一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
ICM2019-5(2019-05)

3-tuple フロー中の 5-tuple フロー数による低帯域 L3, L4 DDoS 検知特徴量

林　裕平† 　鈴木　彦文‡ 　西岡　孟朗†

† 日本電信電話株式会社 ネットワークサービスシステム研究所 〒180-8585 東京都武蔵野市緑町 3-9-11

‡ 信州大学 総合情報センター 〒380-8553 長野市若里 4-17-1

E-mail: †{yuuhei.hayashi.mr, takeaki.nishioka.nh}@hco.ntt.co.jp, ‡ h-suzuki@shinshu-u.ac.jp

あらまし 近年, L3, L4 のプロトコルを悪用し, パルス状に DDoS を行う高度な攻撃が新たに観測されている. この攻撃は短時間の攻撃を繰り返し行うため, 観測される攻撃通信帯域の時間平均が小さい値として観測される場合がある. 一方, ネットワークにはルータ等の転送装置が既に広く配置されており, NetFlow 等から得られる通信フローの情報を用いて最新の DDoS 攻撃を検知できれば, DDoS 対策の水準を経済的かつ迅速に向上させることができる. そのような中, フロー情報から通信帯域を計算し, それを機械学習と組み合わせることで, DDoS 攻撃の検知を行う従来研究が存在する. しかし, これらの手法は攻撃通信の帯域が小さく, 通常通信の帯域と有意な差が現れない場合は, 攻撃検知が難しい課題がある.

本研究では従来研究の課題解決のため, 低帯域の L3, L4 DDoS 攻撃を検知可能とする特徴量とその高速な計算手法を提案する. 当該特徴量は (*src_ip, dst_ip, dst_port*)で定義される 3-tuple flow 中に存在する 5-tuple flow 数の分布が通常通信と攻撃通信で異なる考察に基づく. また, 信州大学のネットワークに対し攻撃ツールを用いて低帯域な攻撃を行いつつ取得したトラヒックデータと, WIDE が公開しているトラヒックデータに対し当該特徴量を計算し, Local Outline Filter (LOF)と組み合わせた際の攻撃検知精度の評価を実施した. 評価の結果, 提案特徴量は偽陰性率及び偽陽性率を低く抑えつつ攻撃検知が可能であることが解った.

キーワード DDoS, 検知, 低帯域, NetFlow, sFlow, 機械学習

Feature Value for Low-Bandwidth L3, L4 DDoS Detection based on Number of 5-tuple Flows in 3-tuple Flow

Yuhei HAYASHI† Hikofumi SUZUKI‡ Takeaki NISHIOKA†

† NTT Network Service System Labs 3-9-11, Midori-cho Musashino-shi, Tokyo 180-8585

‡ Shinshu University Integrated Intelligence Center , Wakasato, Nagano 380-8553

E-mail: †{yuuhei.hayashi.mr, takeaki.nishioka.nh}@hco.ntt.co.jp, ‡ h-suzuki@shinshu-u.ac.jp

Abstract Recently, new sophisticated attacks such as pulse-wave DDoS has been observed. The DDoS attack repeats short duration attacks, so the time-averaged bandwidth of the attack traffic can be observed as low rate. On the other hand, routers are already deployed in their network and it can send traffic flow information by using NetFlow etc. Level of DDoS countermeasure can be raised economically and quickly if the attacks can be detected by the flow information. Some researchers proposed to detect DDoS attack by calculating bandwidth from the flow information and collaborating it and machine learning. However, in a case where the bandwidth of attack is low so there is no significant difference between attack traffic and normal traffic in terms of bandwidth, the conventional approach is not effective.

To make up for the disadvantage of the conventional method, we propose a new feature value and its fast calculation method for detection low-bandwidth L3, L4 DDoS attacks. This feature value is based on a consideration that the number of 5-tuple flows existing in 3-tuple flow defined by (*src_ip, dst_ip, dst_port*) differs between normal traffic and attack traffic. In addition, we evaluated attack detection accuracy when our proposed feature value and Local Outline Filter (LOF) collaborate. Under the evaluation, we used the dataset obtained by carrying out attacks on the Shinshu University network. We also used the dataset obtained at the transit link of WIDE. The evaluation results show that the proposed feature value is effective to detect low-bandwidth L3, L4 attack while suppressing false negative and false positive.

Keywords DDoS, Detection, Low-bandwidth, NetFlow, sFlow, Machine Learning

1. はじめに

近年，社会におけるインターネットの重要性は高まりつつある．そのような中で，インターネットを脅かす DDoS 攻撃がより高頻度化・高度化している．攻撃頻度については，月に 1 回以上 DDoS を観測したサービスプロバイダが 8 割以上もあるとの報告がある[1]. また，高度化については，短時間の攻撃を繰り返し行うパルス状 DDoS 攻撃が新たに観測されている[2].

このように高頻度，高度化する DDoS 攻撃の対策は企業や大学，サービスプロバイダのネットワーク運用者等にとって喫緊の課題である. しかし, 最新のパケット分析装置を用いた DDoS 攻撃対策は，その装置準備のために多大なコストがかかり, かつ装置設置のために長期の時間を要してしまうため, 運用者にとって大きな負担となる. 一方, ネットワークにはルータ等の転送装置が既に広く配置されており, かつ NetFlow[3]や sFlow[4]を用いて通信フローの情報を取得することが可能である. これらの情報を用いて, 最新の DDoS 攻撃を検知できれば, ネットワークの DDoS 対策の水準を経済的かつ迅速に向上させることが可能となる.

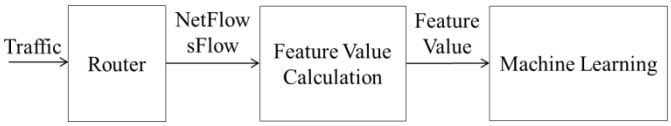


図 1　NetFlow，sFlow と機械学習の組み合わせによる DDoS 攻撃検知

NetFlow や sFlow のフロー情報から通信帯域を計算し，それを機械学習と組み合わせることで，攻撃検知を行う従来研究が存在する. 図 1 にその組み合わせの概念図を記載する. これらの手法は帯域に顕著な特徴が出る攻撃通信に対しては有効なアプローチである. しかし，攻撃通信の帯域が小さい値として観測され，通常通信と攻撃通信の帯域に有意な差が現れない下記の様な場合は，攻撃検知が難しい課題がある.

・低帯域でも攻撃が成立するリソース消費型 L3，L4 DDoS 攻撃[5]を検知したい場合.

・帯域を計算する際の時間スパンを大きくすると通信帯域の値が均されて小さい値となる可能性がある，パルス状の L3，L4 DDoS 攻撃[2]を検知したい場合.

本稿では従来研究の課題を解決するため，低帯域の L3，L4 DDoS 攻撃検知を可能とする新たな特徴量を提案する．提案する特徴量は(*src_ip*，*dst_ip*，*dst_port*)の 3-tuple で定義されるフロー(以下，3-tuple フローと呼ぶ)中に存在する 5-tuple フロー数の分布が，通常通信と攻撃通信で異なる考察に基づいている．また，実際に運用されているネットワークのデータセットに対し，Local Outline Filter (LOF)を用いて提案手法の有効性評価を実施した．なお，データセットについては，信州大学のネットワークに対し低帯域な L3, L4 DoS 攻撃を行い取得したキャプチャデータと，WIDE が公開しているキャプチャデータを使用した.

本稿は次のように構成されている．2 章でまず関連研究の紹介を行い，3 章にて特徴量及びその計算手法を提案する．そして，4 章で提案特徴量の評価及び考察を行い，5 章で本稿をまとめる.

2. 関連研究

本章では NetFlow や sFlow と機械学習の組み合わせによる DDoS 検知手法に関する関連研究を紹介する.

Hou らは，NetFlow データからパケット数，バイト数といった帯域に関わる特徴量を抽出し，機械学習と連携させることで DDoS 攻撃を検知する方式提案を行っている[6]．更に，当該特徴量と Random Forest を組み合わせたシステムを構築し，様々な攻撃ツールを用いて攻撃を発生させた際に，当該システムにより攻撃検知を行うことができたと報告している．しかし，帯域に関する情報のみを特徴量として用いているため，帯域観点で攻撃通信と通常通信に顕著な差がない場合，提案されている特徴量は有効ではない.

Rukavitsyn らはネットワーク構成が頻繁に変わるクラウドの仮想環境において，NetFlow データを基に検知モデルを柔軟に変える提案を行っている[7]．この研究も，特徴量としては単純な帯域を用いているため，低帯域の攻撃の検知は難しい.

Terzi らは NetFlow データを *src_ip* で集約し，*src_port*, *dst_ip*, *dst_port* のユニーク数や，byte，パケット数を特徴量として用いることを提案している[8]．また，CTU のデータセットに対し，k-mean 法と当該特徴量を組み合わせた際の攻撃検知性能の評価を実施している．しかし，当該特徴量は基本的に帯域ベースであり，また評価における帯域に関する前提条件が具体的に示されていない．そのため，低帯域の攻撃に対する本提案の有効性については不明である.

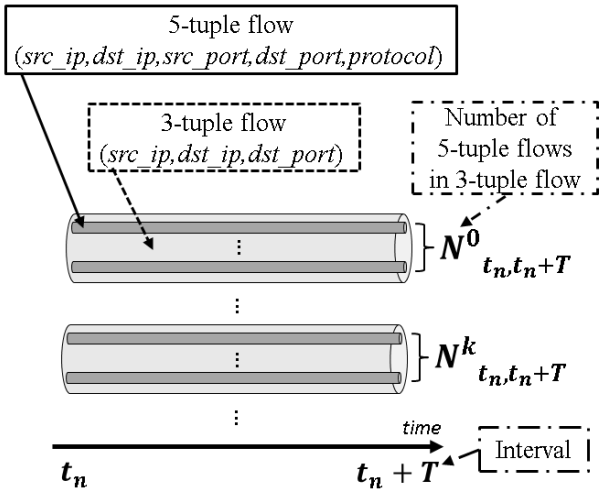
Kemp らは NetFlow と機械学習の連携により,L7 の DoS 攻撃である Slow DoS 攻撃を検知する研究を行っている[9]．当該研究では，TCP flag，フロー継続時間，帯域を特徴量として用い，Random Forest 等の機械学習と組み合わせる提案を行っている．しかし，彼らはサンプリングレート変化に対する有効性評価を行っていない．一般に NetFlow を用いる際はルータの負荷を抑える目的でパケットサンプリングを行うため，サンプリングレートが検知において重要なパラメタとなる.

Wagner らは NetFlow データと OCSVM を組み合わせて攻撃検知を行う方式を提案している[10]．受信した NetFlow データを時間分割し，各分割の同一性を特徴量として提案している. なお，ある 2 つの NetFlow データ同一性を計算する際は Netflow データ内の IP アドレスのサフィックスの長さを用いている. しかし，IP アドレスのサフィックスの長さが攻撃検知に有効である理由の記述が無いことが課題である.

Zhao らは NetFlow と機械学習の組み合わせによりリアルタイムなアノマリ検知を行うシステムの提案を行っている[11]．当該システムでは(*src_port*，*dst_ip*)もしくは(*src_ip*,*dst_ip*)で定義されるフロー数や，(*dst_ip*)もしくは (*src_ip*)で定義されるフローのユニーク数を特徴量としている. また，Apache Hadoop や Apache Storm，Apache Kafka を用いて特徴量を計算し，機械学習と組み合わせる実装を行いアノマリ検知精度の評価を行っている. しかし，具体的なアノマリ検知の目的が記載されておらず，DDoS 検知への応用性については不明である.

3. 特徴量及び計算手法の提案

本章ではまず， 3-tuple フロー内の 5-tuple フロー数の観点における，通常通信と攻撃通信の差異について考察する．その後，低帯域 L3, L4 攻撃検知に向けた特徴量及びその計算手法を提案し，更に計算手法の計算量について考察する.



Proposed Feature Value

$$\max_k N^k_{t_n,t_n+T}$$

図 2 提案特徴量

3.1 特徴量の提案

一般的に，通常のアプリケーションプログラムが Internet Protocol を用いて通信を行う際は，OS のソケット関数を用いて，接続先のアプリケーションプログラムとセッションを構築した上で通信を行う．この際，接続元ホストのアプリケーションが接続先ホストに対し，同時に大量のセッションを構築して通信を行うことは無い．つまり，通常通信における 3tuple フロー中には，5tuple フローが多く含まれない．

一方，攻撃パケットを生成するツールは，あるホストからあるホストへ攻撃を行う際に，*src_port* の値を様々な値に変化させながらパケットを生成するものが多い．つまり，攻撃通信における 3-tuple フロー中には多数の 5-tuple フローが多く含まれる．下記に公開されている攻撃ツールの例を挙げる．

・G3M[12]はオプションにより設定された *src_port* の範囲に基づいて，*src_port* の値を設定しつつ攻撃パケットを生成する．

・T50[13]は *src_port* をランダムな値で設定し，攻撃パケットを生成する．

・hping3[14]はある初期値から徐々に増加させた値を *src_port* に設定し，攻撃パケットを生成する．

以上より，通常時と攻撃時においては 3-tuple フロー中の 5-tuple フロー数に違いがあることが想定される．本稿では，上記の考察に基づき，図 2 のようにある時間区間における 3-tuple フロー中の 5-tuple フロー数の最大値を特徴量として用いることを提案する．なお，最大値を使用する理由は，多数の通常通信の 3-tuple フローに紛れ込んでいる攻撃通信の 3-tuple フローを確実に発見することを目的としているためである．

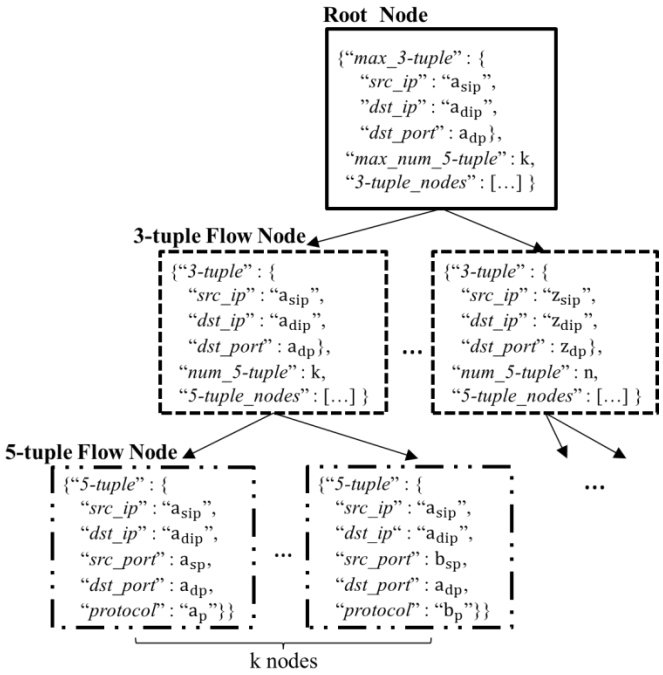


図 3 特徴量計算のためのツリーグラフ

3.2 特徴量計算手法の提案

ルータが NetFlow を用いてフロー情報を送付する際は，まずパケットサンプリングを行うことが一般的である．更にサンプリングされたパケットを設定で定義されたフロー毎にパケット数・バイト数を集計し，一定の間隔でフロー情報として送付する．一方，ルータが sFlow を用いてフロー情報を送付する際も，まずパケットサンプリングを行うことが一般的である．その後，サンプリングされたパケットのヘッダの一部を一定間隔でフロー情報として送付する．これら NetFlow 及び sFlow はどちらもルータを通過するフローの 3-tupe 及び 5-tuple の情報を含んでいる．本稿では，これらの情報を用いて当該特徴量を計算する手法を提案する．特徴量計算時に用いるツリーグラフを図 3 に，具体的な特徴量計算アルゴリズムを図 4 に示す．

提案アルゴリズムは「初期化」「フロー受信」「特徴量送出」の 3 つのアルゴリズムに分かれる．

初期化：ハッシュツリーのような適当なデータ構造を用いて，図 3 に示すような有向ツリーグラフを生成する．当該ツリーは 3 種類のノードにより構成される．

1) 5-tuple Flow Node：5-tuple 情報を保有する．

2) 3-tuple Flow Node：3-tuple 情報と，当該 3-tuple に包含される 5-tuple 情報を持つ 5-tuple Flow Node を持つ．また，5-tuple Flow Node のノード数を持つ．

3) Root Node：全ての 3-tuple Flow Node と，それらの中での 5-tuple Flow Node 数の最大値を保有する．更に，その最大値を保有する 3-tuple Flow Node の 3-tuple 情報を保有する．

Algorithm 1 Initialize

Output: *Root_Node*

```
/* Initialize Root Node */
Construct Root_Node
Root_Node.max_3-tuple ← NULL
Root_Node.max_num_5-tuple ← NULL
Root_Node.3-tuple_nodes ← NULL
Return Root_Node
```

Algorithm 2 Receive Flow Data

Input: *Flow_Data*, *Root_Node*

Output: *Root_Node*

```
/* Extract 3-tuple and 5-tuple from Flow Data */
flow_3-tuple ← get3-tuple(Flow_Data)
flow_5-tuple ← get5-tuple(Flow_Data)
if flow_3-tuple does not exists in Root_Node.3-tuple_nodes then
    /* Initialize 3-tuple Flow Node and append it to Root Node */
    Construct 3-tuple_Flow_Node
    3-tuple_Flow_Node.3-tuple ← flow_3-tuple
    3-tuple_Flow_Node.num_5-tuple ← 0
    3-tuple_Flow_Node.5-tuple_nodes ← NULL
    Append(Root_Node.3-tuple_nodes, 3-tuple_Flow_Node)
end if
3-tuple_Flow_Node ← Root_Node.3-tuple_nodes[flow_3-tuple]
if flow_5-tuple does not exists in 3-tuple_Flow_Node.5-tuple_nodes then
    /* Initialize 5-tuple Flow Node and append it to 3-tuple Flow Node */
    Construct 5-tuple_Flow_Node
    5-tuple_Flow_Node.5-tuple ← flow_5-tuple
    Append(3-tuple_Flow_Node.5-tuple_nodes, 5-tuple_Flow_Node)
    /* Increment num_5-tuple in parent 3-tuple Flow Node */
    3-tuple_Flow_Node.num_5-tuple ← 3-tuple_Flow_Node.num_5-tuple + 1
    /* Update max_num_5-tuple in Root Node */
    if 3-tuple_Flow_Node.num_5-tuple > Root_Node.max_num_5-tuple
    then
        Root_Node.max_num_5-tuple ← 3-tuple_Flow_Node.num_5-tuple
        Root_Node.max_3-tuple ← 3-tuple_Flow_Node.3-tuple
    end if
end if
Return Root_Node
```

Algorithm 3 Output Feature Value

Input: *Root_Node*

Output: *max_num_5-tuple*, *max_3-tuple*

```
max_num_5-tuple ← Root_Node.max_num_5-tuple
max_3-tuple ← Root_Node.max_3-tuple
Free Root_Node
Return max_num_5-tuple, max_3-tuple
```

図 4 特徴量計算アルゴリズム

フロー受信：フロー情報を受信した際は，有向ツリーグラフを成長させる．まず，フロー情報内の 3-tuple 及び 5-tuple 情報を抽出し，Root Node 内に当該 3-tuple 情報を保有する 3-tuple Flow Node が存在するか確認を行う．仮に存在しない場合は，3-tuple Flow Node を新たに生成する．次に 3-tuple Flow Node 内に当該 5-tuple 情報を保有する 5-tuple Flow Node が存在するか確認を行う．仮に存在しない場合は，5-tuple Flow Node を新たに生成する．その際に，3tuple Flow Node 内の 5-tuple Flow Node 数をインクリメントし，インクリメント後の 5-tuple Flow Node 数を Root Node 内の 5-tuple Flow Node 数の最大値と比較する．インクリメント後の 5-tuple Flow Node 数の値が Root Node 内の 5-tuple Flow Node 数の最大値より大きい場合，Root Node 内の 5-tuple Flow Node 数の最大値及びその最大値を保有する 3-tuple 情報を更新する．

特徴量送出：Root Node の情報を出力し初期化する．

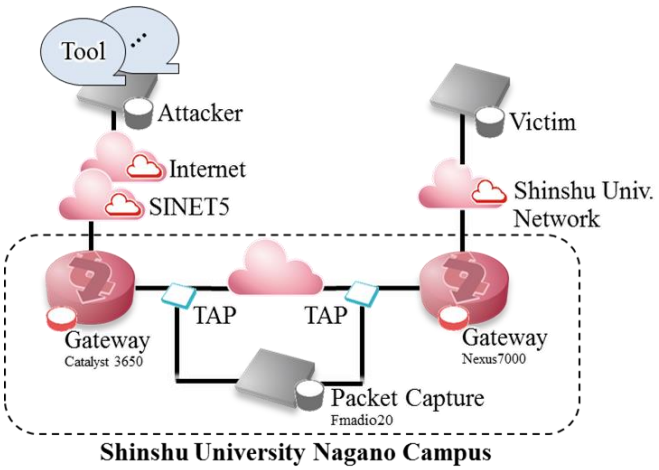


図 5 データ取得環境

表 1 攻撃印可条件

攻撃種	ツール	印可帯域 (3-tuple フロー, 1 分毎)	使用時間
TCP SYN Flood	G3M	1.78 Mbps	2 sec
	T50	2.31 Mbps	2 sec
UDP Flood	G3M	1.24 Mbps	2 sec
	T50	1.78 Mbps	2 sec

3.2 特徴量計算手法の計算量

「初期化」及び「特徴量送出」に関わる計算量は明らかに$O(1)$である．また，「フロー受信」に関わる計算量については，有向ツリーグラフのデータ構造として，ハッシュツリーを用いることで，ツリー内のノード数に関わらず一定時間処理$O(1)$とすることができる．

4. 提案特徴量の評価

本章では 3 章で提案した特徴量の有効性を確認するため，DDoS 攻撃検知における偽陰性及び偽陽性の評価を行った．評価の際は，まず信州大学及び WIDE のネットワークを流れるパケットのキャプチャデータを収集した．その後，フローの 3-tuple 情報と 5-tuple 情報を含む疑似フロー情報を生成し，更に提案した特徴量を計算した．最後に提案特徴量と LOF[15]を用いて異常スコアの時系列値の計算を行った．

4.1 偽陰性の評価

当該特徴量の偽陰性，つまり攻撃発生時に攻撃を見逃すことが無く攻撃検知が可能か評価を行うため，まずパケットキャプチャデータを取得した．

図 5 に示すように，パケットキャプチャ装置を信州大学ネットワークの GW に導入し，ネットワークを流れるパケットのキャプチャを取得し通常相当のデータとした．更に，攻撃相当のデータを取得するため，信州大学ネットワークに攻撃通信が通過するよう，表 1 の条件にて攻撃ツールを使用して攻撃を行い，同じくパケットのキャプチャを取得した．なお，攻撃を行う際は，表 1 の条件で攻撃ツールを短い時間で使用した．

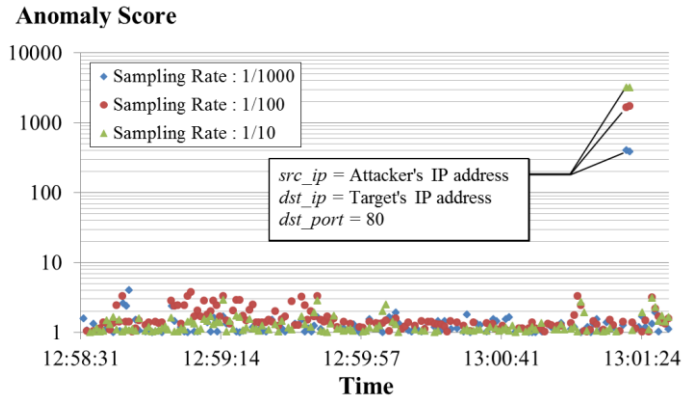


図 6　異常スコア(TCP SYN Flood / G3M)

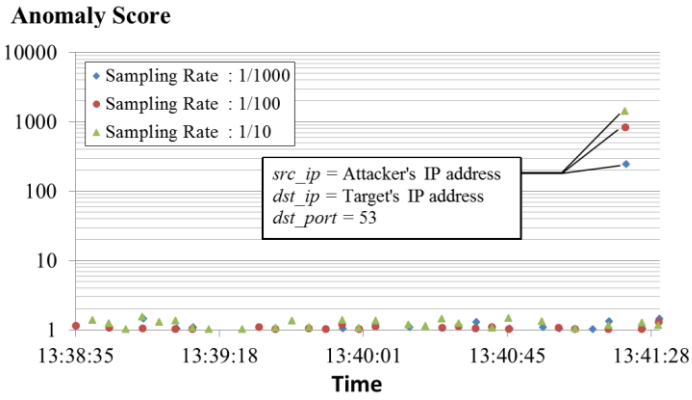


図 8　異常スコア(UDP Flood / G3M)

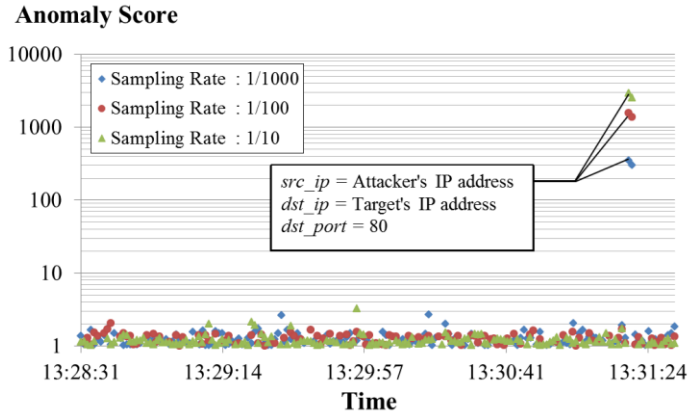


図 7　異常スコア(TCP SYN Flood / T50)

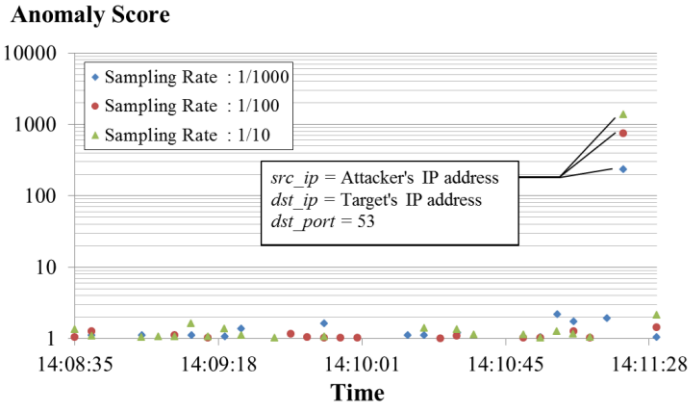


図 9　異常スコア(UDP Flood / T50)

また，DoS 攻撃を検知可能であることは DDoS 攻撃も検知可能であることを意味するため，本評価においては高々数ホストから DoS 攻撃を生成し，取得したパケットキャプチャデータで評価を行った．

通常相当及び攻撃相当のパケットキャプチャデータを取得した後に LOF を用いて異常スコアを計算した．その際はまず，通常時のキャプチャデータからサンプリングレートを 1/10，1/100，1/1000 にそれぞれ変化させながらサンプリングを行った後に，1 分間隔で疑似フロー情報を生成した．その後，フロー情報を基に提案した特徴量を計算し，LOF の学習を実施した．学習後に，攻撃を行った時間帯周辺のキャプチャデータに対しても同様に特徴量を計算し，学習済の LOF に入力することで異常スコアを計算した．図 6～図 9 に異常スコア値のグラフを示す．

図 6～図 9 より，攻撃を行う前後の異常スコア値はオーダが 1 と小さい値である一方，攻撃を発生させた瞬間における異常スコア値はオーダが 100 以上と大きい値となっている．つまり，提案特徴量を基に計算した異常スコアの観点では，通常通信と攻撃通信で明確に差があることがわかる．このことから提案特徴量は，帯域が低い L3, L4 DDoS 攻撃通信を見逃すことが無く，検知が可能であることがわかる．

4.2 偽陽性の評価

当該特徴量の偽陽性，つまり攻撃が発生していない際に誤検知することが無いか評価を行うため，WIDE samplepoint-F[16]で得られたパケットキャプチャデータを収集した．表 2 に評価で使用したデータを記す．まず，表 2 内の#0 のパケットキャプチャデータからサンプリングレートを 1/128，1/512，1/2048，1/8192 に変化させながらパケットサンプリングを実施した．そして，サンプリングされたパケットを用いて 1 分間隔で疑似フロー情報を生成し，疑似フロー情報を用いて提案特徴量を計算した．更に，計算した特徴量を基に LOF の学習を実施した．その後，表 2 内の#1,#2 のキャプチャデータに対しても同様に提案特徴量を計算し，学習済みの LOF を用いて異常スコアの計算を行った．図 10～11 に異常スコアのグラフを示す．

図 10 を見ると，#1 のデータに対する異常スコア値は非常に低いことが解る．このことから，提案した特徴量は偽陽性の観点において，同じ曜日であれば日変動に対してロバスト性があることが言える．一方，図 11 を見ると，#1 のデータから算出された異常スコアと比較して，異常スコアが若干高い値となっている．これについて考察を行う．

表 2　キャプチャデータの使用用途

	日付及び曜日	時間帯	用途
#0	2019/1/20(日)	From 14:00:00 To 14:02:30	LOF 学習用
#1	2019/1/27(日)	From 14:00:00 To 14:15:00	別日，同曜日の異常スコア評価
#2	2019/1/21(月)	From 14:00:00 To 14:15:00	別日，別曜日の異常スコア評価

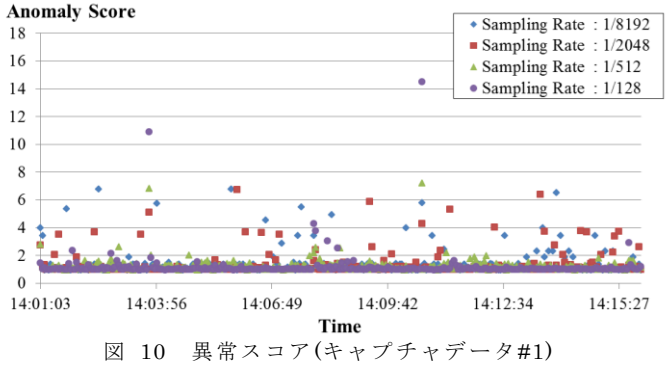


図 10　異常スコア(キャプチャデータ#1)

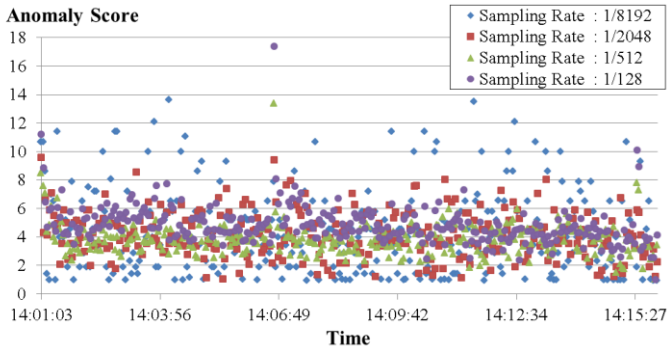


図 11　異常スコア(キャプチャデータ#2)

現在の Web ブラウザは Web サーバに対し，データのやり取りを高速化するため複数コネクションを同時に確立する[17]．そのため，http や https の通信における 3-tuple フロー内の 5-tuple フロー数は 1 から 6 に変動する．また，全ての通信のパケット数に対する http 及び https の通信のパケット数の割合が，平日と休日において異なる[16]．そのため，休日と平日における 3-tuple フロー内の 5-tuple フロー数の傾向が若干異なると考えられる．

4.1 章の結果より，攻撃時における異常スコアの値は，サンプリングレートが 1/1000 の時でさえもオーダが 100 以上の値となっていることが解る．一方，図 10，図 11 における異常スコア値はオーダが 10 程度の値となっている．つまり偽陽性に関して，提案する特徴量は日が変動しても誤検知が少ないと考えられる．

5. おわりに

本稿では，　(*src_ip*, *dst_ip*, *dst_port*)で定義される 3-tuple flow 中に存在する 5-tuple flow 数が通常通信と攻撃通信で異なることに着目し，攻撃検知のための新たな特徴量及びその計算手法について提案した．また，信州大学のネットワークに対し低帯域な攻撃を行いつつ取得したトラヒックデータと，WIDE が公開しているトラヒックデータに対し当該特徴量を計算し，LOF と組み合わせた際の攻撃検知精度の評価を実施した．評価の結果，提案特徴量は偽陰性率及び偽陽性率を低く抑えつつ攻撃検知が可能であることが解った．

文　献

[1] “NETSCOUT Arbor’s 13th Annual Worldwide Infrastructure Security Report,” *NETSCOUT* [online] Available at: ¡https://www.netscout.com/report/¿[Accessed 7 Mar. 2019]

[2] “Attackers Use DDoS Pulses to Pin Down Multiple Targets, Send Shock Waves Throgh Hybrids,” *IMPERVA* [online] Available at: ¡https://lp.incapsula.com/¿[Accessed 7 Mar. 2019]

[3] B. Claise, “Request for Comments: 3954 Cisco Systems NetFlow Services Export V9,” *IETF* [online] Available at: ¡http://www.ietf.org/rfc/rfc3954.txt¿[Accessed 7 Mar. 2019]

[4] P. Phaal, “Request for Comments: 3176 InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks,” *IETF* [online] Available at: ¡https://www.ietf.org/rfc/rfc3176.txt¿ [Accessed 7 Mar. 2019]

[5] M. De Donno, A. Giaretta, N. Dragoni, and A. Spognardi, “A taxonomy of distributed denial of service attacks,” *2017 International Conference on Information Society (i-Society)*, pp. 100-107, Jul. 2017.

[6] J. Hou, P. Fu, Z. Cao, and A. Xu, “Machine Learning Based DDos Detection Through NetFlow Analysis,” *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, pp. 1-6, Oct. 2018.

[7] A. Rukavitsyn, K. Borisenko, and A. Shorov, “Self-learning method for DDoS detection model in cloud computing,” *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pp. 544-547, Feb. 2017.

[8] D. S. Terzi, R. Terzi, and S. Sagiroglu, “Big data analytics for network anomaly detection from NetFlow data,” *2017 International Conference on Computer Science and Engineering (UBMK)*, pp. 592-597, Oct. 2017.

[9] C. Kemp, C. Calvert, and T. Khoshgoftaar, “Utilizing NetFlow Data to Detect Slow Read Attacks,” *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, pp. 108-116, Jul. 2018.

[10] C. Wagner, J Francois, and T. Engel, “Machine learning approach for ip-flow record anomaly detection,” *International Conference on Research in Networking*, Springer, vol. 6640, pp.28-39, 2011.

[11] S. Zhao, M. Chandrashekar, Y. Lee, and D. Medhi, “Real-time network anomaly detection system using machine learning,” *2015 11th International Conference on the Design of Reliable Communication Networks (DRCN)* , pp. 267-270 , Mar. 2015.

[12] “g3m,” [online] Available at: ¡https://github.com/WorkaroundTech/g3m¿[Accessed 7 Mar. 2019]

[13] “T50,” [online] Available at: ¡https://github.com/foreni-packages/t50¿[Accessed 7 Mar. 2019]

[14] “hping3,” [online] Available at: ¡https://github.com/NullHypothesis/hping3¿[Accessed 7 Mar. 2019]

[15] M. Breunig, H. Kriegel, Raymond, T. Ng, and J. Sander, “LOF: identifying density-based local outliers,” *2000 ACM SIGMOD international conference on Management of data (SIGMOD)*, pp. 93-104, May 2000.

[16] “MAWI Working Group Traffic Archive,” *WIDE* [online] Available at: ¡http://mawi.wide.ad.jp/mawi/¿[Accessed 7 Mar. 2019]

[17] M. Belshe, “Request for Comments: 7540 Hypertext Transfer Protocol Version 2 (HTTP/2),” *IETF* [online] Available at: ¡https://www.ietf.org/rfc/rfc3176.txt¿[Accessed 7 Mar. 2019]