L-033

ファイル操作による情報伝搬の追跡機能と可視化機能
Tracing and Visualization Function of Information Propagation by File Manipulation

中務 亮† 山内 利宏† 谷口 秀夫†
Ryo Nakatsuka Toshihiro Yamauchi Hideo Taniguchi

1. はじめに

これまでに，Linux 上で機密情報が拡散する経路を追跡し，機密情報の漏えいを防止する機密情報の拡散追跡機能[1]，および，機密情報の拡散経路を可視化する機能[2]を提案した．しかし，情報漏えいは，利用者が多い Windows で起きているため，Windows での対策が必要である．

本稿では，Windows におけるファイル操作のログを取得し，そのログから情報伝搬を追跡する機能，および情報伝搬を可視化する機能について述べる．

2. 情報伝搬の追跡機能

2.1 基本方式

本研究の目的は，機密情報の拡散追跡機能を Windows で実現することである．このために，ファイル操作による情報伝搬の追跡機能の基本方式と設計について述べる．

情報伝搬の追跡機能はファイル操作のログを取得し，そのログから指定したファイルの情報伝搬を追跡，表示する機能である．情報伝搬の追跡機能の基本機構を図 1 に示す．I/O マネージャとは，AP から発生した I/O 要求をファイルシステムが処理できる形に変換するインタフェースである．I/O 要求は I/O マネージャにより，IRP（I/O Request Packet）と呼ばれる構造体に変換される．以下にログの取得から追跡までの処理の流れを示す．

(1) フィルタマネージャは I/O マネージャからファイルシステムへ送られるすべての IRP をフックする．
(2) ログ収集機構は，フックした IRP のログを取りテキスト形式でログファイルに出力する．
(3) ログ解析機構は，ログ収集機構が出力したログファイルから，指定されたファイルの情報伝搬を追跡し，追跡結果をテキストファイルに出力する．

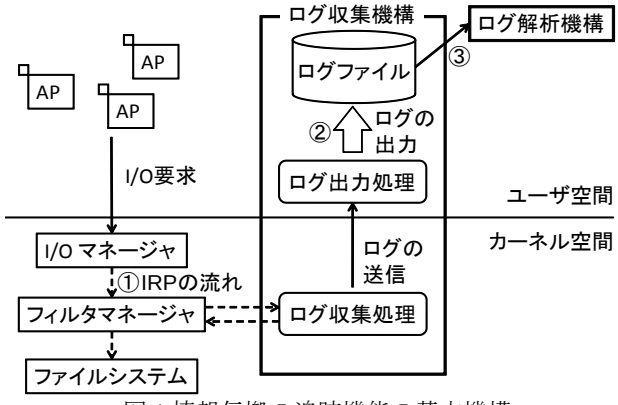

図 1 情報伝搬の追跡機能の基本機構

† 岡山大学大学院自然科学研究科
Graduate School of Natural Science and Technology, Okayama University

2.2 ログ収集機構

ファイルの読み込みや書き込みなどの操作は，すべてフィルタマネージャ[3]を通り，ファイルシステムへ送られる．

そこで，ログ収集機構の実現には minispy minifilter driver（以降，minispy）を用いた．minispy はファイルシステムへの IRP をフックし，ログを取ることができるミニフィルタドライバである．ミニフィルタドライバはフィルタマネージャと協調して IRP に処理を行うことができる．minispy はカーネルモードで動作するプログラム（minispy.sys）とユーザモードで動作するプログラム（minispy.exe）の 2 つで構成されている．

minispy がログをファイルに出力する手順を以下に示す．

(1) minispy.sys はフィルタマネージャから受け取った IRP のログを収集する．収集されたログはバッファに保存される．minispy.sys により収集されるログは，IRP コード，PID，および I/O 要求のターゲットであるファイルやディレクトリのフルパスを含む．
(2) minispy.exe は収集したログの送信要求メッセージを minispy.sys へ送る．
(3) minispy.sys は minispy.exe からログの送信要求メッセージを受信すると，ログを minispy.exe へ送信する．
(4) minispy.exe は minispy.sys から送られたログをファイルに出力する．

2.3 ログ解析機構

ログ解析機構はログ収集機構で得られたログを解析し，指定したファイルの情報伝搬を追跡する．ログの解析の際に必要となる 3 つの情報を以下に示す．

(1) IRP コード

情報の伝搬は追跡対象のファイルの読み込みと，追跡対象のプロセスによる書き込みが行われたときに発生する．そこで，情報伝搬の発生する IRP を判別するために，IRP に含まれる IRP コードを利用する．具体的には，IRP コードが IRP_MJ_READ（読み込み）と IRP_MJ_WRITE（書き込み）の IRP を，情報伝搬の契機として監視することで，情報伝搬を追跡することができる．

また，追跡対象のファイルの移動や，名前の変更が行われたとき，追跡対象ファイルの情報も更新しなければならない．しかし，minispy はファイルのパス変更時に取得しなければならない情報が不足しているため，本稿では今後の課題とする．

(2) プロセス ID

プロセス ID（PID）はプロセスが生成されてから終了するまでの間，変化しない一意な識別子である．情報伝搬が発生する可能性があるのは，特定のプロセスが追跡対象のファイルを読み込み，他のファイルに書き込みを行ったときである．このため，PID を用いて情報伝搬を追跡する．

(3) ファイルのフルパス
ファイルが追跡対象であるかを検査するときに必要となる．

3. 可視化機能

3.1 目的

現在，情報伝搬の追跡機能は追跡結果をテキスト形式で表示している．しかし，これでは情報の伝搬が複雑になった場合，以下の問題点がある．
（問題点 1）情報伝搬の経路の把握が困難となる．
（問題点 2）情報伝搬の追跡機能が正常に動作しているかの把握が困難となる．
これらの問題を解決するために，情報伝搬の経路を可視化する．

3.2 可視化機能に求められる要件

（要件 1）表示する情報が簡潔であること
（要件 2）表示する情報から追跡対象のファイルと PID の情報伝搬の関係を容易に把握できること
（要件 3）表示する情報に漏れと誤りがないこと
3.1 節であげた（問題点 1）を解決するためには，利用者にとって必要な情報だけを表示する必要がある．このため（要件 1）を満たす必要がある．また，どういった経路で情報が伝搬したかを明確にするために（要件 2）を満たす必要がある．（問題点 2）を解決するためには，可視化機能で表示する情報に漏れと誤りがあってはならない．このため，（要件 3）を満たす必要がある．

3.3 基本方式

図 1 の基本機構において，ログ解析機構は情報伝搬の追跡結果をテキスト形式で出力する．可視化機能は，追跡結果をテキスト形式で出力する代わりに，グラフ作成用 AP への入力データを出力する．このため，ログ収集機構に漏れと誤りがない限り（要件 3）を満たすことができる．
追跡対象であるファイルの情報伝搬の経路図は，有向グラフで表示する．これにより，追跡したいファイルの情報伝搬の経路が明確になり，（要件 1）と（要件 2）を満たすことができる．そこで，グラフ作成用 AP には Graphviz[4]を用いる．Graphviz は，DOT 言語で記述されたテキスト形式のファイルを読み込み，図を作成する AP である．
有効グラフは追跡対象ファイルと追跡対象 PID をノード（接点）で表現し，情報が伝搬した向きをエッジ（辺）で表現する．以下に各ノードとエッジに表示する情報について述べる．
(1) 追跡対象ファイルノードに表示する情報
(2) 追跡対象 PID ノードに表示する情報
(3) エッジに表示する情報
(1)には利用者がファイルを識別できるように，ファイルのフルパスを表示する． (2)はプロセス名と PID を併記して表示するのが望ましい．しかし，minispy はプロセス名を取得していないため，本実装では PID のみを表示する．また，ファイルとプロセスを区別するために(1)のノードは四角形で，(2)のノードは楕円形で表示する．(3)について，情報は一つの追跡対象に対して複数個伝搬する可能性がある．このため，エッジの横には時刻を表示する．また，追跡対象に情報を伝搬させた IRP の IRP コードもエッジの横に表示する．

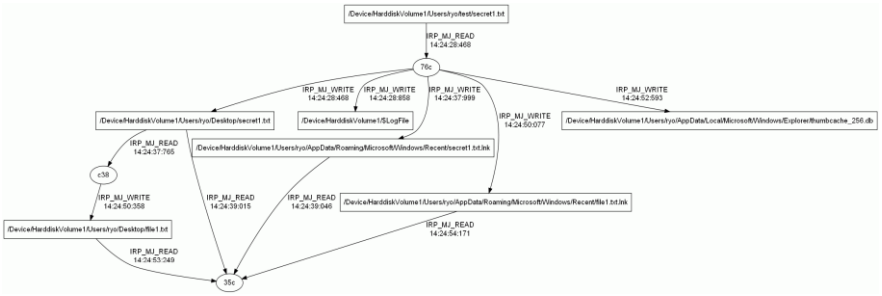

図 2 情報伝搬の経路図

以下の操作を行ったときの情報伝搬の経路を図 2 に示す．
(1) エクスプローラにより secret1.txt をデスクトップに複写する．具体的には，secret1.txt のアイコン上にポインタを合わせ右クリックし，メニューからコピーを選択する．次にデスクトップにカーソルを合わせ右クリックし，メニューから貼り付けを選択する．
(2) メモ帳でデスクトップ上の secret1.txt を編集し，file1.txt という名前で保存する．
(1)，(2)の操作で，ログ収集機構が出力したログは約 70,000 行である．このログから，指定したファイルの情報伝搬を追跡するのは困難である．また，情報伝搬の追跡結果をテキスト形式のファイルで確認するためには，ファイルの内容を一行ずつ追っていかなければならない．可視化機能では，有効グラフで情報伝搬の追跡結果を表示するため，情報伝搬の経路を視覚的に確認できる．これは（要件 1）と（要件 2）を満たしている．また，図 2 より可視化機能は（要件 3）を満たしている．

4. おわりに

ファイル操作による情報伝搬の追跡機能の設計と基本方式について述べた．また，情報伝搬の経路を可視化した．ログの取得は，フィルタドライバを用いることで，ファイルシステムへ送られるすべての I/O 要求のログを取得した．また，情報伝搬の経路を可視化に必要な要件を示し，考察した．これにより，機密情報の拡散追跡機能の Windows での実現の可能性を示した．
残された課題として，ファイルのパス変更時の情報の取得，プロセス名や日付の情報の取得がある．

謝辞

本研究の一部は，科学研究費補助金若手研究(B) (課題番号：21700034)による．

参考文献

[1]田端 利宏, 箱守 聰,大橋 慶,植村 晋一郎,横山 和俊,谷口 秀夫, “機密情報の拡散追跡機能による情報漏えいの防止機構”, 情報処理学会論文誌, Vol. 50, No. 9, pp.2088–2102 (2009).
[2]福島 健太, 田端 利宏, 谷口 秀夫, “機密情報の拡散追跡機能における可視化機能の設計”, 電子情報通信学会 2009 年総合大会講演論文集, Vol.2009, No.4, p.67 (2009)
[3]File System Filter Drivers, WHDC-Windows Hardware Developer Central,http://www.microsoft.com/whdc/driver/filterdrv/default.mspx.
[4]Graphviz, http://www.graphviz.org/.