IPSの実現とネットワークエミュレータ上での評価

2006MI046 伊藤 遼平　　2006MI157 嶋田 伊吹
指導教員　後藤 邦夫

1　はじめに

近年, インターネットの急速な普及の結果, ネットワークは不正アクセスやウィルスの脅威にさらされ, その被害についての報道が多い [3]. これらの攻撃に対応するために侵入検知システム (以下, IDS; Intrusion Detection System とする) の研究が活発化し, 商用化も進んでいる. しかし, IDS は侵入を検知することを目的としており, 防御という点から考えると完全ではないのが現状である [4]. また, IDS に防御システムを搭載した侵入防御システム (以下, IPS; Intrusion Prevetion System とする) 等も開発されているが, 既知の攻撃にしか対応できないことなどの問題点も多く, 完全な防御を目的としたシステムは未だ開発されていない.

そこで本研究では, パーソナルコンピュータ (以下, PC とする) を利用して, 未知の攻撃を検出する可能性のある IDS を用いた IPS を提案する. 本研究の IDS では, 通常時には起こり得ないパケット量の急激な変化に着目して異常を検出する. 攻撃情報に関わらずパケットの量を監視するので, 未知の攻撃を検出できる可能性がある.

また, そのアラートの情報を元に後藤研究室で開発中の通信制限システム (以下, GK とする) [1] にフィルタリングルールを送信し通信を制限することで IPS を実現する. なお, 本研究では既存の GK をそのまま利用する. 伊藤は主に実験を担当し, 嶋田は主に IDS の実装を担当した.

2　システム概要

本節では, 本研究で提案する IPS の配置と処理の流れ, IDS と GK の連携の手法について述べる.

2.1　全体の流れ

GK を外部ネットワークと内部ネットワーク間で IP アドレスをつけずにブリッジとして動作させ, フレーム通過時に通信を制限する. IP アドレスをつけないので攻撃対象にならないという利点を持つ (図 1 参照). また故障時には GK に接続した LAN ケーブルを直結してバイパス出来る. IDS はネットワーク上を流れるパケットを監視し, 検知した攻撃の種類, 攻撃下の IP アドレスなどの情報を直結した GK に渡す. GK はこの情報に基づいてリアルタイムに通信を制限する. 外部ネットワークから内部ネットワークに対しての攻撃を制限することを目的としているが, その逆である内部ネットワークから外部ネットワークへの攻撃も同時に制限することができる.

2.2　IDS の処理の流れ

IPS を実現するために, 侵入を検知する IDS を実現しなければならない. そこで, 本研究では未知の攻撃を検出できる可能性があるしきい値モデルを用いた IDS を提案する. パケットをキャプチャし, 単位時間毎にパケットの平均と分散を保持する. この時保持した値から正常範囲を計算し次の単位時間のパケット数と比較する. 正常範囲から外れていなかった場合は, 平均, 分散の値を更新する. また正常範囲から外れていた場合に異常と認識し, アラートを生成する.

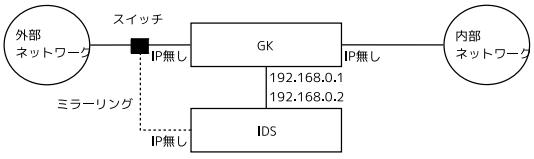

図 1　ネットワーク構成

2.3　異常トラフィック検知方法

システムやユーザの振る舞いに関して数値で表されるデータに正常範囲を設定し, その範囲を越えた場合に異常として検出するしきい値モデルを利用して異常を検出する. このモデルで最も多く用いられている平均と標準偏差を用いて自動的にしきい値を設定する方法を利用する. このモデルは過去に取得したデータの平均値と現在の観測値の差を比較して異常を検出するが, 検出時に標準偏差を用いることにより過去のデータの散らばり具合を考慮した評価をすることが出来る.

単位時間当たりに流れるパケットを, IPv4/IPv6, プロトコル, 発信 IP アドレス (srcIP) , 宛先 IP アドレス (dstIP) , 発信ポート (srcPort) , 宛先ポート (dstPort) 別にカウントし, プロトコルの場合は TCP, UDP, ICMP 等で分類する. ICMP の場合はタイプ別に扱う.

観測値は互いに独立で, 観測値は一般的に正規分布として扱われる. 実際, 観測値の分布が正規分布に当てはまることは少ないが正規分布に近似すると考える. 平均値 $\overline{x}$ と標準偏差 S が明らかになっている場合, 次の観測値 x_{n+1} のしきい値は, 式 (1) のようになる.

$$\overline{x} - (d \times S) < x_{n+1} < \overline{x} + (d \times S) \qquad (1)$$

ただし, 下限は 0 とする. 観測値がしきい値から外れたとき異常とする. この時, 下限は 0 とする. d は観測値がしきい値に収まる相対頻度を定めるための値である. d の値を変えることで任意の信頼区間を与えることができる. d の値と相対頻度の関係を表 1 に示す.

表 1　d の値と相対頻度の関係

相対頻度	90%	95%	96.44%	99.74%
d の値	1.645	1.9645	2	3

また, IDS の平均, 分散の更新には, 現在の平均, 分散と観測値に重みをつけて合わせた値を用いる. 更新後の平均を $\overline{x}$, 更新前の平均を $\overline{x_1}$, 観測値を $\overline{x_2}$, α を重みとすると, 式 (2) のようになる.

$$\overline{x} \Leftarrow (1-\alpha)\overline{x_1} + \alpha\overline{x_2} \tag{2}$$

なお, $0 < \alpha < 1$ とする.

また更新後の分散を V, 現在の分散を V_1 とすると, 分散の更新式は式 (3) のようになる.

$$V \Leftarrow (1-\alpha)V_1 + \alpha(\overline{x_1} - \overline{x_2})^2 \tag{3}$$

$\overline{x_1}$ の方が重みが大きくなるように, α には 0.1 などの小さい値を設定する.

なお, 本研究で用いる d, α の適切な値については 4.1.2 の実験で求める.

2.4 GK のフィルタリングルールの設定方法

IDS において異常を検知すると GK にフィルタリングルールを追加しリアルタイムに通信を制限する. GK の通信制限として以下の 5 種類がある.

- THROUGH (素通し) : 通信制限無し
- DELAY (遅延) : 任意の秒数の遅延を起こす
- THROTTLE (スループット制限) : 帯域幅を絞り, 通信速度を制限する
- LOSS (フレーム損失) : 任意の確率でフレームを破棄し, フレーム損失を起こす
- DROP (フレーム破棄) : 全てのフレームを破棄する

フィルタリングルールの設定は GK のフィルタリングルール設定方法に従い, 通信制限の流れは下のようにする.

THROUGH $\longleftrightarrow$ DELAY $\longleftrightarrow$ THROTTLE $\longleftrightarrow$ LOSS $\longleftrightarrow$ DROP

通信制限の遷移のタイミングはトラフィックが上限を越した時である. IDS において現在施している通信制限の状態を保持しており, トラフィックが上限を越す度に右の段階の通信制限へ遷移する. また連続で上限を越さなかったトラフィックについては単位時間毎に一段階通信制限の状態が左へ遷移する.

3 システムの実現

本節では, システムの実現方法について述べる.

3.1 IDS の構成と処理の流れ

本研究で用いる IDS を実現するため C++ でプログラムを書く. またマルチスレッドのために GNU common C++ クラスライブラリを利用する. 作成した各クラスの概要を表 2 に示す.

Pktcap クラスでキャプチャしたパケットを CountTable クラスを継承した DB クラスと PatternTable クラスに渡し管理する. これらのクラスで管理しているトラフィックデータから MSD クラスと PAlert クラスが異常検知をする. 異常検知の結果アラートが生成されると通信制限をするために AddRule クラスから GK へフィルタリングルールが送信される. PortScan クラスはポートスキャンを受けていないか監視する. List クラスは IDS をリアルタイムで動かしている時に各種情報を閲覧するための CUI である.

表 2 クラスの説明

クラス名	説明
AddRule	GK へフィルタルールを送信する
CountTable	ハッシュテーブルの操作をする
DB	全ての到着パケットを管理する
List	DB クラスで管理しているパケットデータを参照する
MSD	全体のトラフィックから異常を検知する
PAlert	分類別のトラフィックから異常を検知する
PatternTable	分類別のパケット数を計数する
Pktcap	パケットをキャプチャする
PortScan	ポートスキャンを検知する

4 実験

本節では, 本研究で実装した IDS の評価, そして IDS を GK と連携させ, IPS として動作しているか確認する. まず適切な検出をするために 2.3 節で述べた d と α に適切な値を決める実験をした. しきい値を少しでも外れた場合にアラートが出るので, 重要度という値を定義しフィルタリングをして, 適切な値を $d = 3$, $\alpha = 0.1$, 重要度は 1.5 とした. 適切な設定値を決める実験については 4.2 節で詳しく述べる. また, この設定ををした IDS に著者の自宅で収集したパケットログを読み込ませ性能評価をした. 性能評価の結果, 攻撃の可能性があるパケットを検知する事ができた. IDS の性能評価の結果は 4.3 節で詳しく述べる. また, IDS から GK にフィルタリングルールを送信し通信制限が正しく施されたことで, IPS として動作することを確認できた. GK との連携の実験結果は 4.4 節で詳しく述べる. これにより, 未知の攻撃にも対応できる IPS ができたのではないかと考えられる.

4.1 IDS の評価

実験に試作した IDS を実験して評価する. 評価項目は以下の通りである.

- IDS が正常に異常を検知するか.
- GK と連携させ, IPS として動いているか.

上記の点に注目し, 実験を行った. まず, 本研究で実装した IDS に収集したログファイルを読み込ませ, 読み込んだログファイルに異常があった場合正確にアラートを生成するかを確認する. また, 同じログファイルを読み込

んだ Snort のアラートと比較をする．最後にネットワークエミュレータ上で GK と連携しリアルタイムでシステムを動作させ, 正しくフィルタが操作され IPS として正しい動作をしているかを確認する．この実験では IDS でネットワーク上を流れているパケットを監視し, 評価をする．評価用のデータには著者の自宅のそれぞれのパケットログを利用する．著者の自宅のパケットログは, 伊藤は 2009 年 9 月 11 日から 2009 年 12 月 30 日, 嶋田は 2009 年 9 月 7 日から 2010 年 1 月 4 日まで収集した．また, 指導教員の後藤邦夫教授の自宅で収集した 2009 年 10 月 8 日から 2009 年 10 月 24 日までと 2009 年 12 月 11 日から 2010 年 1 月 4 日までのパケットログを提供していただき, このパケットログも利用した．パケット数は伊藤宅は約 1 億 8000 万パケット, 嶋田宅は約 7200 万パケット, 後藤教授は約 3700 万パケットである．実装した IDS に収集したパケットログを読み込ませ, 正確にアラートが生成できるどうかの実験をした．観測地がしきい値に収まる相対頻度を定める値である d は 2 と 3 を代入する 2 パターンを実験する．このときの相対頻度は $d = 2$ の時は 96.44%, $d = 3$ の時は 99.74% である．また, 重みであるの値 α には 0.1, 0.3 の時の 2 パターンの計 4 パターンでアラート数の違いを比較する．

4.2 IDS 適切設定実験結果

まず実験結果の表のそれぞれの値を説明する．ALL はキャプチャした全てのパケットの値で, IPv4 はキャプチャした値の内の IPv4 アドレスを用いた通信の数である．本研究で実装した IDS は IPv4 の通信しか対応できず, この値が実際に IDS で検出するパケット数である．MSD は 5 分単位にパケット到着数を管理し, 定めた上限の値を越えてアラートが発生した回数である．P1, P2, P3 はそれぞれの管理している値の上限を越えた時にアラートが発生した数で, P1 は Protocol, SrcIP, DstIP, SrcPort, DstPort の組み合わせを監視対象としている．P2 は P1 を集約し, Protocol, SrcIP, DstIP, DstPort の組み合わせを監視対象としている．P3 は P2 をさらに集約し, Protocol, DstIP, DstPort の組み合わせを監視している．P1 と P2 は TCP と UDP のみの対応で, P3 は TCP, UDP に加えて ICMP などその他のプロトコルも対応している．SrcIPDstIP は SrcIP と DstIP の組み合わせを監視しているが, 往復の通信を監視している．

表 3　d=2, α=0.1 の実験結果

Name/date	後藤教授宅	伊藤宅	嶋田宅
MSD	453	1,592	1,717
P1	1,246	26,421	6,024
P2	1,962	50,664	13,020
P3	2,713	71,541	21,468
SrcIPDstIP	968	28,456	8,452

PortScan で検知したアラート数は, 後藤教授宅が 25,884, 嶋田宅が 316,328 であった．なお, 伊藤宅は固定 IP アドレスでないため PortScan クラスが期待通りに動作しなかったため 0 とする．

表 4　d=3, α=0.1 の実験結果

Name/date	後藤教授宅	伊藤宅	嶋田宅
MSD	312	956	2,118
P1	521	13,415	4,112
P2	909	25,515	8,059
P3	1,387	41,380	12,015
SrcIPDstIP	583	14,594	4,955

表 5　d=3, α=0.3 の実験結果

Name/date	後藤教授宅	伊藤宅	嶋田宅
MSD	547	1,473	1,579
P1	495	21,468	4,130
P2	1,111	39,155	9,757
P3	1,816	55,053	15,394
SrcIPDstIP	802	21,867	6,524

精度を求めるならば d の値を少なく, α の値を大きくすることが必要であることがわかった．しかし, 実験の結果ではアラート数がもっとも多く, 正しいパケットもアラートとして検出している可能性がある．実験より, IDS の設定は d=3, α=0.1 がよいのではないかと考えた．

また, 上記の実験は上限を少しでも越えてしまった場合にアラートを出力する設定だったためアラートの数が多い．そこで,

$$\text{重要度} = \left| \frac{(\text{今回の観測値}) - (\text{これまでの平均})}{d \times S} \right| \quad (4)$$

の計算式でフィルタリングをした．式 4 の値の絶対値が大きいほど異常として重要と考えられる．この絶対値を以下重要度と述べる．今回の実験では重要度が 3 以上, 2 以上, 1.5 以上の時に異常として検出させ, 再度同様に実験を行う．この手法で影響が出るのは P1, P2, P3, SrcIPDstIP の 4 つで MSD には影響が少ないため, 以下の実験結果は影響が大きい 4 つのみを結果として示す．次の表は $d = 3$, $\alpha = 0.1$, 重要度が 1.5 の時の結果である．フィルタリングをした結果, アラートの発生数が大きく減った．重要度を大きく設定するほど検知数は少なく, 大きく正常範囲から外れた重要度の高いパケットを検出することができる．絶対値の値が小さいほど検

表 6　d=3, α=0.1, f=1.5 の実験結果

Name/date	後藤教授宅	伊藤宅	嶋田宅
P1	331	9,594	2,656
P2	538	17,436	4,850
P3	815	24,430	6,809
SrcIPDstIP	345	9,561	2,848

知数はフィルタリングなしの値に近づくが, 精度の高い結果が期待できる. しかし単純に重要度を高くして, 検知数を少なくした結果では, 到着数が少ない攻撃を検出できない. また重要度を低くして, 検知数を多くした結果では, 正しいパケットを異常として検出してしまう可能性がある. 以上より, 本研究における IDS の適切な設定は $d = 3$, $\alpha = 0.1$, 重要度は 1.5 とする.

4.3 IDS 実験結果サンプル

実際に IDS で抽出されたアラートの中で, 後藤教授の自宅のサーバーへの辞書攻撃の可能性がある攻撃が検出された.

辞書攻撃の可能性があるアラート

```
18:04:46(2009/10/22) Proto:6
Sip:58.20.61.67 Dip:218.227.161.45
Dport:22 1896 -1.100000<4.900000
Warnning!!
```

ポート番号 22 は ssh での通信である. この攻撃はサーバに SSH ログインをする時のパスワード認証で想像しやすいパスワードを総当たりで入れていく攻撃である. 中国から 9 件, トルコから 2 件, 台湾から 1 件の攻撃が確認された. Snort ではこの攻撃が検出されなかった.

また, 本研究の IDS と Snort のアラート数を比較したところ, 本研究の IDS のアラート数はほとんどの分類が Snort のアラート数より多く, 本研究の IDS のアラートには Snort では検知できない攻撃を含んでいる可能性が高いと考えられる.

4.4 GK フィルタ操作

IDS から GK のフィルタ操作が正常に行えているかどうかを ping コマンドで実験した. IDS がフィルタを操作するためにはアラートを生成させなければならない. そこで $d = 0$, $\alpha = 0.1$, 重要度 0, 異常検知の間隔を 10 秒とし, 故意にアラートを生成させるような設定にした. また, 異常検知は P3 のみとした. 本研究ではネットワークエミュレータとして Goto's IP Network Emulator(以下, GINE とする)[2][5] を用いて実験をする. GINE 上で NameSpace を 4 つ作成し, それぞれを外部ネットワーク, GK, IDS, 内部ネットワークと見立て, 環境を構築した.

4.5 GK フィルタ操作実験結果

以下に GK の実行結果を示す.

GK 実行結果 (IN ルール一部)

```
 Rule_num 30:Src 192.168.0.2/ffffffff Dst
192.168.0.4/ffffffff
proto 1 sport 0 dport 0
  action DROP
  by 192.168.0.3 time inst 1261470168 last
0 exp 10 refcount 0
  comment From IDS.
```

IDS を実行し, 外部ネットワークである 192.168.0.2 から内部ネットワークである 192.168.0.4 へ ping コマンドを送る. IDS でアラートが生成されると GK のフィルタリングルール設定方法に従って段階的に通信制限が行われた. 以上のことから, フィルタ操作がリアルタイムに行われ GK に適応されていることが確認できた.

5 おわりに

本研究は, PC で IPS を実現しネットワークを攻撃から防御することを目的としている. そのために, しきい値モデルを用いた IDS を実装し既存の GK と連携させることで防御することが可能となった.

運用レベルでの効果測定として実際のネットワークを流れている総パケット数約 2 億 9000 万のパケットログを用いて IDS の性能評価をした. 実験結果からも分かるように, トロイの木馬の可能性がある通信を検出できたり, ポートスキャンを検知することができたことで IDS の有効性を示すことができた. さらに, IDS がアラートを生成した際にその異常に合わせて GK のフィルタリングルールの操作を正しく行うかどうかも確認できた.

これらより, 本研究で実現した IPS でネットワークを攻撃から防御することができると考えられる.

今後の課題は以下の 4 つである.

- IPv6 への対応
- しきい値モデル以外や時間帯や曜日毎での異常検知手法の実装
- 異常の内容に合わせた最適な通信制限の選択
- 実ネットワークで IPS の実験

参考文献

[1] 福井麻美, 末吉昭仁：セキュリティのための段階的通信制限システムの評価と改良, 卒業論文, 南山大学数理情報学部情報通信学科 (2008).

[2] Ihara, A., Murase, S. and Goto, K.: IPv4/v6 Network Emulator using Divert Socket, *Proc. of 18th International Conference on Systems Engineering(ICSE2006)*, pp. 159–166 (2006). Conventry.

[3] 警察庁：平成 21 年上半期におけるインターネット治安情勢 (accessed August2009). http://www.cyberpolice.go.jp/detect/pdf/H21_kamihanki.pdf.

[4] 太田耕平, マンスフィールドグレン：インターネットにおける不正アクセス検出技術：NIDS の現状と将来, 電子情報通信学会論文誌. B, Vol. 83, No. 9, pp. 1209–1216 (2000).

[5] Sugiyama, Y. and Goto, K.: Design and Implementation of a Network Emulator using Virtual Network Stack, *Proc. of the Seventh International Symposium on Operations Research and Its Applications (ISORA2008)*, Vol. 8, pp. 351–358 (2008). Lecture Notes in Operations Research.