ファイルアクセスログの異常検知を活用した情報流出発見支援のための可視化手法の提案

矢野 智彦[1,a)]　伊藤 曜[2]　葛野 弘樹[3]　白石 善明[3]

概要：情報流出は，組織の運営に深刻な影響をおよぼす脅威であり，その発生件数は年々増加している．情報流出は外部の攻撃者だけでなく，正当なアクセス権限を持つ組織内部の関係者によっても引き起こされるため，検知が困難である．このため，コピーや移動などの操作を時系列で追跡し，直感的に把握できる可視化手法が求められている．既存研究では，ユーザの正常なファイルアクセスパターンを学習する異常検知手法や，ファイルの出所を分析する手法などが提案されているが，これらの多くは内部不正への対応が不十分であり，ファイル操作の可視化を前提としていない．また，可視化に焦点を当てた研究においても，情報流出の兆候を明確に把握するための視認性の向上が課題として残されている．本研究では，ファイルトラッキングに基づく可視化結果に，ファイルアクセスログの異常検知結果を補助的な指標として統合し，異常度の高い操作を強調表示することにより，調査の優先度が高い操作の直感的な把握を可能にする手法を提案する．提案手法を複数の内部不正シナリオに適用した評価実験により，異常な操作が可視化結果に適切に反映され，セキュリティ管理者による発見支援に有効であることを確認した．

キーワード：情報流出，内部不正，ファイルアクセスログ，可視化

A Visualization Method for Supporting the Discovery of Information Leakage Using Anomaly Detection in File Access Logs

Tomohiko Yano[1,a)]　Hikaru Ito[2]　Hiroki Kuzuno[3]　Shiraishi Yoshiaki[3]

Abstract: Information leakage is a major threat to the operation of organizations, and the number of incidents is increasing every year. Information leakage is difficult to detect because it can be caused not only by external attackers but also by insiders with legitimate access privileges. Existing studies have proposed anomaly detection methods that learn normal file access patterns of users and methods that analyze the provenance of files. However, most of these methods are insufficient in responding to insider threats and do not assume visualization. In addition, even in studies that focus on visualization, improving visibility for clear identification of information leakage signs remains challenging. This paper proposes a method that integrates file tracking-based visualization results with anomaly detection results from file access logs as a supplementary indicator, and highlights highly anomalous operations to enable efficient discovery of signs of information leakage. Evaluation of our proposed method in multiple insider threat scenarios confirmed that anomalous operations were appropriately reflected in the visualization results and that the method is effective in assisting security administrators to discover such operations.

Keywords: Information Leakage, Insider Threat, File Access Log, Visualization

1. はじめに

情報流出は組織にとって重大な脅威であり，情報資産を保護するための効果的な対策が必要である．情報流出は，運営の一時停止，損害賠償請求，社会的信頼の失墜など，さまざまな問題を引き起こす可能性がある．東京商工リサーチ [1] によると，2024 年度の事故件数は 189 件（前年度比 8.0%増）であり，年々増加している．

[1] セコム株式会社 IS 研究所
Intelligent Systems Laboratory, SECOM Co., Ltd.
[2] 横浜国立大学大学院環境情報学府
Graduate School of Environment and Information Sciences, Yokohama National University
[3] 神戸大学 大学院工学研究科 電気電子工学専攻
Dept. of EE, Kobe University
[a)] tomo-yano@secom.co.jp

情報流出の根本原因は多岐にわたる．IPA が公表している組織における情報セキュリティ 10 大脅威 [2] によると，「機密情報等を狙った標的型攻撃」，「内部不正による情報漏えい等」，「不注意による情報漏えい等」，様々な主体による情報流出が上位に挙げられている．また，IBM のレポート [3] によると，攻撃の経路は，フィッシング (15%) などの組織外部の人物による攻撃，悪意のあるインサイダー (7%) などの組織内部の人物による攻撃，偶発的なデータ損失 (6%) などの組織内部の人物の不注意によるデータ損失など多岐にわたる．

組織外部からの攻撃の場合は，様々な戦術を用いて目的となる機密情報にアクセスするのに対して，組織の従業員など組織内部の関係者による情報流出は，正当なアクセス権限を持つ人物により引き起こされるため，検知・発見が困難である．こうした人物による情報流出の効率的な発見には，ログの検知だけでなく，機密情報に対するコピーや移動などの操作を継続的に監視し，情報流出に至る一連の操作を時系列で追跡・把握する手法が必要である．そのため，操作の流れを直感的に把握可能な可視化手法が求められている．

既存研究では，ユーザの正常なファイルアクセスを学習することによる異常検知 [4], [5] や，ファイルの出所分析などのアプローチ [6], [7], [8] があるが，これらの手法の多くは正当な権限を持った人物からの情報流出には対応していないほか，ファイル操作の可視化を目的とした研究ではない．また，著者らはファイルトラッキングと可視化を行う手法を先行提案しており [9]，さらなる視認性の向上が依然として課題である．したがって既存の研究は以下のような課題がある．

課題: 内部不正を含む情報流出対策と可視化の視認性向上
組織内部の人物による攻撃も含めた情報流出を効果的に発見するためには，視認性に優れた可視化手法が求められている．しかしながら先行研究では，正当なアクセス権限を有する組織内部の人物による情報流出に十分に対応していない，ファイル操作の可視化を前提にしていないなどの課題がある．より視認性を向上させ，情報流出を容易に把握可能とするような可視化手法が求められる．

本研究では，著者らの先行研究 [9] を拡張し，ファイルアクセスログの異常検知結果を可視化に統合することで，情報流出の兆候の発見を支援する可視化手法を提案する．具体的には，追跡対象となる機密ファイルを起点としたファイルトラッキングに基づくネットワークグラフに，異常検知で算出した異常度を反映し，通常とは異なる操作を示す異常度の高い操作と，それを含むトラッキングパス (追跡対象の機密ファイルから現在までの一連の操作) を強調表示する．これにより，セキュリティ管理者は多数のノードとエッジを含む可視化結果の中から，調査の優先度が高い操作を直感的に把握可能である．

本研究の貢献は以下となる．

貢献 1. 異常検知を用いた可視化による情報流出発見支援:
本研究は，著者らの先行研究であるファイルトラッキング可視化手法を拡張し，ファイルアクセスログの異常検知で算出した異常度を反映することで，異常度の高い操作を含むトラッキングパスを強調表示する可視化手法を提案する．異常検知結果は，通常とは異なる操作を示す補助的な指標として機能し，セキュリティ管理者が多数のノードとエッジを含むネットワークグラフの中から，調査の優先度が高い操作を直感的に把握可能である．

貢献 2. 内部不正シナリオに基づく提案手法の有効性評価:
複数の内部不正シナリオに基づいてファイルの移動・コピーを行い，提案手法の有効性を評価した．評価結果では，いくつかの内部不正シナリオ中のファイル操作が，重要な操作として反映されることを確認した．

2. 背景知識

2.1 情報流出

情報流出とは，組織や個人が管理する機密情報や個人情報が，意図的または非意図的に信頼できない環境へ漏洩することを指す．情報流出は，その主体や意図性に基づいて以下のように分類される [10].

(1) 悪意のある組織外部の人物による流出
(2) 悪意のある組織内部の関係者による流出
(3) 情報流出の脅威を伴うランサムウェア
(4) 悪意のない組織内部の関係者による流出
(5) ハードウェアの紛失による流出

情報流出が企業・組織に与える影響は多岐にわたる．直接的な経済損失として，システム復旧費用，法的対応費用，被害者への賠償金が挙げられる．また，信頼失墜による顧客離れ，株価下落，事業機会の逸失といった間接的な損失もある．さらに個人への影響としては，プライバシーの侵害，なりすまし被害，経済的損失のリスクがある．

2.2 Windows イベントログ

Windows イベントログは，Microsoft Windows が提供するログ記録機能であり，システムの動作状況やセキュリティ関連の活動を詳細に記録する．システム管理者やセキュリティ担当者は，Windows イベントログを分析することにより，システムの状態や異常の兆候を把握可能である．

セキュリティ監視において，Windows セキュリティログは様々なサイバー攻撃の痕跡調査において重要な役割を果たす．各ログには，ログオン・ログオフ，権限の変更，ファイルアクセス，プロセスの作成などのセキュリティイベントに固有のイベント ID が割り当てられており，この ID により発生した事象の種類が特定可能である．

各ログには，発生日時，イベント ID，イベントの詳細説明のほか，イベント ID に応じた様々な情報が含まれる．オブジェクト（ファイル・フォルダ）へのアクセス試行を示すイベント ID 4663 のイベントログには，アクセスを試行したアカウント名，オブジェクト名，プロセス名，プロセス ID，アクセスマスク（読み込み・書き込みなどの操作種別）などが含まれる．

3. 想定環境とシナリオ

提案手法が適用される環境と想定シナリオを図 1 に示す．

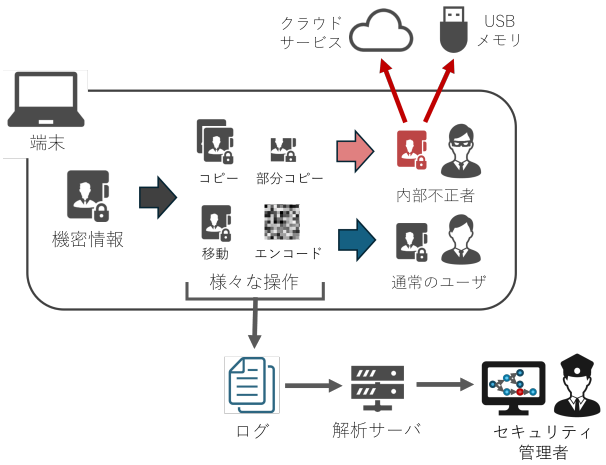


図 1: 想定環境とシナリオ

3.1 想定環境

本研究では，端末およびユーザが組織によって管理されている環境を想定する．端末には機密ファイルが保存されており，Windows イベントログを収集し解析サーバへ送信する機能が搭載されている．また，クリップボードの操作（コピー・ペースト）を監視し，操作時刻および操作を行ったプロセス情報を記録・送信する機能を備える．機密ファイルは，セキュリティ管理者が手動で指定した業務上重要な文書ファイルであり，.txt，.xlsx，その他の一般的なファイル形式が含まれる．解析サーバは，収集したログをもとに提案手法を実行する．

機密ファイルへのアクセスが許可されたユーザは，業務上の正当な目的でファイルを移動またはコピーすることがある．このとき，ファイル全体だけでなく一部の内容をコピーする可能性もある．なお，リモートからダウンロードされたファイルを追跡する必要がある場合には，ファイルの保存場所に応じた追加の実装が必要となる．

3.2 想定シナリオ

本研究では，組織内部の関係者が機密ファイルを移動し，組織の管理外に流出させるシナリオを想定する．このとき，情報流出を行う主体は，流出対象の機密ファイルに対して正当なアクセス権限を有しているものとする．

想定シナリオは，MITRE ATT&CK Enterprise Matrix[11] および NIST SP 1800-29[10] に基づき設計した．情報流出の流れとしては，まずユーザが端末に保存された機密ファイルにアクセスする．次に，ファイルの移動・コピー，クリップボードへのコピー・貼り付け，ファイルの暗号化やエンコードなどの操作を通じてファイルを変換する．最後に，変換されたファイルをクラウドサービスへのアップロードや USB ドライブへのコピーなどの手段によって外部に流出させる．

4. 提案手法

4.1 要件

提案手法は，セキュリティ管理者がファイル操作の流れの中から情報流出の兆候を効率的に把握できるよう支援することを目的とする．特に，異常検知結果を補助的な情報として可視化に統合することで，対応優先度の判断を支援する．要件は以下のとおりである．

要件 1:

提案手法は，Windows イベントログおよびクリップボード操作ログを入力とし，ファイル操作の流れを示すネットワークグラフを構築・可視化する．可視化するネットワークグラフは，ファイル・プロセス・クリップボードをノード，読み込み・書き込み・コピー・貼り付けなどのイベントをエッジとして表現する．

要件 2:

異常検知手法は，通常とは異なる操作に対して高い異常度が付与されるようにスコアを算出する．算出された異常度は，ネットワークグラフ上で操作の強調表示に用いられ，セキュリティ管理者が注目すべき操作を視覚的に把握できるようにする．異常検知は，すべての不正操作を網羅的に検出するものではなく，可視化結果の補助的な指標として機能する．

4.2 提案手法の概要

図 2 に提案手法の概要を示す．提案する手法では，イベントログを入力として，不要な操作を除去したネットワークグラフを構築し，異常度の高い操作を強調表示する．

ネットワークグラフの構築は，ファイル操作の流れを表現する従来手法 [9] を基盤とする．異常検知結果は，操作の重要度を示す補助的な指標としてグラフに統合される．異常度が高い操作は視覚的に強調され，セキュリティ管理者が注目すべき操作を効率的に把握できるよう支援する．

4.3 収集するイベントログ

収集するイベントログ l は以下の情報が含まれているものとする．

- タイムスタンプ：$l.timestamp$
- イベントタイプ：$l.type \in \{ReadEvent, WriteEvent, CopyEvent, PasteEvent, ProcessCreateEvent\}$
- アクセスマスク：$l.amask$
- ユーザ名：$l.user$
- ファイル名：$l.fname$ ($l.type \in \{ReadEvent, WriteEvent\}$)
- プロセス：$l.pid$
- 親プロセス：$l.ppid$ ($l.type \in \{ProcessCreateEvent\}$)

Windows イベントログの場合，$ReadEvent, WriteEvent$ はイベント ID4663，$ProcessCreateEvent$ はイベント ID4688 で取得可能である．また $CopyEvent, PasteEvent$ はクリップボードの情報を読み取るソフトウェアを使用する．

4.4 ネットワークグラフの定義

ネットワークグラフは $G = (V, E)$ で表現される有向グラフである．ここで V は有限ノード v_i の集合，$E \in V \times V$ は有限エッジ $e_{ij} = (v_i; v_j)$ の集合である．

ノード V は，ファイル $f_i \in F$，プロセス $p_i \in P$，クリップボード $c_i \in C$ の集合である．各ノードには以下の情報が

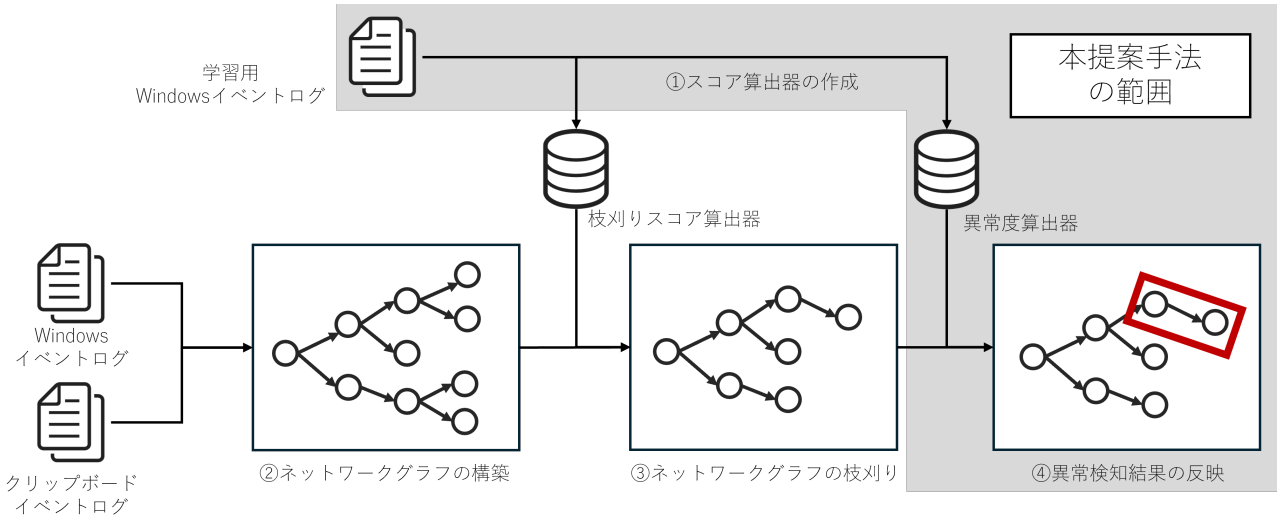


図 2: 提案手法の概要

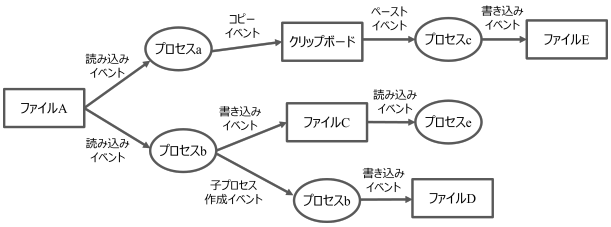


図 3: ネットワークグラフの例

含まれる．

- f_i: ID id，ファイル名 $fname$，アクティブフラグ $active$
- p_i: ID id，プロセス名 $pname$，プロセス IDpid，アクティブフラグ $active$
- c_i: ID id，プロセス ID pid，アクティブフラグ $active$

ここでアクティブフラグはファイル，プロセス，クリップボードで以下の値を取る変数である．

- $f_i.active$: ファイルが存在する場合 1，削除された場合は 0
- $p_i.active$: プロセス実行中の場合 1，プロセス終了済みの場合 0
- $c_i.active$: ノード生成時 1，別のプロセスからの値がコピーされた場合 0

アクティブフラグは 1 から 0 へのみ遷移する．なお，一部の p_i については時間経過によりアクティブフラグが 0 に切り替わる．対象のプロセスについてはプロセスの稼働時間などに基づき経験的に決定する．

エッジ E は以下のイベントリストの集合である．

- 読み込みイベント $read_{f_i p_j}$：プロセス p_j によるファイル f_i の読み込み
- 書き込みイベント $write_{p_i f_j}$：プロセス p_i によるファイル f_j の書き込み
- コピーイベント $copy_{p_i c_j}$：プロセス p_i からクリップボード c_j へのコピー
- 貼り付けイベント $paste_{c_i p_j}$：クリップボード c_i からプロセス p_j への貼り付け
- 子プロセス作成イベント $create_{p_i p_j}$：プロセス p_i による子プロセス p_j の作成

ネットワークグラフの例を図 3 に示す．例えばファイル A からファイル E への操作は，プロセス a によるファイル A の読み込みイベント，プロセス a からのコピーイベント，プロセス c への貼り付けイベント，プロセス c によるファイル E への書き込みイベントが発生している．

4.5 提案手法の詳細

4.5.1 スコア算出器の作成

提案手法では 2 つのスコア算出器を作成する．それぞれのスコア算出器の役割は以下である．

- 枝刈りスコア算出器: バックグラウンドで頻繁に発生する操作を除外するために使用する．手動によるファイル操作と比較して頻度が高いイベントを対象とし，プロセスごとにアクセス頻度の高いディレクトリ，サブディレクトリ，拡張子を学習することにより，該当する操作に高いスコアを付与する．
- 異常度算出器: 通常とは異なる操作を強調表示するために使用する．ファイルアクセスログから特徴量抽出および学習を行い，通常とは異なるファイルアクセスが生じた場合に高いスコアを付与する．

枝刈りスコア算出器

枝刈りスコア算出器は，過去の読み込み・書き込みイベントに基づき，バックグラウンドで頻発する操作を除外するためのスコアを計算する．学習用 Windows イベントログからファイルの読み込み，書き込みに関するログを抽出した後，各プロセス p_i について，ファイルパスからディレクトリパス dir，サブディレクトリパス $pdir$，拡張子 ext を取得し，それぞれの読み込み頻度および書き込み頻度を計算する．これらの頻度情報に基づき，枝刈りスコア算出器は式 1 によりスコアを計算する．

$$\begin{aligned} score_c(l) = {} & w_{dir} H_{dir}(l.pid, l.type, dir) \\ & + w_{pdir} H_{pdir}(l.pid, l.type, pdir) \\ & + w_{ext} H_{ext}(l.pid, l.type, ext) \end{aligned} \tag{1}$$

表 1: イベント処理条件と処理内容

イベント	条件	処理
読み込み	$l.type$ = ReadEvent $f_i = l.fname$ $f_i.active$ = 1 を満たす $f_i \in F$ が存在	・メタデータ $(id, l.pid, 1)$ を持つプロセス p をプロセスリストに追加 ・メタデータ $(f_i, p, score_c, score_a)$ を持つイベントを読み込みイベントリストに追加
書き込み	$l.type$ = WriteEvent $p_i = l.pid$ $p_i.active$ = 1 を満たす $p_i \in P$ が存在	・メタデータ $(id, l.fname, 1)$ を持つファイル f をファイルリストに追加 ・メタデータ $(p_i, f, score_c, score_a)$ を持つイベントを書き込みイベントリストに追加
コピー	$l.type$ = CopyEvent $p_i = l.pid$ $p_i.active$ = 1 を満たす $p_i \in P$ が存在	・メタデータ $(id, 1)$ を持つクリップボード c をクリップボードリストに追加 ・メタデータ (p_i, c) を持つイベントをコピーイベントリストに追加
貼り付け	$l.type$ = PasteEvent $c_i.active$ = 1 を満たす $c_i \in C$ が存在	・メタデータ $(id, l.pid, 1)$ を持つプロセス p をプロセスリストに追加 ・メタデータ (c, p) を持つイベントを貼り付けイベントリストに追加
子プロセス作成	$l.type$ = ProcessCreateEvent $p_i.pidc = l.pid$ $p_i.active$ = 1 を満たす $p_i \in P$ が存在	・メタデータ $(id, l.ppid, 1)$ を持つプロセス pp をプロセスリストに追加 ・メタデータ (p, pp) を持つイベントを子プロセス作成イベントリストに追加

ここで，w_{dir}，w_{pdir}，w_{ext} は重み，$l.dir$，$l.pdir$，$l.ext$ は $l.fname$ から抽出したディレクトリパス，親ディレクトリパス，拡張子である．また，H_{dir}，H_{pdir}，H_{ext} は，プロセス ID，操作種別（読み込み・書き込み），およびディレクトリパス・親ディレクトリパス・拡張子を入力とし，学習用 Windows イベントログから算出された頻度情報を返す関数である．

異常度算出器

異常度算出器では特徴量抽出および異常度の算出を行う．特徴量抽出はユーザ名，ファイル名，アクセスマスク，プロセス名を抽出し，One-Hot エンコーディングを適用する (関数 $OneHotEncoder$)．ただし，ファイル名は主要な親ディレクトリ名 (C:\Users\(ユーザ名)\Documents など) を抽出したパスを使用する．また，プロセス名は頻度の多いプロセス名のみを保持し，その他を「Other」とする．異常度の算出は抽出した特徴量から Local Outlier Factor (LOF) などの異常検出手法を用いて作成する (関数 $AnomalyDetector$)．最終的に異常度 $score_a$ は式 2 により計算する．

$$score_a(l) = AnomalyDetector(OneHotEncoder(l.user, l.fname, l.amask, a.pid)) \quad (2)$$

4.5.2 ネットワークグラフの構築

Windows イベントログおよびクリップボードイベントログからネットワークグラフを構築する．ネットワークグラフの構築は，Windows イベントログを時系列順に読み込み，特定の条件下でノードとエッジを追加する．各イベントにおけるログ l の条件および，処理内容を表 1 に示す．ただし，$score_c$ は式 1，$score_a$ は式 2 を使用して計算する．

4.5.3 ネットワークグラフの枝刈り

構築したネットワークグラフから，枝刈りスコア算出器を用いた枝刈りを行う．まず $score_c \leq TH_c$ となるようなエッジに接続したノード v_{TH_c} をすべて抽出する．ここで TH_c は閾値であり，あらかじめ設定する値である．次に，v_0 から v_{TH_c} へのトラッキングパスを，ダイクストラ法にて計算する．ここでエッジの重みは $score_c$ を用いる．最終的にトラッキングパスに含まれるすべてのノードとエッジ以外をすべて枝刈りする．

4.5.4 異常検知および可視化結果への反映

枝刈りしたネットワークグラフから，異常検知結果を反映する．まず異常度 $score_a \geq TH_a$ となるようなエッジをすべて抽出する．ここで TH_a は閾値であり，あらかじめ設定する値である．次に抽出したエッジの描画条件を変更することにより，可視化結果を強調する．

5. 評価

提案手法を実装した後，3 つの情報流出シナリオの下でファイルを移動およびコピーした際のネットワークグラフの結果を観察することにより評価を行った．

5.1 実験環境

実験には Intel(R) Core(TM) i7-1255U (1.70 GHz, x86_64) CPU と 32 GB DDR4 RAM，Windows 11 pro をインストールしたホスト PC および，ホスト PC 上に Windows 11 Enterprise（評価版）をインストールした仮想マシンを導入した．ホスト PC では，収集したログの解析および可視化，仮想 PC では解析対象のログの生成と収集を行った．

仮想マシンでは，Taro ユーザの作成，ファイルやディレクトリの作成，グループポリシーの設定，学習用・評価用データの生成を行った．ファイルやディレクトリは業務環境を想定し，様々なファイル・ディレクトリの作成，ソフトウェアのインストールを行った．また追跡対象のファイルとして C:\Users\Taro\Documents\list 以下に customerlist.ods を作成した．また監査ポリシーの設定として，ファイルの共有，ファイルシステム，リムーバブル記憶域の監査（成功・失敗）を有効にした．ホスト PC では，ネットワークグラフの構築および可視化を行う Python コードを実装し，動作させた．

5.2 学習・評価用データの作成

学習・評価用データとなる Windows イベントログを収集するために，正常タスクおよび内部不正タスクを定義して実行した．正常タスクとしてファイルへのアクセス・編集・保存，コードの作成と実行，メールのやりとり，Web ブラウジング，アプリの実行などのタスクを定義し，ラン

ダムで実行した．またバックグラウンドで動作させるタスクとして，スケジューラの実行，ウイルススキャン，自動バックアップの実行を行った．

内部不正タスクとして以下に示す 3 つのシナリオを定義した．一部の操作を除き，C:\Users\Taro 以下で操作を実施した．

シナリオ 1

シナリオ 1 は，追跡対象ファイルのコピーおよび Base64 でのエンコードを行い，USB メモリにコピーして情報流出を行った．

(1) Documents\list\customerlist.ods を LibreOffice (soffice.bin) で開き，別名 (Documents\list\list.ods) で保存する．
(2) certutil.exe を用いてデータを Base64 エンコードし，Documents\list\enc_list として保存する．
(3) explorer.exe を用いて USB メモリ (\Device\HarddiskVolume7\customer\enc_list) へ移動する．

シナリオ 2

シナリオ 2 は，追跡対象ファイルの一部をクリップボード経由で別ファイルにコピーし，メールに添付することにより情報流出を行った．

(1) Documents\list\customerlist.ods を soffice.bin で開き, クリップボードにコピーする．
(2) Documents\Personal\Certification_Study\list.txt を notepad.exe で開き，クリップボードの内容をペーストして保存する．
(3) Outlook を開き list.txt をメールに添付して送信する．このとき，Outlook が作成したプロセス msedgewebview2.exe による list.txt の読み込みが発生する．

シナリオ 3

シナリオ 3 は，追跡対象ファイルが含まれたバックアップファイルからファイルを別名でコピーし，Google Drive にアップロードすることにより情報流出を行った．

(1) Documents\list\customerlist.ods が含まれた Backups\backup_204\9\09.zip を作成する．この処理はバックグラウンドで作成される．
(2) Backups\backup_204\9\09.zip 内の customerlist.ods を explorer.exe を用いて Desktop\tmp に移動する．
(3) Desktop\tmp\customerlist.ods を soffice.bin で開き，Desktop\tmp\c.ods として保存する．
(4) Microsoft Edge (msedge.exe) を開き, Desktop\tmp\c.ods を Google Drive にアップロードする．このとき，msedge.exe による c.ods の読み込みが発生する．

学習用データとして 2 時間，評価用データとして 1 時間正常タスクを実施し続けることにより Windows イベントログを生成・収集した．また評価用データ作成中に上記 3 つのシナリオを実行した．最終的に学習用データとして 42,677 件，評価用データとして 24,500 件のファイルアクセスログが得られた．評価用データ中の内部不正シナリオに関するログは，シナリオ 1 が 7 件，シナリオ 2 が 4 件，シナリオ 3 が 7 件の計 18 件であった．

表 2: 異常度上位 N 件 (Top-N) に含まれる各シナリオに関するログ数 (全ログ件数=24,500)

異常検知手法	Top-N	シナリオ 1	シナリオ 2	シナリオ 3
One-Class SVM	20	1/7	0/4	0/7
	200	4/7	3/4	2/7
	2,000	5/7	4/4	2/7
	10,000	5/7	4/4	2/7
Isolation Forest	20	0/7	0/4	0/7
	200	1/7	0/4	0/7
	2,000	1/7	0/4	2/7
	10,000	4/7	3/4	7/7
Local Outlier Factor	20	2/7	0/4	1/7
	200	7/7	4/4	7/7
	2,000	7/7	4/4	7/7
	10,000	7/7	4/4	7/7

5.3 評価内容

実験では以下の項目を評価した．

(1) 異常検知結果: 全ファイルアクセスログの異常度を降順に並べ，内部不正シナリオに関連するログの検出順位を評価する．
(2) 可視化結果: 異常度を反映したネットワークグラフが，視認性や調査対象の絞り込みに寄与する効果を評価する．

5.4 評価結果

5.4.1 異常検知結果

異常検知結果を表 2 に示す．表は異なる異常検知手法（One-Class SVM，Isolation Forest，LOF）を適用した場合の結果を示している．各手法について，異常度上位 20，200，2,000，10,000 件に含まれる，各内部不正シナリオに関連するログの件数を示している．例えば LOF の場合，24,500 件のログの異常度上位 20 件の中に，シナリオ 1 に関係するログが 2 件含まれている．

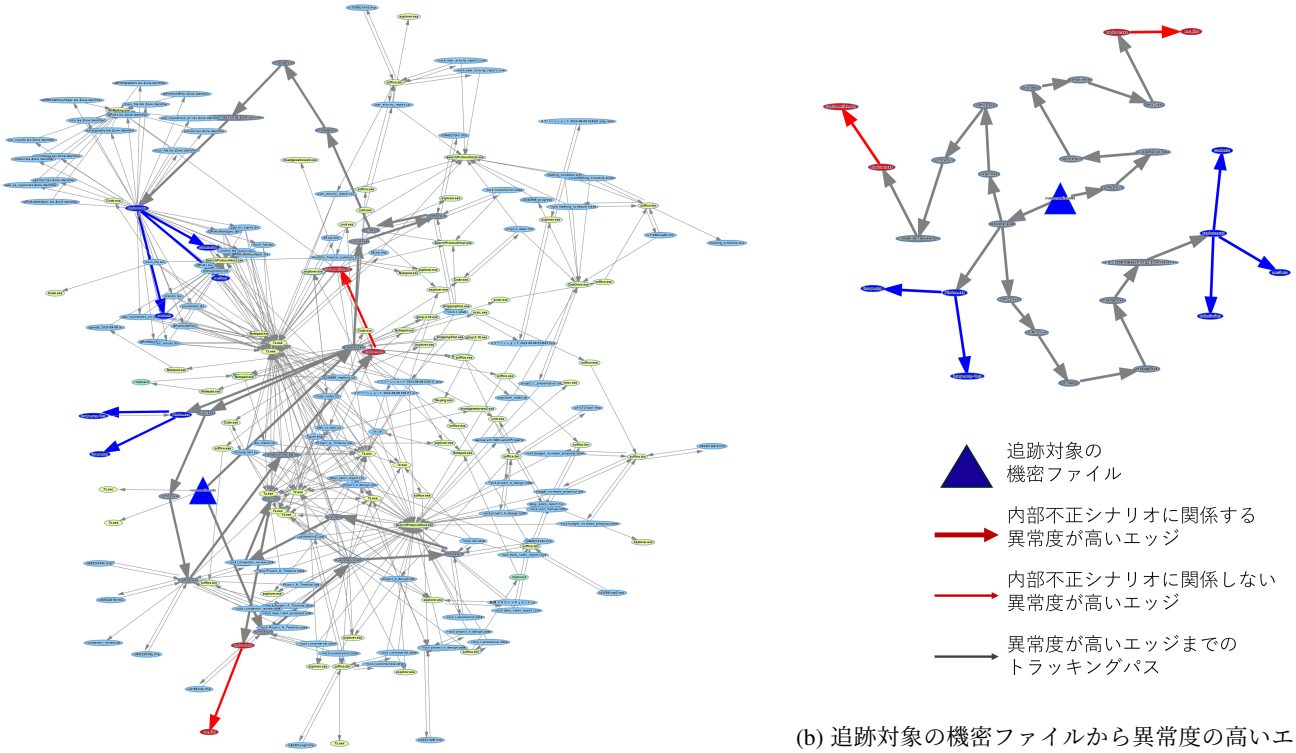
5.4.2 可視化結果

可視化結果を図 4 に示す．図 4a は LOF を適用した場合の枝刈り後のネットワークグラフであり，ノード数 216 個，エッジ数 529 個であった．また，追跡対象の機密ファイルから異常度の高いエッジまでのトラッキングパスのみ残したネットワークグラフを図 4b に示す．図中では，青い三角が追跡対象ファイル，赤および青の太線が異常度の高いエッジ，灰色の太線が追跡対象ファイルから異常度が高いエッジまでのトラッキングパスを表している．赤い太線は内部不正シナリオに関係するエッジ (2 個)，青い太線は実際には内部不正シナリオに関係しないエッジ (5 個) であった．また灰色のエッジは 21 個であった．

6. 考察

6.1 異常検知結果に対する考察

表 2 に示す LOF の Top-20 では，異常度上位 20 件中にシナリオ 1 に関するログが 2 件，シナリオ 3 に関するログが 1 件含まれていた．したがって，セキュリティ管理者は異常度上位のログを見ることにより，効率よく内部不正を発見できる可能性がある．また上位 200 件中には内部不正に関するログがすべて含まれていたが，これは同一の異常度



(a) 枝刈り後のネットワークグラフ

(b) 追跡対象の機密ファイルから異常度の高いエッジまでのトラッキングパスのみ残したネットワークグラフ

図 4: 可視化結果

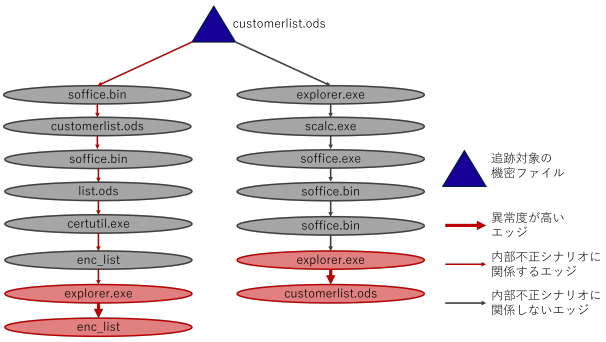


図 5: 内部不正シナリオに関係する異常度が高いエッジまでのトラッキングパス

を示すログが多数存在したことに起因する．そのため，今後は特徴量抽出および異常検知手法の改善が必要である．

6.2 可視化結果に対する考察

可視化結果は枝刈り後であっても，ノード数 216 個，エッジ数 529 個と多く，一目で全体を把握するのは困難である．一方で異常度が高いエッジは 7 個，トラッキングパスを含めても 28 個であり，そのうち 2 個は実際に内部不正シナリオに関連していた．したがって，セキュリティ管理者は太線で強調されたログを確認することで，効率的かつ直感的に注目するべき箇所を特定できると考えている．

6.3 トラッキングパスに対する考察

内部不正シナリオに関係する異常度が高い 2 つのエッジ (赤い太線で強調されたエッジ) までのトラッキングパスを図 5 に示す．図 5 左側では，soffice.bin による customerlist.ods の読み込みと書き込み，soffice.bin による customerlist.ods の読み込みと list.ods の書き込み，certutil.exe による list.ods の読み込みと enc_list の書き込み，explorer.exe による customerlist.ods の読み込みと enc_list への書き込みが含まれる．これらの多くはシナリオ 1 に関連するログである．また図 5 右側では，explorer.exe による customerlist.ods の読み込み，scalc.exe，soffice.exe，soffice.bin の子プロセス作成，soffice.bin による competitor_review.odt の書き込み，explorer.exe による competitor_review.odt の読み込みと customerlist.ods の書き込みが行われている．最後の操作はシナリオ 3 に関連するが，途中のパスは内部不正シナリオと無関係である．以上から，シナリオ 1 に関してはトラッキングパスに推定がほぼできているのに対して，シナリオ 3 に関してはトラッキングパスの推定に失敗している．トラッキングパスの推定はアクセス頻度だけではなく，タイムスタンプや他のファイルやプロセスのメタ情報などを用いて推定する必要があり，今後の検討課題である．

7. 関連研究

7.1 正常なユーザのプロファイリング

内部不正の検出として，異常検知を基盤とした検出手法が多数提案されている．異常検知ベースの手法では，ファイルアクセス履歴などの操作ログを用いて，ユーザの正当な行動パターンをプロファイルとして学習し，学習された

プロファイルとの相違度を計算し，異常な行動を検出する [4], [5].

7.2 ファイルの出所分析

ファイルの出所分析は，サイバー攻撃の調査に用いられる手法の一つであり，ログやシステムコールなどの情報をもとに，ファイルなどの起源と使用履歴を特定する．関連する研究として，イベントループ分析による攻撃原因究明手法 [6]，過去のイベント頻度を用いて脅威アラートをランク付けし，ログを効率的に削減する手法 [7] や，ATT&CK 技術との相関関係を活用する手法 [8] などが提案されている．これらの研究では，脅威アラートを起点として関連するイベントを効率よく抽出することを目的としている．本研究では，セキュリティ管理者が定義した機密ファイルを起点とし，正当な操作を含む一連の操作を可視化することに重点を置いている．提案手法では注目するべき操作を決定するために異常検知手法を用いることにより，視認性の向上を図っている．

7.3 ログの可視化

情報流出対策において，不審な操作の発見を支援するためにログを可視化することが重要である．特にネットワークグラフ型の可視化は，イベント間の関係性を視覚的に把握するために有効である．代表的なツールとして，ログオンイベントの可視化ツール [12]，タイムライン解析ツール [13]，認証ログの可視化と異常検知を組み合わせた手法 [14] などがある．これらのツールは，対象とするイベントの種類に応じて設計されており，可視化の目的や対象に応じて異なるアプローチが採用されている．

8. おわりに

本研究では，ファイルアクセスログの異常検知結果を可視化に統合し，情報流出の兆候を効率的に発見する可視化手法を提案した．提案手法は，従来のファイルトラッキングの可視化に異常検知結果を補助的に組み込むことで，セキュリティ管理者が注目すべき操作を視覚的に把握しやすくすることを目的としている．

評価実験では，複数の内部不正シナリオに対して提案手法を適用し，通常とは異なる操作の可視化結果への適切な反映を確認した．特に，異常度の高い操作の強調表示により，重要な操作を効率的に抽出できることを示した．

今後は，異常検知結果が可視化に反映されないケースへの対応を強化するため，特徴量抽出や異常検知手法の改善を進める．また，トラッキングパスの推定精度を向上させるため，アクセス頻度に加えて時系列情報やファイル・プロセスのメタ情報を活用した推定手法の導入を検討する．

参考文献

[1] 株式会社東京商工リサーチ, “2024 年「上場企業の個人情報漏えい・紛失事故」調査,” https://www.tsr-net.co.jp/data/detail/1200872_1527.html, , accessed Jul. 30. 2025.

[2] 独立行政法人 情報処理推進機構, “情報セキュリティ 10 大脅威 2025,” https://www.ipa.go.jp/security/10threats/10threats2025.html, accessed Jul. 30. 2025.

[3] International Business Machines Corporation, “Cost of a Data Breach Report 2025,” https://www.ibm.com/reports/data-breach, accessed Jul. 30. 2025.

[4] S. Mehnaz and E. Bertino, “Ghostbuster: A fine-grained approach for anomaly detection in file system accesses,” Proc. Seventh ACM on Conference on Data and Application Security and Privacy, CODASPY ’17, New York, NY, USA, pp.3–14, Association for Computing Machinery, 2017, DOI:10.1145/3029806.3029809.

[5] F. Toffalini, I. Homoliak, A. Harilal, A. Binder, and M. Ochoa, “Detection of masqueraders based on graph partitioning of file system access events,” 2018 IEEE Security Privacy Workshops (SPW), pp.217–227, 2018, DOI:10.1109/SPW.2018.00037.

[6] K.H. Lee, X. Zhang, and D. Xu, “High accuracy attack provenance via binary-based execution partition,” Network and Distributed System Security (NDSS) Symposium, 2013.

[7] W.U. Hassan, S. Guo, D. Li, Z. Chen, K. Jee, Z. Li, and A. Bates, “Nodoze: Combatting threat alert fatigue with automated provenance triage,” Network and Distributed System Security (NDSS) Symposium, 2019, DOI:10.14722/ndss.2019.23349

[8] W.U. Hassan, A. Bates, and D. Marino, “Tactical provenance analysis for endpoint detection and response systems,” 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, pp.1172–1189, 2020, DOI:10.1109/SP40000.2020.00096.

[9] T. Yano, H. Kuzuno and K. Magata, “File Tracking and Visualization Methods Using a Network Graph to Prevent Information Leakage,” in IEICE TRANSACTIONS on Information, vol. E106-D, no. 9, pp. 1339-1353, 2023, DOI: 10.1587/transinf.2022ICP0014.

[10] W. Fisher, R. Eugene Craft, M. Ekstrom, J. Sexton and J. Sweetnam, “Data Confidentiality: Detect, Respond to, and Recover from Data Breaches,” NIST SPECIAL PUBLICATION 1800-29, 2024.

[11] MITRE Corporation, “MITRE ATT&CK Enterprise Matrix,” https://attack.mitre.org/matrices/enterprise/, accessed Jul. 30. 2025.

[12] JPCERT Coordination Center, “LogonTracer: Investigate malicious Windows logon by visualizing and analyzing Windows event log,” https://github.com/JPCERTCC/LogonTracer, accessed Jul. 30. 2025.

[13] J. Berggren, “Timesketch: Collaborative forensic timeline analysis,” https://github.com/google/timesketch, 2019, accessed Oct. 14. 2022.

[14] H. Studiawan, C. Payne, and F. Sohel, “Graph clustering and anomaly detection of access control log for forensic purposes,” Digital Investigation, vol.21, pp.76–87, 2017, DOI:10.1016/j.diin.2017.05.001.