リアルタイムバースト検出手法による即応性を考慮したDDoS攻撃検知手法

臼崎　翔太郎[a)]・山場　久昭[b)]・油田　健太郎[c)]・岡崎　直宣[d)]

Highly Responsive Detection Method of Distributed Denial-of-Service Attacks Using Data Mining Technique

Shotaro USUZAKI, Hisaaki YAMABA, Kentaro ABURADA, Naonobu OKAZAKI

Abstract

The damage caused by DDoS (Distributed Denial-of-Service) attack is a big threat for modern society. It is expected that the damage will become bigger, therefore effective attack detection system is desired. In general, DDoS attack detection methods are roughly divided into signature type and anomaly type. The signature type has signature database that stores a pattern of an attack packet. This method detects the attack by comparing its characteristics with the signature every time a packet arrives. However, the more the pattern of registered attack increases, the more the responsiveness decreases because of computational complexity of pattern matching. On the other hand, the anomaly type detects the attack by using statistical information. This method detects attack by comparing statistical information of the current packet series and those of normal case for each window size. However, it has the trade-off relationship between detection accuracy and responsiveness. This is because it is necessary to widen the window size in order to improve the detection accuracy. The detection process is not performed until the window size is exceeded. In order to solve the problem, we propose the anomaly-based DDoS attack detection method using a data mining technique that can process when event occurs, while maintaining sufficient data necessary for detection processing. In this research, we evaluate the detection accuracy and the processing efficiency of the proposed method.

Keywords: network security, DDoS attack detection, data mining

1. はじめに

インターネットが社会基盤となっている現代社会ではDDoS(Distributed Denial-of-Service) 攻撃による被害は大きな脅威となっている。最近ではIoT機器からのDDoS攻撃も確認されていることから[1)]、被害がますます大きくなっていくことが予想され、効果的な攻撃検知システムが望まれている。

DDoS攻撃の検知システムは、シグネチャ型とアノマリ型の二つに大別される。シグネチャ型は、攻撃パケットのパターンをあらかじめ署名と呼ばれるデータベースに保存しておき、パケットが到着するたびに、その特徴を署名と比較することで攻撃を検知する手法である。しかし、このタイプのシステムでは登録した攻撃のパターンが増加するとパターンマッチングの計算量が多くなり、即応性が低下する。また、未知の攻撃にも対応できない問題点がある。それに対して、アノマリ型は統計情報を用いて検知する手法である。アノマリ型では、あるパケット系列の統計情報を、正常時のものと比較することで攻撃か否かを判定する。比較対象となるパケットの系列（以降、窓幅と呼ぶ）は通常、時間かパケット数を単位とする。この手法では、窓幅を大きくすることで検知精度を向上させる。しかし、決定した窓幅を超えるまで検知ができないことから、窓幅を大きくするとその分即応性に欠けてしまうという問題点がある。

そこで我々は蛯名らのデータストリームのバーストを検出する手法[2)]をDDoS攻撃検知に利用することを考えた。本研究では、リアルタイムな解析を可能とし、大量のイベント発生に強いリアルタイムなバースト検出手法を利用し、即応性の高いDDoS攻撃検知手法について検討する。本手法の有効性を確認するために評価実験を行い、検知精度と処理性能について議論する。

2. リアルタイムバースト検出手法

リアルタイムバースト検出手法[2)]はオンラインニュース、ブログといったデータストリームのバーストを解析する手法である。この手法ではデータストリームの特徴となるバーストを検出することを目的としている。バーストとは、イベントの集中発生状態を指す。この手法の利点はイベントの発生ごとに処理を行うことによりリアルタイムな解析を可能にしている点と、ある期間に集中したイベントデータを圧縮することにより大量のデータにも対応できる点である。

2.1 Aggregation Pyramid

本手法で重要なAggregation Pyramidと呼ばれるデータ構造を説明する。このデータ構造は図1のように複数のセルで構成されており、nの階層を持っている。レベルhには$n-h$個のセルが存在しており、新たにセルが生成されるたびに各階層の右側に追加されていく。ここでセルの終了時刻をtと

[a)]工学専攻機械・情報系コース大学院生
[b)]情報システム工学科助教
[c)]情報システム工学科准教授
[d)]情報システム工学科教授

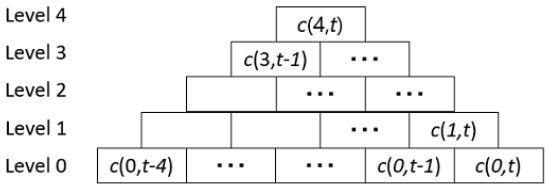

図 1. $n = 5$ の場合の Aggregation Pyramid

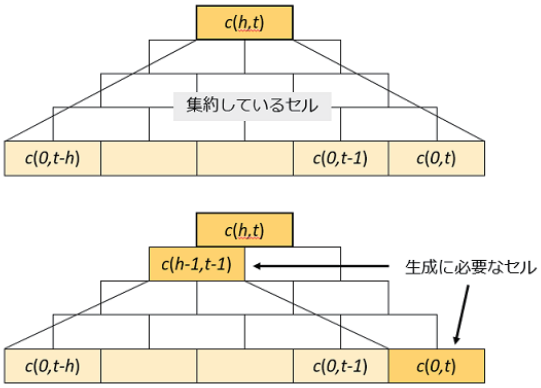

図 2. 上位レベルセルの特徴と生成方法

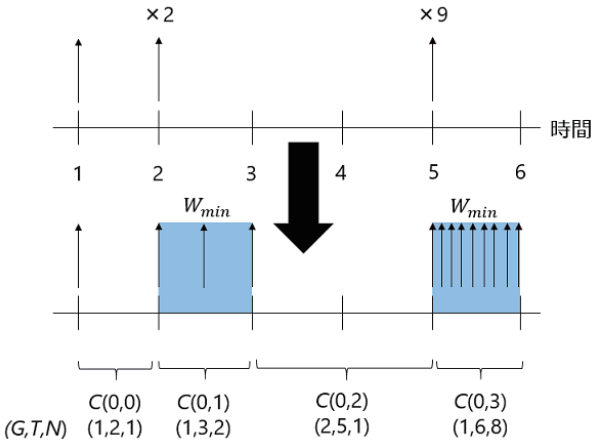

図 3. $W_{min} = 1.0$ の場合の圧縮処理

すると、レベル h のセルは $c(h,t)$ と表現され、それぞれイベントの合計到着間隔 G、到着時刻 T、間隔個数 N の 3 つのイベント情報を保持する。

イベントが発生するたびに、生のイベント情報を保持するレベル 0 のセル $c(0,t)$ を生成する。続いて図 2 上段のように $c(0,t-h)$ から $c(0,t)$ までのセルデータを保持するレベル h のセル $c(h,t)$ を、$c(h-1,t-1)$ と $c(0,t)$ のセルデータを集約することで生成する (図 2 下段)。

2.2 圧縮処理

本手法ではイベントが発生するたびに処理を行うが、短時間に大量のイベントが発生した場合に負荷が大きいため、パラメータ W_{min} を設定し、この間に到着したイベントの情報をひとつのセルに圧縮する (図 3)。このようにイベントの情報を圧縮することで、大量のイベントが発生しても効率よく処理を行うことができる。

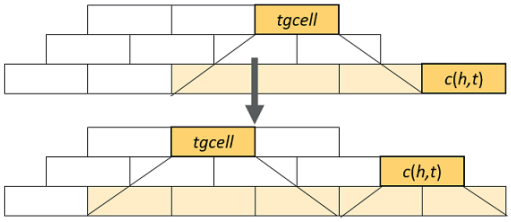

図 4. バースト判定処理における比較対象のセル ($tgcell$)

2.3 バースト判定処理

バースト判定処理ではセル $c(h,t)$ が生成されるたびに、セル $c(h,t)$ のイベント平均到着間隔と、集約している期間が重複していない直前のセル $c(N-1,t-1-h)$(以降、tgcellと呼ぶ) のイベント平均到着間隔を比較する (図 4)。イベント平均到着間隔とはセルで集約しているイベントの到着間隔の平均値である。イベントが発生した時に判定処理を行うため、一定時間ごとに解析を行う手法に比べて無駄な計算が削減され、バーストのリアルタイムな解析が可能となる。

ここで、各セルを同じ状況で比較するためのイベント平均到着間隔関数を式 (1) として、バーストの強さを表現するバースト性 [3] を式 (2) と定義する。

$$avg(c(h,t)) = \frac{G(c(h,t))}{N(c(h,t))} \tag{1}$$

$$brt(c(h,t)) = \frac{avg(c(h,t))}{avg(tgcell)} \tag{2}$$

そして、バーストを判定するパラメータ $\beta(0 < \beta < 1)$ を設定し、閾値として式 (3) が満たされる時、バーストと判断する。

$$brt(c(h,t)) \leq \beta \tag{3}$$

次にパラメータ A_{min} を設定する。バースト判定処理は $N(c(h,t)) \geq A_{min}$ を満たした場合に行う。これによって、過剰なバースト検出を抑制する。

3. 提案手法

本研究では、リアルタイムバースト検出手法の利点に着目し、監視イベントをパケットの到着とすることで DDoS 攻撃の検知に利用する。これによって、即応性と処理効率の高い攻撃検知が可能となる。以前の研究 [4] では、式 (3) によってバーストと判定される時に攻撃としていた。しかしリアルタイムバースト検出手法を単に利用するだけでは、長時間継続する攻撃に対応できない、過剰に攻撃判定処理をしてしまうという問題点がある。ここからは本研究が追加あるいは変更した箇所について説明する。

3.1 攻撃の開始判定と継続判定

式 (3) を用いた攻撃検知手法は短時間に発生する攻撃の検知には向いているが、長時間継続する攻撃の検知には向いていない。なぜならば、攻撃が継続した際には直前のセルと現在のセルのパケット平均到着間隔の差が次第に小さくなってしまい、結果的に検知できなくなってしまうからである。そこで、本研究では最初に式 (3) によって攻撃の開始を判定したのちに、攻撃継続判定処理に遷移するようにした。攻撃継続判定処

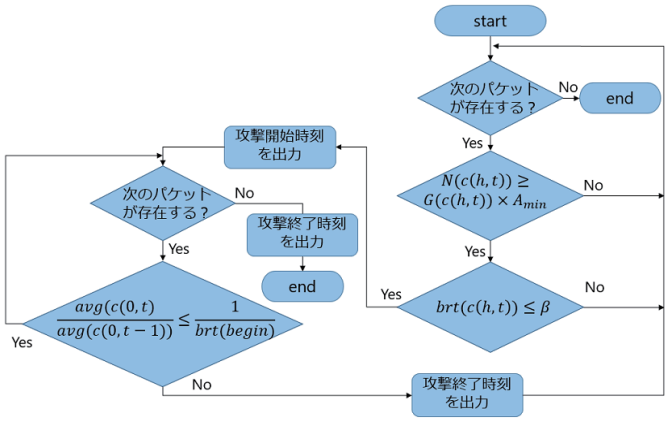

図 5. 提案手法の攻撃検知処理の流れ

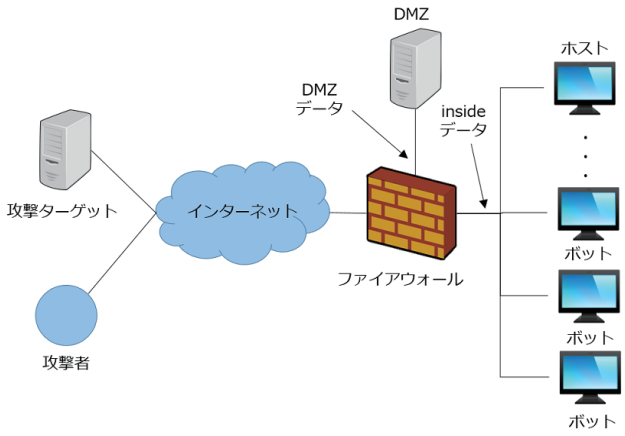

図 6. DARPA2000 の環境の模式図

理では式 (4) を満たした際に攻撃が継続していると判定する。満たさない場合は攻撃が終了したと判断し、$T(c(0,t-1))$ を攻撃終了時刻として出力した後、攻撃開始判定処理に遷移する。提案手法の攻撃検知の流れを図 5 に示す。

$$\frac{avg(c(0,t))}{avg(c(0,t-1))} \leq \frac{1}{brt(begin)} \tag{4}$$

$brt(begin)$ は攻撃開始を検知した際のバースト性を示している。比較対象のセルは n 個前までのセル情報を集約した $tgcell$ ではなく、直前のレベル 0 セルである。これは継続を判定する場合には遠い過去の情報を必要としないためである。本研究では、攻撃開始を検知した時刻と攻撃継続が終了した時刻までを攻撃判定期間とする。なお、攻撃開始と判定した直後の次のセルで攻撃終了と判定したために攻撃開始時刻と終了時刻が等しくなった場合は攻撃と判定しなかった。

3.2 A_{min} の調整

過剰な攻撃判定を防ぐため、攻撃の疑いがある間隔個数である時のみに攻撃検知と判定するように、本研究ではパラメータ A_{min} を非攻撃時における 1s 当たりの間隔個数と定義する。

元々の手法では、全てのセルにおいて合計到着間隔 G を考慮せずに間隔個数 N と A_{min} の値とを比較していた。しかし高いレベルのセルになれば集約しているセル数が多くなることから間隔個数が増加し、異常時でなくとも $N(c(h,t)) \geq A_{min}$ の式を満たしてしまう。そこで G を考慮して A_{min} の調整を行うようにした。本研究では、$N(c(h,t)) \geq G(c(h,t)) \times A_{min}$ を満たすときに攻撃開始判定処理を行う。

4. 評価

検知精度と処理性能を比較して提案手法の有効性を確認する。実験結果は本研究と同じく即応性に着目した小島らの手法 [7] と比較する。実験環境を表 1 に示す。Ubuntu 16.04.1 が稼働する PC 上に、Ubuntu 12.04.5 が動作する仮想環境を構築し、ゲスト OS 上で実験を行った。このようにした理由は、小島らの手法と環境が極力同じになるようにするためである。ゲスト OS はメモリ 1GB、CPU1 コアで稼働している。

実験データとして MIT Lincoln Laboratory が人為的に作成した、DDoS 攻撃のデータセットである DARPA2000[5] を用いた。DARPA2000 では DDoS 攻撃を行うシナリオとして次の 5 段階を想定している。

表 1. 実験環境

CPU	Intel(R) Core i7-4770 @ 3.40GHz
メモリ	8GB
OS	ホスト OS: Ubuntu 16.04.1 LTS, ゲスト OS: Ubuntu 12.04.5 LTS(メモリ 1GB 1 コア)
開発環境	C++

Phase1 ターゲット組織に対して IP スキャンを行い、稼働しているホストを調査。

Phase2 稼働しているホストに対して sadmind デーモンの有無を調査。

Phase3 sadmind の脆弱性を利用しシステムに侵入。

Phase4 3 台のホストに DoS 攻撃ツールの mstream をインストールしボット化。

Phase5 ボットに、外部組織に対する DDoS 攻撃の開始を指示。

DARPA2000 では組織のファイアウォールの内部で観測された inside データと外部で観測された DMZ データが提供されている (図 6)。本研究では、総パケット量の多い inside データを利用し、DDoS 攻撃部分のみを攻撃とみなして実験を行う。

パラメータは全ての実験において、$n = 50, \beta = 0.01, W_{min} = 1.0, A_{min} = 490$ に設定している。A_{min} の値は、通常時の特徴を抽出する王らの手法 [6] を用いて通常時の間隔個数を求めたものである。具体的には、DARPA2000 と同程度の規模のバックグラウンドトラフィックが流れるデータセット DARPA1999[5] を利用し、攻撃が含まれない火曜日のキャプチャデータを学習させて計算を行った。火曜日のデータを利用しているのは、DARPA2000 が火曜日に観測されたデータであるためである。

4.1 提案手法の検知精度

攻撃検知システムにおいて、誤検知である False-Posive(FP) と False-Negative(FN) は、一般的に一方を少なくするともう一方が増加する傾向にある。そこで、本研究では FP と FN を総合的に評価する F 尺度を検知精度の評価に利用する。F 尺度の計算式を式 (5) に示す。F 尺度は値が大きいほど良いとされる。

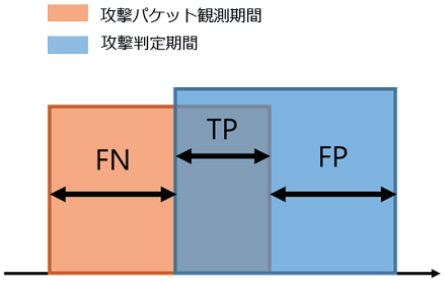

図 7. True-Positive、False-Negative、False-Positive の求め方

表 2. F 尺度の結果

適合率	再現率	F 尺度	既存手法の F 尺度 [7]
0.895	0.917	0.906	0.993

$$F = \frac{2PR}{P+R} \tag{5}$$

ここでの P と R はそれぞれ適合率 (Precision)、再現率 (Recall) と呼ばれる値でそれぞれ式 (6)、式 (7) によって定義される。

$$P = \frac{tp}{tp+fp} \tag{6}$$

$$R = \frac{tp}{tp+fn} \tag{7}$$

F 尺度は、提案手法によって出力された攻撃検知期間と実際の攻撃観測期間を比較し図 7 のように TP(True-Positive)、FP、FN を計算することで求まる。

実験の結果、攻撃パケット観測期間と攻撃判定期間とを図示したものが図 8 である。縦軸はパケット数、横軸はキャプチャデータのうちパケットが最初に観測された時刻からの経過した時間（秒）である。赤く網掛けされている箇所が実際に攻撃が観測されている期間で、青く網掛けされている箇所が提案手法によって攻撃だと判定した期間である。F 尺度の結果を表 2 に示す。提案手法の F 尺度は 0.906 となっており、既存手法に比べて低い値となっていることから検知精度が下がってしまっていることが分かる。しかし、提案手法においても適合率、再現率がともに 0.9 付近の値になっていることから十分な精度は持っていると考える。また、本研究ではパケットの到着情報のみを利用して検知を行っているため、他の統計情報を利用することで検知精度が向上すると考えられる。

4.2 提案手法の処理性能

処理性能は、DARPA2000 の inside データの総パケット (649,787 個) の処理に要した時間で評価を行う。提案手法の値は、実験を 10 回行った際の平均値となっている。結果は表 3 に示す通りである。結果から提案手法は既存手法よりも 60.873s 速く処理を終えており、既存手法時点との CPU クロック周波数の違いを考慮しても、処理性能は提案手法の方が高いことが分かる。また、1 パケット当たりの処理時間を計算すると 0.195μs となっているため、即応性は十分に高いと考えられる。

表 3. 総処理時間の結果

提案手法	既存手法
0.127s	61s

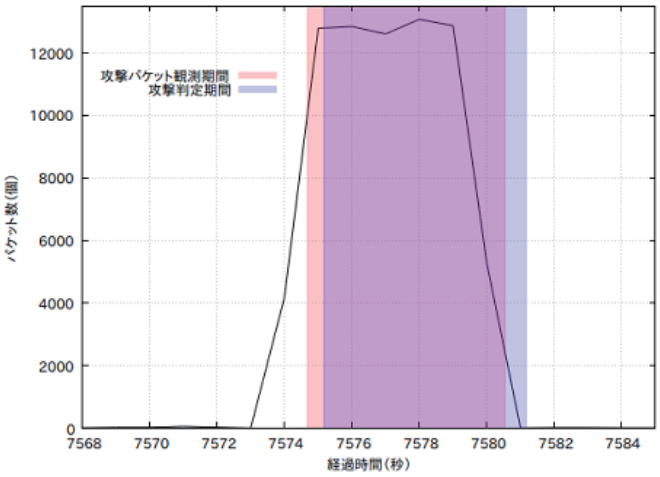

図 8. 攻撃観測期間と提案手法による攻撃検知期間

5. まとめ

本研究では、リアルタイムバースト検出手法を利用した DDoS 攻撃の検知手法を検討した。パケットが到着するごとに攻撃検知処理を行うことによりリアルタイムな解析が可能となる。データ圧縮処理によって大量のパケットが到着しても効率的に対応できる。評価実験の結果、検知精度は既存手法と比較すると低いものの、十分な検知性能を持っていた。現在は攻撃検知にパケット到着の情報のみを利用しているため、他の統計情報を利用することで検知精度が向上すると考えられる。また、既存手法よりも処理性能が高く、1 パケット当たりの処理時間が 0.195μs となっていることから、圧縮処理によって複数のパケットを処理することを考慮しても即応性が高いと分かる。今回 A_{min} 以外のパラメータ値は経験的に設定したものであるため、今後の課題としてパラメータ値の有効な値を自動的に獲得する仕組みが望まれる。特に β は通常時と攻撃時のパケット流入量によって大きく変動するので、検討が必要である。また、今回は全てのパケットを監視対象としているが、ある攻撃に特有なパケットを監視することで多様な攻撃に対応できると考えている。

参考文献

1) 鈴木聖子: 史上最大級の DDoS 攻撃に使われたマルウェア「Mirai」公開、作者が IoT を悪用, ITmedia エンタープライズ 〈http://www.itmedia.co.jp/enterprise/articles/1610/04/news046.html〉 (accessed 2017/01/25).

2) 蛯名亮平、中村健二、小柳滋: リアルタイムバースト検出手法の提案, 日本データベース学会論文誌,Vol.9, No.2, pp. 1–6, 2010.

3) 蛯名亮平、中村健二、小柳滋: リアルタイムバースト解析手法の提案, 情報処理学会論文誌 データベース,Vol. 5, No. 3, pp. 86–96, 2012.

4) 臼崎翔太郎、橘弘智、有川佑樹、高塚佳代子、山場久昭、久保田真一郎、岡崎直宣: リアルタイムバースト検出手法を利用したパケット到着間隔による DDoS 攻撃検知手法の検討, 電気・情報関係学会, 第 69 回電気・情報関係学会九州支部連合大会, p. 270, 2016.

5) MIT：DARPA Intrusion Detection Evaluation Data Set

⟨https://www.ll.mit.edu/ideval/data/index.html⟩ (accessed 2017/01/25).

6) 王サン、フォンヤオカイ、川本淳平、堀良彰、櫻井幸一: 挙動に基づくポートスキャン検知の自動化に向けた学習アルゴリズムの提案とその性能評価, 情報処理学会論文誌, Vol. 56, No. 9, pp. 1770–1781, 2015.

7) 小島俊輔、中嶋卓雄、末吉敏則: エントロピーベースのマハラノビス距離による高速な異常検知手法, 情報処理学会論文誌, Vol. 52, No. 2, pp. 656–668, 2011.

8) S. Usuzaki, Y. Arikawa, H. Yamaba, K. Aburada, S. Kubota, M. Park, and N. Okazaki: A Proposal of Highly Responsive Distributed Denial-of-Service Attacks Detection Using Real-Time Burst Detection Method, Journal of Information Processing (in printing).