作できる領域からバイパスさせることでキーロガーの動作を無効化させる．また，その手法を用いて構築したシステムの実験を行い，提案する手法の有効性を示す．

Proposal and Development of Invalidation Technique Against Key Logger

Chih Woei Chen, Hiroyuki Yasui, Minoru Matsuyama
Musashi Institute of Technology

Abstract This research divides key logger into three types: message-steal type, memory-monitor type, and embedded type at the standpoint of how key logger records information. This paper mainly focus on the invalidation technique against message-steal type which is implemented on user-mode. Concretely, a filter driver is used to bypass the area at which the key logger can work. From the experimental results, it is shown that the technique proposed here is effective against message-steal type key logger.

1. はじめに

キーロガーとは，キーボードからの入力を監視，記録するソフトウェアを指し，もともとはデバッグなどの目的で作られたものである．しかし，近年はその仕組みを悪用し，ユーザがキーボードから入力した情報を搾取する犯罪が多発している[1]．

現在，市販のアンチウィルスソフトウェアやスパイウェア対策ソフトウェアでは，一部のキーロガーの検出及び駆除ができるようになっている．また，特定のタイプのキーロガーの動作を無効化することができるキーロガー対策ソフトウェアもある．しかし，これらの対策法のほとんどが不正でないアプリケーションに悪影響を及ぼす可能性がある．システムに影響しないものもあるが，それらについてはユーザモードで動作するものにしか効果がないという欠点がある．

そこで，本研究では不正でないアプリケーションに悪影響をもたらすことなく，キーロガーそのものを無効化させる手法を提案し，その手法を実装した対策システムの構築を行う．本手法は，キーロガーが入力情報をどのように監視，記録するかという観点から，メッセージ横取り型，メモリ監視型，組み込み型の三種類に分類し，その中のメッセージ横取り型を対象とするものであり，入力情報をメッセージ横取り型キーロガーが動作できる範囲に流さないようにするものである．具体的には，コンピュータの中でも低レベルで動作するフィルタドライバを用いて，入力情報をOSに渡さず，直接フィルタドライバからアプリケーションに送るバイパス入力関数をアプリケーション開発者に提供し，この関数を利用することで，メッセージ横取り型キーロガーから入力情報を守る．

本稿では，クライアントPCのOSとして広く普及し，キーロガーの被害が拡大しているWindows環境を対象に，ユーザモードで実装されたメッセージ横取り型キーロガー対策として構築したシステムの実験結果を示し，本システムの有効性について報告する．

2. キーロガーの種類

キーロガーは大きくハードウェアタイプとソフトウェアタイプに分けることができる．ハードウェアタイプは，一種のメモリ装置で，キーボードのPS/2端子やUSB端子とコンピュータの間に設置し，キーボードから物理的に発

生する信号がコンピュータに渡される前に記録するハードウェアである．それに対し，ソフトウェアタイプは一種のプログラムで，コンピュータにインストールし，OS 内部で発生する情報を記録するソフトウェアである．

ソフトウェアタイプのキーロガーについては，いろいろな分類方法があるが，本研究ではキーロガーが入力情報をどのように監視，記録するかという観点から，以下の三種類に分類している．以下に各種類の特徴について説明する．

➢ メッセージ横取り型

ある機構 A から別の機構 B に渡される入力情報が機構 B に渡される前に，その通信の外部から情報を横取りし，記録してから機構 B に渡す方法を用いて実装したキーロガーである．ユーザモードではアプリケーションとして実装され，OS がアプリケーションに渡すさまざまな Windows メッセージを横取りするメッセージフック関数を利用して作られたものがほとんどである．また，カーネルモードではフィルタドライバとして実装され，キーボードのデバイスドライバに取り付けることによって，デバイスドライバと OS の間の通信を横取りするように作られている．

メッセージフック関数はさまざまな言語でライブラリ関数として提供されていることが多く，このタイプのキーロガーは他のタイプと比べて，より少ない知識で作成することができるなどの理由で，インターネット上に最も出回っているタイプのキーロガーである．本研究を進めるにあたり調べた結果，50 を超えるキーロガーの中で，40 以上がこのタイプであることが確認できた．

➢ メモリ監視型

メモリ監視型キーロガーは，通信する機構が利用しているメモリ領域などを監視することで，入力情報を記録する．ユーザモードではアプリケーションが確保したメモリ領域を監視し，カーネルモードではデバイスドライバのシステムバッファを監視する．このタイプは作成するためにはメモリ分析など，高度な知識が必要で，インターネット上ではあまり出回っていないが，メモリそのものを監視しているため，高い精度でキー入力を記録できる上，対策も難しい．

➢ 組み込み型

通信する機構そのものがキーロガーとなっているタイプである．このタイプは前述の二種類のように通信の外部から情報を記録する仕組みとは異なり，通信の内部に存在する仕組みになっている．通常ではキーボードのデバイスドライバ，PS/2 や USB バスドライバなどに仕込まれることが多い．このタイプのキーロガーを作るには，非常に高度な知識が必要となるため，簡単に作ることはできない．しかし，これらのドライバは OS の構成の中でももっとも大切な役割を果たしているため，対策が非常に難しい．現在，このタイプのキーロガーへの対策は存在していない．このタイプについて調べた限り有効な対策法がない上，ドライバタイプキーロガーが仕込まれるということは，すでに管理者権限レベルのパスワードが漏洩しているということを意味している．これは非常に重大な事態で，もはやキーロガーだけの問題ではない．

3. 既存研究

ここでは，現在キーロガー対策に使われている技術を三つに分類して，関連研究について述べる．

➢ パターンマッチ

アンチウィルスソフトウェアや，アンチスパイウェアソフトウェアがウィルス及びスパイウェアを検出するために用いる手法である．これは，不正プログラムが行う振る舞いと，不正プログラムが持つバイナリコードのデータベースを持ち，データベースにあるデータと一致，もしくは似ている場合は，そのプログラムを危険性の高いものとして検出，駆除するものである．この手法はカーネルモードで実装されたキーロガーを除くすべてのキーロガーに有効であり，代表なものに Symantec AntiVirus[2]がある．また，動的 API 検査方式によるキーロガーの検知方法も提案されている[3]．しかし，パターンマッチ手法共通の欠点として，未知なタイプの不正プログラムを検出できないのと，振る舞いが非常に似ている不正でないプログラムを不正プログラムとして誤判定することが挙げられる．

➢ 機能無効化

キーロガーが利用する機能を無効化することでキーロガーの動作を無効化する手法である．ユーザモードで動作するメッセージ横取り型キーロガーの場合，メッセージフック関数機能を利用できなくすることで，キーロガーの動作を完全に防ぐことができる．この手法はアプリケーションとして実装されたメッセージ横取り型キーロガーにだけ有効であり，代表的なものにノーロガー[4]がある．しかし，メッセージフック関数は IME（Input Method Editor，文字入力を補助するソフトウェア）を含む多くのアプリケーションが利用しているため，メッセージフック関数機能を無効化することにより，これらの不正でないプログラムの動作にも影響を及ぼすことになる．

➢ バイパス

キーロガーが動作できる領域をバイパスすることで，キーロガーの動作を無効化する手法である．ユーザモードで動作するメッセージ横取り型キーロガーはメッセージフック関数を用いて OS からアプリケーションに渡す

入力情報を記録するが，このタイプのキーロガーが動作できる領域，つまりOSより上の領域に入力情報を渡さないようにすることで，キーロガーの動作を防ぐことができる．機能無効化手法はメッセージフック関数を直接利用できなくするのに対し，バイパス手法はメッセージフック関数を間接的に利用できなくすることでキーロガーの無効化を果している．そのため，この手法もユーザモードで実装されたメッセージ横取り型キーロガーにだけ有効である．一例として LocalSSL[5]がある．また，本研究で提案する手法もこの分類に属するが，LocalSSL は特定のサイトへのアクセスにだけ効果を発揮するものであり，アプリケーション開発時に一つの部品として提供する本研究のスタンスとは異なり，実装方法と利用形態も異なる．なお，LocalSSL はユーザモードで実装されたキーロガーにしか対応できない．

ここで分類した各対策手法が対応できるキーロガーと，不正でないアプリケーションに影響をもたらすかどうかを表1に示す．

表1　既存対策法の効果と影響

キーロガーの分類	動作レベル	*1	*2	*3
メッセージ横取り型	ユーザ	○	○	○
	カーネル	×	×	×
メモリ監視型	ユーザ	○	×	×
	カーネル	×	×	×
組み込み型	ユーザ	○	×	×
	カーネル	×	×	×
不正でないアプリケーションへの影響		○	○	×

○：効果あり　×：効果なし

*1:パターンマッチ　*2:機能無効化　*3:バイパス

4.　提案手法について

キーボードフックタイプキーロガーの動作をより明確にするために，まずキー入力の流れを確認する必要がある．通常システムのキー入力の流れを図1に，ユーザレベルで実装されたメッセージ横取り型キーロガーを仕込んだシステムのキー入力を図2に示す．

キーボード上のキーが押下されると，スキャンコードが発生する．そのスキャンコードはデバイスドライバによって，OSに渡される．OSはそのスキャンコードを仮想キーコードに変換し，アプリケーションに渡す．メッセージ横取り型キーロガーでは，メッセージフック関数を用いて，OSとアプリケーションの間に流れる情報を記録してから，アプリケーションに渡すという手順で，入力情報を横取りする．その特性に注目し，OS が提供した通常ルートと違うルートを作り，アプリケーション開発者がキーロガーに記録されたくない特定の重要な入力情報にだけ，新しいルートを使って情報をアプリケーションに渡すようにする．つまり，アプリケーション開発者には，アプリケーションとデバイスドライバの間をバイパスし，スキャンコードをOSに渡さないシステムを提供する．

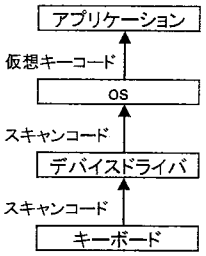

図1　キー入力の流れ

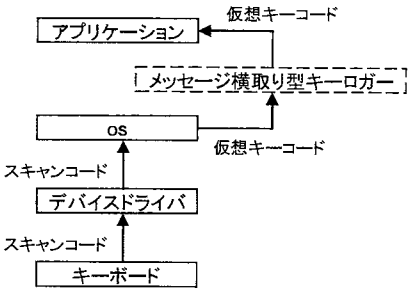

図2　メッセージ横取り型（ユーザレベル）キーロガーを仕込んだシステムのキー入力の流れ

このシステムを構築するにあたって，別ルートを作成する機構をアプリケーション開発者に利用しやすい形態で提供することが必要である．別ルートを作成する機構は，標準システムではカバーしきれないデバイスドライバレベルの機能を補うために使われている中間ドライバの一種，フィルタドライバで構築し，アプリケーション開発者にはこの別ルートを用いて情報を伝達するためのライブラリ関数を提供する．よって，本研究で構築するシステムは単体のプログラムとして動作するものではなく，ライブラリ関数とフィルタドライバのペアとして提供され，ライブラリ関数がほかのプログラムに呼び出されることによって動作するものである．

5. システム構成

本研究で構築したシステムの構成を図3に示す．OSを立ち上げた時点で，フィルタドライバがロードされ，待機状態になっており，ライブラリ関数の制御によって入力情報の処理を行う．ライブラリ関数が呼び出されていないときは，フィルタドライバに流れた情報にはバイパス処理を加えず，そのまま制御を下位のデバイスドライバに返す．ライブラリ関数が呼び出されたとき，フィルタドライバに流れた情報にバイパス処理を加え，アプリケーションに渡す情報の形式を整える処理を行い，本研究で提供したルートに処理後の情報を流すとともに，システムバッファ内の情報をリセットし，制御を下位のデバイスドライバに返す．

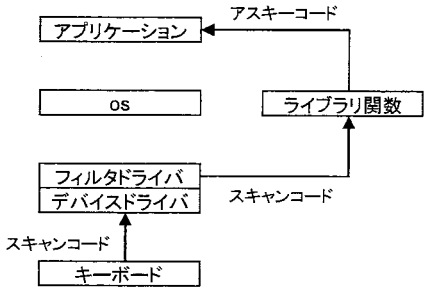

図3 システムの構成

このリセット処理は，入力情報を下位ドライバに渡さないようにするための処理である．また，フィルタドライバは他のデバイスドライバと同じく，OSがシャットダウンするまでは常駐状態であり，OSがシャットダウンするときにアンロードがされる．このシステムの大まかな処理は以下のようになる．

(1) アプリケーションレベルでのライブラリ関数呼び出しによりシステムが起動される
(2) フィルタドライバの初期化
(3) 下位のデバイスドライバからスキャンコードを受け取り，システムバッファに情報を入れる
(4) スキャンコードが発生したときのShiftキーの状態をスキャンコードに付与し，ライブラリ関数に渡す
(5) Shiftキーの状態に基づいて，スキャンコードをアスキーコードに変換する
(6) ライブラリ関数を呼び出したアプリケーションにアスキーコードを渡す

また，アプリケーションがライブラリ関数を呼び出し，入力処理を行うときのシーケンスを図4に示すようになる．

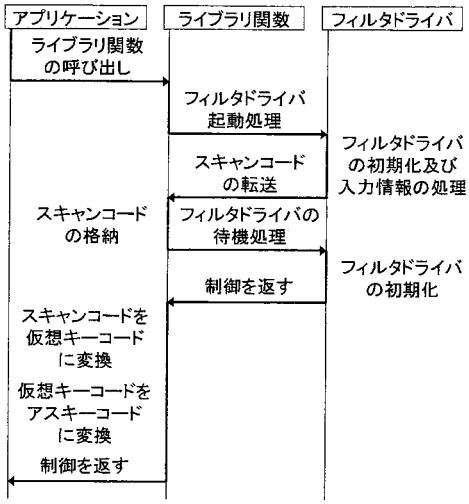

図4 システム動作のシーケンス図

なお本システムは，コンソールプログラムなど，CUIプログラムのために構築されており，現時点ではウィンドウズプログラムなどのGUIによる利用はできない．また，入力情報がOSに渡された後に国固有の文字（漢字，仮名など）に変換されるIMEによる入力も，入力情報をOSに渡さない本システムを用いて実装したアプリケーションでは利用できない．

システムを利用するためのライブラリ関数を表2に示す．第二引数であるエコーバックのタイプについて説明する．本システムでは三種類のエコーバックを提供し，アプリケーション開発者は状況によってエコーバックの種類を選択することができる．第二引数とエコーバックタイプを表3に示す．

表2 ライブラリ関数の説明

BypassInputSystem	
関数型	int BypassInputSystem(char*, int, int);
引数	入力バッファのポインタ，エコーバック
返戻値	成功した場合は0，失敗した場合は-1
機能	指定した入力バッファに入力された
#include	BIS.h

表3 第二引数とエコーバックタイプ

第二引数	エコーバックのタイプ
NO_ECHO	エコーバックなし
CHAR_ECHO	文字エコーバック
ASTERISK_ECHO	アスタリスクエコーバック

6. 実験と結果及び考察

6.1 実験

本研究で提案する手法を用いて構築したシステムがメッセージ横取り型キーロガーに有効であるかどうかを検証するために，該当するタイプのキーロガーを実際に仕掛け，本システムによる入力が記録されるかどうかの実験を行った．なお，本研究で構築したシステムと既存のキーロガー対策法を比較するために，機能無効化手法を用いたノーロガー[3]と，パターンマッチ手法を用いたSymantec AntiVirus[2]でも実験を行った．インターネット上でダウンロードできる各メッセージ横取り型キーロガーを起動した状態で，本システム及びノーロガーの利用と，Symantec AntiVirus によるスキャンの計三種の環境で実験を行った．また，本研究で使用したPCの仕様を表4に示す．

表4　実験環境

実験機	
OS	Microsoft Windows XP Professional
CPU	Intel (R) Pentium (R) 4 2.00GHz
メモリ	1024MB

なお，本システムは前述のように CUI プログラムでしか利用できないので，テストはコンソールプログラムを用いて行った．また，ノーロガーを起動した状態ではコマンドプロンプトが正常に動作しないため，ノーロガーの実験では，Windows が提供するメモ帳を用いて行った．さらに，Symantec AntiVirus はスキャンによりキーロガーを検出・駆除するため，実験に用いるプログラムを用意する必要はなく，キーロガーを仕掛けた後にスキャンを掛ける方法で実験を行った．

本システムとノーロガーでは，入力した情報がキーロガーに記録されなかった場合，そのキーロガーに対して有効であると判断する．また，Symantec AntiVirus では，仕掛けたキーロガーを検出・駆除できた場合，そのキーロガーに対して有効であると判断した．

6.2 実験結果

6.1 に示した実験を行った結果を表5に示す．なお，この実験は，本システムと既存の対策システムを比較し，どれが優れているかを示すことが目的ではなく，キーロガーに対して有効であるかどうかを示すものである．

計 21 種類のメッセージ横取り型キーロガーを用いて実験を行った結果，本システムの利用により防げるキーロガーは 21 種類，ノーロガーの利用では 18 種類，Symantec AntiVirus のスキャンにより検出・駆除では 14 種類であった．

表5　実験結果

キーロガー名	*1	*2	*3
FamilyKeyLogger[6]	○	○	○
GoldenKeylogger[7]	○	○	○
きいろがぁ[8]	○	○	×
キーのログをとる者[9]	○	○	×
KeyLogger[10]	○	○	×
Parasite[11]	○	○	×
PC Activity Monitor Pro[12]	○	×	○
WingKey[13]	○	○	×
Boss Everyware[14]	○	○	○
Free KGB Key Logger[15]	○	○	○
Golden Eye[16]	○	○	○
Handy Keylogger[17]	○	○	○
STARR PC & Internet Monitor[18]	○	×	○
All In One Keylogger[19]	○	○	○
Powered Keylogger[20]	○	×	○
SC-KeyLog Free[21]	○	○	×
SC-KeyLog Pro[21]	○	○	○
Spytech SpyAgent[22]	○	○	○
SpyBuddy[23]	○	○	○
Stealth Keylogger[24]	○	○	○
MixLogger[25]	○	○	×

○：効果あり　×：効果なし

*1:本システム　*2:ノーロガー　*3:Symantec AntiVirus

6.3 考察

本研究で提案する手法を用いて構築したシステムは 21 種類のキーロガーすべての動作を防げたのに対し，ノーロガーは 18 種類しか防ぐことができなかった．本研究で提案する手法は，OS に Windows メッセージを渡さないようにすることで，メッセージフック関数の機能を間接的に無効化するものであるが，ノーロガーはメッセージフック関数の機能を直接的に無効化する手法で実装されている．一見これら2つの手法は似ており，結果も同じになると思われたが，このような違いがでた．一般的にメッセージフックは Windows が提供した関数を用いて実装することが多いが，Windows が提供した関数を用いない実装方法もないとは言い切れない．そのため，ノーロガーが防げなかった種類のキーロガーは，一般的なメッセージフックの実装と違う方法を用いたものである可能性が高い．しかし，本システムは，OS に Windows メッセージを渡さないようにする手法であり，Windows メッセージが OS に流れなければ，メッセージフックしても情報が記録されることはない．よって，本システムはメッセージフックの実装方法にかかわ

らず，ユーザレベルで実装されたすべてのメッセージ横取り型キーロガーの動作を防ぐことができると考えられる．また，ノーロガーを起動した状態でキーボードを操作するときは，入力情報が正しく反映されないなどの問題を確認できたが，本システムではこれらの問題はなく，正常にキーボードの操作が行えた．

Symantec AntiVirus は 21 種類のキーロガーの中から 14 種類しか検出・駆除することができなかった．それらのキーロガーは脅威性が低いと判断されたものか，未知パターンのものだと考えられる．前述したパターンマッチ手法の欠点は，実験結果からも確認できた．本システムはパターンマッチ手法ではないため，本システムと Symantec AntiVirus を直接比較することはできないが，Symantec AntiVirus が検出できないメッセージ横取り型キーロガーを本システムでは防げるようになったことから，未知パターンのメッセージ横取り型キーロガーについても，本システムはその効果を発揮することができると考えられる．

以上のことから，本研究の提案手法で構築したシステムは，不正でないアプリケーションに悪影響をもたらすことなく，ユーザモードで実装されたメッセージ横取り型キーロガーの種類を問わず，動作を無効化できると考えられる．

7. まとめ

本稿では，ユーザモードで実装されたメッセージ横取り型キーロガーの活動領域をバイパスすることで無効化させる手法を提案した．実験から，当手法がキーロガーの無効化に有効であることを示した．しかし，本研究で構築したシステムはユーザモードで実装されたメッセージ横取り型キーロガーに対してだけ効果があり，カーネルモードで実装されたキーロガーには効果がない．同じメッセージ横取り型キーロガーでも，ユーザモードのものとカーネルモードのものは実装方法に大きな違いがある．また，本システムの問題点として，コンソールプログラムにしか利用できないことと，本システムを用いたアプリケーションは IME による入力に対応できないことが挙げられる．今後はカーネルモードで実装されたメッセージ横取り型キーロガーでも無効化できる手法の提案と対策システムの構築とともに，これらの問題点を改善し，実運用に向けてシステムの改良を行う必要がある．

参考文献

[1]オンラインソフト学習塾, http://fine.tok2.com/home/heto2/02600security2/0212.htm

[2]Symantec, http://www.symantec.com/ja/jp/index.jsp

[3]ノーロガー, http://www.asahi-net.or.jp/~ee7k-nsd/readme9.htm

[4]松本隆明・高見知寛・鈴木功一・馬場達也・前田秀介・水野忠則・西垣正勝, 動的 API 検査方式によるキーロガー検知方式, 情報処理学会論文誌, Vol.48, No.9, pp.3137-3147(2007)

[5] LocalSSL, http://www.bluegemsecurity.com/localssl.htm

[6]KMiNT21 Software, http://www.kmint21.com/familykeylogger/

[7] KMiNT21 Software, http://www.goldenkeylogger.com/index.html

[8] sword のソフト倉庫, http://sword01.web.infoseek.co.jp/key/key_new.htm

[9] Topozo Free, http://www.mtst.jp/topozo/soft.html

[10]野田工房, http://www.urban.ne.jp/home/noda/program2/index.htm?http://www.urban.ne.jp/home/noda/program2/vb/keylogger.html

[11] Vector, http://www.vector.co.jp/soft/winnt/util/se327656.html

[12] Raytown Corporation LLC, http://www.keyloggers.com/pcacme.html

[13] Vector, http://hp.vector.co.jp/authors/VA033002/WingKEY.htm

[14] Boss Everyware, http://www.bosseveryware.com/

[15] ReFog Keylogger Software, http://www.refog.com/free-keylogger/key-logger.html

[16] Inisia, http://www.inisia.co.jp/products/GoldenEye/index.html

[17]WideStep Security Software, http://www.handy-keylogger.com/more-information.html

[18]Global General Software, http://www.genuine-software.com/business/Legal/80894.html

[19]Relytec, http://www.relytec.com/

[20]Eltima Software GmbH, http://www.mykeylogger.com/keystroke-logger/powered-keylogger/

[21]Soft-Central, http://www.soft-central.net/keylog.php

[22]Spytech Softwae, http://www.spytech-web.com/

[23] Buy SpyBuddy, http://www.buy-spybuddy.com/

[24]Ampluanet, http://www.amplusnet.com/products/stealthkeylogger/overview.htm

[25] MixLogger, http://mixlogger.the-ninja.jp/