イベントログとネットワークパケットの時系列情報を組み合わせたIDSの構築

中久木 達哉[1,a)]　青木 茂樹[1,b)]　宮本 貴朗[1]

概要：近年頻繁に観測される標的型攻撃は，組織内ネットワークに侵入するために入念な調査を行い，侵入可能な方法を探し出し，通常の通信に紛れて侵入する．そのため，マルウェアの侵入を検知することが困難であり，マルウェア感染後の活動を検知する手法の重要性が高まっている．標的型攻撃のマルウェアに感染すると，ホスト上で C2 サーバと通信するためのプロセスが定期的に実行される．そのため，プロセス名の時系列情報に特徴が表れると考えられる．また感染後の端末は，C2 サーバと定期的に同じ内容の通信を行うため，定期的にサイズ等が一定のパケットが観測されると考えられる．そこで本研究では，プロセス名だけでなく，パケットの時系列情報にも着目することで高精度に標的型攻撃を検出する手法を提案する．まず，イベントログからホスト毎に実行されるプロセスのプロセス名等を抽出し，更にネットワーク上のパケットのヘッダからパケットサイズ等の特徴を抽出し，Doc2Vec でベクトル表現する．その後，抽出したベクトルを LSTM で学習することで不審な通信を検知する．MWS データセットを対象に実験を実施し提案手法の有効性を確認した．

キーワード：イベントログ，標的型攻撃，LSTM，Doc2Vec，MWS Dataset

Intrusion Detection System focused on time series information of event logs and network packets

Tatsuya Nakakuki[1,a)]　Shigeki Aoki[1,b)]　Takao Miyamoto[1]

Abstract: APT (Advanced Persistent Threats) have been observed frequently, attackers find ways to penetrate a network. They infiltrate the organization network through normal communication. Thus, methods for detecting activities after malware infection are becoming important. Since on the infected host with malware, a process for communicating with C2 server is periodically executed, we focus on the time-series information of process names. In addition, because the infected host periodically communicates with the C2 server in fixed formats, it is expected that the packet will be observed periodically. In this paper, we propose a method to detect APT by focusing on the time-series information of process names and these packets. First, we extract the process name from the event log, and packet size and other characteristic from the packet headers. We express them as vectors using Doc2Vec. Then, the vectors are trained by LSTM to detect APT.

Keywords: Event Log，APT(Advanced Persistant Threat)，LSTM，Doc2Vec，MWS Dataset

1. はじめに

近年，サイバー攻撃が増加傾向にあり，更に攻撃手法が巧妙化している．従来のサイバー攻撃は，不特定多数のホストの中から，サイバー攻撃に対する対策が不十分なホス

[1] 大阪府立大学大学院人間社会システム科学研究科
Graduate School of Humanities and Sustainable System Sciences, Osaka Prefecture University
[a)] saa01178@edu.osakafu-u.ac.jp
[b)] aoki@kis.osakafu-u.ac.jp

トを探索して攻撃を行っていた．そのため，組織や団体はファイアウォールや侵入検知システムを設置し，ホストではウイルス対策ソフトを稼働させることで，組織内ネットワークやホストへの侵入を試みる通信やネットワーク内の不審な通信を検知してきた．しかし，近年頻繁に観測されている標的型攻撃は従来のサイバー攻撃とは異なり，これまでの対策では攻撃を防ぐことが難しい．

標的型攻撃は，情報の窃取を目的に特定の組織内ネットワークに侵入した後，ネットワーク内部での攻撃を繰り返しつつ，長期間に亘って機密情報を窃取し続ける攻撃である [1]．従来のサイバー攻撃とは異なり，攻撃目標を選定するまでに入念な調査活動を行い計画を立案する．調査活動において，攻撃目標とする組織が行うセキュリティ対策や組織の脆弱性を調査して攻撃を実行する．

従来から用いられているウイルス対策ソフトや侵入検知システムではパターンマッチングによりサイバー攻撃の侵入を防いでいる．そのため，すでにパターンファイルに記録された特徴を持つ攻撃に対しては高い精度で検知することができるが，新たな攻撃に対してはパターンファイルにその特徴が記録されていないため検知することができない．標的型攻撃を目論む攻撃者は，組織が設置しているウイルス対策ソフトや侵入検知システムを調査し，パターンファイルに存在しない攻撃手法を計画する．そのため，パターンマッチング手法では標的型攻撃を防ぐことは難しい．

そこで，監視対象の異なるセキュリティ対策を何重にも組み合わせることで，侵入や情報漏洩の可能性を下げる多層防御技術が重要になってきている [2]．多層防御技術は，侵入対策，拡大対策，漏洩対策の 3 段階に分けて考えられることが多い．侵入対策では，ファイアーウォールや侵入検知システムを用いて，ネットワークの外部と内部の境界を監視して不正なアクセスを遮断している．拡大対策では，振る舞い検知型のソフトウェアにより，ネットワーク内における不審な通信を監視している．また，各ホストにおけるパスワードの強化やパッチの監視なども拡大対策に含まれる．漏洩対策では，ファイアーウォールや侵入検知システムによる監視に加えて，アクセスログや送信ログの取得・監視などを行うことによって，機密情報が外部に送信されることを防ぐ役割を担っている．

近年，特に拡大対策として，マルウェア感染後の活動を検知することの重要性が高まっている．マルウェア感染後の活動には，外部からの侵入経路 (バックドア) の構築や，外部から端末を遠隔操作するためにボットなどで行われる外部サーバとの定期的な通信 (ビーコン通信) 等が含まれる．

標的型攻撃のマルウェアに感染すると，ホスト上で C2（Command & Control）サーバと通信するためのプロセスが定期的に実行される．正常なホストでは，OS 起動プロセス実行後，ユーザのログイン情報の管理プロセスやユーザが使用したアプリケーションに関するプロセス等が実行される．一方，感染後のホストでは前述のプロセスに加えて，C2 サーバと通信したり，脆弱な端末を探索するために，特定のプロセスが定期的に実行される．例えば，名前が特定の組織名に偽装されたプロセスや侵入可能な端末を探索するプロセスが定期的に実行される．このように，感染後のホストでは特定のプロセスが定期的に実行されるため，プロセス名の時系列情報にマルウェアへの感染による特徴が表れると考えられる．

また，感染後の端末は侵入経路の構築後，C2 サーバと決まった書式で定期的に通信を行う．正規の Web サーバとの通信であれば，同一サーバとの通信であっても，閲覧するページによってページのサイズ等が異なり，メールサーバとの通信であれば，送受信するメールのサイズによって，サイズ等が異なるパケットが観測される．一方，C2 サーバとのビーコン通信の場合，情報を決まった書式で定期的に送信すると考えられるため，サイズ等が一定のパケットが定期的に観測されると考えられる．

そこで，悪性ファイル実行の周期性と特徴的なパケット送出の周期性の両方に注目することで，標的型攻撃を見つける手法を提案する．本研究では，Windows イベントログからホスト毎のプロセス名等の情報をテキストとして抽出し，更にパケットのヘッダからパケットサイズ等の特徴もテキストの形式で抽出する．そして，類似するテキストを類似するベクトルに変換して表現することのできる Doc2Vec[3] により，イベントログとパケットヘッダから抽出したそれぞれのテキストをベクトルに変換する．その後，抽出したベクトルを LSTM で学習することで不審な通信を検知する手法を提案する．実験では，MWS2018 データセットの BOS データセット [4] を用いて本手法の有効性を確認した．以下，2 節で関連研究について述べ，3 節で提案手法について説明する．4 節で実験と考察について述べ，5 節でまとめる．

2. 関連研究

本研究に関連する従来研究として，プロセスの実行順序に注目した異常検知手法である文献 [5]，Windows イベントログの時系列情報に注目した異常検知手法である文献 [6] と，ヘッダからセッション毎に抽出した特徴を用いた異常検知手法である文献 [7] について述べる．

文献 [5] では，個々のホスト内部で動作するプロセス情報（プロセスリスト）を定期的に取得し，ユーザが利用する可能性が低いプロセスを抽出し異常を検出する手法が提案されている．プロセスの親子関係や実行パスを用いた比較を行うことで，正常プロセスに成り済ましたプロセスでも，親プロセスの違いや実行パスの違いから新規プロセスとして抽出し，不審なプロセスを検知している．文献 [6] では，攻撃発生時に，複数のイベントが一連の流れで発生

する特徴に注目し，Windows イベントログから標的型攻撃の痕跡を調査する手法が提案されている．Windows イベントログとは，システムやアプリケーションが記録する Windows 標準のログであり，当該コンピュータ上で発生した事象（イベント）が記録される．各イベントには（イベント ID 5140：ネットワーク共有オブジェクトへのアクセス）のように，カテゴリ毎にイベント ID が割り当てられている．特定のイベント発生前後のイベントをイベント群として抽出し，イベント群に不審なイベントが複数含まれている場合に，攻撃として検知している．これらの手法では，プロセス名の時系列情報に特徴が表れる C2 サーバと通信するためのプロセス等が検知できると考えられる．しかし，プロセス名は容易に擬装できるため，これらの情報のみを利用した手法では，様々な標的型攻撃に対応することは難しい．

文献 [7] では，パケットのヘッダから得られる特徴を用いて標的型攻撃を検知する手法が提案されている．この手法ではまず，通常通信のトラフィックデータをセッションごとに分割し，抽出したパケット数やパケットサイズ等の通信挙動が学習時の挙動と類似しない場合に，不審な通信挙動として検知する．この手法では，パケットヘッダに特徴が表れる C2 サーバとのビーコン通信等を検知できると考えられるが，ビーコン通信の特徴は同一の Web ページを複数回閲覧している場合と類似する．また，観測したパケットをセッション単位にまとめ直し，特徴を抽出するため，大規模ネットワークに適用した際に特徴抽出処理に時間がかかる可能性がある．

そこで本研究では，ホストのイベントログから得られる特徴と，ネットワークのパケットヘッダから得られる特徴の両方に注目し標的型攻撃を検知する手法を提案する．本手法により，イベントログの時系列情報に特徴が表れる攻撃とヘッダに特徴が表れる攻撃の両方を検知することができるため，標的型攻撃を高精度に検知できると考えられる．また，セッション毎に特徴を抽出する手法と比較して，特徴抽出の処理にかかる計算コストを削減することができると考えられる．

3. 提案手法

図 1 に手法の概要を示す．本手法では，イベントログ，パケットヘッダから時間順に特徴をテキストとして抽出する．イベントログから抽出する特徴とパケットヘッダから抽出する特徴の形式が異なるため，どちらもテキスト形式に統一する．また，パケット毎に特徴を抽出することで，セッション毎に抽出する手法と比較して，特徴抽出の処理にかかる計算コストを削減している．本手法は学習と検証の 2 つの処理に分かれる．さらに，学習処理はホストのイベントログの時系列情報とネットワークから抽出したパケットヘッダの時系列情報の学習処理に分かれる．まず，

図 1 提案手法の概要
Fig. 1 Outline of Proposed method.

イベントログの時系列情報の学習処理では，学習用イベントログからイベント発生順に特徴を抽出する．その後，Doc2Vec を用いて，抽出したテキストから特徴ベクトル（パラグラフベクトル）を作成し，LSTM により学習する．次に，パケットヘッダの時系列情報の学習処理では，学習用トラフィックデータからパケットの到着順に特徴を抽出する．その後，Doc2Vec を用いて，抽出したテキストから特徴ベクトル（パラグラフベクトル）を作成する．そして，イベントログの特徴ベクトルの学習に利用した LSTM と

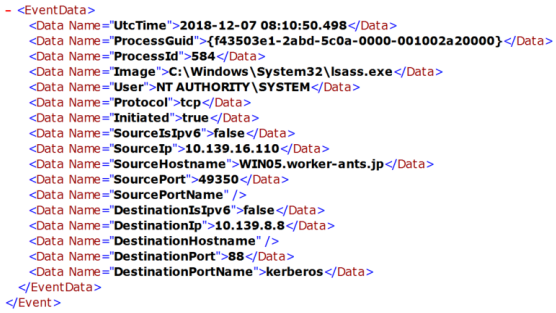

図 2　イベントログの例
Fig. 2　Example of event log.

表 1　イベントログから抽出する特徴量一覧
Table 1　List of features of event logs.

イベント発生時間	実行ファイル名
実行コマンド名	実行ファイルのハッシュ値

は別の LSTM で学習する．

検証時の処理では，検証用イベントログ・検証用トラフィックデータから，学習時と同様の手順で特徴ベクトルを作成し，学習済みの LSTM を用いて予測値を出力し，予測値と実際の特徴ベクトルとの誤差を算出する．その後，誤差を単位時間毎に区切り，それぞれの LSTM で算出された誤差を統合する．区間の誤差が閾値未満であった場合，その区間は正常，閾値以上の場合，異常と判断する．

3.1　学習処理

3.1.1　特徴抽出

ホストで観測するイベントログから特徴を抽出する手順を述べる．まず学習用イベントログをログ単位で読み込む．イベントログには，図 2 に示すように，特定のホストで観測されたイベントが記録されている．2018-01-23 15:00:00 C:\Windows\System32\taskhost.exe のように，ログ単位でイベント発生時刻，実行ファイル名等を抽出する．抽出する特徴の一覧を表 1 に示す．

次に，ネットワークで観測するパケットのヘッダから特徴を抽出する手順を述べる．まず学習用トラフィックデータをパケット到着順に 1 パケットずつ読み込む．標的型攻撃のビーコン通信はパケットサイズ等に特徴が表れると考えられる．加えて，攻撃者がシーケンス番号を予測し，偽のパケットをセッションに差し込むことで送信相手への侵入を試みる攻撃等の多様なマルウェアにも対応するために，表 2 に示す特徴を抽出し，1 バイト毎に 129 0 12・・・のように 10 進数の値で表現する．ここで，変換した数値は 129 0 12 54 129 0・・のようになり，ヘッダが異なる場合にも同じ数値が割り当てられる．それらを異なる特徴として学習させるために eth.type：1290 vlan.id：1254 vlan.etype：1290・・tcp.urgent_pointer：00 data：142 2

表 2　ヘッダ特徴量の一覧
Table 2　List of features of packet headers.

Ethernet タイプ番号 (2bytes)
VLAN ID(2bytes)
VLAN Ethernet タイプ番号 (2bytes)
ip ヘッダ長 (1byte)
輻輳通知 (1byte)
パケット長 (2bytes)
識別番号 (2bytes)
フラグメントオフセット (2bytes)
TTL 値 (1byte)
プロトコル (1byte)
ヘッダのチェックサム (2bytes)
シーケンス番号 (4bytes)
確認応答番号 (4bytes)
輻輳保護 (1byte)
コントロールフラグ (1byte)
ウインドウサイズ (2bytes)
チェックサム (2bytes)
緊急ポインタ (2bytes)
ペイロードの情報 (バイト数はパケットに依存)

21 234 のように，ヘッダの種類毎に文書として抽出する．tcp.urgent_pointer 以降は，ペイロードの情報が格納されている場合があるため，それらを data としてまとめる．なおここでは，ホストに固有の情報（宛先・送信元 IP アドレス，宛先・送信元 MAC アドレス，宛先・送信元ポート番号）は除くことにより，後述する学習処理でホストに固有の情報を学習することを防いでいる．最後に，パケットの到着時間を加えて，2018-01-28 01:35:30 eth.type：1290 vlan.id：1254 vlan.etype：80・・のように，1 パケット毎に 1 文書として抽出する．

3.1.2　Doc2Vec を用いた特徴ベクトルの作成

本研究では 3.1.1 節で構成した，イベントログの特徴を表す文書とパケットヘッダの特徴を表す文書を，別々の Doc2Vec で学習し，それぞれパラグラフベクトルに変換する．Doc2Vec は Le ら [3] によって提案されたモデルであり，Word2Vec[8] の概念を文章全体に拡張した手法である．Word2Vec は，中間層が 1 層，出力層が 1 層のニューラルネットワークのモデルである．そして単語を，文脈を考慮して学習し，ベクトルとして表現することができるモデルである．一方，Doc2Vec は文章全体をベクトルとして表現することができるモデルである．Doc2Vec には 2 種類の学習アルゴリズムが存在するが，本研究では図 3 に示す高精度な DM-PV（Distributed Memory Model of Paragraph Vectors）モデルを用いる．ここで，$\boldsymbol{d}_i$ は文書 ID を表現する One-hot ベクトル，$\boldsymbol{D}$ は文書の重み行列，$\boldsymbol{W}$ は単語の重み行列，$\boldsymbol{w}_t$ は t 番目の単語が 1 でその他の要素が 0 の One-hot ベクトルを表す．

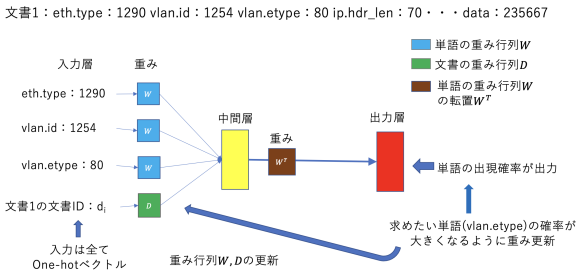

図 3　DM-PV モデルでの学習例

Fig. 3　Example of learning by DM-PV.

まず，文書 ID を示す文書の One-hot ベクトルと，ランダムにサンプリングされた t 番目の単語の周辺単語の One-hot ベクトルを入力層に入力する．そして，出力層に t 番目の単語の One-hot ベクトルが出力されるように，文書の重み行列 $\boldsymbol{D}$ と単語の重み行列 $\boldsymbol{W}$ を更新する．ここで，文書の重み行列に文書 ID を掛けることで，文書 ID が示す文書を表現するパラグラフベクトル $\boldsymbol{t}_n$ として表現される．図 3 には DM-PV モデルの概要と DM-PV モデルで文書を学習する例を示している．図 3 では，文書 1（eth.type：1290 vlan.id：1254 vlan.etype：80 ip.hdr_len：70・・・data：235667）から vlan.etype：80 という単語がサンプリングされた場合の学習過程を説明している．この手法を用いて，3.1.1 節で作成した正常なイベントログ・パケットヘッダの文書 ID と，文書に含まれる単語の One-hot ベクトルを学習する．そして，学習を終えた Doc2Vec モデルを用いて文書をパラグラフベクトルに変換する．

3.1.3　LSTM を用いた深層学習

本研究では，イベントログ・パケットヘッダの時系列情報に注目しているため，時系列データの学習に適した機械学習モデル LSTM[9] を異常検出器として用いる．LSTM とはネットワーク内部での短期記憶を長期間活用できる構造を持つ RNN の一種である．RNN との大きな違いは，記憶セル（memory cell）を導入し，中間層の状態を長期間伝播できるようにしていることである．記憶セルは入力ゲート（input gate），出力セル（output gate），忘却ゲートを持ち，それぞれ，入力から記憶セルに書き込む量，記憶セルから出力する量，直前の時刻の記憶セルの内容を保持する量を調整している．LSTM の内部構造を図 4 左側，概要を図 4 右側に示す．

時刻 n の入力 $\boldsymbol{t}_n$ に対して，入力ゲート $\boldsymbol{i}_n$，忘却ゲート $\boldsymbol{f}_n$，記憶セル $\boldsymbol{c}_n$，出力ゲート $\boldsymbol{o}_n$，中間層 $\boldsymbol{h}_n$ を式 (1)，(2)，(3)，(4)，(5) で算出する．

$$\boldsymbol{i}_n = \sigma\left(\boldsymbol{W}_{it}\boldsymbol{t}_n + \boldsymbol{W}_{ih}\boldsymbol{h}_{n-1}\right) + \boldsymbol{b}_i \tag{1}$$

$$\boldsymbol{f}_n = \sigma\left(\boldsymbol{W}_{ft}\boldsymbol{t}_n + \boldsymbol{W}_{fh}\boldsymbol{h}_{n-1}\right) + \boldsymbol{b}_f \tag{2}$$

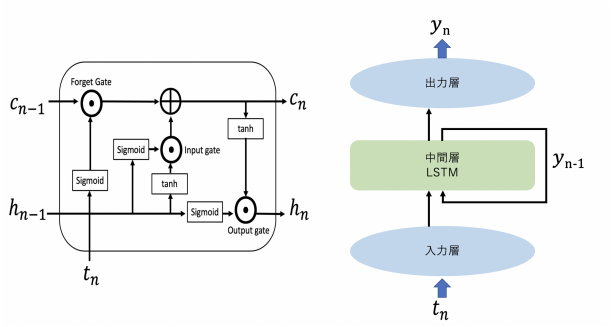

図 4　LSTM の内部構造・概要

Fig. 4　Internal structure and outline of LSTM.

$$\boldsymbol{o}_n = \sigma\left(\boldsymbol{W}_{ot}\boldsymbol{t}_n + \boldsymbol{W}_{oh}\boldsymbol{h}_{n-1}\right) + \boldsymbol{b}_o \tag{3}$$

$$\boldsymbol{c}_n = \boldsymbol{f}_n \odot \boldsymbol{c}_{n-1} + \boldsymbol{i}_n \odot \tanh\left(\boldsymbol{W}_{ct}\boldsymbol{t}_n + \boldsymbol{W}_{ch}\boldsymbol{h}_{n-1}\right) + \boldsymbol{b}_c \tag{4}$$

$$\boldsymbol{h}_n = \boldsymbol{o}_n \odot \tanh\left(\boldsymbol{c}_n\right) \tag{5}$$

ここで，$\boldsymbol{W}_{it}$，$\boldsymbol{W}_{ih}$，$\boldsymbol{W}_{ft}$，$\boldsymbol{W}_{fh}$，$\boldsymbol{W}_{ot}$，$\boldsymbol{W}_{oh}$，$\boldsymbol{W}_{ct}$，$\boldsymbol{W}_{ch}$ は学習する重み行列，$\boldsymbol{b}_i$，$\boldsymbol{b}_f$，$\boldsymbol{b}_o$，$\boldsymbol{b}_c$ は学習するバイアスベクトル，σ はシグモイド関数，tanh は hyperbolic tangent 関数，$\odot$ はアダマール関数である．図 4 右側に示す通り LSTM は，入力層，中間層，出力層で構成されている．$\boldsymbol{t}_n$ を入力層に入力し，入力値に対する予測値 $\boldsymbol{y}_n$ を得る．得られた予測値 $\boldsymbol{y}_n$ と入力値 $\boldsymbol{t}_n$ を比較し，誤差を算出する．誤差の算出には式 (6) で算出する平均二乗誤差 (MSE：Mean Squared Error) を使用する．

$$MSE = \frac{1}{n-1}\sum_{i=1}^{n-1}\left(\boldsymbol{t}_i - \boldsymbol{y}_i\right)^2 \tag{6}$$

その後，得られた誤差を基に誤差逆伝播を行い，LSTM の重みを更新する．ここで，事前に定めたエポック数に到達した時，学習を終了する．本研究では，イベントログ $\boldsymbol{t}_{host}$ の特徴ベクトルとパケットヘッダの特徴ベクトル $\boldsymbol{t}_{network}$ を別々の LSTM で学習する．

3.2　異常検出

3.1.3 節で学習した LSTM を用いて異常を検出する．異常検出処理の概要を図 5 に示す．新たなイベントログ・トラフィックデータから特徴ベクトル $\boldsymbol{t}_{host}$，$\boldsymbol{t}_{network}$ を抽出して LSTM に入力し，予測値 $\boldsymbol{y}_{host}$，$\boldsymbol{y}_{network}$ を算出する．予測値 $\boldsymbol{y}_{host}$，$\boldsymbol{y}_{network}$ と入力値 $\boldsymbol{t}_{host}$，$\boldsymbol{t}_{network}$ との平均二乗誤差をそれぞれ異常度 $\boldsymbol{\alpha}_{host}$，$\boldsymbol{\alpha}_{network}$ とする．該当区間内で，イベントが観測できない場合，その区間 j の異常度 $\boldsymbol{\alpha}_{host\_j}$ は，前後の区間の異常度 $\boldsymbol{\alpha}_{host\_j-1}$，$\boldsymbol{\alpha}_{host\_j+1}$ の平均値とする．また，区間 j 内で通信が発生せずパケットが観測できない場合の異常度 $\boldsymbol{\alpha}_{network\_j}$ も同様に算出する．その後，単位時間毎にイベントログの特徴を学習させ

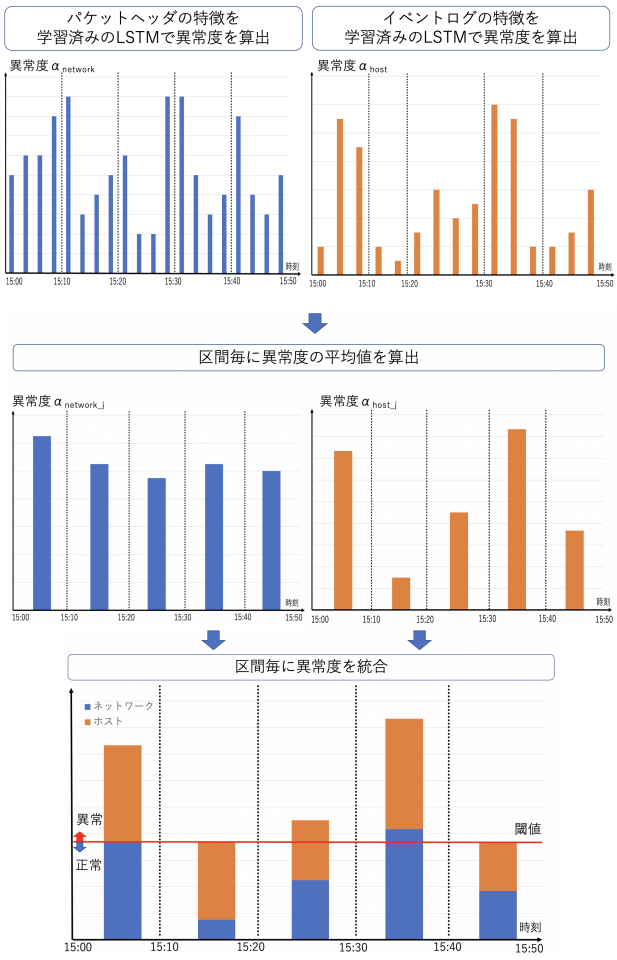

図 5 異常検知処理の概要
Fig. 5 Outline of Anomaly Detection.

た LSTM での異常度 $\boldsymbol{\alpha}_{host}$ とパケットヘッダの特徴を学習させた LSTM での異常度 $\boldsymbol{\alpha}_{network}$ を統合する．それぞれの LSTM で該当区間における異常度の平均値を算出し，式 (7) を用いて該当区間の異常度を統合し，異常度 $\boldsymbol{\alpha}_{mix}$ を算出する．式中の $\boldsymbol{\lambda}$ はハイパーパラメータとする．

$$\boldsymbol{\alpha}_{mix} = \boldsymbol{\lambda} \cdot \boldsymbol{\alpha}_{host} + (1 - \boldsymbol{\lambda}) \cdot \boldsymbol{\alpha}_{network} \qquad (7)$$

区間の異常度が予め設定した閾値 $\boldsymbol{\theta}$ 以上の場合に，不審なイベントの発生，不審な通信を検知し，閾値 $\boldsymbol{\theta}$ 未満の場合に正常とする．

4. 実験

4.1 実験条件

実験に使用する標的型攻撃の通信を含む例として，MWS Dataset 2018，2019 の一つである，組織内ネットワークへの侵害活動を想定した動的活動観測のデータセット BOS Dataset 2018，2019[10] を用いた．ここで，BOS データセットにはマルウェアの進行度が示されており，マルウェアにより通信が発生したか，またどのように通信が行われたデータであるかが示されている．進行度ごとの説明を表 3 に示す．本稿では，BOS Dataset 2019：進行度 1 の 2018 年 12 月 7 日 ~12 月 9 日のイベントログ，pcap データを異常を含まない正常データとして学習データに利用した．そして，BOS Dataset 2018：進行度 8 の 2018 年 1 月 23 日 ~1 月 28 日のイベントログ，pcap データを異常を含むデータとしてテストデータに用いた．

表 3 進行度の説明
Table 3 Explanation of Progress.

進行度	説明
1，2	通信発生なし
3，4，5	通信発生したが，C2 サーバとの攻撃通信不成立
6，7，8	通信発生かつ C2 サーバとの攻撃通信成立

表 4 実験データのパケット数・イベントログ数
Table 4 Number of Event Logs and Packets of Experimental Data.

データ	正常イベントログ数	異常イベントログ数	総イベントログ数
進行度 1	149 個	0 個	149 個
進行度 8	9030 個	364 個	9394 個
データ	正常パケット数	異常パケット数	総パケット数
進行度 1	874224 個	0 個	874224 個
進行度 8	1180470 個	79851 個	1260321 個

テストデータのイベントログに対するラベル付けは，C2 サーバと通信している IP アドレスとの通信で発生したイベントログ，文献 [11] で公開されている不正なプロセス名（実行コマンド名・実行ファイル名）とそのハッシュ値を含むイベントログを異常とした．また，テストデータのパケットに対するラベル付けは，C2 サーバと通信している IP アドレスの通信を異常とした．各実験データの正常イベントログ数・異常イベントログ数・総イベントログ数，正常パケット数・異常パケット数・総パケット数を表 4 に示す．

そして，1 区間に異常イベントログ，異常パケットが P 個以上含まれている場合に異常区間とし，イベントログとパケットの異常度を統合した区間は，どちらかの区間が異常区間であれば異常とした．そして，実験では P を 5，単位時間を 10 分とした．また，学習データにおけるパケットヘッダのデータ量がイベントログのデータ量より多かったため，異常度を統合する際の重み $\boldsymbol{\lambda}$ は 0.3 とし，パケットヘッダの情報の重みを大きくした．

4.2 評価方法

評価方法には Precision，Recall，F-measure を用いた．Precision は異常であると識別されたものの中で，異常であったものの割合を表す．Precision の算出式を式 (8) に示す．

$$Precision = \frac{TP}{TP + FP} \qquad (8)$$

Recall は異常を正しく異常であると識別できた割合を表す．Recall の算出式を式 (9) に示す．

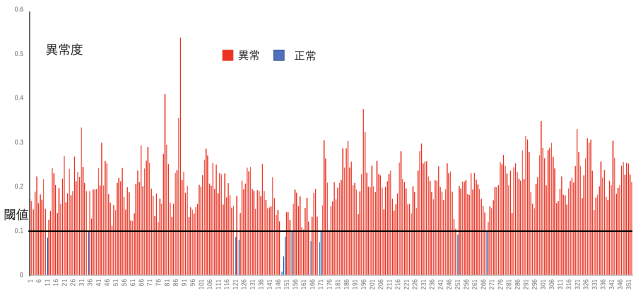

図 6 実験 1 の区間毎の異常度

Fig. 6 Anomaly Score for each section of experiment 1.

表 5 検知結果

Table 5 Result of Detection.

実験	特徴	P	Precision	Recall	F-measure
実験 1	パケットヘッダ イベントログ	5	0.90	0.97	0.93
実験 1α	パケットヘッダ イベントログ	1	0.78	0.85	0.81
実験 2	イベントログ	5	0.75	0.96	0.85
実験 3	パケットヘッダ	5	0.85	0.87	0.86

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

ここで，TP(True Positive) は異常を正しく異常と識別した数，FP(False Positive) は異常と識別したが正常であった数，TN(True Negative) は正常を正しく正常と識別した数，FN(False Negative) は正常と識別したが異常であった数を表す．

F-measure は Precision と Recall の調和平均である．F-measure の算出式を式 (10) に示す．

$$F\text{-}measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{10}$$

4.3 実験結果・考察

学習データとして進行度 1 のイベントログ・トラフィックデータ，テストデータとして進行度 8 のイベントログ・トラフィックデータを用いた実験を実験 1 とする．$\boldsymbol{P}$ の値による検知結果の変化を確認するために $\boldsymbol{P} = 1$ とした実験を実験 1α とする．また，イベントログの時系列情報とパケットヘッダの時系列情報を統合することの有効性を評価するために，比較実験としてパケットヘッダの情報のみを用いた実験 2，イベントログの情報のみを用いた実験 3 を行った．実験 1，実験 1α，実験 2，実験 3 における Precision，Recall，F-measure を表 5 に，実験 1 の異常度のグラフを図 6 示す．図 6 のグラフは，横軸に区間，縦軸に異常度を示している．

4.3.1 学習結果

進行度 1 のイベントログを用いて学習した際のエポック数に対する学習誤差の遷移を図 7 左側に，進行度 1 のトラフィックデータを用いて学習した際のエポック数に対する学習誤差の遷移を図 7 右側に示す．イベントログの特徴ベクトルは，エポックが 30 を越えたあたりから誤差が収束していた．パケットヘッダの特徴ベクトルは，エポック数が 2 や 3 を越えたあたりから誤差の遷移が急に降下し，30 あたりで収束していた．そのため，イベントログの特徴を学習する際のエポック数，パケットヘッダの特徴を学習する際のエポック数はともに 30 とし，実験を行った．

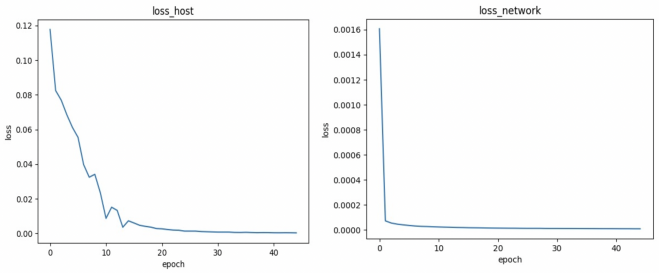

図 7 イベントログ・パケットヘッダの情報を学習させた LSTM の学習誤差の遷移

Fig. 7 Transition of LSTM Training Error of Event Log and Packet header.

4.3.2 実験 1 の異常検知結果と考察

実験 1 の異常検知結果について考察する．閾値を 0.10 として実験を行ったところ，Recall は 0.97 と高い結果を得ることができた．図 6 のグラフからも高精度に正常区間と異常区間の識別が行われたことを確認できる．この結果から本手法で，プロセス名の時系列情報に特徴が表れる不正なイベント，パケットヘッダの時系列情報に特徴が表れる不審な通信を検知できることを確認できた．また，$\boldsymbol{P}$ を 1 とした実験 1α の Recall0.85 と比較して精度が向上していたことから，異常パケット数が多い異常区間の方が正常区間との異常度の差が大きくなったために，精度が向上したと考えられる．さらに，実験 1 の Precision，Recall は，実験 2，実験 3 よりも精度が高いことから，プロセス名の時系列情報とパケットヘッダの時系列情報を統合することが標的型攻撃の検知に有効であることも確認できた．

次に，異常を正しく識別できた特徴と異常を正常，正常を異常と誤識別した特徴について考察する．イベントログから抽出した特徴を確認すると，正常イベントログでは端末の起動や OS の起動に必要なファイルを実行するプロセスが多く，観測期間中は数種類の同じプロセスが実行されていた．一方，異常イベントログでは，C2 サーバとの通信を確立するためのプロセスとして，実行ファイル名に特定の組織名を含んだプロセス，侵入可能な端末を探索する特定のプロセスが定期的に実行されていた．イベントログの情報のみを用いた実験 2 においても，Recall は 0.96 と高い結果を示しており，正常イベントログと異常イベントログのプロセス名の違いを高精度に識別できたと考えられる．しかし，端末内アプリケーションの実行に必要な正常プロセスでも，イベントログの学習データにないプロセスの場合に，誤って異常と識別している場合が存在した．

パケットのヘッダから抽出した特徴を確認すると，進行度 8 のテストデータに含まれる正常パケットと異常パケットのパケットサイズに差があり，異常パケットの TTL 値には正常パケットと異なる TTL 値が設定されていた．パケットヘッダの情報のみを用いた実験 3 においても，Recall は 0.87 と高い結果を示しており，正常パケットと異常パケットにおけるパケットサイズや TTL 値等の違いを高精度に識別できたと考えられる．しかし，正常パケットのパケットサイズと差が小さい異常パケットを正常と識別している場合も存在した．

プロセス名の情報とパケットヘッダの情報を統合した際に，異常と識別できた特徴について考察する．脆弱な端末を探索する活動はプロセスとしては異常，パケットとしては正常と識別されていた．この活動におけるパケットは，接続確認のパケットであるため，正常パケットとのパケットサイズの差が小さく，正常パケットと識別されたパケットも存在した．しかし，このプロセスには正常プロセスに含まれる可能性の低い特定のコマンド名が含まれていたため，統合した区間で異常と識別できたと考えられる．C2 サーバとのビーコン通信はプロセスとしては正常，パケットとしては異常と識別されていた．ビーコン通信時のプロセスは，正常イベントログに含まれるプロセスも実行されており，プロセス名に特徴が表れにくく正常プロセスと識別されたプロセスも存在した．しかし，パケットヘッダの時系列情報に着目すると，正常区間とのパケットサイズの差が大きいことや TTL 値の差から，統合した区間で異常と識別できたと考えられる．

以上より，本手法でマルウェア感染後のプロセスや不審な通信を高精度に検知できることを確認した．本手法の F-measure は既存手法 [7] を超える結果となった．学習データに含まれない実行ファイル名・実行コマンド名を含んだプロセスがある場合，パケットサイズや TTL 値で学習データとの差がある場合に，不審なプロセス，不審な通信として検知できていた．これは異常検知で，イベントログの時系列情報とパケットヘッダの時系列情報を統合したために高精度に標的型攻撃を検知できたと考えられる．しかし，学習データに含まれない，もしくは発生確率の低い実行ファイル名・実行コマンド名を含む通常時のプロセスを誤って不審プロセスとして検知した．パケットヘッダの学習データ数に比べて，イベントログの学習データ数が少なかったために誤検知を招いたと考えられる．また，今回の実験で使用したデータセットは大半の区間で異常なプロセス，パケットが観測されており，本手法の True Negative と False Positive の検証が十分にできていない．今後，今回の実験とは別のデータセットによる実験等で十分な検証を実施したいと考えている．今後の課題として，パラメータ $\boldsymbol{P}$・$\boldsymbol{\lambda}$・単位時間の調整，学習データの増加や新たな特徴の追加等が挙げられる．

5. おわりに

本稿では，ホストのイベントログから得られる特徴と，ネットワークのパケットヘッダから得られる特徴の両方に注目し標的型攻撃を検知する手法を提案した．実験では，MWS データセットを用いて本手法の有効性を確認した．標的型攻撃によるマルウェアの動的観測データを用いた実験では，プロセス名に特徴が現れる不審なプロセスや，ヘッダに特徴が現れる不審な通信を検知することができた．しかし，マルウェア感染後のホストのイベントログ数が少ない場合に異常を検知できない可能性がある．今後の課題として，パラメータの調整，学習データの増加，通常時のイベントログが少数でも異常検知できるようにするための，新たな特徴量の追加などが挙げられる．

参考文献

[1] Le Blond, S. et al.：A Look at Targeted Attacks Through the Lense of an NGO, Proc. 23rd Usenix Security (2014).

[2] McAfee Blog:なぜ多層防御なのか？リスクを最小限にする最強のセキュリティ対策，入手先 ⟨http://blogs.mcafee.jp/defense-in-depth-multilayer-protection⟩(参照 2021/07/27)

[3] Le, Q., and Mikolov, T.：Distributed representations of sentences and documents Proc. of, International Conference on Machine Learning, pp1188-1196(2014).

[4] 高田雄太，他: マルウェア対策のための研究用データセット MWS 2018 Datasets ，情報処理学会研究報告コンピュータセキュリティ，Vol.2018-CSEC-82

[5] 中里純二, et al. ：ホスト型 IDS を用いた標的型攻撃対策，コンピュータセキュリティシンポジウム 2014 論文集，Vol.2014，No.2，pp.466-473(2014)

[6] 藤本万里子，et al.：Event Log を用いた MS17-010 の脆弱性を悪用する攻撃の検知，情報処理学会，第 81 回全国大会講演論文集

[7] 鍛治一祐, 青木茂樹, 宮本貴朗：パケットのヘッダに基づく不審な通信挙動の検知，情報処理学会研究報告インターネットと運用技術，Vol.2018-IOT-40，pp.1-7(2018)

[8] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J.：Distributed representations of words and phrases and their compositionality, Advances in neural information processing systems, pp3111-3119(2013).

[9] M. Ring, A. Dallmann, D. Landes, and A. Hotho, “ Ip2vec: Learning similarities between ip addresses, ” in Proc. of 2017 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 657-666, (2017).

[10] 荒木粧子, 他: マルウェア対策のための研究用データセット ~MWS Datasets 2019~, 情報処理学会, Vol.2019-CSEC-86, No.8, 2019 年 7 月.

[11] Hitachi Incident Response Team：BOS/STARDUST を用いた攻撃者の活動観測，入手先 ⟨https://www2.nict.go.jp/csri/plan/H31-symposium/pdf/terada.pdf⟩(参照 2021/07/27)