ログ解析支援のための文書類似度に基づくイベント ID 割り当て手法の提案

高山 立磨[1,*] 中野 心太[2] 関谷 信吾[2] 折田 彰[3] 岸本 頼紀[4]
早稲田 篤志[4] 花田 真樹[4]

概要：ログ調査において，Windows などのようにイベントに ID が割り当てられていれば，ID の並び順に基づく調査やパターン分析が適用できるが，Linux のログなど ID が振られていない場合はイベント出現順の視認性が悪くなったり機械的な解析に手間がかかる．これに対して Windows のイベントと同様のメッセージは同じ ID を割り当てる方が解析の支援になると考え，事前学習済みモデルである BERT モデルに適切な Linux ログと Windows ログのペアデータセットでファインチューニングを行い，文書類似度を基準として Windows と同等の場合は同じ ID を割り当てる手法とその効果について論じる．

キーワード：セキュリティ，イベント ID, AI

A Proposal of an Event ID Assignment Method based on Document Similarity for Log Analysis

Ryuma Takayama[1] Shinta Nakano[2] Shingo Sekiya[2] Akira Orita[3]
Yorinori Kishimoto[4] Atsushi Waseda[4] Masaki Hanada[4]

Abstract: In log analysis, while systems like Windows assign unique IDs to events, enabling investigation and pattern analysis based on these IDs, Linux logs lack this feature. This can lead to poor visibility of event sequences and a cumbersome mechanical analysis process.
To address this, this thesis proposes a method to assign the same event ID to similar Linux and Windows logs, thereby aiding in analysis. We use a pre-trained BERT model, which is fine-tuned on an appropriate dataset of Linux and Windows log pairs. We will discuss this approach and its effectiveness, focusing on using document similarity as a criterion for assigning Windows event IDs to corresponding Linux logs.

Keywords: Security, EventID , AI

1. はじめに

デジタルフォレンジックでは，ログを分析しサイバー攻撃の発生日時や攻撃の手法について調査する．近年業務システムでは Windows が主流になっていることもあり，Wiindows のイベント ID を基準にした調査支援システムが提案されている[1].

しかし，社内システムでも Linux を使用している場合も多い．しかし，Windows のログにはイベント ID が割り当てられていることに対して，Linux のログではイベント ID が割り当てられていない．フォレンジック解析支援システムでは，Windows を対象としイベント ID を基準とするものも少なくない[1].

そこで，Linux やイベント ID が割り当てられたいないログのイベントログに既存のシステムを適用するために，イベント ID の自動割り当てを提案する．

イベント ID が割り当てられていないものに, Windows と同等のイベント ID を割り当てることができれば，既存システムへの適用ができ，様々なログの解析支援となる．

本論文では，LLM を用いたイベント ID が割り当てられていないイベントへのイベント ID 自動割り当て手法について提案し，実際の例に適用した結果について論じる．

2. 関連研究

“A Machine Learning Approach for RDP-based Lateral Movement Detection” [2]では，サイバー攻撃におけるラテラルムーブメント（内部での横展開）を検知するため，Windows イベントログを分析し，LogitBoost という機械学

1 東京情報大学大学院 総合情報学研究科
Graduate School od Informatics. Tokyo University of Information Sciences
2 株式会社日立システムズセキュリティ技術 R&T センタ
Hitachi Systems, Ltd.Security Technology R&T Center
3 株式会社日立システムズ セキュリティリスクマネジメント本部
Hitachi Systems, Ltd.Security Risk Mangement Division.
4 東京情報大学 総合情報学部
Faculty of Informatics, Tokyo University of Information Sciences

習モデルを適用する手法が提案されているこれはWindowsイベントログからIDを検索キーとして利用し，ID 4624（ログオン）と4634（ログオフ）から「RDPセッション」を構築し，その継続時間といった特徴量の分析を行い，このアプローチにより，既存手法を上回る高い検知精度と，未知の脅威に対する頑健性を実現している．

また，”機械学習を用いた異常ログ可視化のための誤検知された正常ログ対策の検討”[3]では，サイバー攻撃のタイムライン分析において，イベントIDの類似度を用いて異常ログを可視化する手法が提案されている．

両システムは一定の成果はあるものの，適用できるのは，基本的にログにイベントIDが付与されているWindowsのイベントログのみであり，Linuxなどの IDを持たないログへの適用ができない．これに対して，本手法を適用することでIDを持たないイベントログに関しても上記の手法が適用できると考えられる．

3. 考え方

3.1 提案手法

本研究ではイベントIDを割り当てられたログとしてWindowsのイベントログを，イベントIDを割り当てられていないログとしてLinuxのログを対象とする．イベントIDの割り当てのために，イベントの類似としてイベントメッセージに着目する．イベントメッセージでは対象となるイベントに固有のメッセージが記録される．この為，類似をtfidfのような単語出現による類似度で分類し，類似度が高いイベントのIDを割り当てることも考えられる．しかし，Windowsのイベントログでは異なるメッセージでも同一のイベントIDが割り当てられる場合がある．このため単純な一致では判別ができない．

そこでLLMを用いた文書類似度に着目する．LLMを用いれば大規模な文書データを元に類似文書を判別できる．しかし，ログのメッセージは通常の文書と異なり特殊な用語やパラメータなどを含むため，そのまま適用するとうまくいかない可能性が考えられる．これに対して，ファインチューニングによりログメッセージに対応させる．本目的はログ解析を対象としているため，攻撃痕跡に対応するファインチューニングが望ましい．そこで，WindowsとLinuxに対して同様の攻撃を行い，その結果出力されるログを抽出する．これらの対応を目視で確認し，対応するログメッセージをファインチューニング用データとすることで，ログ解析に対応した分類ができる．

3.2 ログの対応関係の評価

ファインチューニング用のペアデータを作成，または本システムを評価において　Linuxログと Windowsログ対応関係の基準が必要になる．そこで，次の３つの原則に基づき関係分析を行う．

①操作の一致

イベントの核となる動詞が一致している必要がある．

例えば,「作成（create, new, add）」は「作成」,「削除（remove）」は「削除」,「成功（success, accepted）」は「成功」,「失敗（failure, failed）」は「失敗」に対応している意味を持つメッセージである必要がある．

②対象の一致

操作の対象となる目的語が一致している必要がある．

例えばユーザーアカウント（user account）」は「ユーザーアカウント」であり，「グループ（group）」や「プロセス（process）」,「コンピュータアカウント（computer account）」とは明確に区別する必要がある．

③主たるイベントの優先

ある一つの Linux ログは，類似するときに複数のWindowsイベントを誘発する可能性がある．その中で「最適」と判断されるのは，ログが示す最も主要な出来事を記述したものであるべきである．副次的な影響や，システム内部の技術的な実装詳細は，最適とは見なさない．

例として次のイベント対応分析を示す．

Linux ログ：login[4684]: pam_unix(login:session): session opened for user msfadmin

Windows ログ：An account was successfully logged on.

イベントID：4624

① Linuxの session opened（セッションが開かれた）とWindowsの successfully logged on（正常にログオンした）は，どちらも「ログインの成功」という同じ操作を指しており，完全に一致する．

② Linux の user msfadmin（ユーザーmsfadmin）とWindowsの An account（あるアカウント）は，どちらも「ユーザーアカウント」という同じ対象を指しており，完全に一致する．

③ このログが示す最も重要な出来事は「ユーザーがシステムに正常に入室した」ことであり，両方のログがこの主たるイベントを最も的確に表現している．

本論文では対象となるLinuxログデータに対してあらかじめ最適な対応関係である Windows のイベントログと IDを定義しておく．そのリストに基づいて調査結果の評価を行い，同じ結果が出れば「正解」，そうでなければ「不正解」とする．

4. システム概要

4.1 システム構成

本システムの構成を次に示す．

開発環境：Anaconda

使用言語：python3.10
使用ライブラリ：

- ・panndas
- ・re
- ・os
- ・torch
- ・tqdm
- ・sentens-transformers

使用データ：

類似用 Windows セキュリティログデータ　414 件
ファインチューニング使用ペアデータ　63 件
検証用 Linux auth_log ログデータ　101 件

図 1 に本システムの構成を示す.

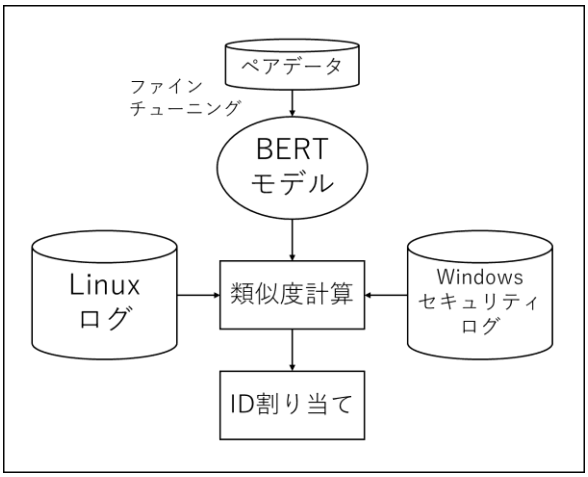

図 1　意味的類似度検索モデルの構成

4.2 使用データ

本システムでは 3 種類のログデータを使用する．類似用の Windows ログ，ファインチューニングのペアデータ，検証用の Linux の auth_log データである．Linux ログデータは似た手順で攻撃をして得た異常ログと正常時のログが混在したデータである.

Windows ログは, 本論文で対象となる Linux のログが auth ログであるため，今回はイベントソースが Microsoft-Windows-Security-Auditing であるイベントログ 414 件に絞り実験を行う．また，今回の対象である Linux のイベントメッセージは英語で構成されているため，Windows の英語言語の仮想環境で取得した英文メッセージを使用する.

モデルの学習に使用するペアデータについては，上記の原則に基づき「最適な適応関係」であると判断した，Linux の auth ログと Windows ログのペアから成るファインチューニング用のデータを用いる．なおこのペアデータは事前実験で両環境に同じような攻撃を仕掛けて取得した Linux と Windows のログデータのペア 63 件から成る.

4.3 モデルの学習 と推論

モデルには，事前学習済みの bert-base-uncased を SBERT として採用した．損失関数には MultipleNegativesRankingLoss を使用し，作成したポジティブペアデータで学習を行う．この手法は，各データバッチ内において，ペア以外のログを自動的に困難な不正解例と見なして学習させる[4]ため，効率的にログ間の微妙な意味の違いを捉えることが可能となる．学習データはシャッフルされた上でバッチ化され，DataLoader を通じてモデルに供給される．sentence-transformers の model.fit()メソッドを用い，ファインチューニングを実行する.

4.4 推論

まず候補となる Windows セキュリティログ を読み込み, その内容を学習済み SBERT モデルを用いて事前に全てベクトル化しておく.

次に，評価対象の各 Linux ログを一つずつモデルに入力し，そのベクトルを生成する．生成されたクエリベクトルと, 事前に計算しておいた全 Windows ログのベクトル群との間でコサイン類似度を総当たりで計算し，最も類似度スコアが高いベクトルを持つ Windows ログを特定し，ID を割り振りを行う.

5. 結果

対象とした auth ログとの結果を表 1 に示す.

表 1　モデルの結果

カテゴリ	件数	割合
正解 (Correct)	73 件	72.3%
不正解 (Incorrect)	28 件	27.7%

評価基準: 予測された Windows イベント ID が，事前に定義した対象である Linux ログとの最適な対応関係リストと完全に一致した場合のみを「正解」とする.

評価結果: 評価基準に基づき, 評価対象の Linux ログ全 101 件に対して評価を行った結果，モデルは全体の 72.2%にあたる 73 件のログにおいて，正確なマッピングを実現した. モデルは, 正常な利用状況と攻撃シナリオの両方において, 主要なイベントを正しく対応付けることに成功した. 特に, session opened という表現を持つ，主体やホスト名が異なる複数のログインイベントを，一貫して Windows の「An account was successfully logged on. (4624)」に正しくマッピングした.

成功例: ユーザーのログイン成功

- 入力 (正常): pam_unix(login:session): session opened for user test...
- 入力 (攻撃): metasploitable login[4684]: pam_unix(login:session): session opened for user msfadmin...

- 予測 (共通): An account was successfully logged on. (ID4624)

同様に，sudo コマンドによるプロセス生成や usermod によるグループへのメンバー追加など，攻撃シナリオの中核をなす多くのイベントも正確に対応付けられた．

一方で，学習データに含まれない概念のログについては，不正解な予測がなされた．

失敗例: システムサービスの起動

- 入力 (Linux): Loading rules from directory /etc/polkit-1/rules.d
- 予測 (Windows): An object in the COM+ Catalog was modified. (ID5888)
- 正解 (Windows): System audit policy was changed.(ID4719)

このほかにも，ユーザのパスワードの変更などのイベントが正確に紐づけられなかった．

6. 考察

以上の結果からの，SBERT を用いた意味的類似度検索モデルは，正常ログと攻撃ログが混在する現実的な環境においても 70%を超える高い精度で異種 OS 間のログを対応付けられることを実証した．

この成功は，モデルがログに含まれるユーザー名やホスト名といった変動要素に影響されず，session opened と successfully logged on のように，構文が全く異なっていてもその背後にある本質的な意味(＝ログイン成功)を理解し，関連付ける能力を持つことを示している．これにより，攻撃活動の連鎖を可視化する上で本モデルが有効であると言える．

しかし，失敗例から，モデルの性能は学習データに強く依存するという課題も明らかになった．Polkit サービスのルール読み込みのような，学習データセットに含まれていない未知の概念に対しては，モデルは正しい推論を行えず，関連性の低い予測を行った．これは，モデルの知識が学習データによって形成された意味空間に限定されるためである．また，学習内容に近いログの内，いくつかのメッセージ (パスワード変更など) は正しい ID が振り分けられなかった．この理由に関してはログメッセージを一般化できなかったためであると考えられる．例えば次例のように可変情報がノイズとなった可能性がある．

metasploitable passwd[4813]: ... password changed

- metasploitable (ホスト名)
- passwd (コマンド)
- [4813] (プロセス ID)

7. 結論

本論文では BERT を用いたイベント ID が割り当てられていないイベントへのイベント ID 自動割り当て手法について提案した．意味的類似度検索モデルは学習データの範囲では高い精度を発揮した．このような結果は今後のデジタルフォレンジックにおいて，一つの OS に限定せず様々なログデータを一元管理し，分析することにつながると考える．一方で，その範囲外のデータの予測に関しては課題が残る結果となった．今後の展望として，学習したにもかかわらず正しい ID を割り振れなかったため，より精度を高める手法やデータのクリーニングについて検討していく必要がある．また，auth ログに限らず様々なログを分類できるためのより高品質なペアデータセットの構築，モデルの改善などを行っていく．

参考文献

[1] “ManageEngine EventLog Analyzer https://www.manageengine.com/products/eventlog/eventlog-analysis.html,
（ 参照 2025 年 5 月 16 日）

[2] Tim Bai, Haibo Bian, Abbas Abou Daya, Mohammad A. Salahuddin, Noura Limam and Raouf Boutaba David R. Cheriton School of Computer Science, University of Waterloo, Ontario, Canada ,「A Machine Learning Approach for RDP-based Lateral Movement Detection」,2020 年 2 月 13 日, IEEE https://rboutaba.cs.uwaterloo.ca/Papers/Conferences/2019/BaiLCN19.pdf

[3] “磯野 怜, 中野 心太, 関谷 信吾, 折田 彰, 岸本 頼紀, 早稲田 篤志, 花田 真樹. 「機械学習を用いた異常ログ可視化のための誤検知された正常ログ対策の検討」, 情報処理学会コンピュータセキュリティシンポジウム 2024（CSS2024）, 4G2-2, 2024 年 10 月, https://ipsj.ixsq.nii.ac.jp/login/?next=%2Frecords%2F240998%3Fdownload%3Ddownload-aed84490-27e6-47e0-a4c3-1505da96439c

[4] “**Sentence-Transformers - Loss Functions**:”. https://www.sbert.net/docs/package_reference/sentence_transformer/losses.html, (参照 2025 年 7 月 17 日)