5ZA-06

トラフィック傾向に基づいた パケットフィルタ型ファイアウォールルール群の再構築

若林　慶 [1,†1]　　小谷　大祐 [1,†2]　岡部　寿男 [1,†3]
京都大学 [1]

1. はじめに

ネットワークの高度化，IoT の発展によりネットワークに流通するトラフィクデータの量は爆発的に増大している[1]. 同時に悪意あるトラフィックの数も増加しており，これらのパケットを破棄するための，効率的なパケットフィルタリングは必要不可欠となっている.

パケットフィルタ型ファイアウォールのうちルータなど広く一般的に実装されている Access Control List(ACL)は，予めネットワーク管理者が設定したルールのリストに基づいて，パケットの通過・拒否の判断を行う.このルールリストは人手で作成されることが多く，ACL のルールリストの質はネットワーク管理者のスキルに依存する. その結果，マッチするパケットが存在しない非効率なルールや，一つにまとめることが可能な冗長なルールが存在し，効率的にルールの検索が行われていないことも多い.

効率的なルールの探索という課題には，これまでのところ多くの研究がなされている. 有向無閉路グラフのソートを用いた並び替え[2] や，トラフィックが多いルール順にルールリストを動的に並び替える手法[3][4] などが挙げられる. 一方で非効率なルールや冗長なルールを減らすことを目的とした研究は見受けられない.

本稿では，効率的なルールの検索という点ではなく，非効率なルールや冗長なルールを減らすことに主眼を置き，ACL によって許可・拒否されたトラフィックのデータからトラフィック傾向に基づいた効率的なルールリストを再構築する手法を提案する.

具体的には，ある ACL のルールリストと ACL 適応前のトラフィックデータから, ACL のエミュレータを利用して，トラフィックデータに含まれる各パケットに通過・拒否のラベルを付与する. これを入力データとして，分類できるパケット数が多いルールが上位に来る決定木を構築する. 出力された決定木は探索を用いてリストに変更することで，新たな ACL のルールリストを構築する.

2. Access Control List

本節では，本研究のパケットフィルタリングに用いる Access Control List(ACL)に関して説明する. はじめに一般的な ACL について言及し，続いて本研究で仮定する ACL について述べる.

2.1 一般的な ACL

通信アクセスを制御するため，各パケットに対して設定されたルールのリストに基づいて規定された操作を行うパケットフィルタリング手法の一つである. 主にルーターで設定され，通過するパケットを許可(Accept)するか拒否(Deny)するかを決定する. 一般的に，各ルールはヘッダの値の組 (ワイルドカードを含む) からなるルールにマッチするパケットの条件と，マッチしたパケットに適用する処理 (Accept, Deny など) からなる. 設定されたルールとマッチするかを上から順に検索する線形探索モデルの ACL が使用される. ルールの書き方にはいくつかフォーマットが存在し，Accept・Deny 以外の処理が行えるものもある.

2.2 本研究で仮定する ACL

本研究では，上記の ACL 全てを対象とはせず，以下の ACL を仮定する.

- ルールリストは線形探索される
- パケットに対する処理は Accept か Deny のみ
- ワイルドカードによるマッチングを含む
- ルールのマッチング対象フィールドは，送信元 IP アドレス，宛先 IP アドレス，送信元ポート，宛先ポート，プロトコル番号の 5 つとする(下図参照).

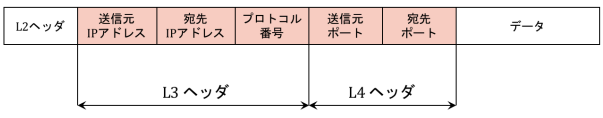

図 1. マッチング対象

2.3 ACL の問題点

ACL の問題点はいくつか存在するが，本研究で着眼する

Reconstraction of Packet Filter Rules Based on Traffic.
†1 KEI WAKABAYASHI
†2 DAISUKE KOTANI
†3 YASUO OKABE
1 Kyoto University.

問題点は，人手で作成されることによる非効率なルールや冗長なルールの存在である．

非効率なルールの定義は，トラフィックが考慮されず，処理できるパケットの少ないルールである．
例を挙げると，以下は 10.1.1.1 の Web サーバが使われていないトラフィック傾向がある場合，非効率である．

access-list 100 permit tcp host 10.1.1.1 eq 80 any (10.1.1.1 の Web サーバから全ての宛先への通信を許可)

冗長なルールとは他のルールで判定可能なパケットのみを対象とするルールである．例を挙げると，以下は一つ目のルールで accept されるパケットを二つ目のルールで同様に Accept 判定してしまっているため冗長である．

access-list 100 permit tcp any host 10.1.1.1 eq 80 (10.1.1.1 への全ての HTTP アクセスを許可)
access-list 100 permit tcp host 172.16.1.0 host 10.1.1.1 eq 80 (172.16.1.0 から 10.1.1.1 への HTTP アクセスを許可)

3. ルールリストの再構築

本節では，本研究で提案するルールリストの再構築手法について説明する．本研究の目的は，非効率あるいは冗長なルールの少ない，簡潔なルールリストを構築することである．

本手法では，大きく分けて二つのアルゴリズムを用いる．一つ目は，再構築対象の ACL とトラフィックデータから，分類できるトラフィックの多いルールが上位に来る決定木を構築するアルゴリズムである(図 2 参照)．二つ目は，先のアルゴリズムで出力された決定木をフラットなリストに変更し，ACL のルールセットに変換するアルゴリズムである．

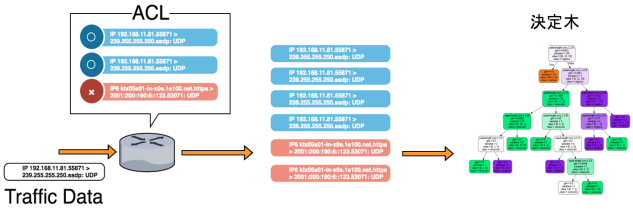

図 2. ACL とトラフィックデータから決定木を構築

3.1 データセットを決定木に変換するアルゴリズム

はじめに，ACL と ACL 適応前のトラフィックデータから，教師ラベル(Accept・Deny)が付与されたデータセットを用意する．具体的には，実際の ACL ではなく，ACL のルールリストから通過・拒否の判定を行い，入力パケットに Accept・Deny ラベルをつけるエミュレータに上記トラフィックデータを入力し，データセットを作成する．

ラベル付きデータセットの取得後，当該データセットから決定木を構築する．決定木の作成に当たっては， 2.2 節で研究対象として絞った各フィールドを説明変数として設定し，Accept・Deny に分類する分類木を構築する．考えなしに決定木を構築するだけでは，木のサイズが膨大になり，簡潔なルールセットとは言えない．ルールを簡潔にするためには枝の剪定(pruning)が必要不可欠であるが，過多な剪定を行うと判定の偽陽性・偽陰性が上がってしまうため，調整が必要となる．

3.2 決定木をルールリストに変換するアルゴリズム

上述のアルゴリズムで作成された決定木を ACL のルールリストの形に変換する．決定木を構築した際に，ACL の各フィールドを説明変数に設定したため，Accept・Deny どちらかの葉に到達するまで根から探索することで一つのルールを求めることができる．同一フィールドに対するノードが複数個存在する場合は，これを分割あるいはワイルドカードに変換することで対応する．

以上の流れを根から全ての葉まで行うことで，ルールリストが作成できる．この時，入力であるトラフィックデータの中で該当する葉まで到達したトラフィックの数を保存しておき，その値に応じてルールのリストを並び替えることでトラフィックの傾向に基づいたルールの設定が可能となる．

4. まとめ

本研究では，非効率や冗長な Access Control List のルールリストをトラフィック傾向に依存した簡潔なルールリストに書き換えることを目的とし，決定木を用いた ACL の再構築手法を提案した．具体的には，対象の ACL とトラフィックデータをデータセットとして用い，Accept・Deny されたトラフィックのデータから決定木を構築し，ルールリストへの変換することで ACL の再構築を試みた．

本研究の今後の課題として，再構築したルールリストの偽陽性・偽陰性の低減が挙げられる．現状再構築されたルールリストでは，偽陽性・偽陰性の高いルールリストが生成されてしまい，ネットワーク管理者が設計時に想定していた悪意のあるトラフィックを全て遮断することはできない．そのため，セキュリティ確保の観点から出力されたルールリストを加工せずに利用することは難しい．そこで，決定木構築のアルゴリズムの修正やトラフィックデータの拡充などにより，偽陽性･偽陰性を低下させる必要がある．

参考文献

[1] 総務省『平成 30 年版情報通信白書』，URL：http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h30/pdf/30honpen.pdf
[2] Ashish Tapdiya, Errin W. Fulp, “Towards Optimal Firewall Rule Ordering Utilizing Directed Acyclical Graphs,” icccn, pp.1-6, 2009. Proceedings of 18th International Conference on Computer Communications and Networks, (2009).
[3] Hazem Hamed and Ehab Al-Shaer. On autonomic optimization of firewall policy organization. Jour- nal of High Speed Networks, Vol. 15.3, pp. 209–227, 2006.
[4] Ken Tanaka, Kenji Mikawa, and Kouhei Takeyama. Optimization of packet filter with maintenance of rule dependencies. IEICE Communications Ex- press, Vol. 2013, pp. 80–85, 2.2.