社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－CSEC－33 (10)
2006／5／12

マトリックス分解によるパケットフィルタリングルールの分析
－多段分析と同一性分析－

松田 勝志
NEC インターネットシステム研究所

概要

企業や組織のネットワークを外部からの不正なアクセス等から守る方法の一つにパケットフィルタリングがある．パケットフィルタリングを適切に運用管理するには，複雑で多数のルールを正確に把握しなければならない．更にフィルタリングシステムが複数存在する場合や，書き方の異なる設定があるため，フィルタリングシステム管理のコストは総じて高価である．

本稿では，パケットフィルタリングのルール集合を詳細に分析することができるマトリックス分解とそれを用いた分析手法について述べる．多段分析は直列に配置された複数のパケットフィルタリングシステムの挙動をシミュレートする．また同一性分析は書式の異なるルール集合間でフィルタリングの意味が同じかどうかを検証する．これらの分析を実装したシステムを用いることで複数のパケットフィルタリングシステムを管理することが容易になる．

A Packet Filtering Rules Analysis by Decomposing into Matrixes
－ Series and Identity Analysis －

Katsushi MATSUDA
Internet Systems Research Laboratories, NEC Corp.

Abstract

Packet filters are essential for organizations that are connected to the Internet. Network administrators have to understand precisely complicated rules to manage the packet filter. Moreover, there are plural filtering systems and several configurations. Management cost keeps high.

In this paper, we describe a novel model called "matrix decomposition" which enables to analyze rules of filtering and two analysis applications using the model. First application called series analysis provides virtual packet filtering systems combined plural filters. The other one called identity analysis verifies the identity of filtering meanings between plural filters written with different fashions.

1 はじめに

企業や組織のネットワークを外部からの不正なアクセス等から守る方法の一つにパケットフィルタリングがある．パケットフィルタリングは，複数のネットワークを接続するゲートウェイやルータに設置されるネットワーク機器またはソフトウェアであり，ネットワークを流れるパケットの属性をルールと照合することで，そのパケットの通過の可否を決定し，内部ネットワークを守る．

内部ネットワークを守るためにはパケットフィルタリングのルール集合を適切に設定・管理する必要がある．しかしながら，各ルールの条件は複数の属性の組み合わせになるため，ネットワーク管理の専門家ですらルール集合全体としてどうなっているのかを理解することは難しい．

よくあるフィルタリングシステム管理における問題に，ルール数が多くなってしまい，メンテナンスができなくなったとか，設定を少し変えたら，下部組織から苦情が来たとか，フィルタリングシステムの機種変更をしたら，意図した通りに動作しない，等がある．これらの問題のうち，最初の増大したルール数による管理コストの低減に対しては，不要ルール削除とルール統合によってルー

ル数を大幅に削減するルール圧縮の方式を提案済みである[1][2]. そこで本稿では, 残りの 2 つの問題に対してフィルタリングシステム管理者を支援する方法について述べる. すなわち, 複数のフィルタリングシステムからなるネットワークにおけるパケット通過判定と, 意味的に同じであるべき複数のフィルタリングシステムの比較を行う分析技術について述べる. 前者を多段分析, 後者を同一性分析と呼ぶ.

筆者らは, [1]および[2]でパケットフィルタリングのルール集合を詳細に分析することを可能にするマトリックス分解の手法を報告している. このマトリックス分解を複数ルール集合に拡張することで多段分析と同一性分析を実現する.

2. マトリックス分解

パケットフィルタリングのルール集合を分析するために適したデータ表現形式がマトリックス空間データであり, ルール集合からマトリックス空間データを作成する方法がマトリックス分解である. 本節では, これらについて概観する.

マトリックスとは, ルール集合で作る多次元空間を, 各ルールの各条件属性の範囲指定された全ての境界点で区割りしてできる最小の多次元立方体である. このマトリックスにより, ルール集合全体を表現したものがマトリックス空間データである. そして, ルール集合からマトリックスを作成し, ルールとマトリックスの対応付けを行うことでマトリックス空間データを生成することをマトリックス分解と呼んでいる. 以下, マトリックス分解について詳細に述べる.

2.1 マトリックス生成

ルール集合からマトリックスを作成するのがマトリックス生成フェーズである. 一般的なパケットフィルタリングシステムでは, 5 種類以上の条件属性 (SrcIP , SrcPort , DestIP , DestPort , Protocol) が用いられているが, ここでは簡単のため, 2 種類で説明する. また, それらの条件属性は 0 から始まり, 比較的小さな値(例えば, 13 や 15 等)で終わるものとする.

図 1 に 8 個のルールを持つルール集合の例とそれを 2 次元平面で表現する. 優先順位は R1 が最も高く, R8 がデフォルトルールである. ルールの書式は(*第 1 属性の範囲, 第 2 属性の範囲, アクション*)とする. アクションは A が accept (通過を許可), D が deny (通過を禁止)を表し, 2 次元平面では A が白, D が灰色となっている.

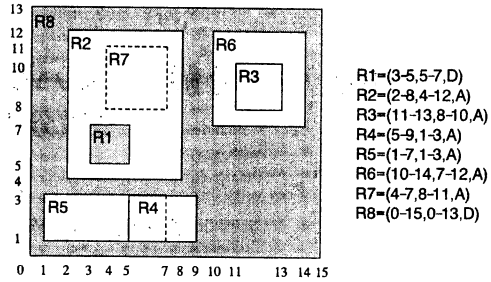

図 1. ルール集合の例

マトリックス生成は, まず条件属性毎にルール集合の各ルールの条件範囲の開始点と終了点(以降, 境界点と呼ぶ)を集め, 重複をなくした上でソートする. そして各属性を軸として, 境界点の交点で極小の多次元立方体を作る. この多次元立方体がマトリックスである. 図 1 のルール集合の例では, X 軸(第 1 属性)の境界点が 14 個, Y 軸(第 2 属性)の境界点が 11 個であるため, 130 個 (=(14-1)*(11-1)) のマトリックスが生成される.

マトリックスは, ルールと類似した書式を持つ. すなわち, (*第 1 属性の範囲, 第 2 属性の範囲, ルールリスト*) である. 以降, この書式のデータをマトリックスデータと呼ぶ.

2.2 ルールとマトリックスの対応付け

マトリックスを生成した後, ルールとマトリックスの対応付けを行う. すなわち, ルールにはそのルールを構成するマトリックスのリストを, マトリックスにはそのマトリックスに関係するルールのリストを列挙することである.

この対応付けによって, ルールの書式は(*第 1 属性の範囲, 第 2 属性の範囲, アクション, マトリックスリスト*)となる. 以降, この書式のデータをルールデータと呼ぶ.

ルールデータのマトリックスリストにはそのリストの順序には意味がないが, マトリックスデータのルールリストの順序はルール集合の優先順位に準拠している必要がある. 例えば, 図 1 の平面図の R1 の部分で説明すると, R1 の下に R2 があり, 更にその下に R8 があるため, R1 を構成するマトリックスデータのルールリストは, (R1,R2,R8) となる.

2.3 マトリックス空間データ

ルールと対応付けたマトリックスデータ, マトリックスと対応付けたルールデータを総称したもの

がマトリックス空間データである．このマトリックス空間データを用いることで様々な分析が可能となる．例えば，分析の例としてパケット通過テスト(packet classification)がある．

2.1節では，マトリックスの開始点と終了点はそれぞれルール集合で作られる境界点としていたが，実際のマトリックス空間データでは開始点は-0.5，終了点は+0.5 の値をそれぞれ加えている．ルールに記載される条件範囲の両端の値もその条件に含まれるためであり，このように境界をずらすことで両端の値も正確に扱うことができる(詳細は[1]を参照)．

またマトリックスの特性上，ルール数が増加すると，生成されるマトリックス数も指数的に増加する．この増加を抑制するために，不要なマトリックスを削除している．すなわち，デフォルトルールにしか関係しないマトリックスをマトリックス空間データから削除する[1]．このマトリックス削除によって，生成されたマトリックスの 1/4～1/5 に削減できる．

3 多段分析

本節では，複数のパケットフィルタリングシステムからなるネットワークにおけるパケット通過判定を行う多段分析について述べる．

ある程度の規模以上の組織のネットワークには，組織間の機密保持等を目的として，フィルタリングシステムが設けられることが多い．組織の規模が大きくなるに従い，フィルタリングシステムの台数は増える．フィルタリング動作の特性から，下流のネットワーク(組織の末端に近いネットワーク)は上流のネットワーク(インターネット等の外部に近いネットワーク)に比べて通過できるパケットの条件範囲が狭く，上流で通過を禁止されると，下流でどのような設定にしようともその範囲のパケットは通過しない．そのため，上流での設定ミスは下流のネットワークに大きな影響を与えることになる．多段分析は，このような設定ミスを削減することを目的としている．

3.1 マトリックス分解の拡張

2 節で述べたマトリックス分解は，1 つのルール集合からマトリックス空間データを生成する方法である．しかしながら，多段分析では，複数のルール集合の合成によるマトリックス空間データを生成する必要がある．以下，複数のルール集合から一つのマトリックス空間データを生成する方法について述べる．

2 節のマトリックス分解では，条件属性毎にルール集合の各ルールの境界点を集め，ソートと重複の除去を行い，マトリックスに分解していた．複数のルール集合(XとYとする)の場合は，それらのルール集合に含まれる全てのルールを用いて境界点を作成する．ルールとマトリックスの対応付けでは，ルールデータに付与するマトリックスリストは単一ルール集合と同様である．一方，マトリックスデータに付与するルールリストは，ルール集合の上流下流を考慮する必要がある．例えばルール集合 X が上流で Y が下流とする．あるマトリックスに関連するルール集合 X のルールが R1 と R6，Y のルールが R3 と R4 と R7 の場合，ルールリストは(X-R1,X-R6,Y-R3,Y-R4,Y-R7)となる．ここで X の R6 と Y の R7 はデフォルトルールである．

3.2 通過判定

各マトリックス単位にパケットの通過判定を行う．ルール集合が 1 個の場合は，マトリックスデータのルールリストの先頭ルールのアクションがそのマトリックスの通過の可否となるが，ルール集合が 2 個以上の場合は，ルールリストを走査する必要がある．具体的には，ルールリスト中のルール集合別にパケット通過の可否を求め，その論理積を求めることで，多段になったフィルタリングシステムの最終的な通過可否を判定する．

次のような例を考える．あるマトリックスデータのルールリストが，以下のようになっているとする．

(X-R1,X-R6,Y-R3,Y-R4,Y-R7)

今，X-R1 のアクションが accept，Y-R3 のアクションが accept の場合は，A & A = A となり，このマトリックスのアクションは accept(通過を許可)となる．一方，X-R1 または Y-R3 のアクションが deny の場合は，マトリックスのアクションは deny (通過を禁止)となる．ルール集合が 3 個以上ある場合でも同様に論理積を求めることで通過判定が可能である．

4 同一性分析

本節では，複数のパケットフィルタリングシステムのルール集合が意味的に同じであるかを比較する同一性分析について述べる．

4.1 利用シーン

同一性分析を用いる場面は，比較的限られて

おり，以下のような場面が考えられる．

(1) 多層防御時の設定確認
(2) 機器リプレース時の設定確認
(3) ポリシーとの同一性確認
(4) ルール集合整理時の同一性確認
(5) ルール変更時の設定確認

設定ミスや機器に脆弱性があることを考慮し，幾重にも防御壁を築くことで，侵入や攻撃からネットワークを守ることを多層防御または多重防御と呼ぶ．多層防御を行う場合，脆弱性の存在を考慮して[1]，各層の機器は別の種類を用いる．各層で同じフィルタリングを行うことを確認する必要がある．

フィルタリング機器を入れ替える場合，ルールの記述形式の違いから，旧機器と新機器が同じ設定になっているかどうかを確認することは困難である．それぞれの機器でサポートする記述形式が異なれば，単なる1対1のルール変換ではなくなるため，より難しくなる．

組織によっては，フィルタリングのポリシーを作成していることがある．そのような場合，定期的にフィルタリングシステムの設定がポリシーに則しているかどうかを確認する必要がある．

フィルタリングシステムの管理者が交代した場合，新しい管理者が設定の意味を確認することや自身のスタイルに変更することを目的として，ルール集合を書き直すことがある．例えば，日々の運用で増加したルール数を削減することを目的に不要なルールを削除したり，前任者がデフォルト許可で作成していたルール集合をデフォルト禁止に書き直したりである．

最も多く考えられる場面が，ルール変更時の確認のためである．すなわち，ルール集合間の同一性を確認するのではなく，異なる箇所を確認するのである．例えば，あるポートに対するアクセスを禁止するルールを追加した場合，追加前と追加後のルール集合を比較し，異なる箇所を特定し，その箇所が追加ルールと一致しているかどうかを確認することができると非常に有益である．

[1] 例えば，1層目の機器がその機器固有の脆弱性で攻撃された場合でも，2層目が別の機器ならば，同じ方法では攻撃が成功せず，侵入に時間がかかるため．

4.2 マトリックス分解の拡張

同一性分析では，複数のルール集合間でそれらの動作が同一であるかを比較するため，マトリックスを共通のものにする必要がある．

多段分析と同様に複数のルール集合に含まれる全てのルールを用いて境界点を作成する．ただし，マトリックス空間データは各々のルール集合毎に作成する．その上で，ルール集合毎にマトリックスとルールの対応付けを行う．これによって，マトリックスの構成が共通な複数のマトリックス空間データが作成される．

4.3 同一性判定

対応するマトリックスを順に比較することで複数のルール集合の動作の意味が同一かどうかを判定することができる．前述のようにマトリックスデータは共通なため，単純に1対1の比較が可能である．

具体的には，比較対象のルール集合のマトリックスデータを順に走査することになる．例えば，ルール集合XとYの同一性を判定する場合，Xの1番目のマトリックスデータとYの1番目のマトリックスデータのそれぞれのルールリストの先頭ルールのアクションを比較，以降2番目，3番目と続く．全てのマトリックスのアクションが等しい場合は，XとYはフィルタリング動作が同一であると言える．一方，アクションが異なるマトリックスがある場合，それを表示することで同一でない箇所を示すことができる．

5 フィルタリングルール管理システム

多段分析と同一性分析を行うフィルタリングルール管理システムを開発した．管理システムは，Microsoft Windows Xp のアプリケーションで，Microsoft Visual C++ 6.0 で開発している．ルール集合の記述は，アクセス制御やフィルタリング設定や侵入検知システム設定を共通のフォーマットで記述できる汎用ポリシー記述言語SCCML[3]を用いている．

5.1 多段分析

図2に，SCCMLに変換したパケットフィルタリングルール集合(FW1.xml)を読み込んだシステムの画面を示す．システムはMDIを用いており，一つのルール集合を一つのウインドウで表示している．ウインドウ上部がルール集合の内容である．デフォルトルールはルール番号000として先頭に表示している．また，読み込んだルール集

合を用いてパケットテスト(Packet Classification)を行う機能も装備している(FW シミュレータ). またルール集合の読み込みと同時に簡単な分析も行っている. 図 2 のルール集合の場合, ルール 004, 005, 007 がそれぞれ別のルールと条件範囲が一部重なっていることを検出している.

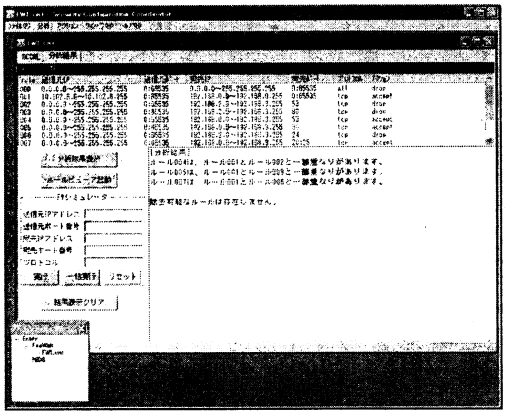

図 2. ルール集合単体の分析例

図3に2つ目のルール集合(FW2.xml)を読み込ませた後, 多段分析を実行した結果の画面を示す[2].

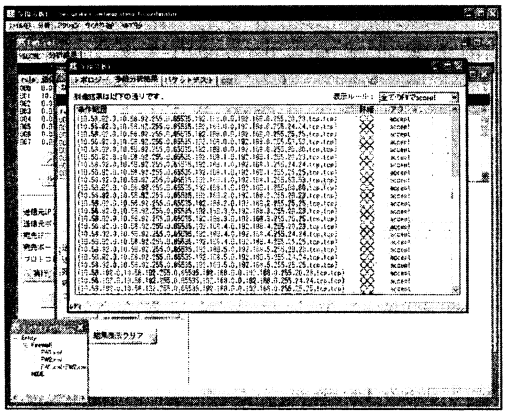

図 3. 多段分析の実行例

管理システムは, これら 2 個のパケットフィルタリングシステムを通過するパケットの領域をマトリックス単位で表示する. 図 3 の各行がマトリックス単位の通過の可否を示している. 2 列目の「詳細」とは, 上流のフィルタリングシステムから順に通過を許可しているか禁止しているかを示している(○が許可で, ●が禁止). この例では最終的に通過が許可されるパケットの領域を示しているのでいずれの行も「○○」となっている.

また, ルール集合単体でのパケットテスト機能と同様にルール集合が複数ある場合でもそれを大きな一つのルール集合と見なしてパケットテストが可能である(図3の「多段分析 1」ウインドウに「パケットテスト」タブがある. これをクリックするとFW シミュレータが表示される).

管理システムには, エンティティツリーというウインドウが設けられている(図 2 や図 3 の左下). このウインドウには現在読み込まれているルール集合をエンティティという形で表示している. 図 2 では FW1.xml のみが表示されており, 図 3 にはそれに加えて FW2.xml が表示されている. 更に FW1.xml+FW2.xml というエンティティが追加されている. これは多段分析によって生成された仮想的なルール集合であり, 文字通り 2 つのルール集合を連結したものである. 上述の複数ルール集合に対するパケットテストもこの仮想的なルール集合を用いている.

5.2 同一性分析

同一性分析は, ルール集合同士を比較するものであるが, ここでは 5.1 節の多段分析結果(仮想的なルール集合:FW1.xml+FW2.xml)と別のルール集合(policy.xml)との比較の例を示す. 同一性分析の結果を図 4 に示す.

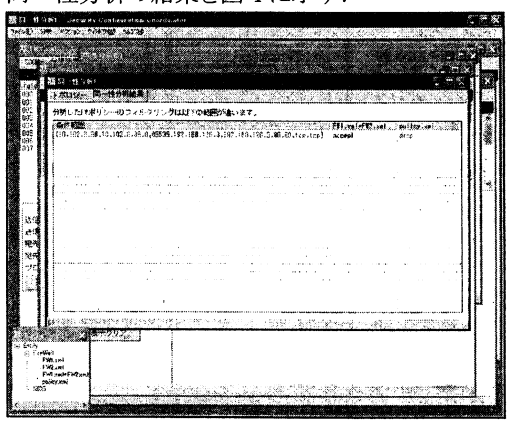

図 4. 同一性分析の実行例

図4は, 1箇所で差異があることを示している. この同一性分析で用いたルール集合のルール数は FW1.xml が 10 ルール, FW2.xml が 21 ルール, policy.xml が 39 ルールである. ルール集合のルール数も違えば, ルール自体もそれぞれ異なる.

[2] ここでは FW1.xml が上流, FW2.xml が下流として多段分析を行っている.

```
#sample01
(10.56.192.0/24,any,192.168.0.0/24,53,tcp,accept)
(10.56.192.0/24,any,192.168.1.0/24,53,tcp,accept)
(10.56.192.0/24,any,192.168.0.0/24,80,tcp,accept)
(10.56.192.0/24,any,192.168.1.0/24,80,tcp,accept)
(10.56.192.0/24,any,192.168.0.0/24,20-25,tcp,accept)
(10.56.192.0/24,any,192.168.1.0/24,20-25,tcp,accept)
(10.56.192.0/24,any,192.168.4.0/24,20-25,tcp,accept)
(10.56.192.0/24,any,192.168.5.0/24,20-25,tcp,accept)
(10.56.192.0/24,any,192.168.2.0/24,20-23,tcp,accept)
(10.56.192.0/24,any,192.168.2.0/24,25,tcp,accept)
(10.56.192.0/24,any,192.168.3.0/24,20-23,tcp,accept)
(10.56.192.0/24,any,192.168.3.0/24,25,tcp,accept)
(0.0.0.0/0,any,0.0.0.0/0,any,all,drop)
```

```
#sample02
(10.56.192.0/24,any,192.168.2.0/23,53,tcp,deny)
(10.56.192.0/24,any,192.168.2.0/23,80,tcp,deny)
(10.56.192.0/24,any,192.168.0.0/22,53,tcp,accept)
(10.56.192.0/24,any,192.168.0.0/22,80,tcp,accept)
(10.56.192.0/24,any,192.168.2.0/23,24,tcp,deny)
(10.56.192.0/24,any,192.168.0.0/22,20-25,tcp,accept)
(10.56.192.0/24,any,192.168.4.0/23,20-25,tcp,accept)
(0.0.0.0/0,any,0.0.0.0/0,any,all,drop)
```

図 5. 同一なルール集合の例

図 5 は，フィルタリング動作が同一である簡単なルール集合の例(sample01とsample02)である．ルールは，(*SrcIP*, *SrcPort*, *DestIP*, *DestPort*, *Protocol*, *Action*)の書式で記述している．ルールの条件属性のうち，SrcIPとSrcPortとProtocolは全て同じなので，これらのルール集合が同一かどうかを検証するには，DestIP と DestPort のみを考慮すれば良く，比較的簡単な例である．しかしながら，この程度のルール数であっても，条件範囲を図示等しながらでなければ難しく，人間にとっては非常に大変な作業であることが分かる．

6 関連研究

ルール集合単体に対するパケットテストは，packet classification と呼ばれ，非常に活発な研究領域である．packet classification では，あるパケットがどのルールと照合するかを速度とメモリ使用量の点から追求している([4]に詳しい)．一方，本稿で提案している複数のルール集合に対するパケットテストやルール集合同士の比較に関連する研究はまだ見当たらない．

7 おわりに

パケットフィルタリングのルール集合をマトリックス空間データによって表現することで，複数のフィルタリングシステムからなるネットワークのパケット通過判定(多段分析)や意味的に同じであるべき複数のフィルタリングシステムの比較を行う同一性分析が容易に実現できることを示した．

組織のフィルタリングシステムを管理するセキュリティ管理者やネットワーク管理者は，複数のフィルタリングシステムでどのようなフィルタリングが実現できているのか，ポリシー通りになっているのか等を常に確認・管理している．この作業を多段分析と同一性分析は支援することが可能であり，フィルタリングシステム管理コストを大幅に削減することができる．

本稿では，多段分析と同一性分析について報告したが，マトリックスを用いることで更に様々な分析が可能である．他の分析についても引き続き検討する予定である．

参考文献

[1] 松田，「マトリックス分解によるパケットフィルタリングルールの分析－不要ルールと冗長条件ルールの検出－」，情報処理学会研究報告 Vol.2005, No.122, pp.1-6, 2005.

[2] 松田，「マトリックス分解によるパケットフィルタリングルールの圧縮」，情報処理学会研究報告 Vol.2006, No.26, pp.43-48, 2006.

[3] 岡城他，「セキュリティ運用管理のためのセキュリティポリシー言語 SCCML」，情報処理学会研究報告 Vol.2004, No.129, pp.89-94, 2004.

[4] Gupta and McKeown, "Algorithms for Packet Classification", IEEE Network, Vol.15, No.2, pp.24-32, 2001.