様　式　Ｃ－１９、Ｆ－１９、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成 ２７ 年　 ６ 月 １７ 日現在

機関番号：１２５０１
研究種目：基盤研究(C)
研究期間：2012～2014
課題番号：２４５００７４

研究課題名（和文）ＩＰＳに特化した探索アルゴリズムに関する研究

研究課題名（英文）A Pattern Matching Algorithm Suitable for IPSs

研究代表者
今泉　貴史（IMAIZUMI, Takashi）

千葉大学・統合情報センター・教授

研究者番号：７０２４２２８７

交付決定額（研究期間全体）：（直接経費）　　3,700,000円

研究成果の概要（和文）：IPSでの利用に適した文字列探索アルゴリズムと、拡張文字列を用いたパターンマッチアルゴリズムを提案した。IPSでの利用を考える場合、若干の誤りがあっても許容される。そこで、高速化のために誤りを含むことを許容した。この誤りの影響を低減させるために、パターンマッチングアルゴリズムでは逆方向走査を導入した。さらに、誤り率を低減させるため、正規表現を拡張文字列に変換する際のルールについても検討した。

研究成果の概要（英文）：I proposed string matching algorithm and pattern matching algorithm using extended string. I designed to use these algorithms by an IPS. For some slips to be included in a measure result of the IPS, when using these algorithms by an IPS, a few errors are permitted. So to speed algorithm up, I decided to permit some errors. I introduced reverse scan into pattern matching algorithm to make them reduce influence of these errors. I also considered the rule when translating a regular expression to an extended string, to make them reduce the errors which occurs in case of pattern matching.

研究分野：ネットワークセキュリティ

キーワード：ネットワークセキュリティ　文字列アルゴリズム

１．研究開始当初の背景

ネットワークを安全に利用してゆくためには、確実さや容易さを考慮すると、個々の端末において脅威への対処を行うことが望ましい。しかしネットワークに接続される機器が多様化するのに伴い、個別の端末すべてに脅威への対処を実装することが困難となってきている。このような状況下では、古典的な方法ではあるが、ネットワークを内部と外部とに分離し、その接続点に流れるトラフィックを監視することで、内部ネットワークに接続される端末等を脅威から守る手法が有効である。

ネットワークを分離して監視する機器としては、ファイアウォール、侵入検知システム、侵入遮断システム、UTM などがある。ファイアウォールのパケットフィルタリングに見られるような簡単なルールで分離をするだけでは十分ではなく、ファイアウォールもパケットの内部情報を利用してトラフィックを判断するなど、各機器の間の差異はどんどん小さくなってきている。特に、侵入検知・遮断システムにおいては、脅威を認識するためにシグネチャマッチング方式が現在の主流となっている。シグネチャマッチング方式では、あらかじめ脅威の特徴を抽出しておいたシグネチャと、機器を通過するパケットのマッチングを取ることで、侵入を検知する。このマッチングは、基本的には文字列やパターンの探索であり、ネットワークの通信量が増えるに伴って、リアルタイムにすべての処理を行うことが困難になってきている。そのため、比較的深刻度の低い脅威に対するシグネチャを除去して比較するシグネチャの量を減らしたり、すべてのパケットについて処理を行うのをあきらめたりする事態も増えている。現状では、一般家庭に導入されているネットワークは 100Mbps 程度までであるため特に問題にはならないが、1Gbps やより広帯域のネットワークが用いられるようになると、対応は困難になる。データセンター等の大容量の通信が行われる組織においては、シグネチャマッチングの速度がすでに問題になってきている。そのため、文字列やパターンのマッチングアルゴリズムの高速化が急務となっている。

２．研究の目的

本研究の目的は、高速な文字列マッチングアルゴリズムやパターンマッチングアルゴリズムを提案することである。しかし、文字列マッチングアルゴリズムやパターンマッチングアルゴリズムはすでに長く研究されてきており、現状の考え方のままで改良を行っても大きな性能の改善は期待できない。そのため、今回アルゴリズムを適用しようとする IDS 等の特性を活かしてアルゴリズムの高速化を図る。

シグネチャマッチング方式の IDS の特性として、誤検知を完全になくすことはできない。これまで、誤検知を除去するために多くの研究がなされてきた。しかしそのいずれもが完全に満足のいく結果を出せてはいない。多くの研究では、IDS の発する警告を、機械学習手法などを用いながらさらに分類・処理することにより、ユーザが欲している警告かどうかを調べ、実際に警告するかどうかを判断している。この手法では、ユーザが望む動作に対応できるように学習操作が必要になり、誰にでも簡単に使えるとはいえない。また、侵入操作を検知することに加え、さらに警告の選択処理を行う必要があり、リアルタイムの処理には向いておらず、ネットワークの広帯域化に伴う処理能力不足の問題を解決することはできない。また、われわれは誤検知の定義がユーザにより異なる点に着目し、それぞれのユーザが考える脅威を定義することから始めて、その脅威を過不足無く検知できるシステムが誤検知の無いシステムだと定義する方法を提案した。この手法により、複数のシステムを同じ基準で比較することは可能になったが、脅威の定義からそれを検知するシステムを得るには至っていない。

IDS における誤検知の問題は、誰でもが納得するような誤検知が明確に定義されておらず、まったく同じ動作をしても利用者によって誤検知と感じる場合があることなどに起因しており、同じ動作をするシステムで、万人が納得する、誤検知の無いシステムは存在し得ない。つまり、IDS において、誤検知は完全にはなくすことのできないものである。逆に考えると、IDS に関しては、シグネチャマッチングにほんの少しの誤りが含まれていても、システム自身の誤検知率に関する性能の劣化はほとんど無いことになる。

この許容できる誤りを許すことで、シグネチャマッチングの速度に関する性能を大幅に引き上げることができる可能性がある。これが可能になれば、IDS において、単位時間当たりに処理できるパケットのサイズや量を現状に比べて飛躍的に増やすことが可能となり、より高速・大容量のネットワークに対しても現在使われているような安価なシステムが適用可能になると考えられる。

３．研究の方法

まず、固定文字列の探索アルゴリズムの高速化について、アプリケーションドメインを考慮しながら、高速だが誤りを含むアルゴリズムを提案する。さらに、より複雑なパターンマッチングに関して、誤りを許容しながらアルゴリズムを並列化する方向で研究を進める。最終的には、既存の侵入遮断システムに探索アルゴリズムを組み込むことを目指す。

固定文字列の探索アルゴリズムについて

は、既存の文字列探索アルゴリズムについて理解を深め、あいまいさを導入するために、Shift-Or 法（R. A. Baeza-Yates and G. H. Gonnet. A new approach to text searching. Proceedings of the 12th International Conference on Research and Development in Information Retrieval, 168-175. ACM Press, 1989.）をベースにする。Shift-Or 法は、文字列を受理する非決定性有限状態オートマトンの状態をビット列で表現しながらビット演算により状態遷移を表現して、文字列を受理できるかどうかを判定する。基本的なアルゴリズムでは、ビットごとに並列に処理を行っているが、これを文字単位に拡張し、文字の代わりに複数個の文字をまとめた q-gram を用いる方法も提案されている（L. Salmela, et al. Multi-pattern string matching with q-grams. ACM Journal of Experimental Algorithmics, Vol. 11, 2006.）。このアルゴリズムに対し、侵入遮断システムにおけるシグネチャの特性、および、入力としてのパケットペイロードのコンテンツの特性を考慮して、実用上問題がない程度に誤り率が低く、かつ、高速に処理可能なアルゴリズムを提案する。

まず Shift-Or 法を並列に行うよう拡張する。この拡張により、複数の文字列を同時に検査することが可能になる。しかし、アルゴリズムを単純に並列化しただけでは探索結果に誤りが含まれるようになってしまう。これは、文字単位で見たら 1 文字目も 2 文字目も探索文字列に含まれていても、その両方が同じ探索文字列に含まれていない可能性があるために生じる。誤り率を下げるために、ハッシュ法などの計算負荷の小さいアルゴリズムと組み合わせる。2 つの方法が独立であれば、方法 A による誤り率が p、方法 B による誤り率が q の場合、両者を組み合わせた結果でも誤りが生じる可能性は 1 - (1 - p) ×(1 - q)であり、誤り率を抑えることは可能である。また、ハッシュ法の場合にはハッシュ値のビット数を調整することで誤り率（衝突率）を調整できる。

パターンに関しては、一般のパターンを扱うのは適当ではない。汎用のパターンマッチアルゴリズムとしては、正規表現から DFA を作成し、1 文字ずつ入力を読みながら状態遷移機械で処理をする方法が基本となる。この手法はかなり高速に動作する。文字列マッチングの場合には、単一の文字列に対する処理と複数の文字列に対する処理が基本的に異なるため、この部分を用いて高速化を図ることが可能である。しかしパターンマッチの場合には、正規表現の 1 つに「選択」演算があり、基本的に単一パターンの検索と複数パターンの検索に違いはない。そのため、並列化による恩恵も簡単には受けられないため、固定文字列の場合と同じ手法は適用できない。

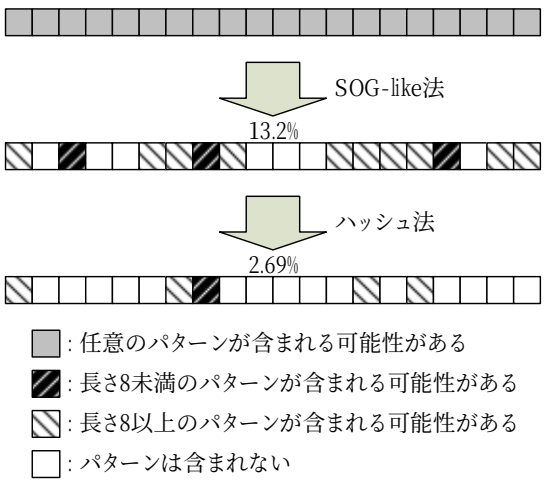
パターンとして一般に用いられているのは正規表現だが、正規表現よりも表現能力の小さなパターンを用いることで、アルゴリズムを高速化できる可能性がある。ただし、単に能力の低いパターンを使うだけではパターンの変換の際に生じる誤りの率が高くなってしまう。そこで、パターンを変換する際にパターンを複数のパターンに分割するなどの処理を加えることで誤りの発生率を下げる工夫も加える。また、IDS での利用を考えたときに、誤りの発生方向が問題となる。IDS としての利用であれば False Positive は許容できるが False Negative は許容できない。逆に、IPS として利用する場合には False Negative は許容できても False Positive は許容できない。これらの点を考慮し、誤りが発生してしまう際には、どちらの方向での誤りを許すのかを指定しておくことで、都合の悪い誤りの発生を抑える。

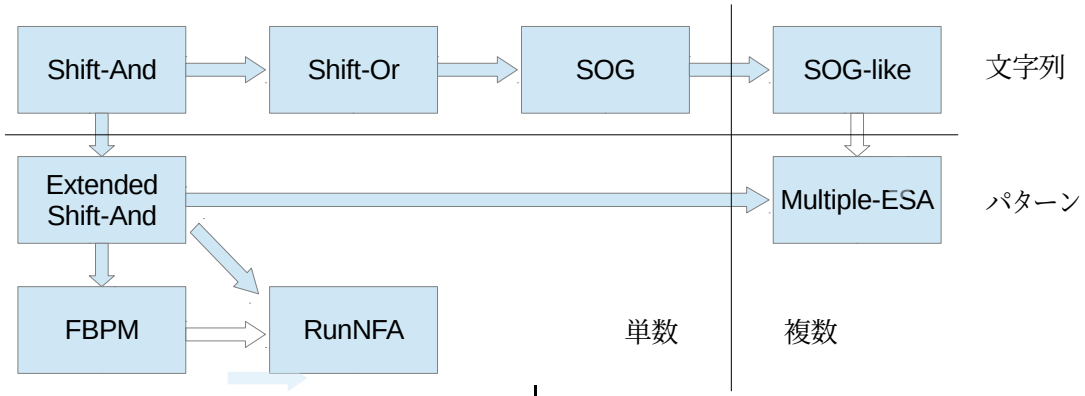
４．研究成果

（１） 固定文字列のマッチングに関して

固定文字列のマッチングに関しては、MI/SH 法を提案した。MI/SH 法は、Shift-Or 法を n-gram に拡張した SOG 法を、さらに扱う文字列を複数に拡張した SOG-like 法と、Rabin-Karp 法に似たハッシュを用いたアルゴリズムを組み合わせたものである。SOG-like 法は、単一の文字列を処理する速度は Shift-Or 法と変わらないが、複数の文字列とのマッチングを 1 度の処理で検査できるため、マッチングスループットを上げることができる。

SOG-like 法では、Shift-Or 法で探索対象の文字列に対して作成する各種テーブルを、文字列の数だけ重ね合わせる。これにより、「abc」と「ade」という文字列を検索するときには、「abe」が出現しても探索成功となる。この段階で探索成功となった文字列に対して、該当部分のハッシュ値を計算しておき、それと比較することでさらに誤りを排除する。ランダムなデータと入力に対してこの手法を適用すると、SOG-like 法の段階で全体の 13.2%を選び出し、さらにハッシュ法を適用することで 2.69%のみが選択できた。この際、





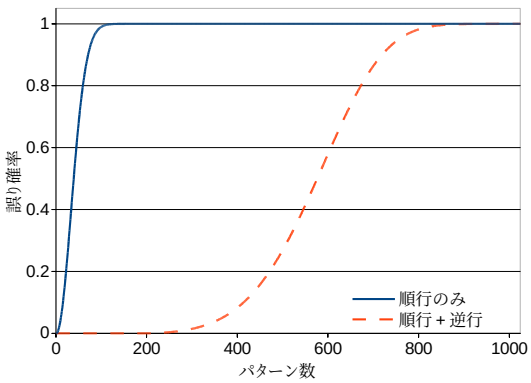
最終的に選ばれたものに False Positive は含まれていない。

実際の動作をデータで確認するだけでなく、MI/SH 法の誤り率の数値的な解析も行った。ある入力がパターンの文字列に一致していないのに一致していると誤って報告される可能性は $1/2^{34}$ となり、IDS の誤り率が数%といわれているのと比較しても、十分に低い値となった。

ここで算出した値は、パターンの長さやパターンをいくつ組み合わせるのかによっても変わってくる。実際にシステムに組み込む際には、誤り率を上げずにマッチング速度を確保できるようなパターン文字列の組み合わせ方が重要となる。しかし IDS の場合にはシグネチャの順番が重要になることがあり、パターンマッチングの都合で順序を入れ替えられないこともある。そのため、単にパターンマッチングアルゴリズムを入れ替えるのではなく、シグネチャ指定言語の拡張から検討する必要がある。

（2） パターンのマッチングに関して

パターンマッチングのために、拡張文字列を用いる Multiple-ESA 法を提案した。拡張文字列は、正規表現と比較すると選択と閉包の演算子が制限されている。正規表現では、任意の正規表現に対して選択や閉包の演算を行うことができるが、拡張文字列ではこれらの演算は文字に対してのみ可能となっている。SOG-like 法でベースとして用いた Shift-Or 法のさらに基となった Shift-And 法には、拡張文字列を用いてパターンの検索ができるように拡張された Extended Shift-And 法がある。SOG-like 法で文字列に関するテーブルを重ねたのと同様に、Multiple-ESA 法では Extended Shift-And 法で用いるテーブルを重ね合わせる。これにより、複数の拡張文字列を一度の処理でマッチングすることが可能となり、マッチングのスループットが上げられる。



MI/SH 法では複数の文字列を組み合わせたことによる誤りの増加はそれほど問題にならなかったが、Multiple-ESA 法では組み合わせるパターン数を少し増やしただけで、誤り率が急激に上昇してしまう。これは、拡張文字列においては選択の演算が許された結果、文字クラスを扱うことが可能となっていて、この文字クラスを重ね合わせることで、本来マッチングしない文字とも容易にマッチするようになってしまうことが原因である。さらに、パターンの中には、一部の文字数だけを問題にすることも多く、その場合パターン中には「任意の 1 文字」が文字数分だけ指定されることになる。これを含むパターンを他のパターンと合わせた場合には、その部分で一気に誤りが増加してしまう。

Multiple-ESA 法では、この誤りの増加を少しでも提言するために、拡張文字列の先頭からの一致を見る順行処理に加え、拡張文字列の末端からの一致を見る逆行処理を行う。これにより、長さの異なる拡張文字列に対しては誤り率を下げることが可能になり、重ね合わせられるパターン数が増加した。しかし、誤り率は IDS の誤り率と比べて高く、このままでは実運用に用いることはできないことがわかった。

（3） 拡張文字列への変換に関して

パターンマッチを導入した際の誤り率の確認は、単に正規表現をそのまま拡張文字列に変換したものであった。正規表現と拡張文字列の能力の違いにより、必ずしも性格に変換することはできないが、それらはすべて誤りが発生するとみなした。しかし、変換法を工夫することで誤りの発生率を抑えることが可能な場合もあり、変換の際にそれらを考慮することに関しても検討を行った。

正規表現から拡張文字列への変換に関して、次の 4 つに分類できる。

1. 拡張文字列でそのまま、もしくは等価に

表現できる
2. 複数の拡張文字列を用いることで等価に表現できる
3. アルゴリズムを改良することで表現できる
4. 誤りを許すことで拡張文字列により表現できる

1.のグループには、正規表現に特有な演算を使っていない場合に加え、選択を使っていても対象がすべて文字の場合なども含まれる。2.は主に文字列に対して選択を用いたパターンが含まれる。選択のそれぞれに対して個別の拡張文字列を作ることで等価に表現できる。ただしこの場合、アルゴリズムで処理するパターン数が増えるため、そちらに起因する誤りの増加は生じる。3.は「行頭」や「行末」に一致する特殊記号を用いるパターンに該当する。これらは純粋な正規表現ではないが、パターンマッチングの際にはよく用いられており、シグネチャにおいても利用されることがある。これは、現状の Multiple-ESA 法が対応していないだけで、アルゴリズムを拡張することで対応可能である。

最後の 4 が一番問題となるグループである。False Positive を許すのであれば、どんなパターンでも「ε（空文字列）」に変換できる。空文字列は任意の文字列に含まれていると考えられるため、どんな入力に対しても一致してしまうが、指定された正規表現により表される集合は間違いなく含むことになる。しかしこのような変換では、マッチング結果に含まれる誤りが膨大になり、使い物にはならない。そのため、許容する誤りにも注意しながら変換を行うことが必要になる。たとえば、

a(bc)*d

というパターン（a の後ろに bc が 0 回以上ついて、さらに後ろに d がつく）に対しては、単に

a[bc]*d

という拡張文字列（a の後ろに b か c が 0 回以上ついて、更に後ろに d がつく）で表現することが可能である。しかしこれを

ad
abc[bc]*d

という 2 つの拡張文字列へと変換することで誤りの発生率は抑えられる。

また、この例では False Positive を許したが、False Positive を許さないという状況では、

ad
abcd

などと変換することで対応できる。内側にある bc を複数回繰り返した拡張文字列を準備することで更に誤り率を下げることが可能だが、パターンが増えることによる誤りの増加もあるためトレードオフとなる。

現状では、正規表現を分類して変換方法を提案しているにとどまっているが、今後の課題としては、正規表現に加え、許容される誤りの方向と誤り率を入力とし、拡張文字列群を生成するトランスレータを構築してゆきたい。

５．主な発表論文等

〔雑誌論文〕（計　１件）

① 「IDS での利用に適したパターンマッチアルゴリズム」柳瀬 葵、今泉 貴史、学術情報処理研究、Vol.17、pp.85－92、2013

〔学会発表〕（計　１件）

① 「muiltiple-ESA 法の IDS への適用可能性に関する検討」岡 大貴、今泉 貴史、第 13 回情報科学技術フォーラム (FIT2014)、6pp、2014 年 9 月 5 日、筑波大学筑波キャンパス（茨城県つくば市天王台 1-1-1）

〔図書〕（計　０件）

〔産業財産権〕
○出願状況（計　０件）

名称：
発明者：
権利者：
種類：
番号：
出願年月日：
国内外の別：

○取得状況（計　０件）

名称：
発明者：
権利者：
種類：
番号：
出願年月日：
取得年月日：
国内外の別：

〔その他〕

６．研究組織

(1)研究代表者

今泉　貴史（IMAIZUMI, Takashi）
千葉大学・統合情報センター・教授
研究者番号： 70242287

(2)研究分担者

(3)連携研究者