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

2008－DBS－144 (9)
2008－GN－ 66 (9)
2008／1／24

デジタルフォレンジックのための電子ファイルの伝搬経路視覚化手法

芝口 誠仁[†] 稲場 太郎[‡] 川口 信隆[‡]
田原 慎也[‡] 塩澤 秀和[§] 岡田 謙一[†]

近年では情報化社会の発展に伴い，情報を電子ファイルで扱うことが増えた．しかし電子ファイルは設定ミスや誤操作，管理ミスなどのヒューマンエラーで誤って情報を漏らしてしまう危険性がある．そこで本稿では各ホストのシステムコールを監視することによってエンタープライズネットワーク内で重要な電子ファイルを保持しているホストを把握し，伝搬経路を視覚化する手法を提案する．具体的には，CreateFile や send などの WindowsAPI をフックすることにより実現した．本システムを利用することにより，実際に情報が漏洩したとき，電子ファイルを漏洩をした疑いのある犯人の絞り込みや，電子ファイルの移動に関わっていなかったホストの無実証明などができる．すなわちデジタルフォレンジック行使の際の支援ツールとしての活躍が期待できる．

Visualization of File Transmission Route for Digital Forensics

Seiji Shibaguchi[†] , Taro Inaba[‡] , Nobutaka Kawaguchi[‡] , Shinya Tahara[‡] ,
Hidekazu Shiozawa[§] and Kenichi Okada[†]

In recent years, information technology has been developed and we have come to use electoric files very often. Along with this development, information leakage due to human errors has been increasing. In this paper, we propose a visualization system that identifies where confidential files are in an enterprise network. Our system shows not only hosts that have confidential files, but also transmission routes of the files in the network. The proposed system monitors system calls executed on each internal host by hooking WindowsAPIs, such as *CreateFile* and *send* to detect the transmission of files to the network and removable disks. We can use the system as a digital forensic tool, which indentifies the criminal that leaked important information, or to prove the innocence of hosts that didn't concern confidential files.

1 はじめに

近年では情報化社会の発展は著しく，それに伴ってかつては紙媒体であった情報も，電子ファイル化される機会が多くなった．これにより大量の情報を簡単に管理できるようになった反面，容易にネットワークや USB メモリを用いて機密情報が持ち出されてしまう可能性があり，例え悪意はなくとも，設定ミスや誤操作，管理ミスなどのヒューマンエラーで誤って情報を漏らしてしまう危険性が生じてきた．これは情報漏洩全体の 60%以上を占めていて，情報漏洩の主な原因である [1].

そこで本稿ではエンタープライズネットワーク環境内で重要な電子ファイルを監視し，そのファイルを保持しているホストや伝搬経路をリアルタイムで視覚化する手法を提案する．視覚化により，テキストベースのログではわかりにくいホスト間のつながり，つまり伝搬経路の把握を直感的に把握できる．

本提案手法は，個々のホストで CreateFile や send 等の WindowsAPI をフックし，監視ファイルへのアクセスや外部への通信データ等の監視情報をサーバ側に随時送信する．そしてサーバは各ホストから受け取った情報から，いつどのような方法でどのホストからどのホストに電子ファイルが受け渡されたかを判断することによって実現する．

そして実際に情報の漏洩が発覚したとき，電子ファイルを漏洩をした疑いのある犯人の絞り込みや，電子ファイルの移動に関わっていなかったとして無実証明などに利用する．よって本稿は，情報漏洩時の法的措置行使

† 慶應義塾大学理工学部
Faculty of Science and Technology, Keio University
‡ 慶應義塾大学理工学研究科
Graduate School of Science and Technology, Keio University
§ 玉川大学工学部
Faculty of engineering, Tamagawa University

の補助，すなわちデジタルフォレンジックのための支援ツールの作成を目的としている．

本稿では，まず2章において関連研究に触れ，3章で具体的な電子ファイルの伝搬経路視覚化手法について述べる．そして4章で提案手法の考察を行い，5章は本稿のまとめとする．

2 関連研究

関連研究としては，先ずAPIフックを利用してファイルを監視する研究がある[2][3]．これは，openやwriteシステムコールを監視し，電子ファイルのオープンや，USBメモリなどの外部デバイス媒体へデータを書き出す挙動が検知されたら，GUIによるダイアログで書き出しを行ってよいかを問うアラートを出すものである．この手法は他ホストに重要な電子ファイルが手渡されないようにすることを目的としているが，本稿ではエンタープライズネットワーク内の他ホストに重要なファイルが流出すること事態は構わない．しかし，重要なファイルの受け渡しがいつ誰から誰にどのような手段で受け渡されたというログを取っておく．そして，実際に情報が外部に漏れたときに，視覚化したログから電子ファイルを漏洩をした疑いのある犯人を絞り込んだり，また電子ファイルの移動に関わっていなかったとして無実証明などの法的措置，すなわちデジタルフォレンジック行使に利用することを考えている．

またWindowsAPIを用いたネットワーク通信データを監視する研究としては，送受信間のデータ相違に基づく未知ワームの検知を利用した蔓延防止手法の提案というものがある[4]．これは，ワームが「ネットワークを介した自己増殖機能」を有していることに着目し，未知ワームを検知しようという試みである．具体的にはWindowsAPIのsendやrecv等のデータを監視し，「外部から受信したデータ」と「外部に送信したデータ」の相関を見て，送信データと類似度が高い受信データが得られた場合，ワームの挙動の疑いがあるとして，closesocketを強制発行し，強制的に通信を遮断する手法である．送信データを監視しているという意味では本稿と同じであるが，受信データも共に監視し，2つのデータの相関を見るという点に新規性がある．これは対象が自己複製を行う特徴を持つワームであるが故のものであるが，APIを監視するという点では共通である．

また，電子ファイルを全く違ったアプローチで監視するものがある[5]．これは不正な挙動の検知による内部犯対策として，監視対象を「人間の挙動」として情報漏洩の防止を試みる．具体的には，正規ジョブと不正ジョブを用意し，それぞれの場合で目の動きや，心拍数などの生体情報から不審な挙動を検知する．

3 電子ファイル伝搬経路視覚化手法

本提案手法ではWindowsAPIをフックすることにより，他ホストへ重要なファイルを受け渡す挙動が有るかを監視し，監視データをサーバに随時送信する．そして各ホストからサーバ側に集められたログを解析することにより，伝送経路の視覚化を行っている．

3.1 システム全体像

本提案手法の概略図を図1に示す．

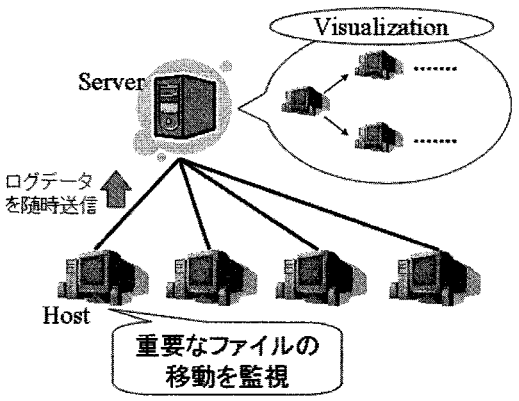

図 1: システム全体像

まず，個々のホストで他ホストに重要な電子ファイルを受け渡す挙動があるか監視する．個々のホストの監視方法はWindowsAPIをフックすることにより実現していて，これは3.2で詳しく述べる．また，電子ファイルの受け渡し手法としては，メール，USB，FTP，メッセンジャーなどが考えられるが，今回は代表的なメールとUSBの2つの手法について考えた．具体的な実装方法については，3.3で詳しく述べている．

そして他ホストへ重要な電子ファイルが受け渡される挙動が見受けられたら，ログデータをサーバ側に送信する．ログデータは電子ファイルが受け渡された時間や自分のホストのIPアドレス等を羅列したテキストデータである．

サーバ側に送られてくるログ情報は主に2つである．

・ファイルの情報

ファイルの情報としては，ファイル名とハッシュ値が送信されてくる．ハッシュ値は重要なファイルの中身を

照合するために使われる．また，用いているハッシュアルゴリズムはSHA-1である．

・メールの送信元，送信先情報

送信元と送信先の情報を知ることで，どのホストからどのホストに重要なファイルが送られたのかがわかる．

サーバ側はこれら2つの情報を整理し，いつどのホストからどのホストへどのような手段で電子ファイルが受け渡されたかを判断する．

そして，最終的には人間の目で見やすいように視覚化する．具体的な視覚化手法は3.4で述べる．

3.2 個々のホストの監視

本提案手法は全プロセスの特定のWindowsAPIをフックすることによって実現されている．フックは，インポートアドレステーブルの書き換えにより実現している．また監視は，WindowsAPIをラッパー関数を設けてフックし，ラッパー関数内で行う．監視対象となる重要な電子ファイルは，誰が持っているか，また誰から誰に受け渡されたのかを明確にしたいファイルであり，例えば社内の機密情報がこれに当たる．

フックによって今回の実装では具体的にメールとUSBメモリでの電子ファイルの受け渡しを監視する．この2つは現在電子ファイルを受け渡すときに用いられる最も一般的な方法であり，多くの電子ファイルの移動を監視できる．

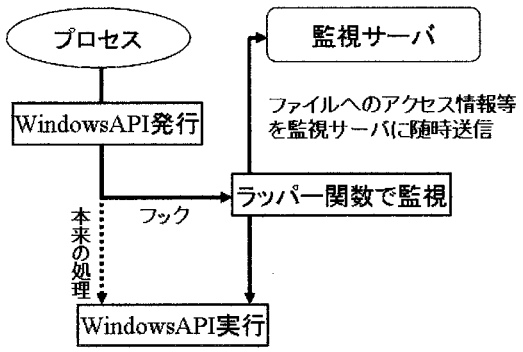

図 2: フック概略図

図2は個々のホストにおいてのAPIフックの概略図である．プロセスがフックしたい特定のWindowsAPIを発行したとき，直接実行せずに，ラッパー関数に一旦処理を渡す．そして引数を解析し該当電子ファイルへのアクセスの有無を監視する．実際にアクセスがあればその旨を監視アプリケーションに通知する．最後に一連の操作が終わったら本物のWindowsAPI関数を呼ぶ．

本手法でフックしているWindowsAPI関数は，CreateFileやsendである．CreateFileはファイルのオープン関数で，Windowsアプリケーションがファイルにアクセスする場合，この関数が呼ばれる．よってCreateFileをフックすれば，ファイルへのアクセスを監視できる．以下にCreateFileのラッパー関数Hook_CreateFileの擬似コードを示す．

```
Hook_CreateFile(PCTSTR fileName,.....){
        if(fileName == 監視対象電子ファイル){
                //ファイルのハッシュを取り
                //ハッシュ値をサーバに送信
        }
        CreateFile(PCTSTR fileName,.....);
}
```

Hook_CreateFile内でfileName引数を見ることにより，どのファイルがオープンされたのかを監視する．そして，重要なファイルがオープンされた場合，ファイルの絶対パスからファイルデータを取得し，ハッシュを取り，そのデータをサーバに送信する．最後に本来の関数CreateFileをラッパー関数Hook_CreateFile内で呼ぶという処理を行っている．

またsendは送信メールを監視する為にフックする．このフックによって，どの宛先にメールを送信したのかを監視する．

以下にsendをフックした場合に実行されるラッパー関数Hook_sendの擬似コードを示す．

```
Hook_send(..,void *buf,..){
        if(実行ファイルがメールソフト){
                //送信データ buf を解析し，
                //送信先アドレスを特定する
        }
        send(..,void *buf,..);//本来の処理
}
```

Hook_send内で先ずsendを呼ぼうとした元のプロセスがメールソフトであるかを判定する．メールソフトであれば，buf引数，つまりメールの送信データを見て，送信先アドレスを特定する．そして，送信先アドレスをサーバに送信する．最後に本来の関数sendをラッパー関数Hook_send内で呼ぶという処理を行っている．

3.3 個々の手法の監視

今回の手法ではメールとUSBメモリでの重要な電子ファイルを監視する.

3.3.1 メールの場合

メールの添付によって電子ファイルが他ホストに受け渡される場合を考える. 今回の実装では, エンタープライズネットワークで統一したメールソフト, Becky!(実行ファイル名 B2.exe) [6] を用いる. 以下に具体的な判定方法を示す.

1. 今回は簡単のためにどのホストがどのメールアドレスを所持しているのか1対1対応しているとして, 監視サーバ上にホストとメールアドレスの対応表を作成する.

2. 各ホストでメールソフト B2.exe が呼ぶ CreateFile を全て監視し, 監視電子ファイルへのアクセスがないかをチェックする. アクセスがあれば, 電子ファイルの添付を行ったとする.

3. send の監視により, メールの送信が検知された場合, 送信先アドレスをサーバ側に送信する.

4. 送信先アドレスを受け取ったサーバは, 送信元ホストと送信先ホストを割り出す. 送信元ホストは送信先アドレスをサーバに送信したホスト. 送信先ホストは送信先アドレスと登録されていたホストとメールアドレスの対応表から判断する. また, 送信先アドレスが対応表に存在しない場合は, 外部に送信したと考え, その旨を表示する.

3.3.2 USBメモリの場合

USBメモリからの情報の受け渡しについては,

- プロセスがUSBメモリに電子ファイルをコピー.
- プロセスがUSBメモリから電子ファイルをコピー.

以上の2つのパターンが考えられる. この2つは以下のチャート図で判定する.

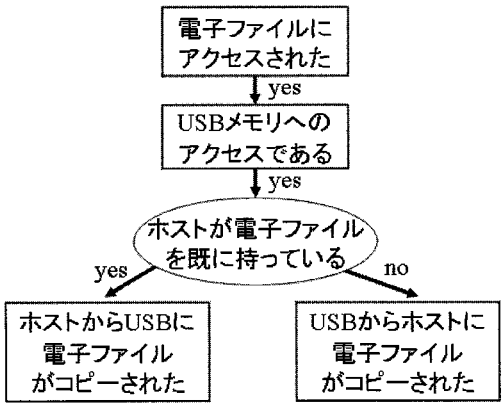

図 3: USB チャート図

1. 重要な電子ファイルにアクセスがあった場合, それがUSBメモリの中にあるファイルかを判定する. 今回の場合はUSBメモリはEドライブやFドライブにあるので, フルパスのファイル名から E:, F: という文字列があるかを判定する.

2. 1. に該当した場合, 次にUSBメモリからデータが入ってきたか出て行ったかを判定する. これはそのホストが既に重要な電子ファイルを持っていたら, USBメモリに電子ファイルがコピーされたと判定し, ホストがまだ電子ファイルを持っていなければ, USBメモリからそのホストに電子ファイルが受け渡されたと考える.

3.4 視覚化

個々のホストでの監視情報を, リアルタイムで随時サーバ側で処理し, 視覚化を行う. 視覚化の際に使われるアイコンの意味を図4に示す. また図5に実際の視覚化の例を示した.

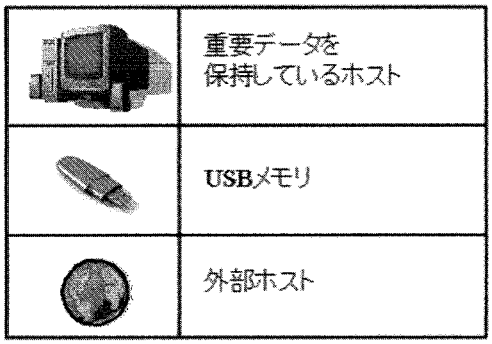

図 4: アイコン対応表

1. **ホスト**
   図5からもわかるように同じホストが一画面に複数登場する可能性がある. これは例えば, 時刻 t1 においてUSBメモリによってファイルを得て,

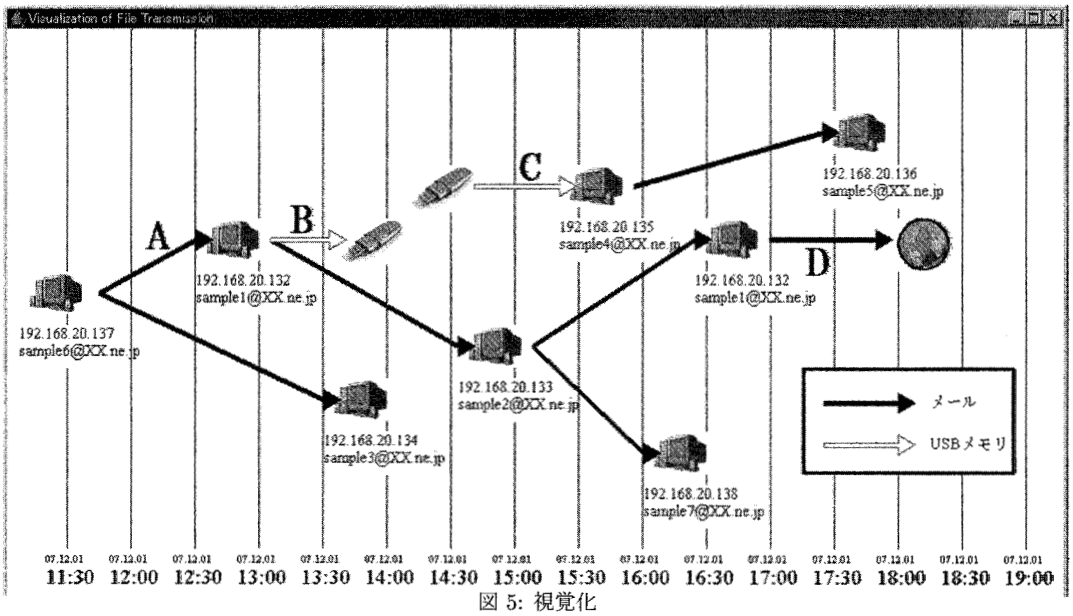

図 5: 視覚化

時刻 t2 にメールでファイルを得た場合に生じる．デジタルフォレンジックの観点から言えば，t1 に送ったファイルではなく，t2 に送ったファイルを使用した可能性があるため，両方の時間が重要であり，両方表示する必要がある．

2. **時間**
ホストが電子ファイルを受け取った時間は横軸に取っている．例えば A の場合だと，ホスト 192.168.20.132 が 12 時 50 分に重要なファイルを手に入れたことを表している．

3. **手段**
以下のような情報伝達パターンが考えられる．それぞれについて記すと共に，各場合の視覚化手法について記す．

A：メールによってあるホストからあるホストに重要な電子ファイルが受け渡された．

図 5 の A で示される矢印ようにホストとホストを線で繋ぐことにより表現する．A の場合は 12 時 50 分にホスト 192.168.20.137 からホスト 192.168.20.132 に重要な電子ファイルがメールで送信されたことを表す．

B：あるホストから重要な電子ファイルが USB メモリにコピーされた．

図 5 の B で示される矢印ようにホストと USB メモリを線で繋ぐことにより表現する．B の場合は 13 時 50 分にホスト 192.168.20.132 から USB メモリに重要な電子ファイルがコピーされたことを表す．

C：USB メモリによってあるホストに重要な電子ファイルが受け渡された．

図 5 の C で示される矢印ように USB メモリとホストを線で繋ぐことにより表現する．C の場合は 15 時 40 分に USB メモリからホスト 192.168.20.135 に重要な電子ファイルがコピーされたことを表す．

D：あるホストがメールによってエンタープライズネットワーク外部に重要な電子ファイルを送信する．

図 5 の D で示されるように，ホストと外部ホストを線で繋ぐことにより表現する．D の場合は 18 時 10 分にホスト 192.168.20.132 から外部ホストに重要な電子ファイルがメールで送信されたことを表す．

これによりわかりにくいログのような文字の羅列ではなく，目で見て直感的に理解しやすいアプリケーションを提供することにより，デジタルフォレンジック支援ツールを提供することが可能となる．

4 考察と今後の課題

デジタルフォレンジックのための電子ファイルの伝搬経路視覚化手法について考察する．

本手法が最も威力を発揮するのは電子ファイルの漏洩が実際に起こったときである．どこに漏洩したか，ファイルの伝搬経路を視覚化によりわかりやすく表示することにより，犯人の絞り込みや特定が容易になる．また保持者がわかるので，本来持つべきではないホストが重要電子ファイルを持っていたら，そのホストに警告を出すことも可能である．さらには，情報漏洩時に電子ファイルの伝搬経路に含まれていないホストは無実といえる．このようなコンピュータに対しての法的問題の解決，ディジタルフォレンジックに対して有効である．

今後はこの提案システムの評価として，システムのオーバーヘッドや，本システムのファイル監視の正確さ，視覚化の画面が人間によって見やすいかといった客観的な評価を取る必要がある．

また，本稿の問題点としては，メールとUSBメモリ以外の他ホストへの情報伝達手段が無いと仮定した上では有効でも，実際にはFTP等の他の他ホストへの情報伝達手段は無数にあるため，今後の研究課題として，他手段でも電子ファイルを監視出来るようにする必要がある．

さらに，重要なファイルをそのまま保存するのではなく，例えば文字情報だけをコピーして，メモ帳などに貼り付けて，その情報を外部に持ち出すようなことをされたら，検知することは不可能である．つまりは，相手が悪意を持って意図的に監視の目を逃れようとした場合，逃れる術があるのが本稿の問題点である．この件に関しては将来ファイル システム フィルタ ドライバ等を用いて，厳密なファイル監視を可能にし，ヒューマンエラーの域に止まらず，ウイルスや悪意を持った情報の持ち出しにも対応出来るようにしたい．

しかしながら，ヒューマンエラーレベルでの監視は十分に可能で，このようなエンタープライズネットワーク環境内で一括して重要な電子ファイルを管理出来る視覚化アプリケーションのプロトタイプを提案したことは，大いに意義がある．

5 まとめ

情報化社会の発展に伴い，かつては紙媒体であった情報も，電子ファイル化される機会が多くなり，悪意はなくともネットワークやUSBメモリを用いて設定ミスや誤操作，管理ミスなどのヒューマンエラーで誤って情報を漏らしてしまう危険性が生じて来た．

そこで本稿ではエンタープライズネットワーク環境内で重要な電子ファイルを監視し，その電子ファイルを保持しているホストや伝搬経路をリアルタイムで視覚化する手法を提案した．

実現方法としては，個々ホストでCreateFileやsend等のWindowsAPIをフックし，監視ファイルへのアクセスや外部への通信データ等の監視情報をサーバに随時送信する．そしてサーバで各ホストから受け取った情報を収集し，いつどのような方法でどのホストからどのホストに電子ファイルが受け渡されたかを把握できるようにする．

つまり，本システムを利用することにより，実際に情報が漏洩したとき，電子ファイルを漏洩をした疑いのある犯人を絞り込みや，電子ファイルの移動に関わっていなかったホストの無実証明などができる．すなわちデジタルフォレンジック行使の際の支援ツールとしての活躍が期待できる．

謝辞

本研究は文部科学省GCOEプログラム，(財) セコム科学技術振興財団の支援を受けて行われた．ここに記して謝意を表す．

参考文献

[1] 日本ネットワークセキュリティ協会 2005年情報セキュリティインシデントに関する調査報告書. http://www.jnsa.org/result/2005/20060803_pol01/2005incidentsurvey_060731.pdf

[2] 大橋 慶. 機密情報の拡散追跡機能を利用した書き出し制御手法. マルチメディア，分散，協調とモバイル(DICOMO2007) シンポジウム，pp. 690-697 (2007.7).

[3] 箱守 聰, 谷口 秀夫. オープン処理に着目した情報拡散追跡法. 情報処理学会研究報告 Vol2005,No79,pp.1-8

[4] 松本隆明, 西垣正勝. 送受信データ間の相違に基づく未知ワームの検知を利用した蔓延防止手法の提案. 情報処理学会論文誌 Vol.47 No.6 pp1941-1954(2006.6)

[5] 丸岡弘和, 西垣正勝. 不正な挙動の検知による内部犯対策 (その 3) ユビキタスネットワーク社会におけるバイオメトリクスセキュリティ研究会，研究発表会予稿集，pp.27-33 (2007.3)

[6] Becky! http://www.rimarts.co.jp/becky-j.htm