Vol.2014-MPS-101 No.1
2014/12/9

Hadoopを用いた遺伝的アルゴリズムによるDDoS攻撃防止システムの検討

水越 大貴[1,a)]　棟朝 雅晴[2,b)]

概要：インターネットにおけるセキュリティ上の脅威の一つとしてDDoS攻撃が深刻な問題になっている。DDoS攻撃は一般に攻撃元の情報が改竄されているため、攻撃元の特定が非常に困難であり防ぐ事が難しい攻撃だといえる。また、攻撃パターンの学習によるパターンマッチングや、異常トラフィック検知などの手法が研究されているが、DDoS攻撃において攻撃者はボットネット等を使用し、常に異なるトラフィックパターンの攻撃を仕掛けてくるため、過去のデータの解析からDDoS攻撃を防ぐ事は非常に難しい。このような背景から、ネットワーク管理者は常に現在どのような攻撃が行われているかを監視、解析し、DDoS攻撃に対処する必要性に迫られる。しかし、近年ネットワーク上を流れるトラフィック量は急激に増加しており、トラフィックの解析にはかなりの時間かかってしまう事が予測される。そこで本稿では、DDoS攻撃に対し迅速に対処するシステムを作る事を目的とし、並列分散処理基盤であるHadoop上での遺伝的アルゴリズムを使用したトラフィックパターンの解析手法を提案する。

キーワード：DDoS攻撃 , Hadoop , 遺伝的アルゴリズム

Mizukoshi Masataka[1,a)]　Munetomo Masaharu[2,b)]

Abstract: DDoS attacks become serious as one of the menaces of the Internet security. It is difficult to prevent because DDoS attacker send spoofing packets to victim, which make the identification of the origin of attacks very difficult. A series of techniques have been studied such as pattern matching by learning the attack pattern and abnormal traffic detection. The pattern matching approach, however; has difficulty because if algorithm learned from past DDoS data, attacker always set attacks of a different traffic pattern. Therefore, the network administrator has to watch what kind of attacks are carried out now, and investigate how to prevent DDoS attacks. The quantity of traffic to flow through the Internet increases rapidly, so the packets analysis takes considerable computation time. In this paper, we design a system to detect and prevent DDoS attacks promptly by proposing a technique to analyze traffic patterns using a genetic algorithm implemented on Hadoop distributed processing infrastructure.

Keywords: DDoS attack , Hadoop , Genetic Algorithm

1. はじめに

クラウドサービスの普及に伴い、可用性の確保はサービス提供者にとって重要な問題である。このような中で、分散サービス妨害（DDoS：Distributed Denial of Service）攻撃は非常に重要なインターネットセキュリティにおいての脅威となっている。DDoS攻撃では攻撃者がネットワーク上でセキュリティの脆弱性のある複数のコンピュータに攻撃用のプログラムを侵入させ、標的のサーバーや回線に対して大量のパケットを送る。標的のサーバーには処理能力を上回るリクエストが送られてくるため、提供するサービスを一時的な遅延または停止させられる事になる。

一般的にDDoS攻撃は攻撃元を特定する事が難しいため、標的となっているホスト側ではなく幅広いルーティング情報を持っているネットワーク側（AS側）での検知、防

[1] 北海道大学情報科学研究科 Graduate School of Information Science and Technology Hokkaido University
[2] 北海道大学情報基盤センター lnformation Initiatlve Center , Hokkaido University
[a)] m.mizukoshi@ist.hokudai.ac.jp
[b)] munetomo@cc.hokudai.ac.jp

ⓒ 2014 Information Processing Society of Japan

止の研究が主流となっている。AS 側での DDoS 攻撃への代表的な対策として、IP トレースバック、プッシュバックアルゴリズムなどの手法があげられる。IP トレースバックは送信パケットに送信元情報をハッシュ関数によりマーキングするダイジェスト方式や、標的となったルーターから最寄りのルーターをたどりパケットの送信元を探知するブラックホール方式など、様々な方法が研究されている [6][18]。プッシュバックアルゴリズムでは各ルータで行う帯域制限を隣接する子ルータに伝えることでネットワーク全体の帯域浪費を抑えようとする技術であり、帯域制限と共に不正アクセスのシグネチャを伝播させる事でルータ同士が強調し不正アクセスのフィルタリング精度を高める研究も行われている [17][12]。しかし、IP トレースバック、プッシュバックアルゴリズムはいずれも検知精度や秘匿性などが、計算資源や専用ルーター導入のコストとトレードオフの関係にある。さらに攻撃元の特定には AS 同士の協力が不可欠であることから、導入には未だ障害が多い。

そこで本稿では、ネットワーク側からの協力無しに、ホスト側での DDoS 攻撃の防止緩和の方法について述べる。ホスト側で行う DDoS 攻撃の対策として、パケット情報のパターン認識によるフィルタリングルールの生成や、パケットの到着パターンから DDoS を検知する手法などが研究されている [16][11][5][4][13]。しかし、DDoS 攻撃には数多くの手法が存在し、攻撃手法は現在でも日々変化を続けているため、フィルタリングの精度をあげる事は難しい。さらに、数千〜数万台のサーバーを要する大規模なクラウド環境下では膨大なトラッフィック量があり、この中からパケットを解析するには膨大な時間と計算コストがかかってしまうという問題がある。

上記のようにホスト側での DDoS 攻撃対策には「新しい手法の攻撃手法に対する順応と検知精度」と「大規模なクラウドシステムに対するスケーラビリティ」の二つの問題が考えられる。本稿で提案するシステムでは、大規模並列分散処理基盤である Hadoop 上で分散型遺伝的アルゴリズムを用いたフィルタリングルールの生成を行う事で、スケーラブルかつ迅速なパケット解析を行い問題を解決する。

2. 関連研究

ここでは過去の研究の中で特に本稿で提案する DDoS 攻撃防止システムに関連する研究を紹介する。

2.1 パターン認識による DDoS 攻撃の検出

パターン認識による DDoS 攻撃検知システムは、各ルータの負荷状況等に基づく経路情報を用いて攻撃パターンを識別する手法と、パケットキャプチャリングによりパケットの中身を解析し分類する手法、またはその二つを組み合わせた手法などが研究されている [4][7]。いずれの手法でもニューラルネットワークやベイジアンネットワークなど、解析には多くの時間がかかる事がボトルネックになっており、パケットやトラフィックパターンの分類の際に入力する学習データを最適化する方法が検討されている事が多い。

2.2 ネットワーク通信解析と Hadoop

近年インターネット上での通信量は爆発的に増加しており、Cisco によると世界の IP トラフィックは過去５年間の間に８倍に増えており、2016 年には 1.3ZB に達するとされている [19]。そのような現状に伴い、トラフィック解析の分野にも大規模並列分散処理基盤である Hadoop を応用した研究が行われている [8][9]。Hadoop1.0 系列では、その特性上バッチ処理によるデータの解析が主であったため、通信解析でも単純な全体の通信量や送信元や頻度などを把握することに使われている。しかし本稿で使用する Hadoop2.0 で動作する Apache Spark[20] ではインメモリ型のデータ処理によりストリーム処理や反復処理など、より包括的な分散データ処理のフレームワークを提供している。

2.3 エントロピーよる DDoS 攻撃の発生検知

パケットの送信元や到着率などからエントロピーを計算し DDoS 攻撃の発生を検知する [15][14][3]。単純な機械の負荷で DDoS 攻撃を受けているか否かを判断しようとすると、フラッシュクラウドと間違えてしまう可能性が高い。そこで、DDoS 攻撃では送信元 IP アドレスが偽造されたパケットが大量に届くため、一定時間での送信元 IP のエントロピーが増大する事を利用し DDoS の発生検知を行っている。

3. 提案手法

ここでは本稿の提案手法である「Hadoop2.0 を基盤とした DDoS 攻撃防止システム」について説明する。はじめの項で述べたように、パターンマッチング手法による DDoS 攻撃フィルタリングルールの生成には以下の 2 つの問題がある。

日々変化し続ける攻撃手法に対しての対応

現在 DDoS 攻撃の手法は多数存在しており、攻撃者の手腕によってパケットパターンやトラフィックパターンを変化させることができるため、フィルタリングルールは環境に応じ常に変化していく必要がある。

スケーラビリティの問題

インターネットの普及に伴いネットワーク上の通信量は急増しており、通信データの解析には膨大な時間を要する。

そこで、「攻撃発生の検知」「トラフィックの分析」「攻撃への対策（パケットフィルタリングアや通信回線の制御）」の３段階を Hadoop を用いたリアルタイムなパケット解析

により自動化し、常に新しいフィルタリングルールを作り続けることで、日々変化する DDoS 攻撃に対応したスケーラブルなシステムを提案する。

3.1 アーキテクチャ

ネットワーク内でキャプチャされたトラフィックは図 1 のような手法で解析し、DDoS フィルタリングルールを生成する。

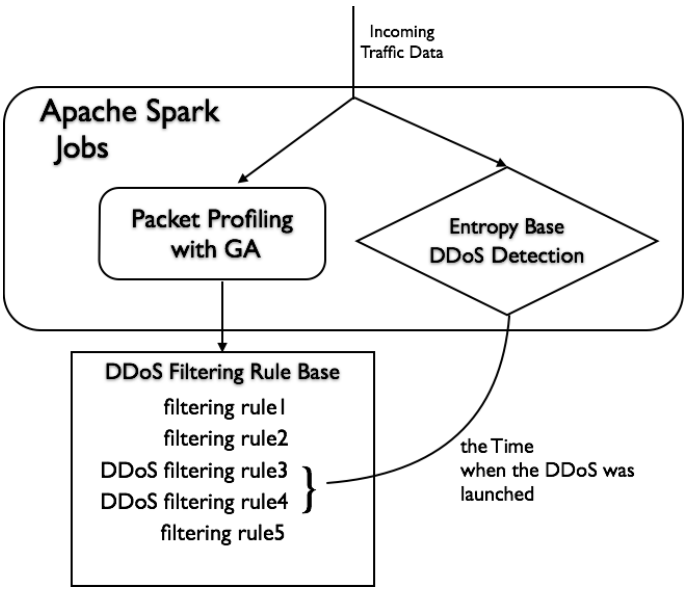

図 1 アーキテクチャ：Apache Spark の中での GA によるパケットプロファイリングとエントロピーを基にた DDoS 発生検知の 2 つのジョブの連携により **DDoS Filtering Rule Base** を作成する。

トラフィックデータは Apache Spark の中のジョブである **Packet Profiling with GA** と **Entropy Base DDoS Detection** の両方に送られる。**Packet Profiling with GA** では、遺伝的アルゴリズムによつパケットの特徴抽出により、現在のパケットの状態をプロファイルしたものをフィルタリングルールとして DDoS Filtering Rule Base へと出力を続ける。**Entropy Base DDoS Detection** では、パケットの送信元や送信頻度の情報を元に、現在 DDoS 攻撃が行われているかどうかを判断し、DDoS 攻撃が行われた時間の情報を **DDoS Filtering Rule Base** に伝える。**DDoS Filtering Rule Base** では、DDoS 攻撃が行われていた時間にキャプチャされたパケットの特徴データを、ミスユース型のフィルタリングルールとし、その他をアノマリ型のフィルタリングルールとして扱う。ここで述べている「アノマリ型のフィルタリングルール」とは、トラフィックの通常状態をパケットプロファイリングにより定義し、通常状態から逸脱したものを検知するルールである。また反対に、「ミスユース型のフィルタリングルール」とは、DDoS 攻撃のパケットをプロファイリングし、型が一致したパケットを DDoS 攻撃として検出するルール（シグネチャ）である。

このようにシグネチャ型とミスユース型の 2 つのフィルタリングルールを混合することで、より精度の高い DDoS フィルタリングルールベースを作成する。

このようにして更新されて行く **DDoS Filtering Rule Base** を実際のネットワーク内のパケットフィルタリングに使う事により、一定時間前に行われた DDoS 攻撃の特徴を正確につかんだパケットフィルタリングを行う事ができる。このシステムにより、長期または断続的に続く DDoS 攻撃を途中で事ができ、ららに一度行われた攻撃と類似した種類の攻撃は未然に防ぐ事が可能であると考える。

3.2 Hadoop2.0 , Apache Spark

Hadoop は Apache Software Foundation で開発、公開がされている分散処理基盤である。Hadoop は数百から数千ノードの計算機を管理し、簡単に分散処理プログラムを実行する事ができるため、ビックデータの処理に優れている。Hadoop2.0 系では HDFS 上に YARN と呼ばれるフレームワークが追加されており、YARN では従来の MapReduce よりもより汎用的な分散処理プレームワークやアプリケーションの作成が容易となっている。本稿ではこの YARN フレームワーク上で動く分散処理基盤である Apache Spark を使用している。Spark は Hadoop を補完してビックデータアプリケーションや様々なデータ処理の迅速な開発を実現するオープンソースの並列処理フレームワークである。Spark ではインメモリー型の処理により処理の高速化を実現しており、また Hadoop1.0 系でサポートされていた分散型のバッチ処理だけでなく反復アルゴリズムやストリーミング処理もサポートされ、ビックデータのより複雑な解析が可能になる。提案手法では、Spark で可能となったストリーム処理、反復処理をパケット解析に利用している。

3.3 エントロピーをもとにした DDoS 検知

ここではエントロピーを用いた DDoS 攻撃の検知手法について説明する。エントロピーを以下のように定義する。

$$P_i = \frac{\text{送信元 IP アドレス (i) の頻度}}{\text{頻度の総数}} \tag{1}$$

$$H = -\sum_{i=1}^{n} P_i \log P_i \tag{2}$$

上式のように、一定時間内のパケットの送信元に対する頻度を求める。さらに、送信元 IP アドレスの頻度を頻度の総数で割ったものを (1) 式の確率として、一定時間内のエントロピーを (2) 式により求める。DDoS 攻撃は多くの場合パケットの送信元が偽造されているため、通常よりも多くのアドレスが観測される事から、DDoS 攻撃を受けた際は上記の式により計算したエントロピーの値が通常時よ

りも大きクなる事が予想できる。よって一定時間ごとにエントロピーを測り続け、時間ごとのエントロピーの変化量を見る事で DDoS 攻撃を検知することが可能になる。本稿のシステムでは Spark 上のストリーミング処理によりエントロピーの計算を行う事で、常にキャプチャし続けるパケットに対してリアルタイムで処理を行う。

3.4 GA によるフィルタリングルールの生成

遺伝的アルゴリズムによりパケットの特徴を抽出する。本稿では Spark の分散処理フレームワークの中で島モデル遺伝的アルゴリズムを実装する事により、大規模なデータに対しても適応可能なアルゴリズムを提案する。

3.4.1 遺伝的アルゴリズム

遺伝的アルゴリズム (Genetic Algorithm , GA) とは、生物の環境適応による進化の過程を工学的に模し、近似解を求める探索手法である。GA では問題に対する解候補の個体を文字列上で表現し、個体集団の中で選択、交叉、突然変異などの遺伝的操作を繰り返す事により最適解を探索する。また問題に応じた評価関数で各個体の適応度を計算し、適応度の高い個体が優先的に選択されることで集団全体が多様性を保ちつつ最適解へと収束していく。

今回パケットの特徴抽出に GA を使用することで、解候補の個体をそのまま検知ルールとして扱う事ができるため軽量なルールベースを作る事ができ、さらに次に説明する島モデル遺伝的アルゴリズムにより大きなデータセットに対して対応可能なアルゴリズムを作る事ができる。

3.4.2 島モデル遺伝的アルゴリズム

島モデル GA は、GA の遺伝子個体の母集団を複数のサブ集団に分割し、それぞれで独立の GA を行う。このままではサブ集団ごとに独立した解の探索が行われるだけなので、一定世代ごとに移住という処理を加える。移住ではサブ集団は移住相手となる別のサブ集団をランダムに選び、一定量の個体を移動させる。こうすることで各集団でそれぞれの局所解に収束してしまうことを防ぎ、多様性が維持される。

島モデル GA ではサブ集団での GA の操作を複数の計算リソースで分担し並列化する事ができ、各計算ノード間の通信も移住世代のみで行われるため計算時間の短縮を測る事ができる。さらに、各集団固有の個体が育つ事で集団の多様性が維持されやすいという特徴がある。

3.4.3 解の表現

本稿では GA により IF/THEN のルールを作る事を目的とする。GA によりパケットの特徴を抽出する際の解の表現方法と評価関数については過去にも研究が行われている [2][10]。通信パケットの中には Time to Live や tcp、udp_window_size などの数値の大きさが意味を持つ値と、port_number や tcp_flags など数値そのものが意味を持つ値が混在する。そのため過去の論文では

port_number=20 and tcp_window_size=16030 ⇒[2 ,0 ,1 ,6 ,0 ,3 ,0]

のように要素を一桁ごとに区切り、遺伝子を表現をしていた。しかし上記の方法でパケットの特徴を網羅的に表現使用とすると遺伝子の数が膨大になり、計算量が多くなってしまう。そのため本稿では値の連続性と解が収束するまでの計算時間を考慮し、パケットの特徴をそのまま実数値として遺伝子に反映した実数値 GA を用いることとする。また、遺伝子にはどんな値にもマッチするワイルドカードを#として表現する。ワイルドカードの存在により、パケットの特徴を表すルールの中で重要度の低いもの (値がランダムに近く、特徴を持たないもの) がワイルドカードで表現される。

[#,6,2,#,34362,557,#,#,3980,2805,#,1,#,0,#,0,#,45717,44,#,0,#,#,0,#,35,#,#,0,#]

上は使用する遺伝子の例である。右から ip.ttl、ip.proto、ip.flags、ip.checksum....etc のように値が入っており、Table2 は上記の遺伝子にパケットの特徴を対応させたものである。Table2 で示すように 30 個のパケットの値をそれぞれの遺伝子の要素に対応させている。

表 1 解の表現：例

iip.ttl #	p.proto 6	ip.flags 2	ip.checksum #
tcp.srcport 34362	tcp.dstport 557	tcp.stream #	tcp.len #
tcp.seq 3980	tcp.nxtseq 2805	tcp.ack #	tcp.flags.ack 1
tcp.flags.push #	tcp.flags.reset 0	tcp.flags.syn #	tcp.flags.fin 0
tcp.flags.urg #	tcp.checksum 45717	udp.srcport 44	udp.dstport #
udp.length 0	udp.checksum #	udp.stream #	icmp.type 0
icmp.code #	icmp.checksum 35	icmp.ident #	icmp.seq #
icmp.seqle 0	tcp.window.size #		

3.4.4 評価関数

実際に観測されたパケットと個体の要素がどれだけマッチしているかにより、以下のように適応度を決める。解の要素がとり得る最大値により要素全体を割る事で、0から1までの値に正規化する。個体の要素の値を x_j、観測されたパケットの要素の値を p_j とする。

$$d_j = x_j - p_j \tag{3}$$

$$F_j = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(d_j - \mu)^2}{2\sigma^2}) \tag{4}$$

ⓒ 2014 Information Processing Society of Japan

$$fitness = \sum_{j=1}^{n} F_j(x) \quad (5)$$

(3) 式により個体要素と観測されたパケットの要素の差を求める。２つの値の近さによって報酬が決めるため (4) 式により得る報酬を計算し、全ての要素の報酬を足し合わせることで最終的な適応度を (5) 式により求める。ガウス関数である (4) 式のパラメータである σ を設定する事で、個体にどの程度の誤差を認めて報酬を与えるかを調整する事ができる。また、個体の要素がワイルドカードである場合は一定の報酬を与える。この報酬が大きすぎると最適解を占めるワイルドカードの割合が多くなり、パケットの特徴を表現できない可能性があるため、今回の実験はこの報酬を小さく設定した。実験では σ=0.2、μ=0、ワイルドカードによる報酬は $d_j - \mu = 0.5$ のときの適応度 ((5) 式) としている。

3.4.5 Spark での GA の実装

Spark の繰り返し処理とメモリベースの分散ファイルシステムである Tachyon を使い島モデル GA を実行する。

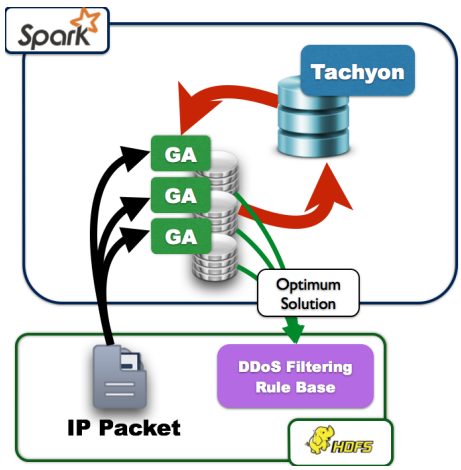

図 2 Apache Spark による島モデル型 GA の実装アーキテクチャ

図 2 のように GA の評価関数として使われる IP パケットが各ノードに分散し、ノードごとに遺伝的操作が実行される。分散データベースである Tachyon に保存したデータは Spark を構成する全てのノードから参照可能になるため、各サブ集団で更新された個体は移住世代になったときに Tachyon に保存する。次の世代の遺伝的操作を行う前に各ノードで Tachyon からサブ集団を読み込む。この時に依然の個体集団の一部を別の個体集団と入れ替えて読み込むことで、個体のサブ集団間の移住を完了させる。ある程度個体が収束するまで上記の操作を繰り返し、サブ集団ごとに出てきた最適解を DDoS のフィルタリングルールベースに書き込み、ルールベースを更新していく。

4. 実験

本稿では GA による DDoS 攻撃フィルタリングの精度を測るための実験を行った。提案手法の検証のためのデータとして DARPA のネットワークログデータセットを使用している。

4.1 実験データ

4.1.1 DARPA データセット

DARPA データセットは侵入検知システムの研究のために MIT の Lincoln Laboratory で作成され、研究者向けに公開されているデータセットであり、大きく 1998 年、1999 年、2000 年の３つのデータセットに分かれている。今回使用するのは 2000 年のデータセット LLS2.0.2(Lincoln Laboratory Scenario (DDoS) 2.0.2) であり、2000 年４月 16 日の 14:45 から 17:28 までおおよそ 2 時間 45 分の通信データが記録されており、この中で 17:06 付近からおよそ５秒間の短い DDoS 攻撃が観測されている。

この DARPA のデータセットのうち、DDoS 攻撃が行われる前の１０分間のデータと DDoS 攻撃開始直後の２秒間のデータを学習データとして扱い、残りの３秒間の DDoS 攻撃データとその後の通常通信のデータをテストデータとしてフィルタリングの精度を測る実験を行う。

4.1.2 WITZ データセット

Waikato University で行われている KAREN の研究で作成されたデータセット。今回、このデータセットの 2006 年 3 月 3 日 13:00 から 13:15 までの１５分間のデータを通常データの学習データとして、その後の１５分間を通常データのテストデータとして使う。この中に、DDoS 攻撃ツールで作成した TCP SYN flood attack、TCP connection flood attack、UDP flood attack、の３つの DDoS 攻撃を同時に行ったデータを混ぜ、検知精度を測る実験を行った。

5. 実験結果

表 2 データセットに対する検知精度

	detection rate	false positive rate	false negative rate
DARPA dataset	0.9842	0.00072	0.00869
WITZ dataset	0.9661	0.0616	0.0200

表 2 が提案手法により作成した DDoS 検知ルールベースの DARPA データセットに対する検知精度を表している。検知精度は 98%と高く、特に誤検知率 (false positive rate) が低く抑えられている。このことから提案手法は DARPA データセットの DDoS 攻撃に対しては有用なものである事がわかる。WITZ データセットに３種類の DDoS パケット

ⓒ 2014 Information Processing Society of Japan

を混ぜたデータに対しては検知精度は 96%と DARPA のときと比べ低くなっている。また、false positive rate も false negative rate も比較的高くなってしまっている。

前述の通りに、DARPA データセットの中では DDoS 攻撃が５秒間しか観測されておらず 、データ内のパケットもあまり多様性のあるものとは言えない。このことから 、今回使用したデータセットはパケットの特徴を捉える事が比較的容易なものであったと考えられる。対して 、３種類の DDoS パケットが混在したデータは 、３つの攻撃の特徴を同時に捉える事が難しく 、検知精度が下がってしまっていると考えられる。

以上の結果から今回の実験では 、提案手法は比較的単純な DDoS 攻撃は高い精度で防ぐ事ができ 、数種類の DDoS パケットが混ざる手の込んだ DDoS 攻撃に対しては検知精度が落ちてしまうことがわかった。今後はさらに手の込んだ DDoS に対しての検知精度の検証ならびに 、複数のパケットの特徴を同時に捉えれるような工夫が必要であると考える。

6. まとめ

本稿では GA によるフィルタリングルール生成によって DDoS 攻撃をフィルタリングする手法を提案した。従来の手法では新しいタイプの攻撃に対応する事が難しかったが 、提案手法では新しく来た攻撃に対し常にフィルタリングルールを更新し続ける事で DDoS 攻撃に対応する事ができると考え実験を行った。

提案手法により単純な DDoS 攻撃に対して DDoS のフィルタリングが可能である事が分かったが 、精度の検証は不十分であり 、さらに巧妙な DDoS 攻撃に対してのシステムの有用性を確認する必要がある。今後は他の様々な種類の DDoS データセットにより再度フィルタリング精度を検討する予定である。さらに 、システムのスケーラビリティを検証するため 、大規模なデータセットに対するシステムのスループットを測る実験も行う予定である。

参考文献

[1] Gong, Ren Hui, Mohammad Zulkernine, and Purang Abolmaesumi. "A software implementation of a genetic algorithm based approach to network intrusion detection." Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005. Sixth International Conference on. IEEE, 2005.

[2] Jeyanthi, N., et al. "An enhanced entropy approach to detect and prevent DDoS in cloud environment." International Journal of Communication Networks and Information Security (IJCNIS) 5.2 (2013).

[3] Kim, Yoohwan, et al. "PacketScore: Statistics-based overload control against distributed denial-of-service attacks." INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies. Vol. 4. IEEE, 2004.

[4] Kim, Yoohwan, et al. ": A Statistics-Based Packet Filtering Scheme against Distributed Denial-of-Service Attacks." IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING 3.2 (2006): 0141-155.

[5] KrishnaKumar, Bharathi, P. Krishna Kumar, and R. Sukanesh. "Hop count based packet processing approach to counter DDoS attacks." Recent Trends in Information, Telecommunication and Computing (ITC), 2010 International Conference on. IEEE, 2010.

[6] Lee, Keunsoo, et al. "DDoS attack detection method using cluster analysis." Expert Systems with Applications 34.3 (2008): 1659-1665.

[7] Lee, Yeonhee, and Youngseok Lee. "Detecting DDoS Attacks with Hadoop." Proceedings of The ACM CoNEXT Student Workshop. ACM, 2011.

[8] Lee, Yeonhee, and Youngseok Lee. "Toward scalable internet traffic measurement and analysis with hadoop." ACM SIGCOMM Computer Communication Review 43.1 (2013): 5-13.

[9] Li, Wei. "Using genetic algorithm for network intrusion detection." Proceedings of the United States Department of Energy Cyber Security Group (2004): 1-8.

[10] Oshima, Shunsuke, Takuo Nakashima, and Toshinori Sueyoshi. "DDoS detection technique using statistical analysis to generate Quick response time." Broadband, Wireless Computing, Communication and Applications (BWCCA), 2010 International Conference on. IEEE, 2010.

[11] Prasad, Reddybathini Durga Siva, and PR Krishna Prasad. "A ROBUST MECHANISM TO MITIGATE DDOS ATTACK USING ENTROPY VARIATION." IJITR 2.1 (2014): 762-766.

[12] Sun, Huizhong, Yan Zhaung, and H. Jonathan Chao. "A principal components analysis-based robust DDoS defense system." Communications, 2008. ICC'08. IEEE International Conference on. IEEE, 2008.

[13] Thapngam, Theerasak, et al. "Discriminating DDoS attack traffic from flash crowd through packet arrival patterns." Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on. IEEE, 2011.

[14] Wagner, Arno, and Bernhard Plattner. "Entropy based worm and anomaly detection in fast IP networks." Enabling Technologies: Infrastructure for Collaborative Enterprise, 2005. 14th IEEE International Workshops on. IEEE, 2005.

[15] Zhong, Rui, and Guangxue Yue. "DDoS detection system based on data mining." Proceedings of the Second International Symposium on Networking and Network Security, Jinggangshan, China. 2010.

[16] 寺田剛陽, 双紙正和, and 宮地充子. "Pushback 機構の一提案とそのモデル化に向けて." (2004).

[17] 門林 雄基 , 櫨山 寛章, 武智洋. "IP トレースバック相互接続 におけるパケットの秘匿性に関する一考察". 電子情報通信 学会 、 Vol. 2006.

[18] Cisco White Paper, " Cisco Visual Networking Index: Forecast and Methodology 2011-2016 , " 入手先 ⟨http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf⟩ , May 2012 , accessed on June 2012.

[19] Apache Spark 入手先 ⟨https://spark.apache.org/⟩

ⓒ 2014 Information Processing Society of Japan