空間的解釈に基づくファイアウォール解析システムの研究

高　橋　直　久　　名古屋工業大学大学院工学研究科教授

1 はじめに

安全で安定したネットワークを実現するためには，ファイアウォールや侵入検知システム（IDS）などのネットワークアクセス検査装置が不可欠である．ネットワーク管理者は，セキュリティの運用指針に従って，これらを正しく設定し，意図した通りに動作させるように維持管理しなければならない．しかし，企業等で実運用されているファイアウォールの設定を調べた結果として，ファイアウォールの設定誤りにより，通過すべき通信を通過させていないなどの状況が数多く発生していることが報告されている[1]．ファイアウォールの設定を正しく維持管理する上で，以下のような問題がある．

（1）ネットワークアクセス検査装置は，一般に，管理者が設定した一連のルール（フィルタと呼ぶ）を手続き型プログラムのように解釈実行することにより，パケットの通過と棄却を制御し，許されていない外部からの侵入を防ぐ．このため，経験豊富な管理者であっても，設定内容から動作を把握することは難しい作業となり，設定の不足や冗長を見落としてしまうことがある．

（2）メールを正しく転送できない，あるいは，意図したホームページにアクセスできないなどのネットワークに係わる異常事象が生じた場合には，その原因として多くの可能性が考えられる．このため，ファイアウォールの設定が起因していたとしても，異常の原因を究明し設定を修復するためには，数多くの試行錯誤を伴い多大な時間を要する場合がある．

（3）ネットワークの規模が大きくなると，これらの装置をネットワーク内に多数分散して設置し，それぞれ異なる管理者により設定されるようになる．このような場合には，相互の影響も考慮して注意深く設定する必要がある．

本研究では，ファイアウォール設定の基本となる IP パケットフィルタをとりあげ，その設定を解析するシステムについて探求する．本研究の目的は，上のような問題に対処するため，フィルタ系列を手続き型プログラムのように扱って各フィルタの動作を順に追いかける作業からネットワーク管理者を解放し，自動的に設定異常を検出して設定を正しく維持管理するための方法論を探求することにある．

2 パケットフィルタの解釈

2-1 パケットフィルタ

IP パケットフィルタでは，IP パケットの通過と遮断を制御する．このため，ネットワーク管理者は，あらかじめ，IP パケットヘッダの特定フィールド（キーフィールドと呼ぶ）の値を用いて通過すべきか遮断すべきか定めたルール（フィルタと呼ぶ）の系列（フィルタ系列）を記述する．たとえば，図 1 は，キーフィールドとして送信元 IP アドレスと送信先ポート番号の二つを用いて，フィルタ f1 から f3 の三つのルール，および，いずれのルールの条件も当てはまらない場合に実行すべきルール（デフォルトルール）を定めた例である．この例では，ネットワーク管理者は，フィルタ f1 により送信元の IP アドレスが 123.4.56.70 以上 123.4.56.90 未満のパケットを通過させ，フィルタ f2 により送信先ポート番号が 10～25 のパケットを通過させ，フィルタ f3 により送信元の IP アドレスが 123.4.56.50 以上 123.4.56.60 未満で送信先ポート番号が 10～25 のパケットを通過させ，その他のパケットを遮断するように指定している．

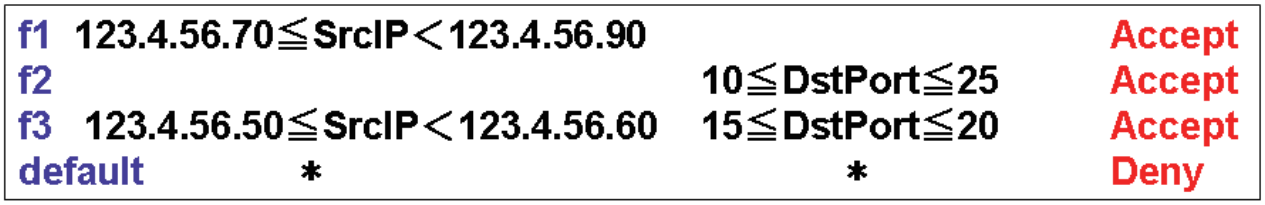

図 1　IP パケットフィルタの例

2-2 パケットフィルタの操作的解釈

多くのパケットフィルタリングシステムでは，パケットが到着すると，フィルタ系列を上から順番に読み出して，そのパケットがフィルタの条件を満たすか調べる．条件を満たす場合には，そのフィルタが指定したアクション（通過又は遮断）に従って，そのパケットを通過又は廃棄する．条件を満たさない場合には，次のフィルタを読み出して調べるという動作を繰り返す．このような動作を if 文により C プログラム風に記述してフィルタ系列の意味を表すと，図１のフィルタ系列は図２のようになる．図２で，P はパケットを，P.X はパケット P のキーフィールド X の値を表す．

```
if (123.4.56.70≦P.SrcIP and P.SrcIP＜123.4.56.90) then Accept P
else if (10≦P.DstPort and P.DstPort≦25) then Accept P
     else if ((123.4.56.50 ≦P.SrcIP and P.SrcIP＜123.4.56.90)
          and (15≦P.DstPort and P.DstPort≦20) ) then Accept P
else Deny P
```

図２　パケットフィルタの操作的解釈の例（図１の IP パケットフィルタの場合）

2-3 パケットフィルタの空間的解釈

フィルタは，キーフィールドの値に従ってアクションを施すべきパケットを指定する．今，キーフィールド数を M としたとき，各キーフィールドの値を座標とする M 次元空間上の点としてパケットを表すような空間（パケット空間と呼ぶ）を考える．このとき，フィルタの条件を，その条件を満たすパケットに対応するパケット空間上の総ての点の集合からなる部分空間（フィルタ空間と呼ぶ）として表すことができる．たとえば，図１のフィルタ系列は，図３のような２次元空間上の領域として表される．

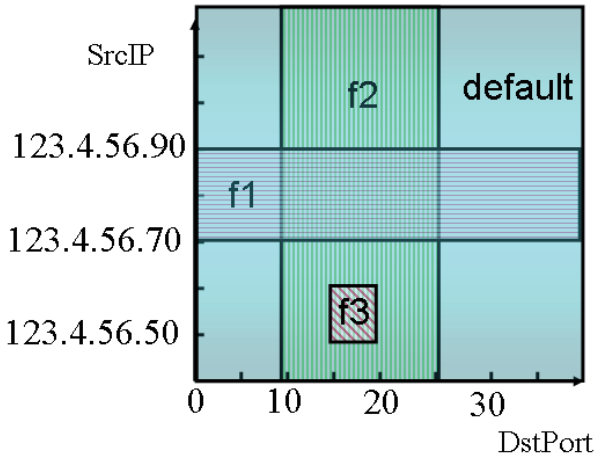

図３　パケットフィルタの空間的解釈の例（図１の IP パケットフィルタの場合）

このようにフィルタの意味を空間的に表すと，パケットフィルタリングにおいて，パケットが到着したときに，適用すべきフィルタを選択する問題は，パケット空間上で，パケットがどのフィルタ空間内に位置するかを求める問題，すなわち計算幾何学の位置決定問題となる．

3 ファイアウォールの設定異常

3-1 先行フィルタにより生じるコンフリクト

図３をみると，f1，f2 のフィルタ空間の一部に重なりがあることがわかる．また，f3 のフィルタ空間が f2 のフィルタ空間に完全に包含されていることがわかる．このようにフィルタ空間に重なりがある場合には，これらのフィルタにより共通して指定されるパケットが存在する．このようなパケットに対しては，先に評価されるフィルタ（先行フィルタと呼ぶ）のアクションが施され，後から評価されるフィルタ（後続フィルタと呼ぶ）のアクションは施されることはない．このような場合に，後続フィルタに対して先行フィルタによるコンフリクトが生じたという．f3 に対して f2 により生じるコンフリクトでは，どのようなパケットに対しても f3 のアクションは決して施されることはない．このようなコンフリクトは設定誤りの結果生じたものである．一方，f2 に対して f1 により生じるコンフリクトでは，f2 が指定するパケットの一部は f2 のアク

ションが施される．f2 をこのようなパケットだけを指定するように変更すれば，コンフリクトは生じない．しかし，f2 の記述が複雑になる，あるいは，複数のフィルタに分割しなければならないという事態が生じる．ネットワーク管理者が，このような問題を回避するために敢えてコンフリクトを生じるように f2 を記述することもあるので，このようなコンフリクトは設定誤りとはいえない場合もある．しかし，この場合でも，意図的にコンフリクトを生じさせたのか，設定誤りか確認するため，ネットワーク管理者にコンフリクトが生じていることを通知する必要がある．

図 2 のような分岐文を含むプログラムにおいて，分岐文の条件によりプログラムの実行経路が変化する．条件部の値がどのように変化しても決して実行されない経路（実行不能経路(infeasible path)と呼ぶ）がある場合には，バグの可能性が高いため，実行不能経路の検出法が研究されている．例えば，図 2 の例の場合には，3 つ目の if 文の条件は決して真にならないので，この if 文の then パートに至る実行経路は，実行不能経路とみなされる．このように，フィルタの意味を操作的に解釈する場合には，コンフリクトの検出は，実行不能経路検出問題となる．一方，フィルタの意味を空間的に解釈する場合には，この問題は，前述のようにフィルタ空間の重なりを検出する問題となる．以下では，フィルタ空間の重なり状態を詳細に分類し，この結果を用いてコンフリクトを詳細に分類して，各コンフリクトの検出法を明らかにする．

パケット空間上に２つのフィルタ空間を表すと，フィルタ間の空間的関係を得ることができる．フィルタ f，g のフィルタ空間を *S(f)*，*S(g)*とすると，f，g の空間的関係 *R(f,g)*は，以下のように分類できる．

$$R(f,g)=\begin{cases} Disjoint:(S(f)\cap S(g))=\phi \\ Equivalent:S(f)=S(g) \\ Inclusion1:S(f)\supset S(g) \\ Inclusion2:S(f)\subset S(g) \\ Correlation:otherwize \end{cases} \qquad (1)$$

図１の例のように，フィルタのキーフィールドが送信元 IP アドレス *SrcIP*，送信先ポート番号 *DstPort* の二つのとき，２つのフィルタの空間的関係は図４に例示するように 2 次元平面上の２つの矩形領域の包含関係として表される．図４の(b)から(e)のように，２つのフィルタ空間がオーバーラップしている場合，コンフリクトがあるといい，フィルタを誤って設定した可能性がある．たとえば，図 4 の(b)の場合には，フィルタ *f* と *g* のフィルタ空間は同じであり，フィルタ系列で下方に記述されているフィルタのアクションは決して実行されない．Shaer らが分類した各コンフリクト[6]を，*R(f,g)*，及び，*f* と *g* のアクションを用いて表現すると以下のようになる．

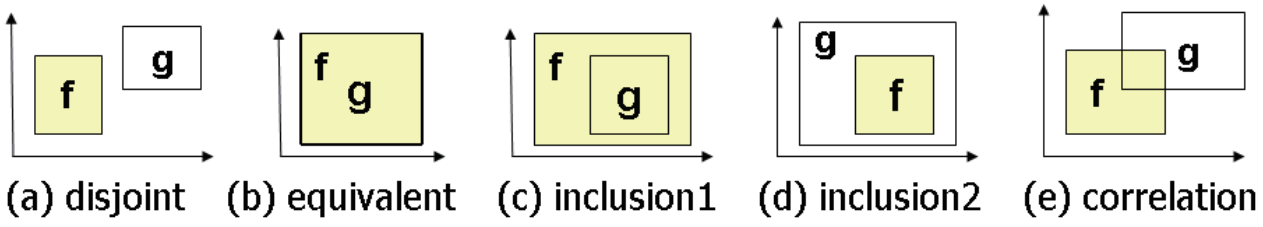

図４　２つのフィルタの空間的関係

（1）　単一ファイアウォールにおけるコンフリクト

(a) リダンダンシーエラー(redundancy error)

前述の f と g が同じファイアウォールに設定されたフィルタであり，*f* が *g* よりも上に記述されていたとする．*R(f,g)*が *Equivalent* または *Inclusion1* のとき，*g* のアクションは決して実行されない．このとき，*f* と *g* のアクションが同じ場合は，*g* で意図した操作は *g* がなくても *f* で実行されるので，*g* はリダンダンシーエラーであるという．

(b) シャドーイングエラー(shadowing error)

f と *g* のアクションが異なる場合には，*g* で意図した操作が *f* の操作に隠れて決して実行されない．このようなコンフリクトをシャドーイングエラーと呼ぶ．

(c) ジェネラリゼーションウォーニング(generalization warning)

*R(f,g)*が *Inclusion2* で，*f* と *g* のアクションが異なる場合には，フィルタの条件を簡明にするため

など，意図的に g のフィルタ空間を大きくしてコンフリクトを発生させた可能性がある．このため，このようなコンフリクトはジェネラリゼーションウォーニングと呼び，エラーとは区別して注意を促すために管理者に提示する．

(d) コリレーションウォーニング(correlation warning)

$R(f,g)$が $correlation$ で，f と g のアクションが異なるとき，上記の場合と同様に注意を促すためにコリレーションウォーニングとして提示する．

（2）複数ファイアウォールにまたがるコンフリクト

図５のように上流と下流にファイアウォールが設置されていて，下流のファイアウォールの設定 (F2) で，あるノードに対するパケットの通過を許可しているときに，上流のファイアウォールの設定（F1）では当該パケットの通過を禁止している場合に，F1 と F2 に矛盾がある． f を F1 に設定されたフィルタ，g を F2 に設定されたフィルタとすると，上記（１）と同様に f と g の空間的関係を調べると，エラー及びウォーニングとなるフィルタを求めることができる．

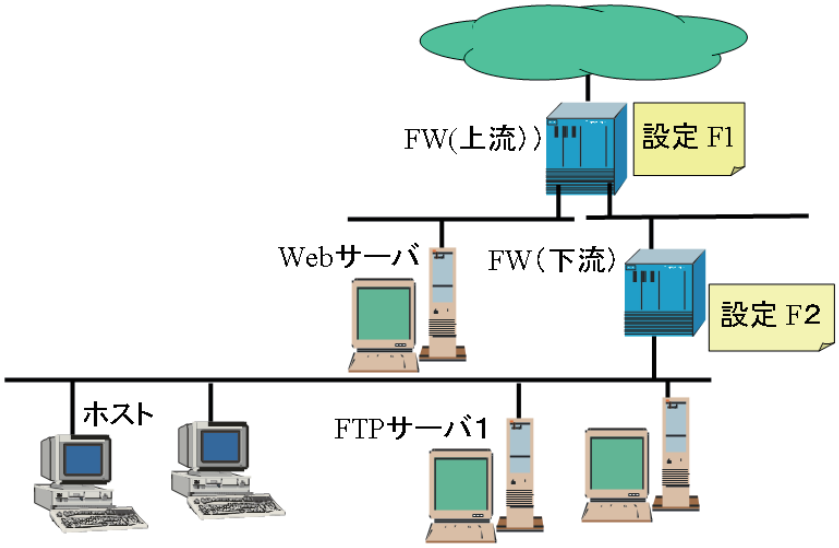

図５　ファイアウォール（FW）が多段に設置されたネットワークの例

3-2 先行フィルタの組合せにより生じるコンフリクト

前節では，ファイアウォールにおいて先行的に評価されるフィルタ f が，後続のフィルタ f に影響を与えるコンフリクトについて述べた．このように，１つのフィルタが別の１つのフィルタに対して生じさせるコンフリクトは，操作的解釈に従ってフィルタ系列を解析しても，空間的解釈による解析の場合と同様にあまり大きな困難を伴わないで求めることができる．しかし，複数のフィルタが先行的に評価されるために，後続のフィルタが影響を受けて決して実行されないような場合など，多対１の関係により生じるコンフリクトを，操作的解釈に従い求めることは困難である．これに対して，空間的解釈を用いると，前節で述べた手法の自然な拡張により，このようなコンフリクトを容易に求めることが可能になる．

図６のように，フィルタ $f1$ と $f2$ が，フィルタ g の前に評価されるようなフィルタ系列を考える．$f1$，$f2$ は，それぞれｇと一部に重なりがあり，コリレーションウォーニングを提示すべきコンフリクトを起こしているが，これだけでは，g が決して実行されないとはいえないので，設定エラーと断定できない．しかし，図７のように $f1$，$f2$，g のフィルタ空間をパケット空間上に表現すると，フィルタ g のフィルタ空間が，フィルタ $f1$ と $f2$ のフィルタ空間の和に包含されることが分り，設定エラーであると断定できる．

f1	SrcIP＜123.4.56.90	DstPort＜20	Accept
f2	123.4.56.78≦SrcIP		Accept
g	123.4.55.0≦SrcIP＜123.4.57.0	10≦DestPort＜20	Accept

図６　フィルタの設定例（フィルタの組み合わせによりコンフリクトが生じる場合）

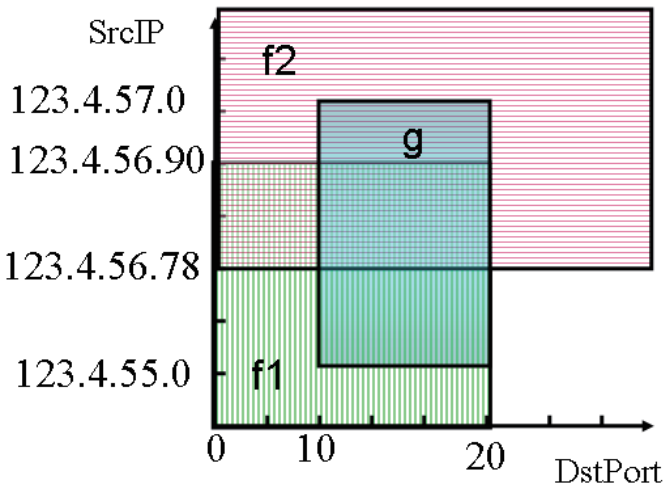

図７　フィルタの空間的関係（図６のフィルタの組み合わせによりコンフリクトが生じる場合）

今，K 個のフィルタからなるフィルタ系列を C_K=C[f1, f2, …, fK]と表し，C_Kのフィルタ空間を以下のように定義する．

$$S'(C_K) = S(f1) \cup S(f2) \cup ... \cup S(fK) \qquad (2)$$

C_Kとフィルタ g の空間的関係 $R'(C_K, g)$は，$R(f, g)$と同様に式（４）のように定めることができる．C_Kのすべてのフィルタのアクションが同一で，すべてフィルタ g よりも先行的に評価されるとき，C_Kと g の空間的関係 $R'(C_K, g)$と C_Kと g のアクションに従って，１対１の場合と同様にエラーとウォーニングとなるコンフリクトを求めることができる．

$$R'(C_K, g) = \begin{cases} Disjoint : (S'(C_K) \cap S(g)) = \phi \\ Equivalent : S'(C_K) = S(g) \\ Inclusion1 : S'(C_K) \supset S(g) \\ Inclusion2 : S'(C_K) \subset S(g) \\ Correlation : otherwize \end{cases} \qquad (3)$$

4 ファイアウォール解析システム

4-1 フィルタ逆引きシステム

パケットフィルタリングシステムでは，フィルタが，そのフィルタのアクションを施す対象となるパケットを定めている．このため，フィルタ系列からフィルタを読み出すと，そのフィルタが対象としているパケットを知ることができる．このような作業をフィルタの意味を調べる操作とみなすと，辞書で単語を調べる操作と同様となるので，フィルタ正引きと呼ぶ．一方，次のように，ある特定のパケットから，それらのパケットを対象とするフィルタを求める操作は，辞書の逆引きのように捉えられるので，フィルタ逆引きと呼ぶ．フィルタ逆引きは，たとえば，自分のホストから特定の外部サーバにパケットを送ることができない，あるいは，自分のホストが受信するはずのパケットが来ないといったトラブルが生じた場合に，その原因を調べる作業を助ける．すなわち，フィルタ逆引き機能は，注目するパケットから，そのパケットを棄却するように記述されたフィルタ，あるいは，通過させるように記述されたフィルタを求める．

本研究では，3-1 節で述べた 1 対 1 の関係に基づく異常に係わるフィルタを求めるフィルタ逆引きシステムを実現した．ここでは，メールを正しく転送できない，あるいは，意図したホームページにアクセスできないなどのネットワークに係わる異常事象を，それらの事象に関連した送信元ホストと送信先ホストの対などからなるフローとして表現し,そのフローを対象とするフィルタを求めるフィルタ逆引き機能を実現した．また，そのフローを実際に通過又は遮断したフィルタ（実効フィルタと呼ぶ）を求める機能を実現した．

本研究では，プロトコル番号，送信元 IP アドレス，送信元ポート番号，送信先 IP アドレス，送信先ポート番号の５つをキーフィールドとする IP パケットフィルタのフィルタ逆引きシステムのプロトタイプを開発した．

4-2 フィルタの組み合わせにより生じるコンフリクトの検出システム

3-2 節で述べたように，先行的に評価されるフィルタの組により後続のフィルタが決して実行されないような設定誤りが生じる場合がある．今，K個のフィルタからなるフィルタ列，$C_K=C[f1,f2,\cdots,fK]$ $(K\geqq 2)$ がフィルタ gに先行し，C_Kのフィルタ空間が gのフィルタ空間と等しい又は包含しているため，gが決して実行されないとする．このとき，$f1,f2,\cdots,fK$ から任意の１つのフィルタを取り除いてできる $K-1$ 個のフィルタの組のフィルタ空間が gのフィルタ空間と等しくなく，かつ包含しないとき，すなわちコンフリクトによるエラーを生じないとき，C_Kを gのエラーに対する必須フィルタ列と呼ぶ．ファイアウォールに設定されているフィルタ系列の中から，このような必須フィルタ列を見つけ出すことは，エラーの遡及範囲を絞り込むので，エラーの原因を究明して修正を加える上で有効である．

我々は，以下の２つの手法を開発し，各手法を用いたコンフリクト検出システムを開発した．いずれも，フィルタ列 C_Kとフィルタ gが与えられたとき，C_Kのフィルタ空間が gのフィルタ空間と等しい又は包含しているとき，C_Kの全部あるいは一部のフィルタからなる必須フィルタ列を求める手法である．

（１）*RFM-T*

トップダウンアプローチにより必須フィルタ列を求める．この手法では，まず，C_Kからフィルタを１つ除いてできる K個のフィルタ列と gの空間的関係を調べる．K個のフィルタ列のフィルタ空間の中に，gのフィルタ空間と等しい又は包含するものがない場合には，C_Kが必須フィルタ列であると判定する．そうでない場合，すなわち，あるフィルタ列のフィルタ空間が gのフィルタ空間と等しいか包含する場合には，そのフィルタ列から更に１つフィルタを除いてできる $K-1$ 個のフィルタ列と gの空間的関係を調べる．このように，フィルタを１つずつ取り除いていき，取り除いてできた総てのフィルタ列のフィルタ空間のなかに，gのフィルタ空間と等しいか包含するようなものがないとき，元のフィルタ列を必須フィルタ列と判定する．

（２）*RFM-B*

ボトムアップアプローチにより必須フィルタ列を求める．この手法では，まず，C_Kからフィルタを１つずつ取り出して，それぞれのフィルタと gの空間的関係を調べる．このとき，gのフィルタ空間と等しいか包含するようなフィルタ空間をもつフィルタがある場合には，そのフィルタが必須フィルタであると判定する．そうでない場合には，gと *correlation* 又は *inclusion2* の関係を有するフィルタだけを C_Kのフィルタから抽出する．次に，これらのフィルタから任意の２つのを組み合わせてできるフィルタ列と g の空間的関係を調べる．gのフィルタ空間と等しいか包含するようなフィルタ空間をもつフィルタ列がある場合には，そのフィルタ列を必須フィルタ列と判定する．そうでない場合には，gと *correlation* 又は *inclusion2* の関係を有するフィルタ列だけを残し，このフィルタ列に，g と correlation 又は inclusion2 の関係を有するフィルタを加えた３つのフィルタからなるフィルタ列を作り，gと空間的関係を調べるというように，必須フィルタ列が見つかるまで，フィルタ数を１つずつ増加させていく．

5 関連研究

インターネットの社会的役割が増すにつれて，ネットワークアクセス検査機能 (NAI) の重要性が認識され，NAI の設定の検証に関する研究，および，セキュリティポリシーから NAI の設定の生成に関する研究が活発になってきている．Guttman らは，文献[2]でセキュリティポリシーを用いてルータのフィルタリング設定の生成と検証を行うアルゴリズムを提案している．しかし，セキュリティポリシー内の不整合を検出できない，実ネットワークとの対応を考慮していないため設定の冗長や不足を検出できないなどの問題がある．また，Hazelhurst らは，ファイアウォールのフィルタリングの設定検証システムを提案し，２つのフィルタを比較し，設定の差を求めることを可能にしている[3]．しかし，セキュリティポリシーとフィルタリング設定との比較検証は考慮していない．

下條らは文献[4] で，ネットワークに分散して存在する複数のファイアウォールを一括で設定するシステムを提案している．この研究では，管理者が与えるネットワークのトポロジー情報とファイアウォールを構成するホスト情報に基づき，ポリシからファイアウォールの設定を生成する．これにより，管理者はベンダによる設定方法の違いを意識することなく，複数のファイアウォール間で整合性を保った設定の作成が可能となる．佐藤らは文献[5]で，ネットワークに適した IDS のシグネチャを生成するシステムを提案している．この研究では，ネットワークのトポロジー情報とセキュリティポリシーを基に，汎用的に作成されたシグネチャからネットワークに適したシグネチャを生成する．これにより，正常なアクセスを不正と判断する誤りを削減し，管理者の負担を軽減することが可能となる．しかし，このためにホストの追加や削除などの実ネットワークの変動に対応するためには管理者が注意深くネットワークを監視する必要がある．

Shaer らは文献[6]で，２つのフィルタの述語の関係に基づいてフィルタの設定異常を詳細に分類する手法を提示し，それらの異常を検出するシステムを実現している．Eronen らは文献[7]で，ファイアウォールの解析用エキスパートシステムを提案している．このシステムは，管理者の質問に基づきファイアウォールの設定を解析し異常を通知する機能を有するが，複数のファイアウォール間の設定を比較し整合性を検証する機能はもたない．以上のように，従来の研究は NAI 設定の生成と検証に関する一部の要素技術の提案にとどまり，多くの課題が残されている．パケットフィルタのコンフリクトに関して，コンフリクトの検出[6,8]，冗長なフィルタの削除[9]，インタラクティブなファイアウォール解析システム[7]など多数の研究がなされている．これらの研究は，いずれも 3-1 で述べた単一の先行フィルタにより生じるコンフリクトを対象としており，複数の先行フィルタの組合せにより生じるコンフリクトは考慮していない．

フィルタを集合論的にとらえて,パケットフィルタリングを計算幾何学の位置決定問題として扱う方式の原型となったのは，筆者らの知る限りでは Degermark らの経路表探索法[10]である．彼らの方式では，経路表のエントリーを送り先 IP アドレスの１次元空間（直線）上の点の集合を表す線分として表し，到着パケットが,どのエントリの線分上にあるか高速に求めるためのデータ構造を与えている.彼らの方式提案の直後に，多次元空間に拡張したパケットフィルタリングの方式がいくつか提案された[11-15]．筆者らの SIERRA 木を用いたパケット分類器も，この一つであり，パイプライン的にパケット空間の次元を１つずつ小さくしていくことによりメモリ量を抑えながら高速にパケットを分類する方式を実現している[14-15]．文献[16-17]では，3-1 節で述べた１対１の関係に基づく設定異常について，SIERRA 木を用いてパケットフィルタのコンフリクトを求める方式を提案している．

6 おわりに

本稿では，空間的解釈により IP パケットフィルタを解析して，設定誤りを検出するシステムに関する研究について述べた．今後，以下の３つの方向に研究を展開する予定である．

（１）設定異常検出機能の適用領域を拡大する

通信状況より動作が変化するステートフルファイアウォールなど，設定内容が時間とともに変化するような場合に対しても適用できるように空間的解釈に基づく解析方式を拡張する．

（２） システムとしての完成度を高める

GUI の整備などを図り，ツールとしての実用性を高める．さらに，学内ファイアウォール等への適用実験を通して，設定異常の分類を詳細化し，システムへ反映させる．

（３）自己管理型ファイアウォールへ発展させる

検出した設定異常を回復するために自動的に設定を変更する機能について検討を進める．

【参考文献】

[1] A. Wool, “A Quantitative Study of Firewall Configuration Errors. Computer, vol. 37, no. 6, pp. 62-67, Jun., 2004.

[2] J. D. Guttman, “Security goals: Packet trajectories and strand spaces,” Lecture Notes in Computer Science,” Vol. 2171, pp.197-263 (2001).

[3] S, Hazelhurtst, “Algorithm for analyzing firewall and router access lists,” Technical Report TR-Wits-CS-1999-5, University of Witwatersrand (1999).

[4] 下條, 衛藤, 門林, ”ポリシによる複数ファイアウォールの一括設定方式の提案と実装”, 電子情報通信学会論文誌, Vol.J87-B, No.10, pp.1616-1625 (2004).

[5] 佐藤, 今泉, ”ネットワーク情報を用いた IDS のシグネチャ構築手法”, 情報処理学会 分散システム／インターネット運用技術シンポジウム論文集, pp.51-56 (2004).

[6] E.Al-Shaer and H. Hamed, R. Boutaba and M. Hasan “Conflict Classification and Analysis of Distributed Firewall Policies,” IEEE Journal on Selected Areas in Communication. Vol. 23, No. 10, pp. 2069-2084 (2005).

[7] P.Eronen and J. Zitting. “An expert system for analyzing firewall rules,” Proc. 6th Nordic Workshop on Secure IT Systems(NordSec 2001), pp. 100-107 (2001).

[8] A. Hari, S. Suri and G. Parulkar, “Detecting and resolving packet filter conflicts,” Proc. of IEEE INFOCOM 2000, pp. 1203-1212 (2000).

[9] A. X. Liu and M.G. Gouda, “Complete Redundancy Detection in Firewalls,” Proc. of 19th Annual IFIP Conference on Data and Applications Security, pp. 196-209 (2005).
[10] M. Degermark, A. Brodnik, S. Carlssson, and S. Pink, ”Small forwarding tables for fast routing lookups,” Prc. SIGCOMM 97, pp.3-14 (1997).
[11] T. V. Lakshman and D. Stiliadis, “High-Speed Policy-based Packet Forwarding Using Efficient Multi-dimensional Range Matching,” Proc. SIGCOMM 98, pp. 203-214 (1998).
[12] V. Srinivasan, G. Varghese, S. Suri and M.Waldvogel, “Fast and Scalable Layer Four Switching,” Proc. SIGCOMM 98, pp.191-202 (1998).
[13] P. Gupta and N. McKeown. Packet classification on multiple fields. SIGCOMM 99, pp. 147–160 (1999).
[14] 高橋，“フィルタ弁別関数の部分計算に基づく実時間パケット分類器，第 2 回インターネットテクノロジーワークショップ論文集，pp.190-197 (1999).
[15] N. Takahashi, "A Systolic Sieve Array for Real-time Packet Classification," IPSJ Journal, Vol.42, No.2, pp.146-166 (2001).
[16] Y. Yin, R.S.Bhuvaneswaran, Y. Katayama and N. Takahashi: "Implementation of Packet Filter Configurations anomaly Detection System with SIERRA", Proc. of 7th International Conference on Information and communications Security (ICICS2005), LNCS Vol.3783, pp.467-480, (2005).
[17] Y. Yin, R. S. Bhuvaneswaran, Y. Katayama, N. Takahashi, "Inferring the Impact of Firewall Policy Changes by Analyzing Spatial Relations between Packet Filters", Proc. of 2006 IEEE Int. Conf. on Communication Technology(ICCT2006), ISBN: 1-4244-0800-8 Volume I, P203-208 (2006).

〈発　表　資　料〉

題　　名	掲載誌・学会名等	発表年月
Detection of Conflicts Caused by a Combination of Filters based on Spatial Relationships	情報処理学会論文誌	2008 年 9 月（掲載予定）
Implementation of Filter Reverse Search System based on Spatial Relationships of Filters	Journal of Convergence Information Technology	2008 年（掲載予定）
ステートフルファイアウォールを有する LAN のあめのフィルタ逆引きシステムの実現	電子情報通信学会技術報告 IEICE-IN2007-153	2008 年 2 月