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

2011SE236 嶋田憲人
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, 不正アクセスの被害がよく取りざたされている. その中でも特に, サーバやネットワーク帯域に対して過剰な負荷をかける DoS 攻撃の被害が一段と深刻となっている [1].

対策として, 侵入防止システム (Intrusion Prevention System, 以下 IPS) を用いてアクセスの制限等がされているが [2], 誤検知が起きた場合様々な弊害が発生する [3].

そこで本研究では, 検知した攻撃の影響度に応じて適切にパケットの損失率を変化させることができる IPS を提案する. トラフィックを監視し, 単位時間あたりのパケットの移動平均がしきい値を越えた場合 DoS と判定し, その値により自動でパケットロス率を決定する. 本研究のシステムにより, IPS でパケット損失率を攻撃元に応じて変化させることができるようになる.

2 システムの概要

IPS システムは図 1 のように LAN と外部ネットワークとの間で IP アドレスなしで Bridge として動作する. IP アドレスを設定しないことによりこのシステム自体は攻撃対象にならないという利点がある.

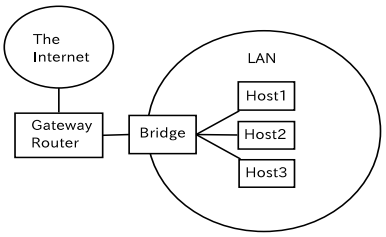

図 1 ネットワーク構成

システムの処理はフレームキャプチャごとに RecvAndSend クラスの rum() ループ内で実行される. その処理フローを以下で説明する.

1. パケットキャプチャ
   ブリッジに到着したフレームをキャプチャし, Frame オブジェクトを作りヘッダ分解をする.
2. 統計情報のカウント
   ヘッダ分解した Frame を用いて, 1 フレームごとに分類計数, 統計情報をデータベースに追加/更新する. 決めた回数ごとに, DoS であるかの判定と, 使わなくなった統計情報の削除をする. DoS であればロス率を生成し, フィルタリングルールに統計情報とロス率を追加/更新する.
3. フレーム送信/破棄
   フレームのロス率をフィルタリングルールから参照し, 乱数を生成して比較することによりフレームを送信, または破棄する.

統計情報は sqlite3 を用いてデータベースで管理する.PRIMARY KEY を srcAddr, dstAddr, protocol, srcPort_Type, dstPort_Code として, フレームの統計情報が PRIMARY KEY と一致すれば分類計数を更新, 一致しなければ新しくデータを追加する.

DoS の検知は, 単位時間毎にパケットの移動平均 frequency[packets/s] を求め, 次の単位時間の移動平均を更新した値がしきい値を越えた場合に判定する. 時刻 [s] を取得し, 現在の移動平均の観測値 [packets/s] は 1/(現在時刻 - 前回到着時刻) により求められる. 時刻 n[s] での移動平均を y_n, 過去の移動平均を y_{n-1}, 現在の観測値を x_n, r を現在の観測値が与える重みとすると, 移動平均の更新値は以下の指数平滑移動平均の式で与えられる.

$$y_n = (1-r)y_{n-1} + rx_n (n = 1, 2, 3, ...) \quad (1)$$
$$y_1 = x_1 \quad (2)$$

また, 重み r は過去の影響が大きくなるよう小さな値が望ましい. そこで, t[s] を前回到着からの経過時間として r を以下の式により与えるようにする.

$$r = 1 - (1-p)^t \quad (3)$$

この式では $0 < p < 1$ として影響度 p の値を変えることにより, 重みを変化させることができる. しきい値は移動平均が異常であると判定できる最小値を設定すればよい.

フィルタリングルールは後藤研究室の先行研究で開発した GateKeeper プログラムの運用例 [4] を参考に設計する. DoS と検知されたフレームは, ロス率を計算してフィルタリングルールに追加/更新する. ルールの管理は list を使用する. ロス率の生成モデルは, 時間あたりの移動平均値が 0 のときロス率 0 で単調増加、移動平均値が無限大に大きくなると 1 に漸近する関数が適切である。本研究ではその例として指数分布関数を用いる。$F(x)$ を生成するロス率, x を移動平均値, λ をパラメータとして以下の式により定義する.

$$F(x) = 1 - e^{-\lambda x} \quad (4)$$

3 実装

DoS は Counter クラスのメンバ関数 checkDoS() で検知する. しきい値を THRESH で定義し, 移動平均 frequency

がしきい値より大きいルールを SELECT する.

DoS 検知

```
double thresh = THRESH; //しきい値
char sqlcommand[1024];
sprintf(sqlcommand,
   "SELECT * FROM PACKETS WHERE frequency > %f"
   " ORDER BY frequency DESC;", thresh);
```

SELECT された統計情報の frequency を x として, 2 節で述べた指数分布関数 (4) 式に代入してロス率 lossProb を生成する. さらに PRIMARY KEY と生成された lossProb をクラス Filter のメンバ関数 setRule() に渡し, フィルタリングルールとして追加/更新する.

Filter の getlossProb() でルールの lossProb を参照できるようにして, RecvAndSend クラスでその値を利用できるようにする. RecvAndSend クラスでは生成した lossProb と drand48() で生成する乱数を比較し, 乱数の方が大きければパケットを送信する.

フィルタリング

```
if (filter == 0 ||
  filter->getLossProb(frame) < drand48()){
   if (outsock > 0){
     int sent = send(outsock,buffer,recvd,0);
     if (sent != recvd)
       throw std::runtime_error("send error\n");
   }
}
```

4 実験

本節では実験環境と試作した IPS システムの実験手順, 結果を説明し, 考察をする.

4.1 実験環境

Ubuntu10.04LTS(32bit OS) をインストールした PC を用意し, ネットワークをエミュレートすることができる CORE にネットワークを構築し実験する. IP アドレスは HostA:10.0.0.21, HostB:10.0.1.10 として HostA から HostB へトラフィックを送る.

4.2 実験手順

1. しきい値 TRESH と $F(x)$ のパラメータ λ を定義する.
2. Bridge で IPS プログラムを実行する.
3. ping, もしくは iperf コマンドで攻撃元からターゲットへトラフィックを送り, パケットロスを観測する.
4. 計算上のロス率と実験結果を比較し, システムの有効性を考察する.

4.3 実験結果

スペースの都合上, ping で IPv4 の結果のみを示す.ping は frequency = 50, 60, ⋯, 100 となるようにオプション-i で送信間隔を 0.02s, 0.017s, ⋯, 0.01s と変更して実験した. frequency = 50 からロスが起こるように THRESH = 50, frequency = 100 のときにおよそ 90% ロスとなるように rate = 0.02 と定義する. ping の送信回数は各 3000 回として, 計算上のロス率と実験の観測値を以下に示す.

表 1 実験結果

frequency[packets/s]	50	60	70
送信間隔 [s]	0.02	0.017	0.014
計算値 [%]	63	70	75
観測値 [%]	64	65	70

frequency[packets/s]	80	90	100
送信間隔 [s]	0.013	0.011	0.01
計算値 [%]	80	83	86
観測値 [%]	73	77	86

frequency = 60 から 90 では誤差が生じているが, これは ping の送信間隔が 1/1000s までしか設定できないことによる, コマンドの仕様上起きた誤差であると考えられる. コマンド上の誤差が発生しない frequency = 50, 100 ではほぼ誤差が起きていないので, この可変パケットロス機能は有効であるといえる.

5 おわりに

本研究では攻撃元のトラフィックの移動平均により自動でパケットロス率を定め, フィルタリングをする IPS システムを試作できた. この機能により, 単純 DoS に対して検知, アクセス制限をすることができるようになる.

今後の課題として, DDoS 攻撃へ対応できる機能の追加が求められる. そのためには, checkDoS() で SELECT した情報から合計をとり, 同 IP からのリクエスト回数を制限する等の操作が必要である.

参考文献

[1] Kenig, R.: How much can a DDoS attack cost your business? (2013). https://blog.radware.com/security/2013/05/how-much-can-a-ddos-attack-cost-your-business/.

[2] McAfee Network Security Platform(旧 IntruShield) : サービス拒否 (DoS) 攻撃防止技術 バージョン 0.1 (2010). https://kc.mcafee.com/resources/sites/MCAFEE/content/live/PRODUCT_DOCUMENTATION/22000/PD22959/ja_JP/NSP_DoS_Prevention_Techniques_0.1_JA.pdf.

[3] 左門至峰 : 情報セキュリティスペシャリスト - SE 娘の剣 - (accessed Jan. 2016). http://sc.seeeko.com/archives/cat_125428.html.

[4] 伊藤遼平, 嶋田伊吹 : IPS の実現とネットワークエミュレータ上での評価, 南山大学数理情報学部 情報通信学科 2009 年度 卒業論文 (2010).