機密情報の拡散追跡機能によるネットワーク上の機密情報の管理と漏洩防止機構の研究

代表研究者　山内利宏　岡山大学大学院　自然科学研究科　准教授
共同研究者　谷口秀夫　岡山大学大学院　自然科学研究科　教授

1 はじめに

近年，計算機で機密性の高い情報を扱う機会の増加とともに，機密情報が可搬記憶媒体やネットワークを経由し，計算機外部へ漏えいする事例が増加している．情報漏えいの主な原因は，計算機の誤操作や管理ミスといった内部的要因が多く，情報漏えい事例の約 70 %を占めている[1]．このような情報の漏えいを防止するには，計算機の利用者が計算機内部の機密情報の利用状況を把握することが重要である．また，利用者にとって，現在どのプロセスが機密情報をどのように利用しているのかを把握することは，不正なプログラムの兆候を把握する上でも重要である．

しかし，既存 OS では，機密情報に着目して，情報の流れを追跡することは行われていないため，計算機内部の機密情報の利用状況を把握するのは難しい．また，機密情報の利用状況を把握する手法として計算機内の機密情報を有するファイル（以降，機密情報ファイルと略す）への操作を監視し，ログとして出力する機能[2] があるが，テキスト情報としてログを保持するため，課題がある．

我々は，機密情報の漏えいを未然に防ぐ手法として機密情報が拡散する契機となるシステムコール発行に着目し，機密情報が拡散する経路を追跡し，計算機外部への漏えいを検知する機能[3]（以降，機密情報の拡散追跡機能と呼ぶ）を提案した．機密情報の拡散追跡機能は，機密情報の漏えいを計算機利用者へ通知するため，利用者は，計算機外部への機密情報の書き出しを制御できる．しかし，機密情報の拡散追跡機能から利用者が確認できる機密情報の利用状況は，テキスト形式のログ，もしくは取得した情報から作成される機密情報を有する可能性のあるファイルとプロセスの一覧のみである．このため，機密情報の利用状況を把握するには，文献 2 と同様にログを解析する必要があり，情報漏えいの原因を特定する時間を短縮できない．また，機密情報の漏えいを検知した際も書き出し制御は，計算機利用者が書き出しの可否を判断する．このため，書き出しの可否を判断する際にその操作で漏えいする可能性がある機密情報ファイルを特定するには，ログを解析し，機密情報の利用状況を確認することが必要であり，利用者への負担が大きい．このとき，確認にミスがあった場合は，情報の漏えいを許す可能性もある．

現在の機密情報の拡散追跡機能には，3 つの課題がある．

一つ目の課題は，ネットワークを介して他の計算機に機密情報を送信した際に，機密情報の伝搬を追跡できないことである．機密情報は計算機内だけでなく，複数の計算機で共有することが多い．しかし，機密情報の拡散追跡機能は，計算機内部の機密情報の拡散しか把握できない．

二つ目の課題は，機密情報の伝搬情報をログとして計算機内部に保存するものの，そのログが改ざんされ，ログ出力機構が攻撃されるなどしてログが消失した場合に，ログが失われ，機密情報の拡散追跡が正確に行えない可能性があることである．サーバやデスクトップ用途で広く利用されている Linux では，ログの書き出しに syslog プログラムを用いており，攻撃者や不正者がファイルに出力された AP によるログ（ユーザログ）やカーネルによるログ（カーネルログ）を改ざんできる．また，syslog のプログラム自体が改変された場合，出力されたログは信頼できなくなる．さらに，短い間に大量のログが出力された場合，カーネル内のリングバッファに格納されたカーネルログが上書きされ，古いログが消失することがある．これに対処し，ログの消失を防ぐ必要がある．

三つ目は，機密情報の拡散情報をログとして内部に保持しているが，これを人間が理解し，機密情報の伝搬の流れを追跡するには，非常に負担が大きいことである．これは，機密ファイルが伝播する情報を取得できたとしても，機密情報の利用状況を把握するにはログを解析する必要があるためである．ログは，発生したイベントが 1 行ごとに記録してあるため，ログの解析に時間を要する．このため，機密情報の流れを即座に把握することは難しく，情報の漏えいが起こった場合の原因の特定に時間を要してしまう．

本研究では，これらの三つの課題に対処する手法について検討した．

一つ目の課題については， 機密情報の拡散追跡機能を分散環境へ適用するための対処法について述べる．具体的には，通信に関するシステムコールをフックし，機密情報の通信の有無を把握する機構を実現する．機密情報の拡散情報の通信がある場合，送信元と送信先計算機で機密情報の伝搬を伝える方式を設計し，機密情報の伝搬を確実に把握できる機構を実現する．

二つ目の課題については,仮想計算機モニタにより,ログの改ざんと喪失を防止するシステムを提案する．提案システムでは，ログを取得する対象の OS（以降，監視対象 OS）を VM 上で動作させる．また，監視対象 OS 上のユーザログとカーネルログを監視対象 OS のソースコードの修正なしに VMM でも取得する．ユーザログは，監視対象 OS のシステムコールを VMM によりフックすることで，ログの出力を検知し，取得する．提案システムは，ユーザプロセスにおけるユーザログ送信システムコールの発行直後にログを取得するため，ユーザログ送信システムコール終了後のログに対する改ざんなどの攻撃の影響を受けない．また，カーネルログは，バッファへログが出力される直前に現在のバッファの内容を VMM が取得する．これにより，次のログがバッファへ書き込まれる際に，前回出力されたログを必ず取得できる．このため，まだ取得されていない古いログがバッファの上書きにより喪失する問題へ対処できる．

三つ目の課題については，計算機での機密情報の利用状況について特定の機密情報ファイルや特定の期間などに着目し，視覚的に把握可能な機密情報利用状況の可視化機能を提案する．また，既存の機密情報の拡散追跡機能を拡張して，提案機能を実現する方式について述べる．さらに，可視化機能を実現した機密情報の拡散追跡機能で取得したログを基にした提案方式の評価を行い，評価結果を報告する．評価と関連研究との比較から，提案方式の有用性と長所を明らかにする．

2 機密情報の拡散追跡機能の分散環境への対処法

2-1 機密情報の拡散追跡機能

機密情報の拡散追跡機能は，機密情報の拡散に関連するシステムコールをフックすることで，機密情報が拡散する経路を追跡し，外部への漏えいを検知する．

ここでは，機密情報の拡散追跡機能の計算機内のソケット通信による機密情報の拡散の追跡について述べる．

以降では，機密情報を保持している可能性があるとして，機密情報の拡散追跡機能が監視する資源を管理対象，情報を送信する側のソケットを送信元ソケット，情報を受信する側のソケットを受信先ソケットと呼ぶ．

(1) 情報を送信するシステムコールをフックする．
(2) 送信を行うプロセスが管理対象か判定する．管理　対象でない場合，システムコール処理を再開する．
(3) ローカルプロセス間の通信か判定する．リモート　のプロセスへの通信の場合，漏えいを検知する．
(4) 受信先ソケットが管理対象か判定する．管理対象　の場合，システムコール処理を再開する．
(5) 受信先ソケットを管理対象にする．

2-2 分散環境への対処法

設計の前提を以下に示す．

ネットワークに接続されたすべての計算機に機密情報の拡散追跡機能が導入されている．

分散環境に対処する際の課題と対処を以下に示す．

(1) 外部の計算機のソケットを管理対象にする方法

既存の機密情報の拡散追跡機能は，計算機内部のソケット通信について，管理対象のソケットから情報を受信する受信先ソケットを管理対象にする．受信先ソケットがリモートの場合も同様に，受信先ソケットを管理対象にすることで，分散環境へ対処できる．

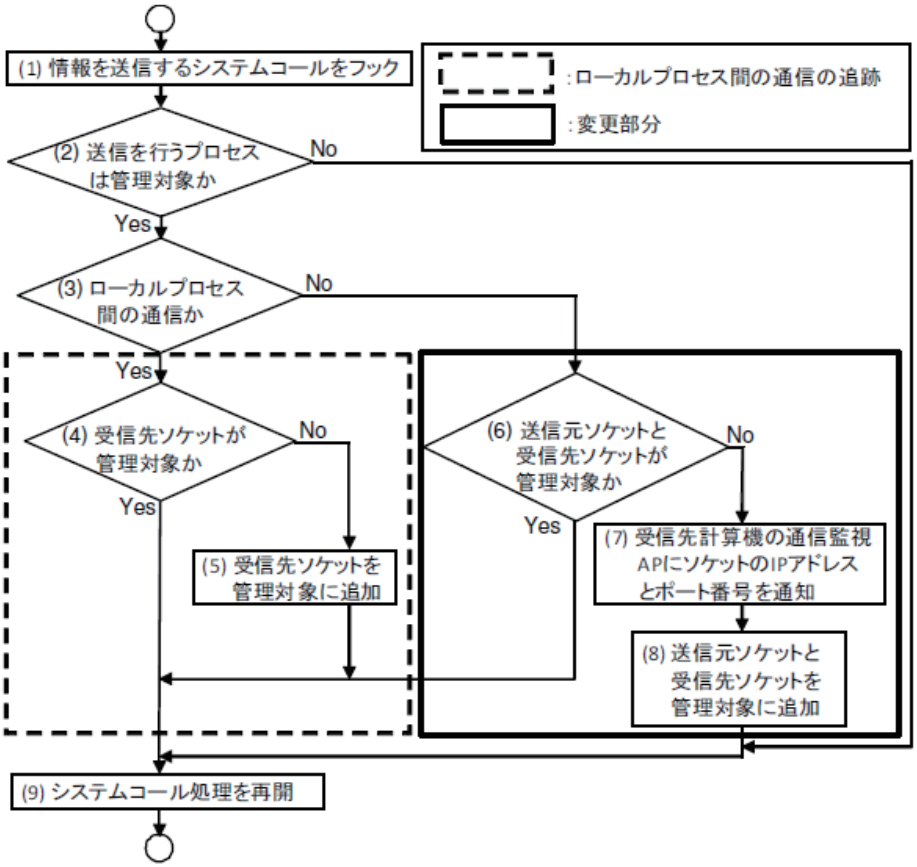

図 1　分散環境への対処による送信元計算機の処理の流れ

分散環境へ対処したソケット通信による機密情報の拡散追跡の処理の流れを図 1 に示し，以下に述べる．以降では，機密情報を送信する側の計算機を送信元計算機，機密情報を受信する側の計算機を受信先計算機と呼ぶ．また，各計算機において，管理対象とするソケットの情報を受け渡す通信監視 AP を起動しておく．

(6) リモートのプロセスへの通信の場合，送信元ソケットと受信先ソケットが管理対象か判定する．なお，両方とも管理対象の場合，システムコール処理を再開する．これにより，通信監視 AP 間の通信を削減する．

(7) 受信先計算機の通信監視 AP に送信元ソケットと受信先ソケットの IP アドレスとポート番号を通知する．

(8) 機密情報の拡散追跡機能は，送信元計算機と受信先計算機のソケットを管理対象にする．受信先計算機では，情報を受信するソケットが管理対象か確認し，管理対象であれば，受信したプロセスを管理対象にする．

2-3 期待される効果

上記の対処により，計算機間の機密情報の拡散の把握と機密情報の計算機間での共有が可能となる．

3 仮想マシンモニタによるログの改ざんと消失を防止するシステム

3-1 システムへの要求

1 章で述べた課題を解決するシステムを提案する．ログへの攻撃への対処では，ログを計算機レベルで隔離することが有効である．計算機内部でログを保護した場合，様々な攻撃によりログが攻撃される可能性がある．また，ログ保護機構への対処では，ログの保護機構自体を AP と OS から隔離することが有効である．AP やカーネル内部でログを保護した場合，ログの保護機構自体が攻撃され，保護したログの信頼性が失われる可能性がある．さらに，カーネルログ消失への対処として，カーネルログを喪失前に取得する必要がある．つまり，リングバッファによるカーネルログの上書きへ対処する方法が必要である．最後に，OS バージョンの限定や導入の困難さ への対処として，OS に依存しない方式の実現がある．これにより，OS の実装を意識する必要がなく，様々な環境への適用が可能となる．また，カーネルに依存したシステムと異なり，カーネ

ルのバージョンアップへ容易に対応でき，常に最新のカーネルを利用できる．

各問題への対処から，提案システムへの要求は以下のようになる．

（要件 1）すべてのログの取得

（要件 2）取得したログの隔離

（要件 3）ログの保護機構自体の安全性の確保

（要望 1）OS の種類やバージョンに依存しないシステムの実現

なお，（要望 1）は，ログの保護の観点からは要件とはならないが，重要な課題である．

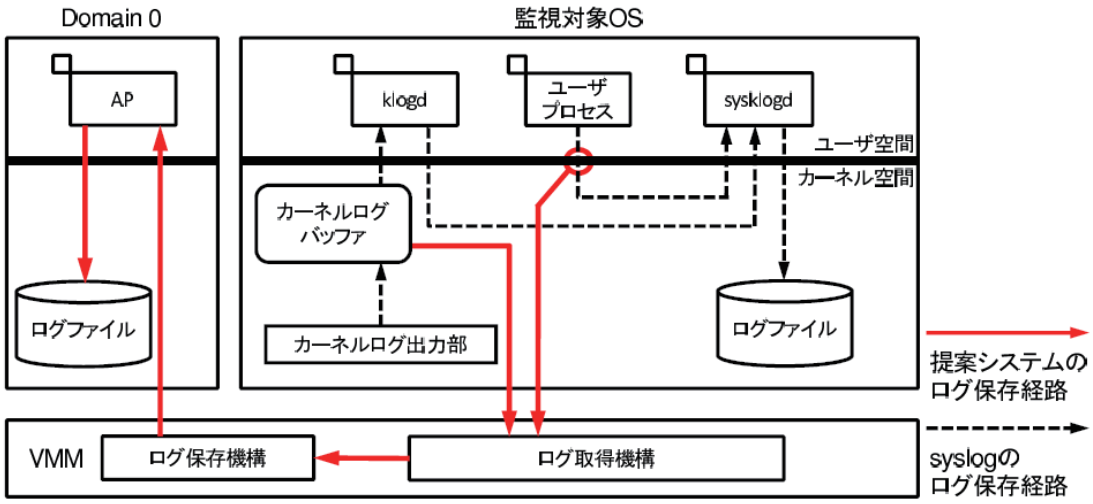

図 2　仮想マシンモニタによるログの改ざんと消失を防止するシステム

3-2 提案システムの構成

提案システムの全体像を図 2 に示す．提案システムでは，監視対象 OS を VM 上で動作させる．ログ取得機構とログ保存機構は VMM 内で動作する．また，監視対象 OS のソースコードを改変しないために，完全仮想化に対応した VMM を対象とする．なお，本研究では，VMM は安全であり，VMM の管理者は不正を行わないと仮定する．

（要件 1）については，ログ取得機構により監視対象 OS 内のユーザログとカーネルログの出力を検知し，取得することで満たす．ユーザログは，/dev/log への send または write システムコールにより出力される．ログ取得機構は，このシステムコールを検知し，ログを取得する．また，カーネルログは，カーネル内部の関数である printk 関数により，カーネルログバッファへ蓄積される．ログ取得機構は，printk 関数の実行を検知し，カーネルログバッファに蓄積されているログを取得する．

（要件 2）と（要件 3）については，ログ取得機構とログ保存機構を VMM 内で実現することで満たす．VMM 内で動作するログ取得機構が監視対象 OS のログを取得し，ログ保存機構へ送信するため，取得したログとログ取得機構は，監視対象 OS の外に隔離されている．このため，ゲスト OS への攻撃により，取得したログとログ取得機構自体が影響を受ける可能性は低い．

（要望 1）は，提案システムを VMM 内で実現し，監視対象 OS を完全仮想化環境で動作させることで実現する．提案システムは，監視対象 OS のソースコードを修正することなく，ログの出力時に VMM に制御を移行できる．ログ取得機構の実現に必要な情報は，監視対象 OS の printk 関数の開始アドレスやカーネルログバッファの領域などのシンボル情報のみである．

3-3 ログ取得機構

3-3-1 ユーザログ取得機能

本機能は，図 2 に示すように，ユーザプロセスが syslog デーモンへログを送信する際に発行するシステムコールを VMM がフックすることでログを取得する．このため，ユーザログ取得機能では，監視対象 OS のソースコードを修正することなく監視対象 OS で発生するシステムコールを VMM により検知する仕組みが必要となる．そこで，提案システムでは，ゲスト OS のシステムコールの発行によりページ例外を発生させる手法を用いた．この手法では，監視対象 OS の sysenter eip msr レジスタの内容を監視対象 OS がアクセス

を許可されていない場所へ書き換える．これにより，システムコールの発行でページ例外を発生させ，VMM へ処理を移行させる（VM Exit）．VMM へ処理が移行した後，監視対象 OS のユーザログを取得し，ページ例外の発生を隠ぺいした後，監視対象 OS の処理を再開させる．これにより，監視対象 OS は，システムコールを VMM にフックされたことを意識せずに動作できる．

システムコールのフックによるユーザログの取得手順は以下の通りである．なお，ログを送信するために，ユーザプロセスは事前にソケットを作成し，/dev/log ソケットファイルに対して connect を発行する．

（１）プロセスごとに，ログの送信に利用するソケット番号を特定する．具体的には，/dev/log ソケットに対する connect システムコールを検知し，connect システムコールの第 1 引数からソケット番号を取得する．

（２）（1）で取得したソケット番号への send と write システムコールを検知し，システムコールの第 2 引数で指定された文字列をログとして取得する．

なお，ログを送信したプロセスの識別には，プロセスごとにユニークな値が設定される CR3 レジスタを用いる．

3-3-2 カーネルログ取得機能

本機能は，監視対象 OS における printk 関数の実行を契機として，ログを取得する．提案システムでは，監視対象 OS 内にブレークポイントを設定する．監視対象 OS において，ブレークポイントを設定した場所に処理が到達すると，ブレークポイント例外が発生し，VMM へ処理が移行する．これを契機とすることで，VMM によるカーネルログ取得が可能となる．

提案システムでは，カーネルログを取得するために，監視対象 OS のカーネルログ出力部にブレークポイントを設定する．ブレークポイントの設定は，メモリ上へロードされているカーネルに INT3 命令を埋め込むことによって実現する．この方式では，メモリ上のデータを書き換えるため，カーネルのソースコードの修正は不要である．なお，INT3 命令の埋め込みは，提供しているすべての VM のメモリ空間を管理できる VMM により実現している．この埋め込み対象のメモリ領域は，監視対象 OS からは書き込み不可のままであるため，安全性は低下しない．

3-4 評価

3-4-1 ログの改ざんの防止

提案システムで取得したログは，監視対象 OS から独立した場所で保持する．このため，攻撃者やマルウェアが監視対象 OS の特権を奪取し，監視対象 OS 内であらゆる操作が可能になった場合でも，監視対象 OS から隔離したログを攻撃することは困難である．

3-4-2 ユーザログの喪失の防止

攻撃者が設定ファイルを改ざんし，syslog の振舞いを変更することで，特定のログを書き出させない状況を想定して評価した．

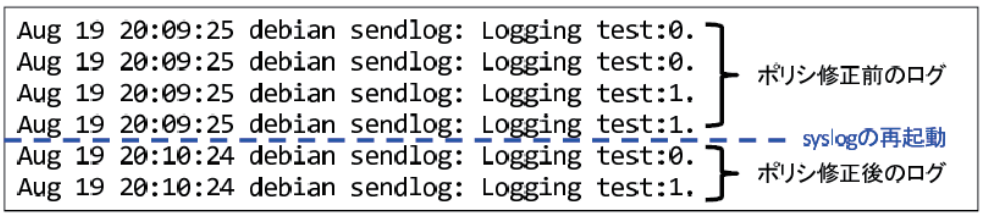

図３　監視対象 OS 上のユーザログ

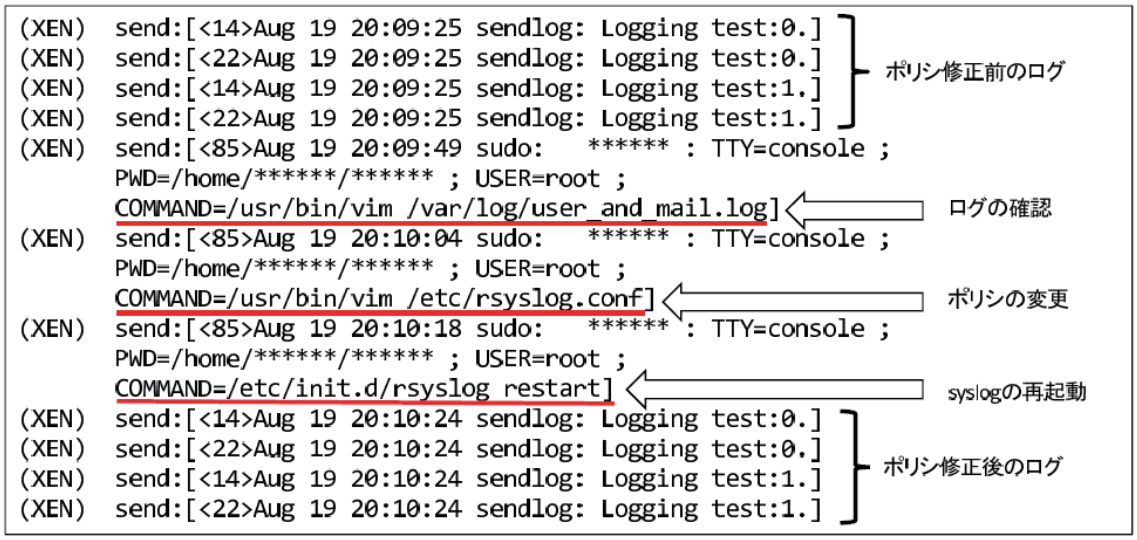

図４　提案システムで取得したユーザログ

図 3 と図 4 は，同じ処理に対する監視対象 OS と提案システムのログである．図 3 には，図 4 の sudo コマンドについてのログが存在しないが，これは，監視対象 OS では，user と mail ファシリティのログしか出力していないためである．図 5 のようにポリシを変更し，syslog の振舞いを変更させた前後でお互いのログを比較した．

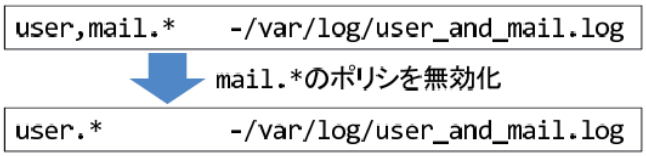

図５　ログ書き出しポリシの変更

図 3 では，ポリシ変更前は user と mail ファシリティのログがそれぞれ 2 回ずつ出力されているのに対し，ポリシ変更後ではログが少なくなっている．これは，ポリシの変更で除外された mail ファシリティのログが出力されていないことを示す．一方，図 4 では，ポリシ変更後もログの量は変わっていない．これにより，提案システムは，syslog の振舞いに関係なく，確実にログを取得できることを確認した．

4 可視化とフィルタリング機能により機密情報の拡散追跡を支援する機構

4-1 可視化が必要な状況

以下に機密情報の拡散追跡機能利用時において機密情報の拡散経路の可視化が必要になると想定される状況について述べる．また，各状況において計算機利用者に求められる作業について述べる．

（状況 1）　機密情報の拡散追跡機能が機密情報の漏えいの可能性を検知した際に，可否判断をする場合

（状況 2）　機密情報が計算機外部へと漏えいした際，漏えいの原因を特定する場合

（状況 3）　機密情報が意図していない拡散をしていないか確認する場合

（状況 1）については，機密情報の拡散追跡機能が計算機外部へのファイル書き出し時に情報の漏えい可能性があるとして検知したファイルを,利用者自身が管理対象ファイルだと認識していなかった場合がある．この場合，どの機密情報ファイルから書き出し対象ファイルに，機密情報が拡散した可能性があるのかを利用者が検証し，計算機外部へのファイル書き出しを許可するか否か判断する必要がある．また，管理対象ファイルを計算機外部に書き出す必要があり，書き出そうとした場合，書き出すファイルに他の管理対象ファイルから機密情報が拡散しているか否かを確認する必要がある．これにより，利用者の判断ミスによる情報漏えいを防止できる．

（状況 2）については，機密情報の漏えいが起こった場合，漏えいの原因を検証することで今後機密情報の漏えいを起こさないように対策する必要がある．これにより，再度の情報漏えいを防止できる．

（状況 3）については，定期的に機密情報の利用状況を確認することで利用者が意図してない機密情報の

拡散を確認できる．これにより，マルウェアなどの不正なプロセスの存在や情報の漏えいを未然に防止できる．

以上の 3 つの状況は，文献 2 や文献 3 などの既存手法では，機密情報ファイルからの情報の伝搬に必要な情報がログに出力されていれば，それぞれ機密情報の利用状況を記したログを解析することで対応できる．しかし，広く利用されている既存 OS は，そのようなログを取得していない．また，文献 3 では，新たに機密情報が拡散する契機でしかログを取得しておらず，一度機密情報が拡散した対象に対して，それ以降の操作のログを取得していないという問題がある．

また，ログの解析には，以下の 2 つの問題点がある．

（問題点 1） ログは時系列順に読んでいく必要があり，ログが増えることで 1 つの管理対象の情報が数百行に渡って分散されて表示される可能性がある

（問題点 2） ログの解析には時間がかかり，迅速に対応することが困難である

（問題点 1）により，解析の際に人為的なミスが発生し誤った解析結果が出る可能性がある．また，（問題点 2）により，計算機利用者の作業効率が低下する．そこで，機密情報ファイルからの伝搬を追跡するのに必要な情報を取得し，ログを自動で解析することで，計算機での機密情報の利用状況の把握を支援でき，過去の利用状況も視覚的かつ詳細に把握可能な機密情報の拡散経路の可視化機能を提案する．

4-2 機密情報利用状況の可視化機能

4-2-1 可視化の目的

以下に可視化の目的を述べる．

（目的 1） 機密情報の拡散経路を既存のログよりも正確に把握可能にする

（目的 2） 機密情報の拡散経路を既存のログよりも迅速に把握可能にする

2.3 節で述べた状況において，利用者の判断ミスによる漏えいを防止し，機密情報漏えい時の漏えい原因を特定するには，機密情報の拡散経路を正確に把握することが重要となる．

そこで,機密情報の拡散経路を既存のログよりも正確に把握可能にすることを 1 つ目の目的とする.また，2.3 節で述べた状況において，利用者の機密情報の拡散経路の認識速度を向上させ作業効率を高めるには，機密情報の拡散経路を迅速に把握することが重要となる．そこで，機密情報の拡散経路を既存のログよりも迅速に把握可能にすることを 2 つ目の目的とする．

4-2-2 考え方

文献 3 で提案した機密情報の拡散追跡機能は，新たに機密情報が拡散したときの操作内容だけをログに出力する．このため，すべての機密情報に関する操作をログとして出力するように機能を追加する必要がある（3.4 節で後述する）．

また，機密情報の拡散に関わった操作に関するログを全て出力したとしても，計算機利用者が機密情報の利用状況を確認する際には，いつも機密情報の利用状況に関する全ての情報が必要になるわけではない．たとえば，ある特定の管理対象ファイルに関連する機密情報の利用状況のみが必要となる場合がある．また，ある特定の時刻における機密情報の利用状況のみが必要になる場合がある．このように，計算機利用者が確認したい情報が限られている際，全ての情報を表示した場合，計算機利用者は必要な情報を自身で探す必要がある．このため，問題点で述べたように解析の誤りと作業効率が低下が起こる可能性があり，計算機利用者が機密情報の拡散経路を正確かつ迅速に把握するのを阻害している．

そこで，可視化機能では，表示する情報をフィルタリングする機能を提供する．フィルタリング機能は，以下の 3 つに分けられる．

(1) 指定したファイルから拡散した機密情報の拡散経路のみを表示

(2) 指定したファイルに拡散した機密情報の拡散経路のみを表示

(3) 指定した期間の機密情報の拡散経路のみを表示

また，3 つのフィルタリング機能は，組み合わせて使うことも可能であり，指定したファイルから指定したファイル間の機密情報の拡散経路を表示したり，指定した時刻における指定したファイルからの機密情報の拡散経路を表示できる．

4-3 機密情報の拡散追跡機能における設計と実現方式

4-3-1 設計方針

目的を達成するために可視化機能に求められる 2 つの要件を以下に述べる．

（要件 1） 表示する情報に漏れと誤りがないこと
（要件 2） 表示する情報が簡潔であること
（要件 1）は，機密情報の拡散経路を正確に把握する上で必要になる．
（要件 2）は，機密情報の拡散経路を正確かつ迅速に把握する上で必要になる．

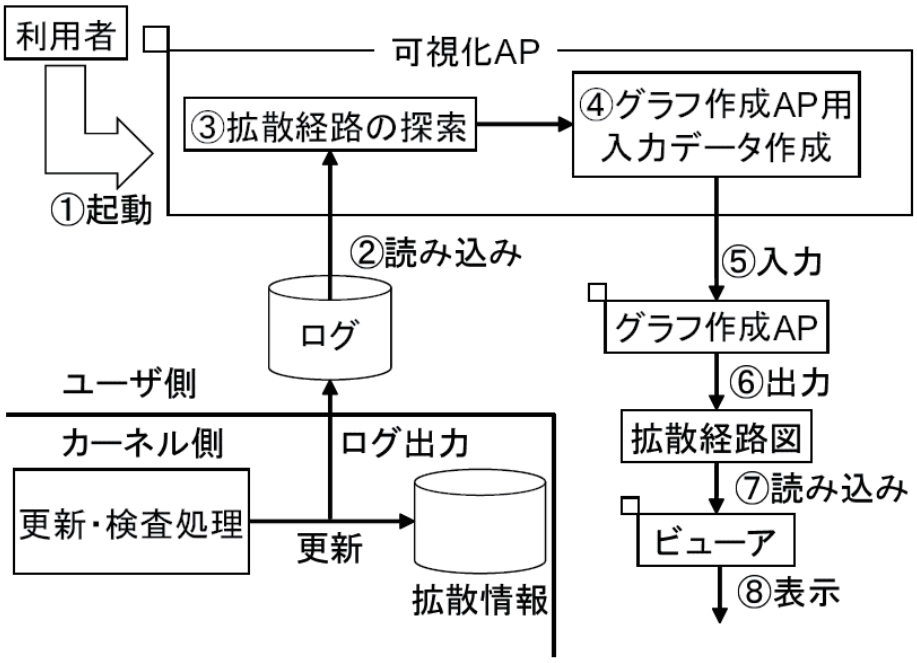

図６　可視化機能の基本機構

4-3-2 基本機構

図 6 に可視化機能の基本機構を示し，図中の番号に対応した処理の流れを以下に述べる．
(1) 利用者が可視化 AP を起動
(2) 可視化 AP がテキスト形式のログを読み込み
(3) 読み込んだログから機密情報の拡散経路を探索
(4) 探索した結果から，グラフ作成 AP への入力データを作成
(5) グラフ作成 AP へデータを入力
(6) グラフ作成 AP が拡散経路図を出力
(7) 利用者がビューアを起動し (6) で出力した図を読み込み
(8) 拡散経路図を利用者に表示

処理 (1) の際，利用者は可視化する対象のファイルや期間を指定できる．可視化機能は，処理 (3) の際，利用者が指定した対象に関する機密情報の拡散経路のみを探索する．また，各機密情報の拡散経路について，機密情報が拡散した時刻の順序関係を用いて，指定された期間に機密情報が拡散した経路か否かを判別する．これにより，表示する拡散経路をフィルタリングできる．

上記の処理のように，ログから機密情報の拡散経路を探索することにより，ログに漏れと誤りがない限り，（要件 1）を満たせる．また，拡散経路探索時に利用者が指定した管理対象に関する情報のみを探索して表示する．これにより，表示する情報が簡潔になるため，（要件 2）を満たせる．

グラフ作成 AP は，可視化 AP が出力したデータから機密情報の拡散経路を表す有向グラフを作成する．有向グラフは，表示する管理対象の数によって図の大きさや各ノードの配置を変える必要がある．このため，表示する管理対象の数に合わせて図のレイアウトを自動で調整できるフリーウェアのグラフ作成 AP である Graphviz を使用する．Graphviz は，DOT 言語と呼ばれる言語で記述されたテキスト形式のファイルを読み込むことで有向グラフを作成できる．

DOT 言語で記述されたファイルは，可視化 AP が出力する．具体的には，可視化 AP が機密情報の拡散追跡機能が出力するテキスト形式のログを読み込む．次に，リスト形式の構造体である管理対象ファイルに関す

る情報を格納する管理対象ファイルリストと管理対象プロセスに関する情報を格納する管理対象プロセスリストを作成する．各リストには，管理対象に機密情報を拡散したファイルまたはプロセスの情報が格納されている．その後，各リストを相互にたどることで機密情報の拡散経路を探索し，DOT 言語で記述されたグラフ作成用のファイルを出力する．

4-4 フィルタリングによる拡散経路図の複雑化防止の評価

4-4-1 目的と評価内容

提案機能は，可視化するログの量が増えるほど表示するノードの量が増加し，ノード間の依存関係が複雑になる．このような表示の複雑化を防止するため，提案機能は，表示する拡散経路のフィルタリング機能を提供する．そこで，取得したからログを提案機能によって全経路を可視化した拡散経路図と，提案機能のフィルタリング機能により特定のファイルに関する経路のみを可視化した拡散経路図に表示されるノード数を比較する．これにより，フィルタリング機能により，表示の複雑化をどの程度防止できるかを評価する．また，評価には約 250 行のログを用いた．このログは，機密情報の拡散追跡機能を実装した計算機を使用し，7 日間に渡って計算機内で機密情報を拡散させることで取得した．

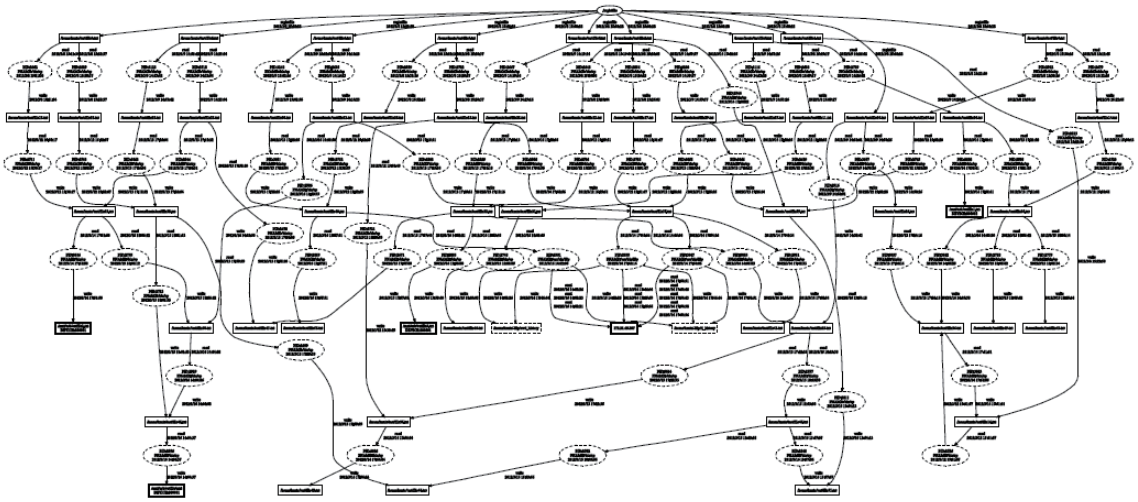

図 7　全経路を可視化した拡散経路図

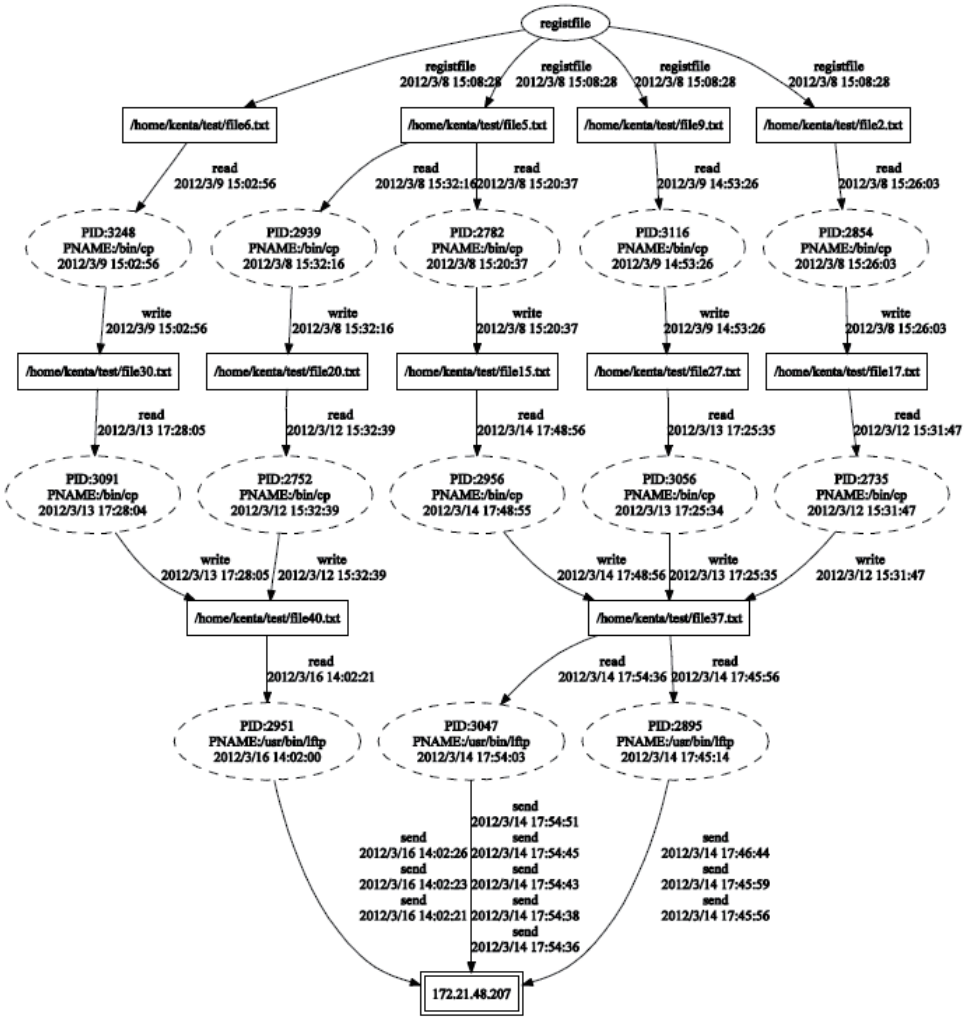

図８　特定経路のみの拡散経路図

4-4-2 評価結果と考察

全経路を可視化した拡散経路図を図 7 に，フィルタリング機能により特定のファイルへの拡散経路のみを可視化した拡散経路図を図 8 に示す．図 7 と図 8 から，ノード数は特定のファイルのみへの拡散経路にフィルタリングすることにより，129 ノードから 25 ノードに約 80%減少しており，表示の複雑化を防止できていることがわかる．

長期間にわたってログを収集した場合でも，提案機能は可視化する期間を指定して，特定のファイルに着目してフィルタリングした結果を可視化できる．このため，可視化する期間を指定するなどフィルタリングの粒度を細かくすることにより表示の複雑化は防止できると考える．

【参考文献】

[1] 日本ネットワークセキュリティ協会：2010 年度情報セキュリティインシデントに関する調査報告書 Ver.1.1, http://www.jnsa.org/result/incident/2010.html, 2011.

[2] Goel, A., Po, K., Farhadi, K., Li, Z., and Lara, D.E: The Taser Intrusion Recovery System, Proc. the 20th ACM Symposium on Operating Systems Principles (SOSP 2005), pp.163–176, 2005.

[3] 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.

〈発　表　資　料〉

題　　名	掲載誌・学会名等	発表年月
ログの改ざんと喪失を防止するシステムの仮想計算機モニタによる実現	情報処理学会論文誌	2012 年 2 月
可視化とフィルタリング機能により機密情報の拡散追跡を支援する機構の実現	情報処理学会論文誌	2012 年 9 月（掲載予定）
機密情報の拡散追跡機能の分散環境への対処法	電子情報通信学会 2012 年総合大会 情報・システム講演論文集 2	2012 年 3 月
VMBLS: Virtual Machine Based Logging Scheme for Prevention of Tampering and Loss	Lecture Notes in Computer Science	2011 年 8 月
機密情報の拡散経路を可視化する機能の提案	コンピュータセキュリティシンポジウム 2011（CSS2011）論文集	2011 年 10 月
仮想計算機モニタによるカーネルログ取得機能の実現と評価	情報処理学会研究報告	2011 年 7 月
仮想計算機モニタによるログの改ざんと喪失防止システムの提案と評価	情報処理学会 コンピュータセキュリティシンポジウム 2010(CSS2010) 論文集	2010 年 10 月