

差分フィルタを用いたトラフィック解析とログマイニングによる ネットワークイベントの自動判断

長尾 真宏[†] 北形 元[†] 菅沼 拓夫[†] 白鳥 則郎[†]

ネットワークイベントの自動検出は、安定したシステム運営のためには重要な課題である。しかし、既存手法では false negative の多さとイベント検出時に提供される情報の少なさが問題となり必ずしも実運用に適しているとは言えない。本稿では、実用上有益なイベント情報を提供することを目的として、差分フィルタを用いたシンプルなイベント検出手法で多様なトラフィックを監視することによる false negative の低減と、イベントを検出したトラフィックの種類を利用してログマイニングを行うことで自動的にイベントの内容を判断する仕組みを提案する。

Automatic Event Interpretation based on Traffic Analysis using Difference Filter and Log Mining

Masahiro Nagao[†], Gen Kitagata[†], Takuo Suganuma[†], Norio Shiratori[†]

Automatic detection of the network event is an important issue of secure system management. However, conventional techniques are not necessarily suitable for the real operation because of a lot of false negative and lacking information offered when the event is detected. In this paper, for the purpose of providing useful event information, we propose the scheme that reduces false negative by monitoring various traffic with simple detection method using difference filter and interprets the content of the detected event by log mining using traffic type.

1 はじめに

近年、インターネットの発展とともにネットワークシステムが広く運用されるようになり、情報化社会における基礎的なインフラストラクチャとして重要な役割を果たしている。そのため、ネットワークシステムは常時安定して稼動することが期待されており、適切な運用管理が不可欠となっている。すなわち、システムに何らかの特殊なイベントが発生した場合には、ネットワーク管理者はそのイベントの発生を検出し、イベントの内容を調べた上で問題があれば必要に応じて対処し、その影響を最小限に抑える必要がある。問題となるイベントには、例えばスイッチの故障やケーブルの断線などによるリン

クダウン、DoS 攻撃やポートスキャンなどの不正アクセス、ワームによるネットワークリソースの浪費などさまざまな種類が考えられる。このようなイベントの発見と対処には、高度な経験と知識を要求されるものが多いため、すべての作業をネットワーク管理者が行うには負担が大きく、作業の自動化を支援する仕組みが期待されている。本稿では、ネットワーク管理者の負担を軽減し、ネットワークシステムの安定運用を補助することを目的に、イベントの検出とその内容の判断をリアルタイムで自動的に行う仕組みを提案する。

本稿の構成は、2章でイベント検出に関する関連研究とその問題点について述べ、それを踏まえて3章で提案手法について述べる。4章で現状の試作システムによる実験について述べ、最後に5章でまとめと今後の課題について述べる。

[†] 東北大学電気通信研究所/情報科学研究科 / Research Institute of Electrical Communication/Graduate School of Information Science, Tohoku University

2 関連研究とその問題点

イベント検出に関する研究の代表的なものに IDS (Intrusion Detection System) がある [1]. これは不正アクセスの検出を自動化するためのシステムで、大きく分けて 2 つの方式、すなわち不正検出 (Misuse Detection) と異常検出 (Anomaly Detection) の 2 種類に大別できる. 不正検出は検出したい不正アクセスの特徴をあらかじめ登録しておくことで、登録したものについては高精度で検出することができる. 逆に異常検出では検出したいものを明確に定義するのではなく、正常状態を何らかの方法によって定義することによって、それに当てはまらないものを異常として検出する手法である. 最近では新種のウイルスやワームが頻繁に発生することから、未知の問題にも対応可能な異常検出の手法が注目され、その精度を上げるための研究がさかんに行われている. 例えば [2],[3] では混合正規分布を用いた統計的なトラフィックのモデル化によって、モデルの変化を異常として検出するという手法がとられている. このような統計的な手法は、基幹ネットワークや大規模なサーバなどの統計的に安定した環境では非常によいモデル化が可能である. 他にも、[4],[5] は、トラフィックの系列を離散時間信号の系列ととらえ、デジタル信号処理の手法を用いて異常検出を試みている. 信号系列の低周波成分には長期的なトラフィックの傾向が、高周波成分には短期的なトラフィックの変動が表れることを利用し、その変化の仕方によって正常な変化か、異常な変化かを区別するというものである.

これらの手法は、実際に効果的に異常なイベントを検出できる場面もある. しかし、異常検出の手法は検出すべき異常を明確に定義しないというその特性上、異常の検出率は低く、false negative と呼ばれる異常の見落としが数多く発生する. [6] によると、高精度な手法でもせいぜい 5 割から 7 割の検出率であり、逆に 3 割から 5 割程度の異常なイベントが未検出となっていることがわかる. 実際の運用上では、システムの障害を最小限に抑えるために false negative はできる限り避けなければならず、これらの手法は十分にイベント検出を任せられる段階には至っていないと考えられる.

もう 1 つ、別の問題として考えられるのが、これらの異常検出の手法が異常なイベントを検出するまでに対象を絞っており、その後の具体的なイベント情報の活用に結び付けられていないということである. ネットワーク管理者にとっては、何かが起こったということだけではその後の対処に結びつけることができず、次の段階としてどのようなイベントが起こったのか、という調査が必要となる.

3 提案

3.1 概要

上述の問題を踏まえて、本研究ではイベントの内容の判断までを自動化した新しいイベント検出手法を提案する. ここのイベントの判断とは、厳密な内容を要求するものではなく、あくまで管理者にその後の対応のための基準を与えるのに十分なものである. このイベント判断をイベント検出に含めることによって、false positive と呼ばれる、不要なイベントの誤検出の悪影響を抑えることができる. なぜなら、イベントの内容がわかるものであれば、管理者は対処が必要ないものはすぐに判断できるようになるからである. これを利用すると、false positive を恐れない積極的なイベント検出手法を採用することができ、false negative の低減にも効果が出せる. 本稿では、false positive を容認する代わりに false negative を抑えるシンプルで高検出率なイベント検出手法と、false positive を含むイベント情報を集約し意味付けすることによってネットワーク管理者のイベント把握を支援するイベント判断手法の 2 つを組み合わせ、新しいイベント検出の仕組みを提案する.

3.2 イベント検出部分の設計

イベント検出モジュールは、第一に false negative を少なくすることが重要である. 異常なイベントは多様な種類が存在するため、その見え方もさまざまである. 例えば SYN Flood 攻撃では、TCP のセッション数やパケット数が急激に増加するが、バイト数にはあまり変化が表れない、という見え方が考え

られる。しかし、例えばFTPサーバがダウンした場合には、TCPのバイト数も大きく減少するであろう。このようなさまざまなイベントに対応するためには、できるだけ多様なトラフィックの値を監視することが必要であり、また多様な値の変化の仕方を組み合わせることによって、その後のイベント判断に活用できる可能性がある。多様なトラフィックとは、例えばIP、TCP、UDP、ICMP、HTTP、SMTPなどといったプロトコル別に加え、それぞれのプロトコルについてもパケット数、バイト数、セッション数などの複数の側面からの見方が考えられる。本手法では、このように多数の監視ソースを用意し、それぞれの値について変化があったことをイベントとして検出する。ここで、変化を検出する具体的な方法については、多数の監視ソースをリアルタイムで同時に扱うため、効率のよい計算が求められる。そこで、シンプルで低コストな高域通過フィルタを設計し、高周波成分、すなわち短い時間間隔での変化の大きさからイベントを判断することにする。入力データ系列を離散時間信号の系列 $x(n)$ とみなすと、デジタルフィルタは次に示すようなN次の線形差分方程式で表される。

$$y(n) = - \sum_{k=1}^N a_k y(n-k) + \sum_{k=0}^N b_k x(n-k)$$

フィルタの周波数特性は、次数Nと係数 a_k 、 b_k によって調整できる。次数Nが大きいほど理想的な周波数特性を得られるが、4次程度でも急峻な周波数特性が得られる。つまり、少ないメモリと計算量で高域通過フィルタを実装可能である。

なお、フィルタ出力からどのようにイベントを判断するかについてはさまざまな方法が考えられ、考察の余地が残されているが、本稿では単純に閾値を設定して出力の大きいものを検出することにする。

3.3 イベント判断部分の設計

イベント判断モジュールは、イベント検出モジュールが検出したイベントそれぞれに対し、どういった原因でイベントと判断されたのかの意味付けを明らかにすることが目的である。また、同時刻や近い時間間隔で複数の監視ソースからイベントが検出され

た場合、それらを組み合わせてイベントの情報を集約することを目指す。例えばTCPとHTTPのトラフィックから同時にイベントが検出されれば、それはHTTPのイベントと判断されるべきである。他にも、HTTPのバイト数とセッション数が同時に増えていた場合、一般にバイト数の増加はセッション数の増加に起因すると考えられるが、HTTPのバイト数が増えてもセッション数が増えていない場合には、セッションあたりのバイト数の増加、つまり大きいファイルの転送に起因する、と予想できる。このように、複数の監視ソースのイベントの有無を情報として使用することで、イベントの内容を判断する助けになる。ただし、このようなトラフィックの監視とその種類の組合せだけでは、十分にイベントの内容を判断することは難しい。そこで、本手法ではアプリケーションやシステムのログを利用する。ログにはこれまでに起こったことが比較的具体的な記述で記録されている。また、多くの場合時刻情報を含んでいるので、イベントの検出時刻と照らしあわせてのマイニングが可能である。

具体的なログの利用方法について、本手法はリアルタイムのイベント検出を補助することを対象にしているため、時間制限のないオフラインでのテキストマイニングとは異なり、限られた計算時間で効果を上げなければならない。そこで方針として、最初にテキストの内容を吟味するのではなく、イベントの時刻周辺のログについていくつかの特徴を定義し、特徴的なログをある程度絞り込んだ後にそのログの内容を検討することにする。ログの特徴については、次の2種類の形式で与える。

(1) ログ1行ごとの特徴量と特徴ベクトル

異常なイベントが発生したときには普段とは異なったログが出力されている可能性があり、そのログを調べることは重要である。特徴量の与え方は、アプリケーション依存のものとアプリケーション非依存のもの2種類に分類する。アプリケーション依存のものとは、アプリケーションが定義した具体的なログの内容の一部を特徴量として使用するものである。例えばステータスコードやエラーコードの類がこれに相当する。アプリケーション非依存のものとは、どのようなログであっても同じように定

義できる特徴量である。例として、ログ1行あたりの文字数が挙げられる。アプリケーション依存の特徴量はログの特徴づけがしやすいというメリットがあるが、あらかじめ内容を知った上で設定する必要があるために汎用性に欠け、これが多くなると設定に負担がかかるというデメリットがある。逆に、アプリケーション非依存の特徴量では、汎用性があり事前の準備が不要というメリットがあるが、ログの特徴づけとしての効果は低い。これらの特性を把握した上で、必要十分な特徴量を定義する必要がある。

(2) ログの時間分布

もしログの内容からでは特徴的なものが見当たらなかったとしても、ログの密度からイベントがわかる可能性がある。例えば通常のアクセスを示すログしか残っていないが、同一時刻に大量のアクセスがあった場合、それはDDoS攻撃の可能性がある。また、エラーログが突然多くなった場合にも不具合や攻撃を示唆する特徴となる。

4 実験

提案手法の効果を確かめるため、実際のネットワークから得られたトラフィックデータとアプリケーションログを使用して、動作を確認する実験を行った。本実験ではWebサーバのイベント検出を対象を絞り、HTTPトラフィックとHTTPサーバのログを使用した。具体的には、HTTPトラフィックとしてWebサーバ稼働しているホストのHTTP転送バイト数、パケット数をそれぞれ内向き、外向きと計4種類の値を監視し、ログはApacheのaccess_logとerror_logを使用することとした。

4.1 イベント検出

差分フィルタは、カットオフ周波数 $\pi/200$ の4次の高域通過フィルタを設計した。フィルタの周波数特性を図1に示す。フィルタの周波数特性をどのように設計するかについても考察の余地があるが、本

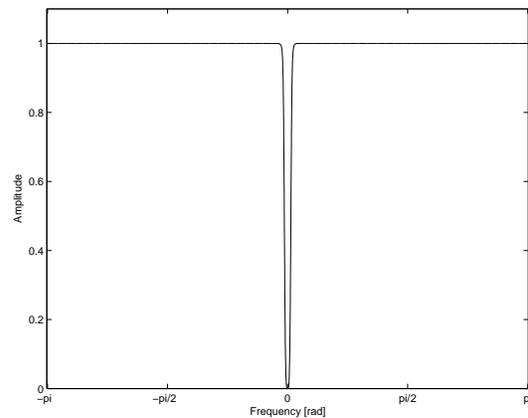


図 1: 設計した高域通過フィルタの周波数特性

実験では高周波成分がイベント検出に有効かどうかを確認することを目的として、この単純な高域通過フィルタの出力からイベントをカウントすることにした。

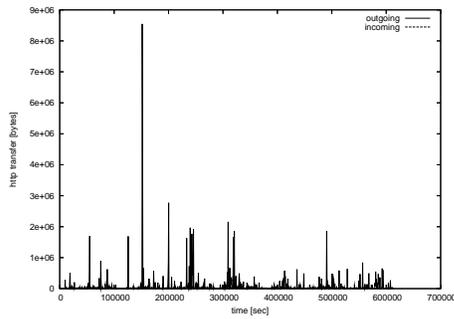
次に、実験に使用したHTTPトラフィックを図2に示す。図2(a)がバイト数、図2(b)がパケット数で、どちらも10分ごとの合計値をプロットしている。これらを差分フィルタに通した際の出力が、図3(a)、3(b)である。前述の通り、この出力からイベントを決定する方法には考察の余地があるが、本実験では単純な閾値の設定により、最も変化の激しかったものから順に数個のイベントを検出した。

4.2 イベント判断

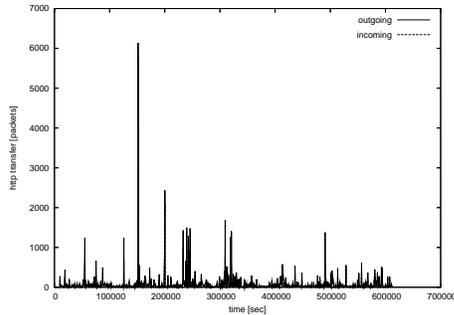
次に検出されたイベントの内容判断であるが、本実験では今回は高度に内容を判断する仕組みが未完成のため、最終的な判断に至る前段階の情報として、具体的なイベントについてそれがどのような見え方をしたか、という形で示す。それぞれのイベントに対し、監視ソースごとのイベント検出の有無の表れ方に特徴があるかどうか、またログの特徴づけが効果的に行われているかどうかについて議論する。

(1) 大容量ファイルのダウンロード

このイベントはバイト数、パケット数の内向き、外向きの4種類全てで検出されている。このイベントでは、特徴量としてaccess_logに記載



(a) HTTP Bytes



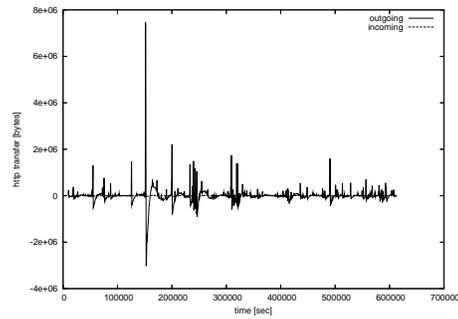
(b) HTTP Packets

図 2: 実験に使用したトラフィックデータ

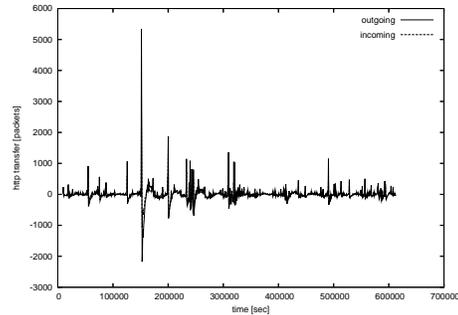
されている content-length の値を採用した場合に特徴的なログが見えてくる。イベント時刻付近の content-length の値を対数スケールで表したものが図 4(a) である。平均と比べて 2,3 桁大きい値を示しているログがあり、この時刻のイベントにとって特徴的なログとなっていることがわかる。実際にこのログの内容を見ると、access_log に動画ファイルへのリクエストが記録されている。これにより、このイベントが検出された理由は動画ファイルの転送開始であると判断できる。

(2) ワームのアクセス

このイベントは内向きのバイト数でのみ検出され、残りの 3 種類では反応していない。特徴量としてログの 1 行あたりの文字数を採用した場合の、イベント時刻付近の特徴量の分布を示したのが図 4(b) である。文字数が他のログに比べて著しく多いログが 1 つ存在していることがわかる。実際にこのログの内容を見る



(a) HTTP Bytes



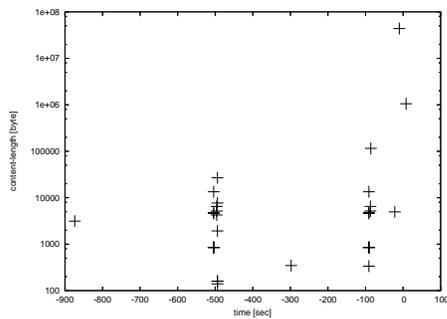
(b) HTTP Packets

図 3: 差分フィルタの出力

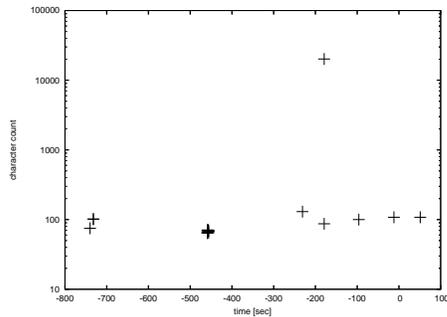
と、非常に大きなリクエストメッセージが記録されている。この大きなリクエストメッセージによって内向きのバイト数は大きく増加したが、内向きのパケット数や外向きのトラフィックには変化がなかった。よってこのイベントの内容はこの大きなリクエストメッセージそのものであると考えられる。なお、この大きなリクエストメッセージはワームのアクセスによるものであった。

4.3 考察

以上の結果から、高周波成分を利用した複数の監視ソースからのイベント検出手法が、イベントの検出だけでなく内容判断にとっても有効に作用しており、さらにログマイニングと組み合わせることによって、イベントの内容をより具体的に判断できることを確認した。これにより、イベント検出時にネットワーク管理者が直ちに問題に対処できるようにな



(a) content-length の分布



(b) 文字数の分布

図 4: ログの特徴量の分布

り、ネットワークシステムの安定運用を補助できると考えられる。

5 むすび

本稿では、シンプルな差分フィルタを用いた高検出率のイベント検出手法に、ログマイニングによるイベントの内容判断を組み合わせることによって、ネットワーク管理者にとって有益なイベント情報を提供する仕組みを提案した。また、小規模な HTTP サーバのイベントでの実験結果を用いて、その有効性を示した。

今後の課題として、高周波成分からイベントを決定するための処理を高度化することに加え、特徴づけられたログからどのようにして高度なイベントの要約を自動的に行うかについての検討が必要である。また、複数の監視ソースの情報を組み合わせるだけでなく、サービスの分散配置に対応するため、複数のホスト上のログ情報についても組み合わせ

る必要があると考えられる。

参考文献

- [1] Snort, <http://www.snort.org/>
- [2] HASSAN HAJJI, BEHROUZ. H. FAR, “Integrated Network Operation Baseline and Adaptive Detection of Faults and Performance Problems”, *IPSJ Journal* vol. 44, No. 2, pp. 386-396, Feb. 2003.
- [3] KENJI YAMANISI, JUN-ICHI TAKEUCHI, “On-Line Unsupervised Outlier Detection Using Finite Mixture with Discounting Learning Algorithms”, *Data Mining and Knowledge Discovery*, 8, pp. 275-300, 2004.
- [4] Paul Barford, Jeffery Kline, David Plonka and Amos Ron, “A Signal Analysis of Network Traffic Anomalies”, *Internet Measurement Workshop 2002*.
- [5] Ding Hui Zhang, Kazuhide Koide, Gen Kitagata, Glenn Mansfield Keeni, Norio Shiratori, “Detection of Network Events based on Digital Filtering”, *IPSJ SIG Technical Reports*, 2005-DPS-122, pp.271-276
- [6] 佐藤 陽平, 和泉 勇治, 根元 義章: “複数の検出モジュールの組み合わせによるネットワーク異常検出の高精度化”, 電子情報通信学会技術研究報告, 2004-NS-144, pp.45-48.