ゲートウェイ絞込み制御による不正フロー対策システム

片山 晃一　尾高 由和　中村 浩　戸辺 義人*　佐々木 良一*
東京電機大学工学部情報通信工学科、*情報メディア学科

１．はじめに

現在、DoS(Denial of Service)攻撃[1]への一般的な対策としては、IDS(侵入検知システム:Intrusion Detection System)と既存のファイアウォール等を併用して、不正アクセスに対処することが考えられる。しかしこうした方法では、未知の種類のフローについて遮断すべきか否か判断がつかないという問題がある。

そこで本研究では、従来の対策にかけている未知の攻撃に対するシステムとして iSHAPER (intelligent SHAPER)を提案し、本稿ではその設計と実装について述べる。

２．研究課題

サービスを提供しているサーバを利用不可にさせることを目的とした DoS 攻撃は、企業活動などへの多大な損害をもたらす。その解決手段して、現在、シグネチャマッチングによるフローの遮断などがあるが、正しいフローまで遮断されてしまうという問題がある。そこで、本研究では、その解決策として、フロー開始直後に観測期間を設け、非適正と判断したフローのみを遮断し、適正と判断したフローは遮断しないゲートウェイを設けることにより、DoS 攻撃を受けてもサービスの提供を続けられる仕組みの実現を考えた。このゲートウェイに付加する機能を iSHAPER と名付けた。

３．iSHAPER

3.1 設計

システムのアーキテクチャ、フロー管理、および iSAHPER の制御機能を含めたシステムの設計について述べる。

図１は iSHAPER の機能を有する iSHAPER ゲートウェイ(iSGW)のシステムの全体を示したものであり、iSHAPER と管理ツールから構成されている。
iSHAPER は

- フロー管理モジュール：FMM
- トラフィック制限モジュール：TLM

の２つのモジュールからなり、その中には次の３つのテーブルが含まれる

- フロー登録テーブル：FET
- 不正フロー登録テーブル：IET
- 許可フロー登録テーブル：PET

FMM がパケットを受け取り、そのパケットが不正フロー(Fi)となることを認識した場合に、TLM がレートを制限することにより未知の攻撃に対処する。

FET は、ネットワークの外側から来たパケットを、フロー（送信元 IP アドレス、送信元ポート番号、受信先 IP アドレス、受信先ポート番号）毎に分類し、登録するテーブルである。PET は、予め通信を許可するフローを登録しておくテーブルである。IET とは、不正であると判断されたフローを登録するテーブルである。

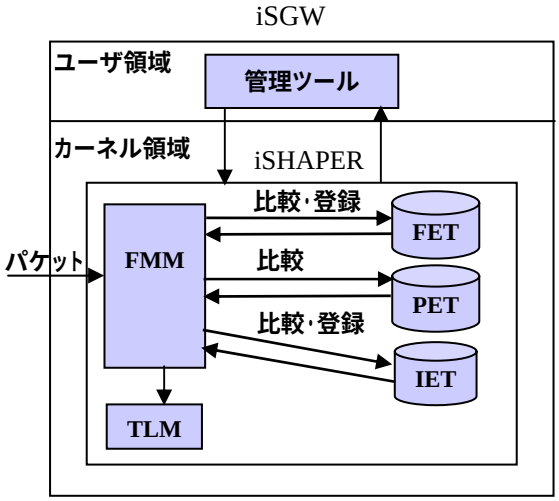

図１ 提案システムの全体図

各々の機能についての詳細は以下のようになる。

(ｉ) FMM

- 新しいパケットの情報を FET に登録する。
- T[s]：観測期間
  P:T 時間内に来るパケット数
  b[bit]：１パケットあたりのビット数
  N[bps]：帯域の閾値
  としたとき、外側からのフローを監視し
  　N　＜＝　bP/T
  となるフローを危険領域フロー(Fh)と認識する。
- Fh と PET の情報を比較して Fi かどうかを判別する。
- フローが Fi と認識された場合、IET に登録し、その情報を TLM に伝える。
- フローが Fi でない場合は、そのままフォワードさせる。

(ⅱ) TLM

- FMM で不正と認識されたフローに、適正テストを行う。

適正テストとは、一時的に許容通過レートを絞る期間を設け、適正フローと非適正フローを判別することである。

ここで、適正フローを、送信レート制御がされているもので、パケット喪失率が高くなれば自動的に送信レートを下げるようなフローと定義する。

また非適正フローは、パケット喪失率が高くなっても、送信レートが下がらないような、送信レ

ート制御がされていないフローと定義する。
・ 適正フローについてはレート制限を解除し、非適正フローは完全に遮断する。

(iii) 管理ツール
・ iSHAPER の ON,OFF、各種フロー登録テーブルのリセットや、情報の表示を行う。

3.2 実装

iSGW の実装として Linux kernel2.4.18 をベースにした。カーネルにおける実装として、/usr/src/linux/net/ipv4 に FMM, TLM の具体的なコードを実装した。基本的に図２のように、ip_rcv()の中で分岐させて、トラフィックを計測して絞込みの判断をさせる。現状では FMM と TLM のレート制限機能が完了しており、適正テスト機能は今後実装していく予定である。また FET は図３のような構造体になっている。

ユーザコマンドは ioctl システムコールを用いて、iSHAPER の ON/OFF、iSHAPER の初期化、FET、IET、PET の表示を行えるように作成した。

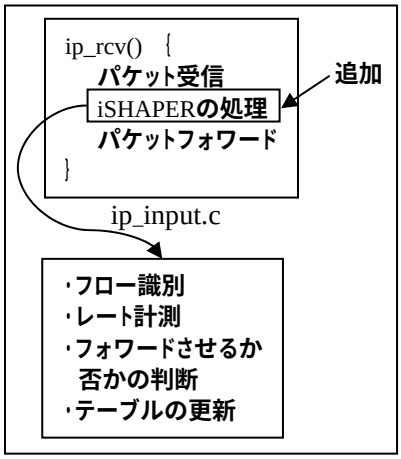

図２　iSHAPER の実装

```
struct flow {
struct flow *next;      /*次のテーブルの位置*/
int         index;      /*テーブル番号*/
__u32       d_IP_addr;  /*受信先IPアドレス*/
__u16       d_port;     /*受信先ポート番号*/
__u32       s_IP_addr;  /*送信元IPアドレス*/
__u16       s_port;     /*送信元ポート番号*/
int         count;      /*パケットのカウント*/
int         first_time; /*危険領域フローの
int         first_count;   判定*/
};
```

図３　登録テーブルの構造体

４．実験

本章では仮想的に DoS 攻撃を生じさせた時の、提案システムの効果について評価する。図４に実験環境を示す。図に示されるように、ホスト H1、H2、GW から構成されている。

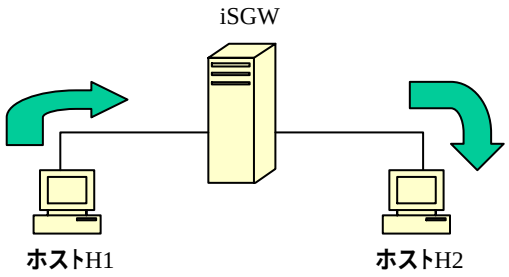

図４ 評価実験の環境

まず、iSGW の iSHAPER を OFF にした状態で、ホスト H1 から H2 に向けてパケットを送信し、そのスループットを測定した。次に、iSHAPER を ON にした状態で、同様にスループットを測定した。実験に利用したフローは以下のものである。
・ プロトコル：ＵＤＰ
・ パケットサイズ：1400 [Byte]
・ 送信パターン：連続送信
・ 送信間隔：100 [ms]
また、２つの状態におけるスループットは表１に示す。

表２ 測定スループット

iSHAPER	受信側スループット
OFF のとき	11.12Mbit/s
ONのとき	4.21Mbit/s

５．考察

表１から解かるように、iSHAPER を ON にした状態では正しくレート制御が実行されており、OFF の状態に比べてスループットが下がっている。
このことから、過剰なフローに対する GW でのレート制御は機能しているといえる。

ゲートウェイにおいてフロー別にレート制御をする技術として CBQ (Class-Based Queueing)[2] が知られている。CBQ では、ネットワーク管理者が予めフロークラス別に許容最大帯域を決定しておく。しかし、この方法においては未知のフロー（ネットワーク内部の者が新たに使用開始したフロー等）に対処することができない。

６．まとめ

本稿では不正フローをＧＷでレート制御するシステムを考案し、有効性を確認した。
今後はさらに研究を進め、本稿では設計だけで終わった、適性テストを行う機能を実装していく予定である。

７．参考文献

[1]Scambray,J., McClure,S., and Krutz,G.：クラッキング防衛大全第二版, pp. 435-458, (2001.5).
[2]Floyd,S. and Jacobson,V.：Link-sharing and resource IEEE/ACM Trans. Networking vol.3,No.4,pp. 365-386 (1995.8).