Snortルールを入力とするネットワーク侵入検知ハードウェアの開発

若葉陽一† 川中洋祐‡ 永山忍‡ 稲木雅人‡ 若林真一‡

†広島市立大学情報科学部情報工学
‡広島市立大学大学院情報科学研究科

1 はじめに

近年，インターネットの伝送速度の高速化に伴い，ネットワークに接続されたコンピュータに対する侵入や攻撃の脅威が年々増加し，深刻な問題となっている．そのような侵入や攻撃に対処するため，ネットワークにはネットワーク不正侵入検知システム (Network Intrusion Detection System, NIDS) を導入することが一般的である [5]．NIDS はネットワークからの不正な侵入を検知し警告を発するシステムである．NIDS の代表的なソフトウェア実装として Snort[1] が知られている．しかしながら，ソフトウェア実装では処理速度が近年のネットワーク速度に追いつくことができなくなってきている．そのため我々は，Snort のウィルスパターン (以下ではルールとよぶ) で定義されるウィルスの検知をハードウェアで高速実行する NIDS を提案した [2]．Snort のルールには提案システムで直接には処理不可能なパターンの表現形式が含まれる．そこで，本稿では Snort のルールを提案ハードウェアで処理可能なパターンに変換する手法を提案する．

2 準備

2.1 Snort

ネットワーク侵入検知システム (NIDS) とは，監視対象となるネットワークに設置された監視システムであり，当該ネットワークに流れる通信パケットとあらかじめ与えられたウィルスパターンとのストリングマッチングを実行することによって，不正侵入や攻撃がないかチェックを行うシステムである．

NIDS に対する代表的なオープンソフトウェアとして Snort が知られている．Snort においてパターンは正規表現 [3] とそのサブクラス [4] で記述されている．

2.2 提案ハードウェア

提案ハードウェアは正規表現のサブクラスをパターンとし，入力として与えられた文字列に対してストリングマッチングを実行する専用ハードウェアであり，1文字単位のマッチングを行う比較セルが1次元配列状に相互接続されたシストリックアルゴリズムとして実現されている．提案ハードウェアではパターンは実行時に設定可能であるため，ウィルスパターンが頻繁に更新される NIDS においてはパターン更新に対して瞬時に対応可能である，という特徴がある．

2.3 正規表現

入力文字系列を構成する文字の集合を $\Sigma = \{a_0, a_1, \cdots, a_n\}$，$R$ と S をおのおの Σ 上の正規集合 R と S を意味する正規表現とする．なお正規表現を定義する記号，ε(空語)，. (ピリオド)，(，)，^，{，}，\$，\ は Σ に含まれないものとする．提案ハードウェアでは以下の演算子を使用できる．

1. $R|S$ $=$R$\cup$U
2. R^* $= \varepsilon \cup$R$^1 \cup$R$^2 \cup \cdots$
3. $R?$ $=$R$| \varepsilon$
4. $R+$ $=$R$^1 \cup$R$^2 \cup \cdots$
5. . $= a_0 \mid a_1 \mid a_2 \mid \cdots \mid a_n$
6. $[\hat{}\ a_i] = a_0 \mid a_1 \mid a_2 \mid \cdots \mid a_{i-1} \mid a_{i+1} \mid \cdots \mid a_n$

ただし，提案ハードウェアでは同一演算子のネスト構造は禁止している．

次に，Snort では上記に加えて，以下の正規表現の演算子を使用できる．

7. $R\{n\} =$R^n ※ R^2の場合，RR とする．
8. $R\{n, m\} =$R$^n \cup$R$^{n+1} \cup \cdots$ R^m
9. $R\{n, \} =$R$^n \cup$R$^{n+1} \cup \cdots$
10. $[a_0 a_i a_j] = a_0 \mid a_i \mid a_j$
11. $[a_0 - a_n] = a_0 \mid \cdots \mid a_n$

7. 8. 9. を量指定子と呼び，10. 11. をクラスと呼ぶ．また，Snort では演算子の任意のネスト構造が許されている．

3 パターン変換ソフトウェア

前節で述べたように，我々が提案した専用マッチングハードウェアは，パターンとして許される正規表現に制約がある．そこで，提案ハードウェアを NIDS のマッチングエンジンとして利用するために，Snort のルールを提案ハードウェアが処理可能なパターンに変換するソフトウェアを開発した．変換にあたっては，(1) 提案ハードウェアでは許されていない演算子を用いた表

Development of a Network Intrusion Detection System for Snort Rules

Yoichi WAKABA†, Yosuke KAWANAKA‡
Shinobu NAGAYAMA‡ Masato INAGI‡
and Shin'ichi WAKABAYASHI‡
†‡Faculty of Information Sciences, Hiroshima City University
3-4-1, Ozuka-higashi, Asaminami-ku,Hiroshima, 731-3194 Japan

現を，それと等価（マッチする文字列集合が同一）で，かつハードウェアが許している演算子を用いた表現に変換し，かつ，（2）2重以上のネスト構造を無くすことによって提案ハードウェアが処理可能なパターンとする．なお，Snortの正規表現は一般の正規集合を表現できるが，提案ハードウェアのパターン表現は制約された正規集合しか表現できない．しかしながら，Snortのルールで実際に表現されている正規集合は正規集合の部分集合であり，提案ハードウェアの制約された正規表現で表現可能である．

以下にパターン変換ソフトウェアの概要を示す．

3.1 クラス変換手法

入力パターンにクラスを見つけた場合に，クラスの中の文字列を解析し，クラスの中に '-' がある場合とない場合の2つの場合にわけて処理する．'-' がない場合は，'[', ']' の間に含まれる文字をそれぞれ '|' でつないだ文字列を作成し，それに '(', ')' をつけた文字列を生成する．'-' がある場合は，'-' の前に位置する文字と後ろの文字で定義される文字集合のすべての文字を '|' でつなぎ，'(', ')' をつけた文字列を作成する．

3.2 括弧展開と量指定子変換手法

括弧展開と量指定子変換手法は以下の3つの手続きによって行われる．

1. 構文解析手続き
   入力：パターン文字列
   出力：1重括弧のみのパターン文字列
2. 括弧展開手続き
   入力：一番内側の括弧のパターン文字列
   出力：展開したパターン文字列
3. 量指定子手続き
   入力：量指定子とその対象となる括弧
   出力：量指定子を展開したパターン文字列

これらの3つの手法に基づくパターン変換手法について述べる．入力パターンを前から1文字づつ見ていき，左括弧を見つけた場合，構文解析手続きが呼び出され，右括弧が来ると括弧展開手続きを呼び出し，括弧内のパターンを渡し，展開されたパターンが出力され，そのパターンを返す．また構文解析手続き内で括弧を見つけた場合，さらに構文解析手続きが呼び出される．それによって一番内側の括弧から展開される．量指定子手続きは量指定子 '{' がある場合に呼び出され，構文解析手続きによって得られた括弧内のパターンを入力として，量指定子の指定する数だけ括弧内のパターンを繰り返した文字列を返す．これら3つの手続きによって与えられたパターンを1重の括弧のみで量指定子を含まないパターン文字列に変換することが可能である．

変換例 変換ソフトウェアにより *Snort* のルールが順次，以下のように変換される．
Snort ルール：$(a|b)(c[a\text{-}d])d\{4\}$
1.クラス変換：$(a|b)(c(a|b|c|d))d\{4\}$
2.括弧展開：$(a|b)(ca|cb|cc|cd)d\{4\}$
3.量指定子変換：$(a|b)(ca|cb|cc|cd)dddd$

表 1: 変換後のルールが必要とするセル数

	セル数（個）
平均	517
最大	24,339
最小	1

4 実験結果と考察

全387種のSnortのルールパターンを変換をした．表1に変換後のパターンが必要とするセル数を示す．また変換する前は提案ハードウェアが扱えるパターン数が75種類に対して変換を行うことによって282種類のパターンが扱えるようになった．また変換後のルールの長さは平均で元のルールの約57倍となったが，変換後のルールをハードウェアで実装した場合のセル数は，元のルールの長さの高々12倍である．

変換を行うことによりオリジナルのSnortルールの約73%が提案ハードウェアで処理可能になった．他のルールが変換できなかったのは，それらのルールに後方参照 [4] など，**2.3** 節で示した演算子以外の演算子を含んでいるためであるが，それらの演算子は提案ハードウェアを拡張することによって扱うことができることを示しており [2]，それに対応するようにパターン変換ソフトウエアを拡張することで，処理可能になる．また，量指定子を直接扱えるようにハードウェアを拡張した場合，実装に必要なセル数の削減も可能である．

5 まとめ

本稿では，我々が [2] で提案した正規表現のサブクラスに対する専用マッチングハードウェアをネットワーク侵入検知システムのマッチングエンジンとしてSnortルールを扱えるようにするための手法を紹介した．またSnortルールパターンを変換した場合のセル数から考察を行い提案ハードウェアが今後行うべき拡張について述べた．今後の課題として，ソフトウェアの変換によるパターンに対する必要セル数削減があげられる．

本研究の一部は科学研究費補助金基盤研究 (C)（課題番号 20500054）による．

参考文献

[1] http://www.snort.org/

[2] 川中洋祐，若林真一，永山忍: ストリングマッチングマシンのFPGAによる実現とネットワーク侵入検知システムへの応用，情報処理学会 DA シンポジウム論文集, 3B-3, 2008.

[3] J. E. ホップクロフト,J. D. ウルマン: 言語理論とオートマトン, サイエンス社, 1971.

[4] http://www.perldoc.perl.org/perler.html

[5] B. L. Hutchings, R. Franklin, D. Cover: Assisting network intrusion detection with reconfigurable hardware, *Proc. 10th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, pp.111-120, 2002.