社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

IP フロー情報を用いた確定時間でのマルウェアトラフィック検知

小松　聖矢† 　桂　　祐成† 　垣内　正年†† 　新井イスマイル†† 　藤川　和利††

† 奈良先端科学技術大学院大学 先端科学技術研究科　〒630–0192 奈良県生駒市高山町 8916–5
†† 奈良先端科学技術大学院大学 総合情報基盤センター　〒630–0192 奈良県生駒市高山町 8916–5
E-mail: †{komatsu.seiya.kt6,katsura.yusei.ky6}@is.naist.jp, ††{masato,ismail,fujikawa}@itc.naist.jp

あらまし　ボットネットやランサムウェア等のマルウェアの活動による被害が社会問題となっている．マルウェアの活動を効率的に検知し被害を低減するため，その活動トラフィックをネットワーク上で検知する研究が提案されている．これらの研究で利用されるトラフィック情報には，パケット情報，IP フロー情報，Interface カウンタ情報の 3 種類がある．IP フロー情報を用いる場合，5 タプルでトラフィックが集約され軽量だが，タイムアウトが発生するかコネクションが終了するまで情報が出力されず，スキャン活動や長期間継続するフローに対し早期の検知が困難である．本研究では，これらのフローが終了する前に検知を行い，特徴量を変更することで，検知性能を従来のシステムと同等に保つことを目指す．本報告では，フローごとのコネクション状態とポート番号，トランスポート層プロトコルの遷移を特徴量として用いる既存の検知手法と，Zeek を用いて IP フロー情報に変換した ISCX botnet データセットを利用し，フロー継続時間の上限を設定しない場合（データセット中の最長フロー 240,418 秒）と 30 秒とした場合で検知性能を調査した．結果，それぞれ 98.1% と 96.1% の F 値で検知が可能であることを確認した．本調査から，30 秒以内（確定時間）でのマルウェアトラフィック検知をわずかな検知性能の低下で実現可能なことを示した．
キーワード　マルウェア，ボットネット，マルウェア検知，侵入検出・検知，ネットワークセキュリティ

Malware Traffic Detection at Certain Time Using IP Flow Information

Seiya KOMATSU†, Yusei KATSURA†, Masatoshi KAKIUCHI††, Ismail ARAI††, and Kazutoshi FUJIKAWA††

† Graduate School of Science and Technology, Nara Institute of Science and Technology　8916–5, Takayama-cho, Ikoma, Nara, 630–0192 Japan
†† Information Initiative Center, Nara Institute of Science and Technology,　8916–5, Takayama-cho, Ikoma, Nara, 630–0192 Japan
E-mail: †{komatsu.seiya.kt6,katsura.yusei.ky6}@is.naist.jp, ††{masato,ismail,fujikawa}@itc.naist.jp

Abstract　The damage caused by the activities of malware such as botnets and ransomware has become a social problem. In order to detect malware activity efficiently and reduce the damage, research on detecting malware activity traffic in the network has been proposed. There are three types of traffic information used in these research: packet information, IP flow information, and interface counters information. In the case of using IP flow information, traffic is aggregated in 5-tuples, which is lightweight, but the information is not output until a timeout occurs or the connection is terminated. Therefore, making it difficult to detect scanning activities or long-lasting flows at an early stage. This research aims to maintain the same detection performance as conventional research by modifying the feature while detecting these flows before they terminate. In this paper, we experiment with existing methods that use connection status, port numbers, and transport layer protocols transition of each flow as features. We used the ISCX botnet dataset converted into IP flow information using Zeek to investigate the detection performance when the upper limit of flow duration is not set (the longest flow in the dataset: 240,418 seconds) and when the upper limit is set to 30 seconds. As a result, we confirmed that detection was possible with an F-measure of 98.1% and 96.1%, respectively. From this research, we showed that it is possible to detect malware traffic within 30 seconds (a certain time) with a slight decrease in detection performance.
Key words　Malware, Botnet, Malware Detection, Intrusion Detection, Network Security

1. はじめに

ボットネットやランサムウェア等のマルウェアによる攻撃活動の被害が社会問題となっている．マルウェアの活動による被害の例として，大規模な Distributed Denial-of-Service (DDoS) 攻撃によるサービスやネットワークの停止，ランサムウェアによるストレージの暗号化と金銭の要求，スパムメールやマルウェアの拡散等が挙げられる [1], [2].

これらのマルウェアの活動を検知し被害を低減するため，マルウェア検知システムが提案されている．マルウェア検知システムはホストベースの手法とネットワークベースの手法に大別される．ネットワークベースの手法は，マルウェアの活動により発生するネットワークフローを検知対象とし，多数のホストのトラフィックをひとつのシステムで監視できる．また，マルウェアが検知システムの検査を回避することが比較的困難である．

ネットワークベースのマルウェア検知システムでは，検知に用いるトラフィック情報として，次に示す 3 種類が用いられる．

（1） ネットワークトラフィック中のパケットの全情報を取得するパケット情報

（2） 主にネットワーク機器が出力し，IP アドレス，ポート番号，トランスポート層プロトコル（5 タプル）で集約された IP フロー情報

（3） ネットワーク機器から Simple Network Management Protocol (SNMP) により得られる Interface グループ MIB 情報中の統計情報（Interface カウンタ情報）

IP フロー情報は，NetFlow version 9 を基に RFC 7011 [3] で標準化された IP Flow Information Export (IPFIX) 以外はペイロード情報を保存できない．また，パケットごとの観測時間やペイロードサイズ，ヘッダフィールドの値等の詳細な特徴量を検知システムに提供できない．しかし，関連研究で提案されている手法 [4] では，IP フロー情報を利用した場合でも，利用する特徴量や検知アルゴリズムの工夫により，パケット情報を利用する場合と同程度の検知性能を達成している．従って，IP フロー情報を利用することで，ストレージ容量や学習の際のメモリサイズ等の要求されるリソースを抑えつつ高性能な検知が可能である．

しかしながら，IP フロー情報はフローの終了もしくはタイムアウトが発生しないとトラフィック情報として出力されない特徴をもつ．そのため，応答が観測されないスキャン，継続的にパケットをやり取りするマルウェア，Remote Access Trojan (RAT) による調査活動，ファイル転送プロトコルを利用した情報窃取等のフローに対して，攻撃活動の早期段階での検知が困難である．この課題を解決するため，指定した時間内に IP フロー情報を出力し検知が可能な時間確定性を有する検知システムが必要である．

本報告では，良性，悪性のそれぞれのフローに対し設定したフロー継続時間の上限値が経過した時点での情報を検知システムに入力することで，確定時間で検知が可能なシステムを提案する．本提案手法は，ネットワークトラフィックやデータセットから得られるパケット情報を IP フロー情報に変換する際に，上限値以降のパケットを用いずに変換を行う．そのため，得られる情報は IP フロー情報と同じ形式で，関連研究で提案された検知アルゴリズムを適用することができる．

提案手法の有効性を確認するための評価実験において，フロー継続時間の上限値を 30 秒に設定し，関連研究 BOTection [4] と ISCX botnet データセット [5] を用いて実験を行った結果，30 秒以内での確定時間検知を行う場合，94.8% の適合率，98.8% の再現率，96.1% の F 値を得た．

本報告の構成は以下のとおりである．2. では既存のマルウェア検知手法について説明する．特に IP フロー情報を利用した手法を取り上げ，検知性能や利用している特徴量について概説する．加えて，IP フロー情報を利用した際の確定時間検知が困難な課題について説明する．3. では提案する確定時間でのマルウェアトラフィック検知手法の要件を示し，その要件を満たす提案手法の概要と実装方法を示す．4. では提案手法の有効性を評価するための実験及びその結果と考察を示し，5. では本報告のまとめと今後の課題について述べる．

2. 関連研究

マルウェアの検知手法は，ホストベースの手法とネットワークベースの手法に大別される．

ホストベースの手法は，エンドポイントで動作し，主にプロセス情報を用いて検知を行う．この手法は，エンドポイントで動作するためプロセス情報やバイナリファイル等のマルウェアの動作に関する情報を多く利用でき，高性能な検知が可能である．しかし，監視対象の全てのエンドポイントに検知システムの導入が必要で，動作するプラットフォームに大きく依存する．

ネットワークベースの手法は，エンドポイント間のネットワークで動作し，正常なトラフィックとマルウェアのトラフィックの差異に着目して検知を行う．この手法は，多数のホストを 1 台の検知システムで監視できるため，ホストベースの手法と比べ導入コストが小さい．また，マルウェアが動作するエンドポイントから隔離されたノードで動作するため，マルウェアが検知システムの存在を検知し，検査を回避することは比較的困難である．

2.1 ネットワークベースの手法で用いられるトラフィック情報

ネットワークベースのマルウェア検知手法は，ネットワークの高速化や大規模化に伴い，より少ないデータ量で検知を行うことが求められている．ネットワークベースの手法で利用されるトラフィック情報として，パケット情報，IP フロー情報，Interface カウンタ情報がある．

パケット情報は，ネットワーク上でやり取りされるパケットから得られる全情報を利用可能である．そのため，保存や検知システムを動作させる際のストレージやメモリ等の要求

表 1　関連研究で用いられる特徴量の比較

	MalClassifier	MalAlert	BOTection	MalPhase
Layer-4 プロトコル	○	○	○	○
ポート番号	○	○	○	○
バイト数	○	○	–	○
パケット数	○	○	–	○
フロー継続時間	–	○	–	○
コネクション状態	○	–	○	–
タイムスタンプ	–	○	–	–
Round-Trip Time	–	–	–	○
エントロピー	–	–	–	○
IP アドレスの種類	–	–	–	○

表 2　関連研究の検知および分類性能（F 値）

	MalClassifier	MalAlert	BOTection	MalPhase
検知性能	–	–	99.78%	98.00%
タイプ分類性能	–	90.00%	–	93.00%
ファミリ分類性能	96.00%	–	99.09%	91.00%

リソースは大きいが，パケットごとのタイムスタンプやペイロードサイズ，ヘッダフィールドの値等の詳細な情報を利用でき，検知時には高い性能を達成できる．

IP フロー情報は，ネットワークフローを 5 タプルで集約し，ペイロード情報を含まないトラフィック情報で，フローごとの IP アドレスやポート番号，トランスポート層プロトコル，フロー継続時間等の情報が含まれる．

Interface カウンタ情報は，インタフェースごとのバイト数やパケット数等の統計情報を SNMP プロトコルにより収集した情報である．

2.2　IP フロー情報を利用した検知手法

IP フロー情報を利用した関連研究では，マルウェアの活動により発生するネットワークフローを検知することを対象としている．AlAhmadi らは, Zeek [6] で生成した IP フロー情報を用いてマルウェアトラフィックを検知するシステム MalClassifier [7] と BOTection [4] を提案している．MalClassifier は IP フロー情報から得られるトランスポート層プロトコルの種類，ポート番号，フロー中のバイト数，パケット数，コネクション状態に N-gram モデルを適用し生成した時系列の特徴量から，Random Forest と k-nearest neighbors を用いて検知する．BOTection はトランスポート層プロトコルの種類，コネクション状態，ポート番号にマルコフ連鎖モデルを適用し生成した時系列の特徴量から，Random Forest を用いて検知する．

Piskozub らは，IPFIX とフローの統計情報を用いてマルウェアトラフィックを検知するシステム MalAlert [8] と MalPhase [9] を提案している．MalAlert は大規模なネットワークにおいて，観測されたフローの統計情報を特徴量として，Random Forest を用いて検知する．MalPhase は IPFIX で得られる IP フロー情報とペイロード中のシャノンエントロピー，Round-Trip Time，IP アドレスの種類（プライベートアドレスかグローバルアドレス）等を特徴量として，Feedfoward Neural Network を用いて検知する．また，検知の際にノイズとなる大多数の良性フローを Auto-encoder で除去する手法を提案している．

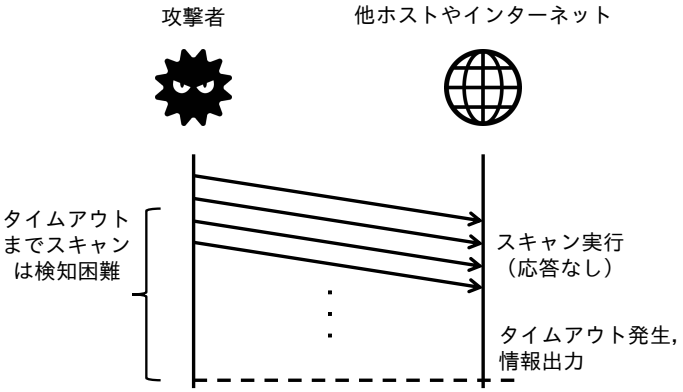

図 1　IP フロー情報利用時の課題（スキャン活動の検知）

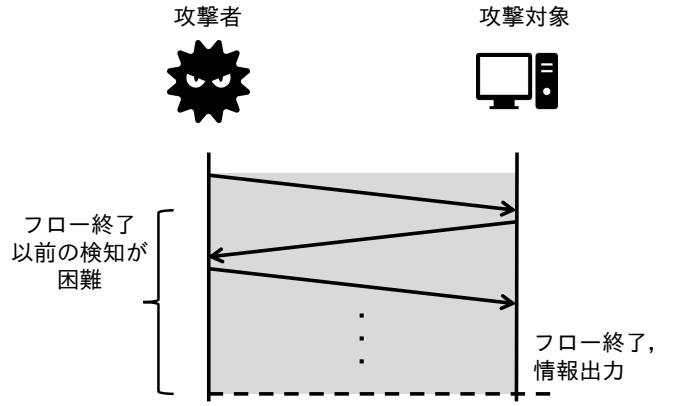

図 2　IP フロー情報利用時の課題（長時間継続するフローの検知）

これらの関連研究の特徴量の比較を表 1 に，検知性能の比較を表 2 に示す．表 2 において，タイプとはワーム，トロイの木馬，スパイウェア等，脅威の挙動の種類を示す．ファミリとは Mirai, Zeus, Rbot 等，同様の挙動を示す脅威をグループ化した固有名を示す．

2.3　IP フロー情報の課題と確定時間検知の必要性

IP フロー情報を利用した検知手法は多く提案されているが，これらの研究は確定時間での検知を考慮していない．NetFlow や IPFIX，Zeek 等の IP フロー情報の多くの形式や変換に用いられるソフトウェアは，タイムアウトの発生時またはフローの終了時にそのフローの情報が出力される．そのため，マルウェアがスキャンを行い応答が無かった場合，タイムアウトまでスキャンを検知することができない．また，RAT による調査活動，ファイル転送プロトコルや隠しチャネルを利用した情報窃取，Command and Control (C&C) サーバとの通信等，比較的長時間のフローが発生する場合，これらのフローが終了するまで攻撃活動の検知ができない．（図 1, 2）．

従って，フローが開始されてから指定した時間内でフローが終了し検知が完了する時間確定性を有する検知手法が必要である．時間確定性を有する検知手法を用いることで，前述した攻撃性のフローに対し攻撃の早期段階で検知が可能となり，マルウェアの活動による被害をより小さくすることができる．

アルゴリズム 1 提案手法のアルゴリズム

Require: t フロー継続時間の上限値
Ensure: 指定した上限時間で終了するフローの集合
  while ネットワークまたはデータセットで観測された各パケット **do**
    if IP ヘッダが存在するか **then**
      if トランスポート層プロトコルが TCP か **then**
        if パケットの 5 タプルが破棄リストに含まれるか **then**
          パケットを破棄
        if パケットの 5 タプルが観測中リストに含まれるか **then**
          if 観測時間は t を超過しているか **then**
            当該フローを終了する
            破棄リストに 5 タプルを追加
        else
          観測中リストにパケットの 5 タプルと観測時間を追加

3. 提 案 手 法

本研究では 2.3 で示した IP フロー情報の課題に対し，確定時間でのマルウェアトラフィック検知手法を提案する．まず，3.1 で提案手法が満たすべき要件について議論し，3.2 で提案手法の概要について述べ，3.3 で提案手法の実装方法を示す．

3.1 提案手法の要件

本報告で提案する手法の要件を下記に示す．

要件 A　全てのフローが指定した継続時間の上限値で終了する
要件 B　特徴量の変更を抑え，検知性能の低下を小さくする
要件 C　特定のソフトウェアや IP フロー情報の形式に依存しない

提案手法は，2.3 で示した IP フロー情報の課題を解決するため，検知対象の全てのフローが指定したフロー継続時間の上限値で終了し，検知アルゴリズムに入力される必要がある．また，提案手法を適用した際に，フロー継続時間やパケット数等の特徴量が変化し検知性能が低下することが予想されるが，その影響を最小限に抑える必要がある．加えて，Zeek や IPFIX 等の特定のソフトウェアや IP フロー情報の形式に依存しない必要がある．

3.2 要件を満たす提案手法

3.1 で示した要件を満たす提案手法を検討する．要件 A をを満たすための処理は，トラフィック情報を取得する際に各フローの継続時間を保持するか，変換に用いるソフトウェアの追加機能として実装することが考えられる．しかし，変換に用いるソフトウェアに追加の実装した場合，要件 C を満たさない．したがって，トラフィック情報を取得する際にこれらのソフトウェアにフローが終了したと認識させるようトラフィック情報を変更し，フローを終了させる必要がある．

また，要件 B を満たすため，コネクションの状態等の変更される特徴量の割合を可能な限り抑える必要がある．そのため，ネットワークトラフィックの特徴を調査し，最も多く観測されているコネクション状態でフロー終了処理を行う．

3.3 提案手法の実装方法

3.2 に基づいた提案手法の実装方法について述べる．提案手法の実装は，検知対象のネットワークやデータセットで発生するフローに対し，設定したフロー継続時間の上限値を経過した際，FIN パケットや REJ パケットを利用し，アルゴリズム 1 に示すアルゴリズムに従ってフローを終了させる．

このアルゴリズムは，t にフロー継続時間の上限値を設定し，破棄するパケットの 5 タプルを保持する破棄リスト，観測された 5 タプルと観測開始時間を保持する観測中リストを用いる．確定時間検知を行う対象のネットワークやデータセットのそれぞれのパケットに対して，そのパケットが IP ヘッダと TCP ヘッダが設定さているか確認し，TCP ヘッダまで設定されていた場合，その 5 タプルが破棄リスト存在するか確認する．存在する場合は，そのパケットを破棄する．存在しない場合は，観測中リストを確認する．観測中リストに存在しない場合，そのパケットの 5 タプルと観測開始時間を保存する．存在する場合は，フロー観測時間が上限値 t を超過していないか確認する．超過していない場合は処理は行わない．超過している場合はそのパケットの 5 タプルを破棄リストに追加し，そのパケットを破棄する．

4. 評 価 実 験

3. で提案した手法の有効性を評価するため実験を行った．本章では，ISCX botnet データセットに提案手法を適用し，BOTection を検知アルゴリズムとして利用した際の，検知性能の変化を調査した結果とその考察を述べる．

4.1 検知アルゴリズムの選定

本実験では，関連研究 BOTection [4] を検知アルゴリズムとして利用し提案手法を評価する．BOTection を選定した理由を下記に示す．

- 検知に Random Forest（木の数が 101）を利用しており軽量である
- 調査した先行研究中で最も高い検知性能を達成している
- フロー継続時間やバイト数，パケット数を特徴量として用いず，提案手法適用時における影響が少ない

4.2 データセット

調査で利用するデータセットとして，先行研究 BOTection で利用されている ISCX botnet [5] を用いた．本データセットに含まれるマルウェアトラフィックの内訳を表 3 に示す．

4.3 実験の設定と評価指標

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{1}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{2}$$

$$F - measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{3}$$

本実験でのフロー継続時間の上限値は，データセット中の最長フローが 240,418 秒と非常に長時間継続している点や，RAT での調査では最長 38 時間の活動が観測されている報告をふまえ，十分早期段階での検知だと考えられる 30 秒に設定した．

表 3　データセットに含まれるマルウェアトラフィック内訳

データセット	フロー数	ファミリ内訳
ICSX botnet Training	6,925,812	Neris (12%), Rbot (22%), Virut (0.94%), Zeus (0.01%), Zeus C&C (0.01%)
ICSX botnet Testing	5,040,068	既知ファミリ：Neris (5.67%), Rbot (0.018%), Virut (12.80%), Zeus (0.109%), Zeus C&C (0.006%)
		未知ファミリ：Menti (0.62%), Sogou (0.019%), Murlo (0.16%), Tbot (0.283%), Zero Access (0.221%), Weasel (9.25%), Smoke Bot (0.017%), ISCX IRC Bot(0.387%)

表 4　データセットの統計量

	提案手法適用前	提案手法適用後
フロー継続時間の中央値	0.3992 秒	0.3992 秒
フロー継続時間の最大値	240418 秒	30 秒
30 秒以上のフローの割合	0.6%	0%

表 5　評価実験結果

	提案手法適用前	提案手法適用後
適合率	97.4%	94.8%
再現率	98.8%	97.5%
F 値	98.1%	96.1%

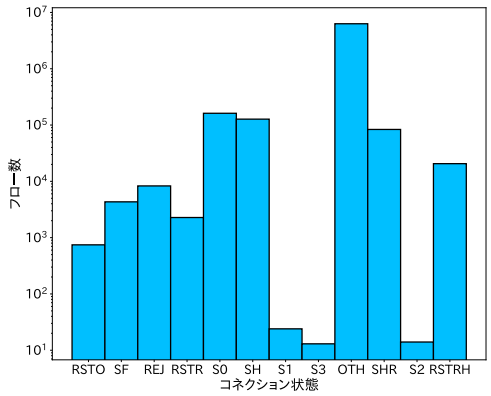

図 3　データセット中のコネクション状態の頻度分布

30 秒以上継続するフローを発生させたマルウェアは，Neris, Rbot，Virut の 3 種類を確認した．本設定時のデータセットの統計量を表 4 に示す．データセットには 30 秒以上継続するフローが 0.6% 存在したが，提案手法を適用することにより全てのフローが 30 未満で終了することが確認できた．また，フロー継続時間の中央値が非常に小さく提案手法の適用後も変化していないことから，大多数のフローはごく短い時間で終了しているといえる．

フローの終了方法は，データセットの分布を調査し正常終了（Zeek では SF）を用いた．データセット中のコネクション状態の頻度分布を図 3 に示す．また，データセットを IP フロー情報に変換するために Zeek を利用した．本実験の結果は式 (1), (2), (3) に示す適合率 (Precision)，再現率 (Recall)，F 値 (F-measure) により評価する．

4.4　実 験 結 果

評価実験の結果を表 5 に示す．本実験では，フロー継続時間の上限値を 30 秒に設定し，提案手法適用後のトラフィックは全て 30 秒以内に終了したため，確定時間での検知が実現できているといえる．また，検知性能は適合率が 2.6%，再現率が 1.3%，F 値が 2.0% 低下する結果となった．検知アルゴリズムとして用いた BOTection は，コネクション状態を特徴量として用いるため検知性能に影響を受けたためだと考えられる．しかし，BOTection が利用する 3 種類の特徴量のうち 1 種類が影響を受けたという点を考慮すると，2.0% の検知性能の低下は非常に小さいといえる．

4.5　考　　察

実験結果から，30 秒以内での確定時間検知が可能なことが示された．しかし，BOTection を検知アルゴリズムに用いた場合，F 値が 2.0% 低下した．この検知性能の低下は，検知に用いたアルゴリズムがコネクション状態に依存しているため，フローの終了処理に伴い，特徴量が変化したためだと考えられる．

実験結果では，データセット中の 0.6% が変化した場合にそれ以上の検知性能（F 値）の低下が発生している．これは，検知に用いた BOTection がマルコフ連鎖モデルを利用して特徴量を生成しており，提案手法により変化が発生した場合，前後のフローとの状態遷移が変化するため，影響が大きくなったと考えられる．また，再現率よりも適合率が低下していることも，同様の理由によるものだと考えられる．

本報告の実験ではフロー継続時間の上限を 30 秒に設定したが，ISCX botnet データセット中に 30 秒を超えるフローの割合は 0.6% と少なく，十分な評価が行えていない．したがって，図 4, 5 に示すフロー継続時間の分布から，調査に適した時間を設定する必要がある．この際，フロー継続時間の上限値を小さくした場合，影響を受けるフロー数は指数的に増加するため，提案手法適用時に変化する特徴量に大きく依存する検知手法は影響を受けることが予想される．

4.6　今後の課題

本提案手法では，フロー継続時間やコネクション状態，フロー中のパケット数等を変更してしまうため，これらを特徴量として用いる場合は検知性能が低下することが予想される．そのため，IP フロー情報出力時に変更を加えたフローにフラグ等を付与し特徴量として利用することにより，検知性能が向上できると考える．

また，フロー上限時間を 30 秒未満に設定した際の検知性能の変化を調査し，検知性能とより早期での検知の両立が可能なフロー上限時間を決定することが挙げられる．

加えて，本提案手法で影響を受ける特徴量を用いない検知

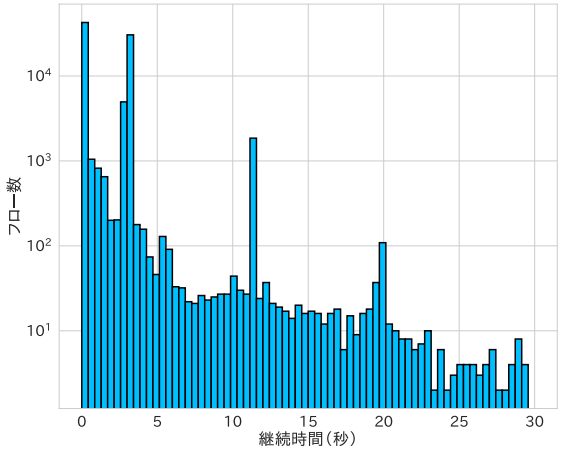

図 4　フロー継続時間の分布（30 秒未満）

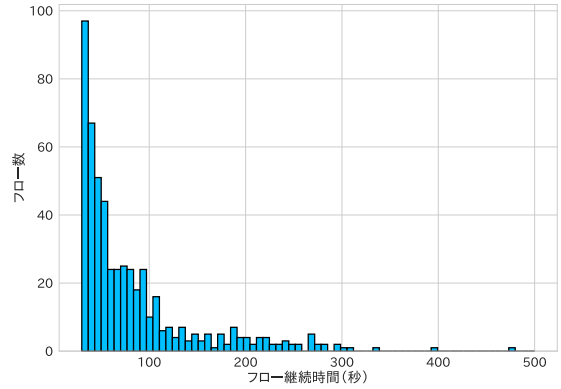

図 5　フロー継続時間の分布（30 秒以上）

手法であれば，高い性能を維持しつつ確定時間での検知が実現できる．したがって，影響を受ける特徴量を用いない軽量な検知アルゴリズムを設計し実装することが挙げられる．

5. おわりに

マルウェアの活動による被害を低減するため，その活動を検知するシステムが提案されている．IP フロー情報を用いたマルウェアトラフィック検知手法は多く提案されているが，それらの研究は確定時間での検知を考慮しておらず，タイムアウトが発生するまでのスキャンや長時間継続するフローに対して早期の検知が困難である．

本研究ではこの課題に対し，設定したフロー継続時間の上限値を超過したフローを終了させ，検知アルゴリズムを適用することにより，確定時間検知を実現する手法を提案した．

提案手法の有効性を評価するための実験では，検知アルゴリズムとして，関連研究の BOTection を，データセットとして ICSX botnet を用いた． 結果，それぞれ 98.1% と 96.1% の F 値で検知が可能であることを確認した．本調査から，30 秒以内（確定時間）でのマルウェアトラフィック検知をわずかな精度低下で実現することを示した．

今後の課題として，IP フロー情報出力時に変更を加えたフローにフラグ等を付与し特徴量として利用し検知性能を向上させること，フロー上限時間を 30 秒未満に設定した際の検知性能の変化を調査すること，本提案手法で影響を受ける特徴量を用いない軽量な検知手法を考案することを挙げた．

文　献

[1] “情報セキュリティ 10 大脅威 2021：IPA 独立行政法人情報処理推進機構,” https://www.ipa.go.jp/security/vuln/10threats2021.html.
[2] “情報通信研究機構 NICTER 観測レポート 2020,” https://www.nict.go.jp/cyber/report/NICTER_report_2020.pdf.
[3] P. Aitken, BenoîtClaise, B. Trammell, “Specification of the IP Flow Information Export (IPFIX) protocol for the exchange of flow information,” RFC 7011, Sept. 2013. https://rfc-editor.org/rfc/rfc7011.txt
[4] B.A. Alahmadi, E. Mariconti, R. Spolaor, G. Stringhini, and I. Martinovic, “Botection: bot detection by building markov chain models of bots network behavior,” Proceedings of the 15th ACM Asia Conference on Computer and Communications Security, pp.652–664, 2020.
[5] E.B. Beigi, H.H. Jazi, N. Stakhanova, and A.A. Ghorbani, “Towards effective feature selection in machine learning-based botnet detection approaches,” 2014 IEEE Conference on Communications and Network SecurityIEEE, pp.247–255 2014.
[6] “The zeek network security monitor,” https://zeek.org/.
[7] B.A. AlAhmadi and I. Martinovic, “Malclassifier: Malware family classification using network flow sequence behaviour,” 2018 APWG Symposium on Electronic Crime Research (eCrime)IEEE, pp.1–13 2018.
[8] M. Piskozub, R. Spolaor, and I. Martinovic, “Malalert: Detecting malware in large-scale network traffic using statistical features,” ACM SIGMETRICS Performance Evaluation Review, vol.46, no.3, pp.151–154, 2019.
[9] M. Piskozub, F. De Gaspari, F. Barr-Smith, L. Mancini, and I. Martinovic, “Malphase: Fine-grained malware detection using network flow data,” Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security, pp.774–786, 2021.

付　録

表 A·1　Zeek のコネクション状態一覧

状態	状態の意味
S0	接続の要求を観測したが応答しない
S1	接続が確立し終了していない（データのやり取り無し）
SF	接続が確立し終了した（データのやり取りが発生）
RET	接続の要求が拒否された
S2	接続が確立し，接続元は終了を試みるが， 接続先からの応答は無し
S3	接続が確立し，接続先は終了を試みるが， 接続元からの応答無し
RSTO	接続の確立後，接続元が接続を中断
RSTR	接続の確立後，接続先が接続を中断
RSTOS0	接続元は SYN パケットを送信後，RST パケットを送信， 接続先からの SYN-ACK パケットは無し
RSTRH	接続先は SYN-ACK を送信後，RST パケットを送信， 接続元からの SYN パケットは無し
SH	接続元は SYN パケットを送信してから FIN パケットを送信，接続先の SYN-ACK パケットは無し
SHR	接続先は SYN-ACK パケットを送信してから FIN パケットを送信，接続元の SYN パケットは無し
OTH	SYN パケット無しや，閉じていないコネクション， 中間のトラフィック等の部分的なコネクション