L-019

アドレスマスクを用いたアクセスリスト集約化手法の提案
Proposal of Access List Aggregation Method with Address Mask

横山 直彦 †
Naohiko Yokoyama

今泉　貴史 ‡†
Takashi Imaizumi

1. はじめに

近年、インターネット端末のセキュリティに関する被害は増加している。そのセキュリティに関する被害の中で、脅威となっているものの 1 つに DoS 攻撃がある。DoS 攻撃はサービス不能攻撃とも呼ばれ、大量のトラフィックを発生させたり、システムの脆弱性を悪用することで、サーバの提供するサービスを提供不能にする攻撃である。またこの DoS 攻撃を複数の拠点から同時に行う DDoS 攻撃による被害も発生している。

DDoS 攻撃の対策としては、既に攻撃者であるとわかっているホストの IP アドレスを収集しておき、それらをファイアウォールのアクセスリストに設定し、パケットフィルタリングでその攻撃者から送られてくるパケットを遮断するといった方法が考えられる。DDoS 攻撃では、同じホストからの大量のパケットを受信することになる。そのため、既知の攻撃者の IP アドレスをアクセスリストに設定するだけで、大量のパケットを処理できるようになるこの方法は、単純であるが有効な方法である。しかし、攻撃者の数が膨大であるとパケットフィルタリングの処理が追いつかないという問題が発生してしまう。

そこで本研究では、1 つのアドレスで複数のアドレスを表すことができるアドレスマスクを用い、効率のよいアクセスリストを作成することで、パケットフィルタリングの処理を軽減し、DDoS 攻撃への耐性を高める手法を提案する。

2. パケットフィルタリング

パケットフィルタリングは、ネットワークから流れてくるパケットを検査して、パケットを通過させるかどうかを判断する機能である。また、このパケットフィルタリングを行う際に、ネットワークから流れてくるパケットの許可・拒否を決定する条件文のことをアクセスリストという。

アクセスリストでパケットの許可・拒否を判断するルールとして設定できるものは、送信元 IP アドレス、送信先 IP アドレス、プロトコル番号、送信元ポート番号、送信先ポート番号などである。またルールには順番があり、先頭のルールから順に条件に合っているか照合していく。

2.1. ワイルドカードを指定するマスク

アクセスリストの設定では、ワイルドカードを指定するマスクを使用することができる (以降、ワイルドカードマスクと呼ぶ)。

ワイルドカードマスクは、マスク部分を ‘1’、それ以外を ‘0’ で表記する。ワイルドカードマスクを用いる利点として挙げられるのが、マスクのビットが最下位ビットから連続している必要がないことである。ワイルドカードマスクでは、自由にマスクビットを定義することができるため、柔軟な設定が可能となる。

3. 提案手法

本章では、提案手法について説明していく。3.1 節で概要を述べ、その後提案手法の具体的な手法ついて説明していく。

3.1. 概要

提案手法では、パケットを受信した際に行うパケットフィルタリングの処理を、より効率的に行うことによって、パケット 1 つ 1 つの処理にかかる負荷を軽減し、単位時間あたりにサーバが処理できるパケット数を増やすことで、DDoS 攻撃に対する耐性を高める。

この DDoS 攻撃では、ボットに感染したコンピュータを踏み台として利用することがある。最近ではボットに感染したホストやスパムメールを送信してくるホストの IP アドレスがインターネットサイトで公開されているところもあり [7]、このようなインターネットサイトから攻撃者の IP アドレスを収集できる。本研究ではこの攻撃者の IP アドレスのリストから、効率的なフィルタリングが可能なアクセスリストを生成する。提案手法では、インターネットサイトから収集した攻撃者の IP アドレスのリストに、ブール関数を簡単化するために用いられる、クワイン・マクラスキー法を適用することで、1 ビットのみ異なっている攻撃者の IP アドレスの組み合わせを見つけ、それらを集約していく。この集約を繰り返し導出したマスクビットを含む IP アドレスは、2.1 節で説明したワイルドカードマスクを用いてアクセスリストを設定することで、一度に複数の IP アドレスをフィルタリングできる。複数の IP アドレスをフィルタリングできるアクセスリストを用いることによって、1 つ 1 つの IP アドレスをアクセスリストとして設定するより、より効率的なフィルタリングが実現できる。

†千葉大学融合科学研究科
‡千葉大学総合メディア基盤センター

3.2. 提案手法の手順

提案手法では、アドレスを集約する手法としてクワイン・マクラスキー法を用いる。クワイン・マクラスキー法は W・J・クワインが提案し、E・J・マクラスキーが発展させたもので、ブール関数を簡単化するための方法であり、同様の目的で使用されるカルノー図よりもコンピュータによる自動化に適している。

提案手法は主に以下の 2 つの段階からなる。

1. アドレスを集約する。
2. 重複しているアドレスを調べ、不要なアドレスを除去する。

ここからそれぞれの段階について、処理の例を通して説明していく。

3.2.1. アドレスを集約する

アドレスを集約する過程を例を通して説明していく。今回の例では、表記を簡単にするため 4 ビットのアドレスを用いた。アドレスのリストを値が ‘1’ であるビットの数ごとにまとめ、列挙したものが図 1 であるとする。

1の数	ビット列表現
0	0000
1	0001
	0010
	0100
2	0110
	1100
3	1011

図 1: 値が ‘1’ であるビットの数ごとに表に列挙したアドレスのリスト

次に 1 ビットのみが異なっている (ハミング距離が 1 となる) アドレスの組を見つけて、その 2 つを集約する。ここでは、集約したビット (マスクビット) は ‘*’ で表現する。これを値が ‘1’ であるビットの数の差が 1 であるすべてのアドレスの組み合わせについて確かめる。一通り確かめた後、これ以上まとめることができなかったアドレスには ‘E’ でしるしをつける。以上の動作が終了したら、できたアドレスを再び ‘1’ の数ごとにまとめ、同様に列挙する。今回の例では、1 回目の比較が終了した後に、値が ‘1’ であるビットの数ごとに列挙したものが図 2 となる。

この操作を再帰的に適用していき、新しいアドレスが作成できなくなった時点で’E’ がついているアドレスすべてがこの段階での最終的な出力となる。今回の例では 2 回目の比較 (図 3) で終了している。

すべての比較が終了し、これ以上まとめることができなかったアドレスに ‘E’ でしるしをつけたものが図 4 である。

1回目の比較	
1の数	ビット列表現
0	000*
	00*0
	0*00
1	0*10
	*100

図 2: 1 回目の比較で作成できたアドレス

2回目の比較	
1の数	ビット列表現
0	0**0

図 3: 2 回目の比較で作成できたアドレス

今回の例では最終的な出力は、1011、000*、00*0、*100、0**0、の 5 つとなった。元のアドレスと区別するため、以後、これらの本節で説明した手順で導出したアドレスを、集約アドレスと呼ぶこととする。

3.2.2. 不要なアドレスの除去

3.2.1 節で導出した最終的な出力は、1011、000*、00*0、*100、0**0、の 5 つであった。この状態でもアドレスを集約して、アクセスリストのルール数を減らすことには成功しているが、この 5 つの集約アドレスはまだ減らすことができる。5 つの集約アドレスに含まれているアドレスの総数を数えてまとめると、図 5 のようになった。

マスクビット数が少ない集約アドレスから順に、その集約アドレスに含まれるアドレスについて、図 5 から総数を調べ、すべてのアドレスの総数が 2 以上である集約アドレスを探す。ここでマスクビット数が少ない集約アドレスから順に調べているのは、できるだけ集約することができたビット数が多いアドレス、つまり 1 つの集約アドレスでより多くのアドレスを調べられる集約アドレスを優先的に残しておくためである。

すると今回の例では、00*0 の集約アドレスに含まれるアドレスすべてが図 5 において、総数が 2 以上になっている。アドレスの総数がすべて 2 以上になっているということは、その集約アドレスに含まれるすべてのアドレスが他の集約アドレスと重複している、つまりその集約アドレスがなくても他の集約アドレスですべ

1の数	ビット列表現
0	0000
1	0001
	0010
	0100
2	0110
	1100
3	1011 E

1回目の比較	
1の数	ビット列表現
0	000* E
	00*0 E
	0*00
1	0*10
	*100 E

2回目の比較	
1の数	ビット列表現
0	0**0 E

図 4: 比較後の結果

アドレス	総数
0000	3
0001	1
0010	2
0100	2
0110	1
1100	1
1011	1

図 5: アドレスリストとそのアドレスの総数

てのアドレスを検査できるということなので、この例では、00*0 の集約アドレスは不要であると判断できる。このような集約アドレスを発見したら、図 5 における集約アドレスに含まれるアドレスの総数をそれぞれ 1 減ずる。

これをすべての集約アドレスについて調べることで、3.2.1 節で導出した集約アドレスの数を、さらに減らすことができ、より効率的なアクセスリストが生成可能となる。

3.3. アクセスリストの生成

アクセスリストに用いる集約アドレスを決定したら、アクセスリストを設定していく。アクセスリストを設定する際は、集約したビットをワイルドカードマスクとして設定する。

実際にアクセスリストを設定する基本コマンドは、cisco の標準アクセスリストの設定 [2] では、
access-list **番号** { **可否** } { **アドレス** }
となっている。**番号**には設定するアクセスリストの番号を入れる。**可否**はその IP アドレスを許可するか、拒否するかを指定する。続く**アドレス**の部分で、送信元 IP アドレスとそのワイルドカードマスクを指定する。

4. 実験

4.1. 実験 1

実際にネットで公開されている IP アドレスに提案手法を適用した場合、どのような結果が得られるかを実験した。

実験 1 では、入力を IP アドレスのリスト、出力を提案手法を適用して導出した集約アドレスとするプログラムを作成し、7 つのインターネットサイト [1][7][8][9][10][11][12] で収集した 10429 個の IP アドレスを用いた。実験 1 の手順は以下の通りである。

1. 10429 個の IP アドレスの中から一定数の IP アドレスをランダムに選択する。
2. 選択したアドレス群に対し提案手法を適用し、何ビット集約できたアドレスが何個生成できたかを求める。
3. 使用したアドレス数から生成できたアドレスの和を引いて削減できたルール数を求める。
4. 削減できたルール数を使用した IP アドレス数で割ってアクセスリストの削減率を求める。
5. 1〜4 の処理を 10 回繰り返し、それぞれの平均を求める。

この実験を IP アドレス数を 100、1000、2000、3000、4000、5000、6000、7000、8000、9000、10000 の 11 パターンで行い、その結果をまとめたものが表 1 と表 2 である。

また、横軸に使用したアドレス数、縦軸に削減率を設定し、実験結果をグラフにしたものが図 6 である。

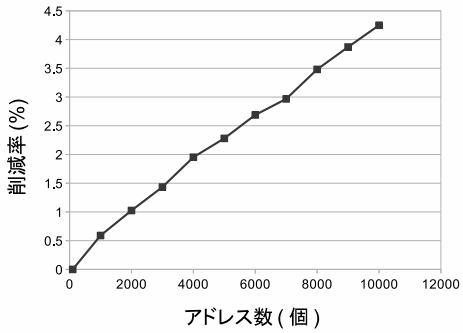

図 6: 実験 1 の削減率のグラフ

そして、最終的にアクセスリストとして設定することになるルールの数をグラフにしたものが図 7 である。

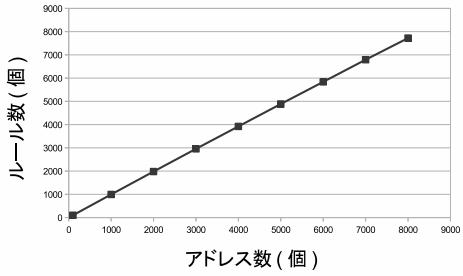

図 7: 実験 1 のアドレス数のグラフ

今回の実験 1 では 10492 個の IP アドレスを収集できたため、使用する IP アドレス数によってどの程度集約できるビット数や主項の数が変化するのかを確認するために、1000 個刻みでデータを算出した。また、使用した IP アドレスが 100 個の場合のデータも算出した。結果としては、2 ビット以上集約できたのは使用した

表 1: 実験 1 の結果 1

使用したアドレス数	100	1000	2000	3000	4000	5000
集約できなかったアドレス数	100.0	987.3	1954.7	2904.9	3823.3	4742.6
1 ビット集約したアドレス数	0	6.8	24.4	51.2	96.2	137.9
2 ビット集約したアドレス数	0	0	0.4	0.9	2.4	5.5
3 ビット集約したアドレス数	0	0	0	0	0	0
4~32 ビット集約したアドレス数	0	0	0	0	0	0
削減できたルール数	0	5.9	20.5	43.0	78.1	114.0
削減率 (%)	0	0.59	1.025	1.433	1.953	2.28

表 2: 実験 1 の結果 2

使用したアドレス数	6000	7000	8000	9000	10000
集約できなかったアドレス数	5645.5	6540.1	7408.3	8290.0	9150.3
1 ビット集約したアドレス数	180.6	231.1	280.6	315.7	363.3
2 ビット集約したアドレス数	12.6	28.4	31.8	43.9	56.0
3 ビット集約したアドレス数	0	0.3	0.9	2.4	5.5
4~32 ビット集約したアドレス数	0	0	0	0	0
削減できたルール数	161.3	207.7	278.4	348.0	424.9
削減率 (%)	2.688	2.967	3.48	3.87	4.249

IP アドレス数が 2000 個以上になったとき、3 ビット集約できたのは使用した IP アドレスが 7000 個になったときで、4 ビット以上集約することはできなかった。

図 6 のグラフを見ると、使用する IP アドレス数が増加すると、アクセスリストの削減率も増加している。しかし、IP アドレスを 10000 個使用した場合でもアクセスリストの削減率は 4.441%という低い値だった。また、図 7 のグラフは右肩上がりのほぼ直線となった。このことから、使用するアドレスの数を増加させると、削減できるアドレスの数も増加するが、大幅に削減できるアドレスの数が増加することもないことがわかる。

これら原因としては、アドレスの総数が多かったことが考えられる。実験で使用した IPv4 アドレスは 32 ビットのアドレスであるため、IPv4 アドレスは全部で 2^{32} 個、つまり約 42 億個ある。今回使用した IP アドレスは全体の 0.00023%と全体の数から考えると、かなり低い割合になる。全体のアドレス数に対する実験で使用するアドレスの割合が低いほど、1 ビットのみ異なっているアドレスの組ができづらくなってしまうため、今回の実験では、あまり多くの集約が行えず、削減率が上がらなかったのではないかと考えられる。

よって提案手法の効果をより高めるには、アドレスの範囲をもっと狭くし、アドレスの全体数を減らすことで、アドレスの全体数に対する使用するアドレスの割合を高くする必要があると考えられる。

4.2. 実験 2

実験 1 の結果より、アドレスの全体数に対する使用するアドレスの割合を高くすれば、提案手法の効果をより高めることができるのではないかと考えられるため、アドレスの範囲を実験 1 の 32 ビットから 16 ビットに狭くし、再度実験を行った。

実験 2 では一部実験 1 とは異なり、16 ビットのアドレスを乱数により一定数生成し、実験を行った。実験の手順は以下の通りである。

1. 16 ビットのアドレスを一定数ランダムに生成する。
2. 選択した 16 ビットアドレスに提案手法を適用し、何ビット集約できたアドレスが何個生成できたかを求める。
3. 使用したアドレス数から生成できたアドレスの和を引いて削減できたアクセスリスト数を求める。
4. 削減できたアクセスリスト数を使用したアドレス数で割ってアクセスリストの削減率を求める。
5. 1〜4 の処理を 10 回繰り返し、それぞれの平均を求める。

この実験をアドレス数を全体の 10%、20%、30%、40%、50%、60%、70%、80%、90%の 9 パターンで行い、その結果をまとめたものが表 3 と表 4 である。

また、実験 1 と同様に実験結果をグラフにしたものが図 8 と図 9 である。

表 3: 実験 2 の結果 1

使用したアドレスの全体に対する割合 (%)	10	20	30	40	50
集約できなかったアドレス数	1235.0	367.6	59.0	6.5	0.8
1 ビット集約したアドレス数	2815.9	4349.2	2557.4	695.4	85.9
2 ビット集約したアドレス数	173.0	1974.9	5381.8	6679.1	3724.9
3 ビット集約したアドレス数	0	12.8	247.3	1785.9	5525.1
4 ビット集約したアドレス数	0	0	0.5	3.1	89.6
5 ビット集約したアドレス数	0	0	0	0	0
6 ビット集約したアドレス数	0	0	0	0	0
7 ビット集約したアドレス数	0	0	0	0	0
8~32 ビット集約したアドレス数	0	0	0	0	0
削減できたルール数	2329.1	6402.5	11414.0	17044.0	23340.7
削減率 (%)	17.652	48.848	58.078	65.019	71.233

表 4: 実験 2 の結果 2

使用したアドレスの全体に対する割合 (%)	60	70	80	90
集約できなかったアドレス数	0	0	0	0
1 ビット集約したアドレス数	4.7	0.1	0	0
2 ビット集約したアドレス数	667.4	24.3	0.1	0
3 ビット集約したアドレス数	7324.3	2699.6	45.9	0
4 ビット集約したアドレス数	1236.0	5558.4	4319.8	5.2
5 ビット集約したアドレス数	0.4	65.2	2636.7	1969.8
6 ビット集約したアドレス数	0	0	3.8	2574.7
7 ビット集約したアドレス数	0	0	0	4.8
8~32 ビット集約したアドレス数	0	0	0	0
削減できたルール数	30088.2	37526.4	45421.7	54426.5
削減率 (%)	76.519	81.803	86.636	92.264

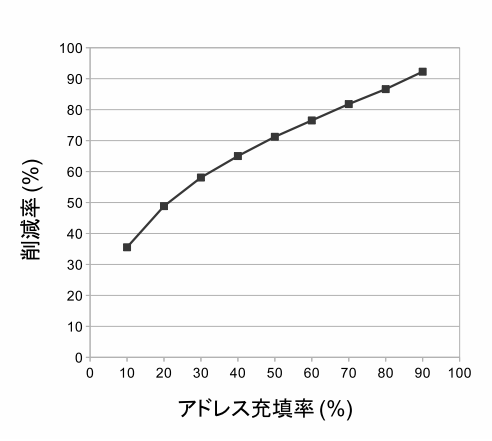

図 8: 実験 2 の削減率のグラフ

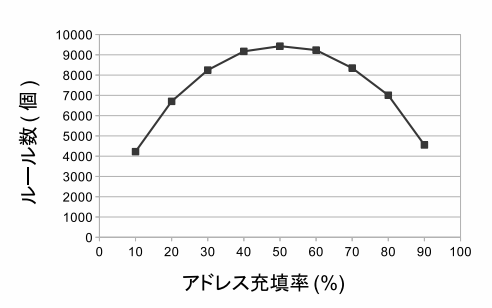

図 9: 実験 2 のアドレス数のグラフ

図 6 と図 8 から、実験 1 で 10000 個のアドレス (全体の約 0.00023%) を使用した場合の削減率と実験 2 で 13107 個のアドレス (全体の 20%) を使用した場合の削減率を比較すると、実験 2 の削減率が実験 1 の削減率を大きく上回っていることがわかる。このことから、ほぼ同じ数のアドレスを使用しても、アドレス全体の総数によって削減率は大きく変化するということがわかった。

今回の実験では、実験 1 は IPv4 アドレスに提案手法を適用した場合、実験 2 は 16 ビットのアドレス、例えばクラス B アドレスなどに提案手法を適用した場合を想定して実験を行った。実験 1 では削減率はあまり大きな値にならなかったため、IPv4 アドレス全体に提案手法を適用しても効果は小さいと考えられるが、実験 2 では削減率は十分大きな値になったため、アドレスの範囲を狭くできるような場合、例えば、同じネットワーク内に属しているアドレス群などに提案手法を適用する場合などに、提案手法の効果はより高まると考えられる。

5. 考察

5.1.IPv6 環境下における提案手法

提案手法は IPv4 環境下でのアドレスを用いたアクセスリストの集約化手法であり、IPv6 環境下でのことを考慮していない。IPv4 と IPv6 の環境で提案手法に関係してくる要素はアドレスのビット数である。アドレスのビット数は、IPv4 では 32 ビット、IPv6 では 128 ビットとなり、アドレスの範囲が IPv6 では大幅に広くなる。提案手法では、ハミング距離が 1 である IP アドレスの組み合わせを見つける、という方法を基本としているため、アドレスの範囲が大幅に広くなれば、ハミング距離が 1 である組み合わせも見つけづらくなると思われる。そのため、IPv6 環境下で提案手法をそのまま適用しても、ほとんど効果がないと考えられる。

5.2. 順番に依存するアクセスリストの特性を利用した集約

3.2 節ではクワイン・マクラスキー法を用いてアドレスリストを集約する手法を提案したが、提案手法を用いて生成した集約アドレスを、アクセスリストの順番の特性を利用し、さらに集約する方法が考えられる。

2 項でも述べたように、アクセスリストは対象となるアドレスに対して、そのパケットの通過の“ 許可 ”と“ 拒否 ”を設定することができる。また、アクセスリストには順番があり、先頭のアクセスリストから順に適用されていく。これらの特性を利用すればアドレスをさらに集約することが可能となる。

図 10 に簡単な例を挙げると、10**、1100、1101、1110 という 4 つの“ 拒否 ”のアクセスリストを、1111 を“ 許可 ”した後で、1***を“ 拒否 ”するアクセスリストに集約化することで、提案手法では集約しきれなかったものをさらに集約することができる。

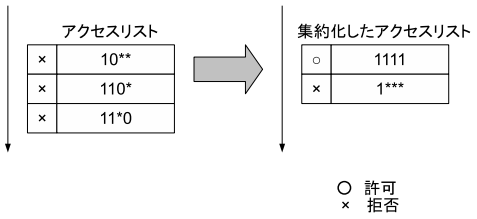

図 10: 順番の特性を利用して集約する例

5.3. 除去しきれなかった重複アドレス

4 章の実験では、アクセスを拒否するアドレスのリストを集約する、という設定で行った。また 4.2 節の実験 2 では、アドレス全体の割合を基準に使用するアドレス数を決定した。しかし、4.2 節の実験 2 のように、アドレス全体に対して多くの割合のアドレスを使

用する場合は、違う見方をすることもできる。例えば、10%のアドレスからのアクセスを拒否することと、残り 90%のアドレスからのアクセスを許可することは、結果的には同じである。また逆に 90%のアドレスからのアクセスを拒否することと、残り 10%のアドレスからのアクセスを許可することも結果的には同じである。結果が同じであるならば、よりアクセスリストのルール数を減らすことができるほうを選択するべきであり、それを調べるため、図 9 に図 9 のグラフを $x = 50(\%)$ の軸を基準に反転させたものを追加したグラフを作成した (図 11)。

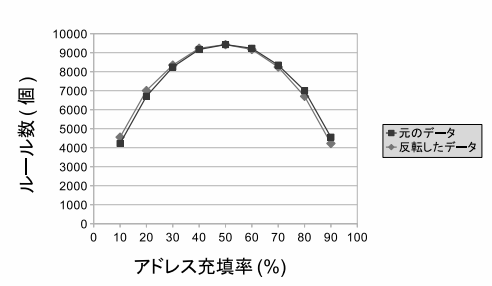

図 11: 反転したデータを追加したグラフ

図 11 を見ると、グラフはほぼ一致しているが、中央から離れるにしたがって少しずつずれていく。しかしこのずれも大きなずれではないため、アクセスを拒否するアドレスのリストとアクセスを許可するアドレスのリストのどちらを使用してもあまり大きな違いはないと考えられる。また、このずれに対処するよりも 5.2 節で述べた方法を利用したほうが、より効率的なアクセスリストを作成できると考えられるため、5.2 節で述べた方法を優先して適用すべきであると思われる。

5.4. 既存研究との比較

パケットフィルタリングの効率化を目的とした既存研究は、パケットフィルタリングのルール、つまりアクセスリストの順序をパケットフィルタリングを行う際の負荷が軽くなるように再配置する手法 [3] や、パケットフィルタリングの統計データを利用して、パケットフィルタリングを行う手法 [4] などがある。

パケットフィルタリングルールの最適配置法と提案手法を比較すると、パケットフィルタリングルールの最適配置法は、既存のアクセスリストを利用してパケットフィルタリングの負荷を減らす手法であるのに対し、提案手法は、新たに効率的なフィルタリングが可能なアクセスリストを生成する手法である。5.2 節でも述べたように、提案手法はアクセスリストの順序を入れ替えることによって、さらに効率的なフィルタリングが可能となるため、パケットフィルタリングルールの最適配置法のような既存手法と組み合わせることで、さらに効率的なフィルタリングができるのではないかと考えられる。

統計データを利用したパケットフィルタリングの手法と提案手法を比較すると、統計データを利用したパケットフィルタリングの手法は、攻撃の意図があると判断され遮断されたパケットに対する、攻撃の意図があると判断されたが、実際は攻撃の意図がなかったパケットの割合がが多少上昇しても、フィルタリングの効率を重視する手法であるのに対し、提案手法は、攻撃の意図がないのに攻撃の意図があると判断され遮断されてしまうパケットを増加させずに、効率的なパケットフィルタリングを行う手法である。そのため、攻撃の意図がないのに攻撃の意図があると判断され遮断されてしまうパケットを増加させないことが、提案手法を用いる大きな利点であると考えられる。

6. おわりに

6.1. まとめ

本研究では、近年増加している DDoS 攻撃に対する耐性を高めるため、クワイン・マクラスキー法を用いたアクセスリスト集約化手法を提案した。

攻撃者の IP アドレスを収集し、それらのアドレスのリストをクワイン・マクラスキー法を利用して集約し、集約したアドレスから不必要なものを除去して、効率のよいアクセスリストを生成した。また、この提案手法が効果がどの程度あるのかを確かめるため、インターネットのサイトで公開されていた IP アドレスを収集し、実験を行ったが、あまり効率のよいアクセスリストが作成できず、DDoS 攻撃への耐性向上にはあまり高い効果が得られないことがわかった。またその原因を考え、アドレス範囲を狭くして実験を行ったところ、アドレス範囲を狭くすれば、提案手法の効果が高まるということもわかった。

6.2. 今後の課題

本研究で、提案手法は 32 ビットのアドレスにはあまり有効ではなく、アドレス範囲をより狭くすることで効果が高まることがわかったため、今後は提案手法の効果が発揮できる活用法、例えば、より限定的なネットワーク内での提案手法の活用などを考えていきたい。また、5.2 節で述べた順番の特性を利用した集約も併用すれば、さらに高い効果が期待できると考えられるため、順番の特性を利用した集約についても研究を進めて生きたいと考えている。

今回はクワイン・マクラスキー法を用いた集約化をアドレスに対して適用したが、提案手法はビット列であれば適用可能であり、ビット列でデータを扱っている様々な場面で提案手法による集約化の活用が期待できるため、それらについても考えていきたい。

参考文献

[1] ”List of Bad IP / Web Scrapers / Content stealers!” http://badip.4rev.net/

[2] ”cisco” http://www.cisco.com/pcgi-bin/imagemap/guestbar .jp?30,6

[3] 嶋 良平, 田中 賢. ”パケットフィルタリングルールの最適配置法” 情報科学技術フォーラム講演論文集 9(4), 253-254, 2010-08-20

[4] 三島 大, 安達 直世, 滝沢 泰久. ”統計的性質を利用した DDoS 攻撃のフィルタリング手法” 電子情報通信学会技術研究報告. IA, インターネットアーキテクチャ 109(137), 7-11, 2009-07-10

[5] 堀川 茂, 今泉貴史，鈴木正人. ”アクセスリスト生成システムの設計とその実装に関する研究” 日本ソフトウェア科学会第 17 回大会 E4-4, 2000

[6] Takashi Imaizumi, Shigeru Horikawa. ”Using an Access-List Compilation System to make Network Reliable” WORLD MULTICONFERENCE ON SYSTEMICS CYBENETICS AND INFORMATICS VOL1 PROCEEDINGS, 1999

[7] ”botscout” http://botscout.com/

[8] ”Stop Forum Spam” http://www.stopforumspam.com/

[9] ”There are the Emerging Threats.net Open rulesets.” http://rules.emergingthreats.net/

[10] ”Autoshun” http://www.autoshun.org/

[11] ”SRI international” http://mtc.sri.com/live_data/attackers/

[12] ”Malware archive, network activity and honeypot statistics” http://www.nothink.org/blacklist.php