可変パケットドロップ機能を持つ IPS の改良
− DDoS 攻撃の検出と自動フィルタ設定 −

2013SE208 竹田 拓哉 2013SE226 堤 泰一
指導教員 後藤 邦夫

1　はじめに

近年，ネットワークの普及に伴い，さまざまなサイバー攻撃の被害が増加している．その中で特にターゲットとなるコンピュータに複数のマシンから大量の処理負荷を与えサービスを機能停止状態へ追い込む DDoS 攻撃の被害が拡大している．現段階では DoS 攻撃への対策として IPS が提案されているが，DDoS 攻撃への対策は難しいとされている．

そこで，本研究では先行研究 [5] をもとに，DDoS 攻撃に対応できる IPS を試作する．

本研究では，先行研究 [3] を参考にベイズ推定より危険度を測定する．ベイズ推定とはベイズ確率の考え方に基づき，観測事象から推定したい事柄を，確率的な意味で推定する方法である．

事前確率を設定し，IP アドレスの数，平均トラフィック量 (bit/s)，パケットの平均到着間隔 (packets/s) などの統計データを元に危険度を推定する．危険度が高ければパケットロス率を高く，危険度が低ければパケットロス率を低くなるようにパケットロス率を更新する．

パケットロス率を更新することで，DoS/DDoS 攻撃されたときに自動的にパケットロスする IPS を目標とした．

パケットの移動をデータベースに保存し，統計を取る．データベースには SQLite3[1] を用いる．研究にはネットワークエミュレータである，Common Open Research Emulater[2] を使用する．以下，これを CORE と示す．

本研究では，outTable の作成，ポートスキャンを検知する過程は竹田が担当し，ベイズ推定を用いてパケットロスする過程は，堤が担当した．要旨は，2 人で作成した．

2　DoS/DDoS 攻撃

DoS 攻撃は特定のコンピュータから，Victim に向けて大量のパケットや大きなデータ量のパケットを送信されることで，そのコンピュータをサービス停止状態にする攻撃である．DDoS 攻撃とは Distributed Denial of Service Attack の略である．これはターゲットとなるサーバに大量のトラフィックを送る DoS 攻撃を，攻撃者が複数の無関係なコンピュータに侵入し，そこから一斉に行う攻撃である．多数のコンピュータからの大量のアクセスがあるので，攻撃者を特定することが難しい．

一般的に DDoS 攻撃は，攻撃に使用するパケット自身はサイズの小さなものが多いが，その数が多ければ 100〜300[Mbps]，さらに組織的な攻撃となれば 1[Gbps] 程度のトラフィックを発生させることができる．多くの企業の契約回線が 100[Mbps] 程度という現状より，DoS/DDoS 攻撃によって簡単に回線を処理の限界を超えた状態にする．以下に代表的な DDoS 攻撃を示す．

- TCP SYN flood 攻撃

TCP SYN flood 攻撃の説明する．TCP SYN パケットを大量に送信する手法である．TCP 通信を用いる場合，3 ウェイハンドシェイクと呼ばれる，3 つのパケット交換を用いてコネクションを完了させる．

DoS 攻撃と DDoS 攻撃の違いは，DDoS 攻撃の方が Src Addr の数が多いことである．大量のマシンから，Victim に向けて通信を送信するからである．つまり，平均到着間隔 (packets/s) や平均トラフィック量（bit/s）も DDoS 攻撃の方が大きくなる．

本研究では，Src Addr の数に着目し，DoS と DDoS を識別する．

3　WAN から LAN への通信の処理フロー

本節では，本研究で試作する IPS の処理の流れ，検知方法，通信の遮断方法について述べる．WAN → LAN の通信では以下のように処理する．

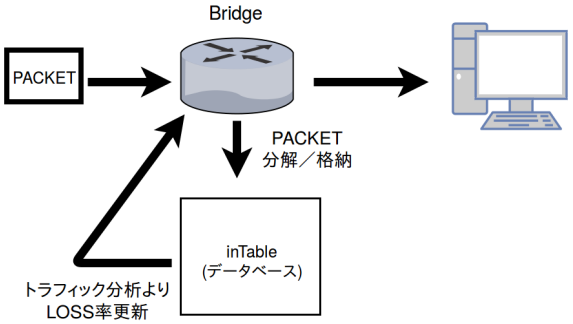

図 1　システムの概要

1. ブリッジに到着したフレームをキャプチャし，ヘッダ分解
2. ヘッダ分解したフレームを分類，情報をデータベースに追加/更新
3. データベース内にあるデータの中から Dst Port 毎に統計
4. 統計されたデータに対しベイズ推定を用いて危険度を計算

5. DoS/DDoS を判定して，loss 率更新

これらの過程により，トラフィックを統計し，不正アクセスを遮断する．

3.1 システムの実現

全体のシステムとして, 先行研究 [5] を参考にした．データベースの table にデータを追加/更新，プログラムで必要なデータを抽出したのち，loss 率を計算するが，その計算過程をベイズ推定に変更した．

3.2 データベースの構成

このデータベースでは Src Addr，Dst Addr，protocol，Src Port, Dst Port を Primary Key として設定する．よって，これらのパラメータがないものは，エラーとして保存させることなく処理される．table に保存したパケットは以下の record のように保存する．

データベース内のパケットの record 例

10.0.0.10 | 10.0.0.11 | 17 | 34671 | 80
srcaddr | dstaddr | protocol | srcPort | dstPort|

183.0 | 1477308568 | 580646 |
count(回) | lastUTCsec(sec)| lastUTCusec(μsec)|

7.967962 | 94213.195287
frequency(packets/s) | traffic(bit/s)

Primary key として保存された Src Addr，Dst Addr，protocol，Src Port，Dst Port の値より，トラフィックを判別する．同じトラフィックと判断した場合，パケットが通過する毎に count の値を 1 ずつ増やす．これにより回数を計測する．パケットを保存する際に，通過した時刻を gettimeofday() 関数によって取得する．取得した時刻によって lastUTCsec，lastUTCusec，移動平均である frequency，1sec のビットの移動量である traffic を計算する．

データベースは，メモリ上に作成する．メモリ上で作成することにより，パケットのデータの読み書きの高速化を図る．tmpfs で mount されたディレクトリにデータベースファイルを置く．tmpfs は，OS のシャットダウン時に自動で削除されるというデメリットがあるので，必要に応じてこのデータベースを別のディレクトリにコピーする．

3.3 DoS/DDoS の識別

本研究では，DoS/DDoS 検知にベイズ推定を用いる．DoS と DDoS の識別は，Src Addr の数で判定する．

- THROUGH（素通し）
  通常の通信処理と同じ役割を果たす．正常なパケットのみをこの状態で通す．
- LOSS（パケット損失）
  任意の確率でパケットを破棄し，パケット損失を起こす．
- DROP（パケット破棄）
  全てのパケットを破棄する．

本研究では loss 率を設定しており，loss 率は 0 以上 1 以下とし，loss=0 のときは THROUGH，loss=1 のときは DROP となる．ブリッジに到着し，データベースに追加/更新されたフレームの分解された情報の到着間隔の平均 (μsec) を着目する．危険度を計算し，危険度に応じて loss 率を更新する．

3.4 フィルタリングルールの設定方法

集計したデータから危険状態を推定する．危険度は時間と共に変化する．そのため本研究の IPS では，集計したデータに対して，ベイズ推定を用いる．危険度を繰り返し更新することにより，危険度の動的な変動にも対応した推定をする．本研究では，次のような確率変数を考える．

- S_c : 攻撃元の IP アドレスの種類の数．S_c の取り得る値は，s_0，s_1，s_2 の 3 種類.
- T_c : 平均トラフィック量．T_c の取り得る値は， t_0，t_1，t_2 の 3 種類.
- F_c : 平均到着間隔．F_c の取り得る値は， f_0，f_1，f_2 の 3 種類.
- E_c : トラフィックの状態．E_c の取り得る値は，正常なトラフィック $Ec = e_0$ と，異常なトラフィック $E_c = e_1$, e_2 の 3 種類.

これらの確率変数は DoS/DDoS 攻撃の特性により採用した．DoS/DDoS 攻撃の規模が大きくなると到着間隔は短くなり，トラフィックの量は大きくなり，IP アドレスの個数は多くなる．これによって危険度を決定する．$P(A)$ を事象 A が発生する確率，$P(B|A)$ を事象 A が発生した下で事象 B が発生する確率とすると，推定する危険度はベイズの定理によって次のように表される．

$$P(E_c|S_c,\ T_c,\ F_c) = \frac{P(S_c,\ T_c,\ F_c|E_c)P(E_c)}{P(S_c,\ T_c,\ F_c)} \qquad (1)$$

この $P(E_c|S_c,\ T_c,\ F_c)$ を e_0，e_1，e_2 それぞれの場合について求め，最大の確率を与える e_n を推定値とする．

3.5 危険度推定の例

事前に $P(S_c, T_c, F_c|E_c)$ および $P(E_c)$ を与える．これは，一般的なトラフィックを参考に決定した．実際には，各ネットワークにおけるトラフィックを計測し決める必要がある．

$P(S_c, T_c, F_c|E_c)$ に設定した値の一部を表 1 に記載する．$P(e_0)$, $P(e_1)$, $P(e_2)$ の値をそれぞれ $\frac{8}{10}$, $\frac{1}{10}$, $\frac{1}{10}$ と与える．

ブリッジで，次のようなトラフィックが観測されたとする．

表 1　$P(T_c,\ S_c|E_c)$ の確率表

$P(s_0,\ t_0, f_0|e_0)$	$\frac{70}{100}$	$P(s_0,\ t_0, f_0|e_1)$	$\frac{30}{100}$
$P(s_0,\ t_2, f_2|e_0)$	$\frac{30}{100}$	$P(s_0,\ t_2, f_2|e_1)$	$\frac{70}{100}$
$P(s_1,\ t_0, f_0|e_0)$	$\frac{80}{100}$	$P(s_1,\ t_0, f_0|e_2)$	$\frac{20}{100}$
$P(s_1,\ t_2, f_2|e_0)$	$\frac{40}{100}$	$P(s_1,\ t_2, f_2|e_2)$	$\frac{60}{100}$
$P(s_2,\ t_0, f_0|e_0)$	$\frac{90}{100}$	$P(s_2,\ t_0, f_0|e_2)$	$\frac{10}{100}$
$P(s_2,\ t_2, f_2|e_0)$	$\frac{50}{100}$	$P(s_2,\ t_2, f_2|e_2)$	$\frac{50}{100}$

観測されたデータ例

TCP|10.0.1.10|80　　S_c:23　　T_c:0.051097[bit/sec]
F_c:0.04212[packet/s]

Dst Port80 へのトラフィックは，Src Addr の種類数が 23，平均トラフィック量が 0.051097μsec，平均到着間隔が 0.04212packets/s と観測された．それぞれの値について，事前に設定したものに当てはめる．ここでは $S_c = s_2$，$T_c = t_2$，$Fc = f_2$ とすると

$$P_0 = P(s_2, t_2,\ f_2|E = e_0)P(E = e_0) = \frac{50}{100} \times \frac{8}{10}$$
$$P_0 = P(s_2, t_2,\ f_2|E = e_1)P(E = e_1) = \frac{0}{100} \times \frac{1}{10}$$
$$P_0 = P(s_2, t_2,\ f_2|E = e_2)P(E = e_2) = \frac{50}{100} \times \frac{1}{10}$$
$$P_1 = 0 < P_2 = \frac{50}{1000} < P_0 = \frac{400}{1000}$$

したがって，Dst Port80 へのトラフィックは，正常なトラフィック $E_c = e_0$ と推定された．システムの実現より，loss 率の設定を e_0 を 0%，e_1 を 50%，e_2 を 100% とし，更に細かく分けていくことを考えている．

また，ここで危険度を更新する．事後確率の合計が 1 となるように計算すると，

$$P_0 = \frac{40}{45} = \frac{8}{9},\ P_1 = 0,\ P_2 = \frac{5}{45} = \frac{1}{9}$$

この事後確率を，新たな事前確率として扱う．

loss 率は，到着平均，通信量，IP アドレスは平均から離れているほど危険であると考えられる．よって，値が離れているほど指数関数的に loss 率を大きい値として設定する．先行研究では，DoS 攻撃のみの対策であったため，DDoS 攻撃にも対応できるよう，改良を加える．パケットのフィルタリングルールをベイズ推定で決定するとき，S_C，F_C を参考にする．現在想定として，DoS は S_c か F_c のどちらかが，2 と判定した場合としこれらに当てはまるときはパケット loss 率は 50%，S_2 と F_2 は DDoS 攻撃と判断しパケット loss 率は 100% とする．

4　LAN から WAN への通信の処理フロー

LAN→WAN の通信は outTable というデータベースの table を作成した．outTable は，ポートスキャンの検知に用いた．本節では，ポートスキャン検知について説明する．

4.1　システムの概要

LAN→WAN の通信は outTable というデータベースの table を作成し，パケットのデータを追加/更新する．この処理を加えるために，inTable 同様に table を作成しようとしたが，1 つのデータベースを扱うに関して，細かい処理が増えたため，”database is locked”のエラーが返されるようになった．そのため，システムを再構築しスレッド処理を増やし，outTable へ書き込みに成功した．

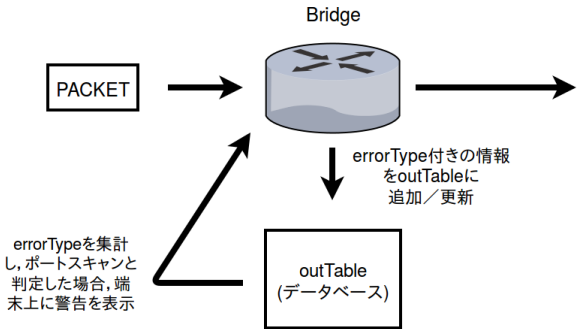

図 2　outTable の処理フロー

4.2　outTable の構成

outTable に関しては，ベイズ推定ではなく，エラーの数により攻撃判定する．そのため，inTable の column に errorType を追加した．これは，ポートスキャンは通信要求に対するエラーが多く返されるという特徴を記録するために作成した．表 2 のように TCP に関しては TCP RST，UDP に関しては，ICMP port unreachable が返される．TCP と UPD のエラーを区別し，table に追加/更新する．errorType と frequency の値を集計し，しきい値を超えたとき，端末上に警告文を表示する．

outTable の record 例

srcaddr | dstaddr | protocol | errorType |

srcPort | dstPort|

count(回) | lastUTCsec(sec)| lastUTCusec(μsec)|

frequency(packets/s) | traffic(bit/s)

4.3　ポートスキャン検知

ポートスキャンとは，DDoS 攻撃などの前段階として用いられ，Port の脆弱性を探る方法である．outTable に関しては，ポートスキャンの検知に使う．Victim から Attacker に対し多くのエラーを返すことになる．この特性を活かし，outTable のパケットも集計し攻撃を未然に防ぐ方法を提案する．

5　実験

ここで，本研究の実験について説明する．

表 2　ポートスキャンにおけるエラー応答

接続の種類	エラー応答
TCP	RST + ACK
UDP	ICMP port unreachable

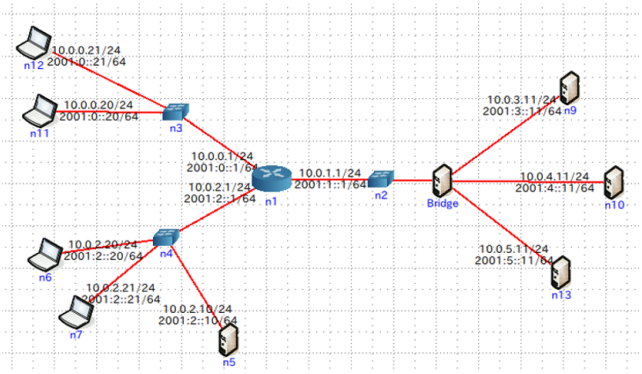

図 3　DDoS 実験用ネットワークの構成図

5.1　DDoS 攻撃の段階的遮断

図 5 のネットワーク構成で実験を行う．n12 を Attacker，n9 を Victim とし，n12 から先行研究 [4] の DDoS 攻撃模擬ツールでトラフィックを送信し，n9 にて tcpdump でトラフィックを確認する．Bridge で設定したロス率と，実際に得られた結果が合うかを確認することで，Bridge の性能を評価する．

まず，DDoS 攻撃模擬ツールでトラフィックを送信し，ベイズ推定により危険度を推定し，正しくパケットロスできるかの実験をした．

S_c の取り得る値は，0 種類以上 33 種類未満 s_0，34 種類以上 66 種類未満 s_1，67 種類以上 s_2，T_c の取り得る値は，100bit/sec 以上 t_0，1bit/sec 以上 100bit/sec 未満 t_1，1bit/sec 未満 t_2，F_c の取り得る値は，1μsec 以上 f_0，100μsec 以上 1μsec 未満 f_1，100μsec 未満 f_2，ロス率は，e_0 で 0%，e_1 で 50%，e_2 で 100%，と設定した．

DDoS 攻撃模擬ツールで，0.2sec 間隔，ランダムな SrcAddr から 200 個のトラフィックを送信した．setRule が実行された時刻と，その間に送信したトラフィックと届いたトラフィックからパケットロス率を求めた結果を次の表 3 に示す．

表 3　パケットロスの評価

経過時間 [sec]	到着/送信（ロス率）	E_c
00.0-09.0	45/45(0%)	e_0
09.2-19.2	28/51(46%)	e_1
19.4-40.0	0/104(100%)	e_2

表 3 より結果として，概ね正確なパケットロスを起こすことができた．

5.2　メモリ上のデータベースの読み書きの評価

次にランダムな 7 文字の配列を 100 個，メモリ上のデータベースに書き込む実験をした．その結果を表 4 に示す．

表 4　10 回の書き込みの平均の速さ

ホームディレクトリ [sec]	メモリ上 [sec]
11.0480	00.0046

表 4 より結果として，コンピュータによって変わるがホームディレクトリでの実験は約 10sec，メモリ上では約 0.05sec で書き込めるという違いが出ることがわかった．

5.3　ポートスキャンの検知

outTable は，ポートスキャンツールを使い，エラー応答を確認した．プロトコルの違いより，エラーが異なるので，エラー毎に outTable に追加更新し，ポートスキャンを検知することができた．

6　おわりに

ベイズ推定を用いて実験することで，危険度を測定し，DDoS 攻撃を検知し，自動的に遮断することができた．また，メモリ上にデータベースを作成し，そこで操作をすることで，データベースの読み書きの高速化に成功した．LAN → WAN の通信については，プロトコルのエラー応答という特徴より検知することができた．今後の課題は，IPv6 での実験と LAN → WAN よりポートスキャン検知後，IP アドレスの範囲指定し，攻撃を遮断できるようにすることである．

参考文献

[1] Hipp, D. R. et al.: SQLite3 (accessed Dec. 2016). https://www.sqlite.org.

[2] U.S. Naval Research Laboratory Networks and Communication Systems Branch: Common Open Research Emulator (accessed Dec. 2016). http://www.nrl.navy.mil/ itd/ncs/products/core.

[3] 安藤尚紀：高速パケットキャプチャによるトラフィックの統計分析，南山大学情報理工学部システム創成工学科 2013 年度卒業論文 (2014).

[4] 川上健人，黒田涼介：DDoS 攻撃模擬ツールの試作，南山大学情報理工学部システム創成工学科 2015 年度卒業論文 (2016).

[5] 嶋田憲人：DoS 攻撃に対する可変パケットドロップ機能を持つ IPS の試作，南山大学情報理工学部システム創成工学科 2015 年度卒業論文 (2016).