2004－CSEC－27　(14)
2004／12／20

ログの統一管理及び異常検出に関する研究

神尾 政和　　石田 常竹
安川情報システム株式会社

抄録

コンピュータネットワークにおける不正アクセスは年々増加しており、Firewall等の導入は一般的になったがそれだけで完全な不正アクセス防止はできていない。そのため随時対策を行う必要があり、有効な対策を実施するためには事象に関する調査が必須である。しかし、このような調査、対策は少数の高度な技術を持ったネットワーク管理者に依存している。特に、大規模な環境においては複数のノードにわたってインシデントが発生するため、効率的、効果的な調査を行うことすら困難である。そこで我々は、不正アクセスなどへの対策としてログの管理、解析に注目し、これらを効率的に実施するためログ研究集中管理システムの研究開発を行っている。

Unified Log Management and Abnormal Log Detection System

Masakazu KAMIO　　Tsunetake ISHIDA
YASKAWA INFORMATION SYSTEMS Corporation

Abstract

In recent years, illegal accesses in computer networks have been increasing, and use of equipments like a firewall became general. But these equipments may not block illegal accesses perfectly. So we have to repeat an investigation and countermeasure. But only engineers who have an advanced technology can implement such investigation and a measure. It is difficult to even conduct efficient and effective investigation, because incidents occur in many nodes in large-scale environments especially. Then, we aim for management and analysis of logs. In order to carry these out efficiently, we are studying and developing the unified log management system.

まえがき

近年、コンピュータネットワークの発達に伴い、ネットワークに接続されたサーバへの不正アクセスなどが社会問題となっている。これに対して、各国では法律[1-8]の制定がなされ、国家間では条約[9]の締結が進められるなど法的な対策が進められている。また、不正アクセスに対する技術的な対応としては Firewall 等の機器によるネットワークへのアクセス制限、ノードへのアクセス権限の設定による制御などは一般的に行われている。更に、最近では不正侵入検知システム（IDS：Intrusion Detection System）によりネットワーク上のトラヒックの監視も行われている[10]。

しかし、前述の技術によって不正アクセスを完全に防止することはできていない。むしろ、ネットワークの拡大によりセキュリティの脆弱なシステムからの不正アクセスが発生するなど、インシデントは増加しているといえる[11]。

このような状況を鑑みると不正アクセスを完全に防止することは極めて困難であると考えられる。むしろ、コンピュータネットワークにおける効果的なセキュリティ、不正アクセス対策を講じるにはいかに素早く問題を発見できるかが重要であると考えられる。コンピュータネットワークにおける異常検出の仕組みとしては IDS などがあるがこれらだけで全ての不正アクセスを発見できるわけではない。また、IDS による異常検出が行われたとしても不正アクセスの対象となったノードやその周辺の調査を行うことは必須である。そして、その調査はノードが出力したログを調査し、実際に発生した現象や IDS などの監視装置の通知内容と照らし合わせて問題を確認するのが基本となる。しかし、このような調査には高度な技術を持ったネットワーク管理者が必要とされる。そのため十分な人的リソースを準備できないことが多く、場合によっては不正アクセスが長期にわたって見逃されることもある。また、多数のノード、複数のネットワークによって構成されるような大規模環境では分散した複数のノードの調査が必要であり困難性が急速に増加する。

この様な観点から、我々はログを統一的に管理し、効果的に異常を検出するための仕組は前述の

ような調査を効率的な実施に有用であり、セキュリティ対策を実施するうえで有効であると考えた。我々は過去の研究[12]においてこのようなログ集中管理システムに要求される機能、要件は次の通りであるとの考察を行った。

(1) 複数ノードのログを形式によらず収集、管理可能であること
(2) 複数ノードからのログを横断的に異常検出が可能であること
(3) ネットワーク、ノード数に対して柔軟に拡張可能であること

本論文では、実現方法の検討結果については「ログ集中管理システムの概要」、ログ集中管理システムを構成する連携方式については「ログ収集解析サーバ間の連携」、ログの収集、解析を行うサーバの構成については「ログ収集解析サーバ」、現在の状況に関しては「現在の進捗状況」、の各段で説明する。

ログ集中管理システムの概要

我々の構想するログ集中管理システムは大規模なネットワーク環境において多種のノードからなるログを統一的に管理可能なことを目的としている。そのためには、ログの高速な集積が必須の条件となるが、数千台のノードからログを同時に集積する場合、ネットワークのトラヒック及びログ集中管理用サーバの処理能力が問題となる。

まず、ネットワーク上のトラヒックについて考える。クライアント/サーバ（C/S）モデル基づき、多数のノードからのログを一ヶ所に集積した場合、そのノードが接続されているネットワークでは大量のトラヒックが発生することが予測される。ネットワークに大量のトラヒックが発生した場合、syslog など信頼性の低いプロトコルを用いたログ転送方式では収集すべきログを消失する可能性が高い。TCP のような信頼性の高いプロトコルを用いたログ転送方式でも再送などによる遅延が発生するなどの問題を生じる可能性が高い。

同様にログ集中管理システムの処理能力について考える。まず、ログ集中管理システムではログの集積が必須の機能である。しかし、ネットワークを介して集積したログはハードディスクなどの二次記憶媒体に記録しなければならない。通常、ハードディスクなどの二次記憶媒体への I/O はネットワーク上の通信に比べて低速である。そのため、大量のログが到達した場合に二次記憶媒体への I/O がボトルネックとなり処理時間に影響する可能性がある。

次に、我々の構想するログ集中管理システムでは複数のログを統一的に扱うことを目的のひとつとしてあげている。そして、ログの収集対象ノードとしてサーバだけではなくルータやスイッチングハブの様な容易に機能追加を行えない機器も対象と考えている。このような機器を対象に導入の容易性を考慮すると、ログ収集対象のノード側でログをログ集中管理システムで扱う形式に変換するのではなく、ログ集中管理システム側で受信した時点でログを内部形式へと変換を行う必要がある。しかし、一箇所にログを集積した場合、大量のログの形式変換にかかるコストは膨大になることが予想される。更に、我々はログ集中管理システムでリアルタイムでの異常検出機能を実装することを考えており、この検査にかかるコストも巨大になると考えられる。

以上の考察から我々は単純な C/S モデルによる一箇所へのログの集積は我々の考えるログ集中管理システムの方法として不適当であると判断した。そこで我々は大量のログが一台のログ収集解析サーバに集中することを避けるため、Peer to Peer（P2P）技術を応用し、複数のログ収集解析サーバが連係動作することによってログ管理システムを構築可能な方式の検討を行った。この詳細については「ログ収集解析サーバ間の連携」で説明を行う。

また、ログの集積を行うログ収集解析サーバについては多様なログの収集、解析が可能である必要がある。まず、様々なログを収集可能であるためには syslog をはじめ、WindowsNT 系の OS で利用される Event Log、ネットワーク機器で利用されることの多い SNMP、IDS などではメールによる通知などを考慮しなければならない。更に、ノードによってはそれぞれが独自に実装した方法でのログ収集に対応する必要もある。しかし、これら全てに対してはじめから対応することが不可能であることは明らかである。つまり、受信可能なログ形式、プロトコルは拡張可能でなければならないといえる。また、受信可能なログを拡張した場合には既存のログ解析機能では対応できないと考えられるためログの解析機能についても同様に拡張可能でなければならない。そこで我々はログ収集解析サーバを 3 階層に分けることで各階層のインターフェースを統一し、必要な機能をモジュールとして容易に追加可能な仕組みとした。これによりモジュール単位での機能の追加、修正、削除

が可能となり拡張の容易性を確保できると考えた。その結果について「ログ収集解析サーバ」で説明する。

このように、我々はネットワーク構成、ネットワーク規模、対象ノードに対して柔軟な構成が可能であり、ログの収集、解析をログ収集解析サーバの連携によって行うことが可能なログ管理システムを構想している。

ログ収集解析サーバ間の連係

前段において C/S 構成を利用し、単純に中央のサーバにログを集積するモデルを用いた場合、ログ収集解析サーバに到達するトラフィックの増大と処理すべきログデータの増大が大きな問題となりえると述べた。この問題を解決する方法として我々はログ収集解析サーバの分散配置を検討した。ログ収集解析サーバを分散配置することができればログ収集解析サーバ一台あたりのトラヒック及び処理すべきログデータの量を軽減することができるからである。

ログ収集解析サーバを分散配置する構成を考えた場合、次のような理由から P2P ネットワークを利用した分散データベース（分散 DB）モデルに利点が多いと考えた。

- ネットワークトポロジに対して柔軟である
- 伝達経路の多重化が可能であり障害に強い
- 各構成要素が機能的に代替可能であるため拡張、構成変更が容易である
- 各ノードの近傍の機器からのログだけを集積するためトラヒックを削減可能である

そこで我々はログ収集解析サーバ同士が P2P ネットワークで接続し、それぞれが自立的に連携することによってログ管理システムを構築可能な構成を考えた。図 1 にそのイメージを示す。

このモデルにおいて各ログ収集解析サーバは次の様に動作する。まず、ログ収集解析サーバは比較的少数のノードごとに配置する。ログ収集解析サーバは自分以外にいくつかのログ収集サーバを近傍として設定を行い、相互に通信可能とする。ログ収集解析サーバは自身に割り当てられたノードのログを収集し、収集したログの異常検出を行う。もし、いずれかのログ収集解析サーバが異常を検出した場合、P2P 通信によって近傍のログ収集解析サーバに通知、関連情報の検索依頼メッセージを送信する。メッセージを受信したログ収集解析サーバは自身に蓄積したログデータの検索を行う。蓄積したログデータ内に関連があると判断可能なログデータを発見した場合はその情報をメッセージ送信の起点となったログ収集解析サーバに返信する。また、メッセージの送信元以外で自身の近傍に設定されたログ収集解析サーバにメッセージを転送する。この繰り返えしによりログサーバ同士の連携によってネットワーク全体に対してログの検査を行い、結果を知ることができる。

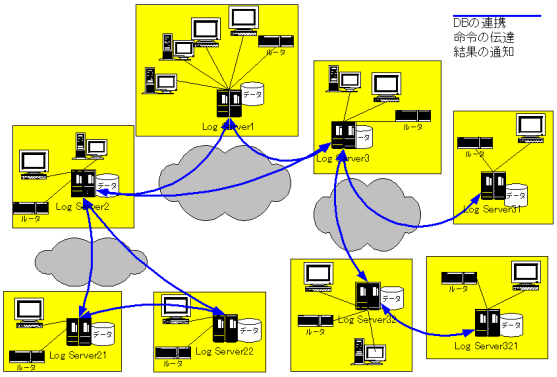

図 1 ログシステム構成概念図

例えば、図 1 ではログ収集解析サーバ LogServer1 は LogServer2、3 と、LogServer2 は LogServer1、21、22 と近傍である。ここで LogServer1 が発したメッセージは LogServer1 から LogServer2、LogServer2 から LogServer21、22 へと伝達され、LogServer1 は直接存在を知らない LogServer21、22 とも連携が可能となる。同様にそのほかの部分とも連携が可能である。

しかし、これだけでは途中にループ構造を含む場合、同一のメッセージが重複して流れることになり余分なトラヒックが発生する。例えば先ほどの例で、図 2 では LogServer21、22 は互いに近傍であるため LogServer1 からのメッセージを互いに送信しあってしまう。P2P プロトコルとしてよく知られている Gnutella プロトコル[13]ではこのような重複メッセージを受け取った場合は一方を破棄する。しかし、それだけではトラヒックは参加する機器の二乗のオーダで増加するため参加するノードが多数になるとネットワーク帯域を圧迫するという問題を生じる。Chord[14]などの P2P プロトコルではデータを重複なく配置し、検索手順を工夫することで P2P ネットワークの柔軟性を生かしながらトラヒックの抑制を実現している。しかし、我々の考えるログシステムではメッセージを全体に通知し、その結果を知る必要があるため Chord の方式では問題があった。また、Chord の方式は経路の帯域幅や状態を考慮しないため低

速な経路を選択してしまうなどの可能性がある。そこで我々はログシステム間の連携を実現するために、近傍ノード間における経路の最適化機能を持った P2P プロトコルを新たに考案した。図 2 は我々の考案した P2P プロトコルでの経路制御手法を説明したものである。

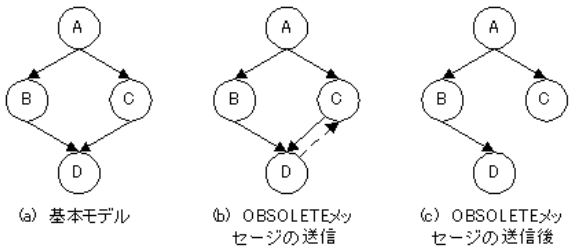

図 2 経路制御方法

図 2 は、ノード A の近傍にノード B、C が位置し、ノード B、C の近傍にノード D が位置する構成を示す(図 2(a))。この構成では A が発したメッセージを D は B、C 両方から受信することになる。前述したように Gnutella プロトコルでは D は B ないし C から受け取ったメッセージのいずれかを破棄する。しかし、このような重複メッセージは常に送信されるため、明らかに無駄である。そこで、あるノードが同一の要求を受信した場合、2 つめ以降の要求を発行したノードに対して、送信停止要求(OBSOLETE)を送信し、OBSOLETE を受信したノードは、次回から要求をそのノードに送信しないという制御を考案した。例えば、D が同一の要求を B、C から受信した場合、D は後からメッセージを送信した側に対して OBSOLETE を送信する(図 2(b))。ここで D が後から受信したメッセージの送信ノードを C とすると、C は、OBSOLETE 受信後、ノード A からの要求をノード D へ送信しない(図 2(c))。この仕組みにより以降の要求伝達には無駄なトラフィックは生じない。そのため OBSOLETE による経路制御前の時点では Gnutella と同様に $O(n^2)$のトラヒックが発生するが、OBSOLETE による経路制御が行われた時点以降のトラヒックは $O(n)$となり効果的にトラヒックを削減する。

しかし、この方法ではメッセージの中継ノードに障害が生じるとその先のノードへのメッセージを伝達できなくなるという問題ある。例えば、前述の例のように A は B および C に、B が D にメッセージを送信するという状態で B に障害が生じた場合、D にはメッセージが伝達されなくなってしまう。この問題はメッセージを送信するノードが送信先のノードの障害を検出した場合、OBSOLETE を無視してメッセージ伝達を行うことで障害を回避してメッセージの伝達を行うことで解決可能である。例えば、B に障害を検出した場合、A は B に接続不能であることを接続確立時に検知することができる。このようにして障害を検出した A は C に OBSOLETE 解除要求を送信したうえでメッセージを送信する。これによって C は D にメッセージを送信可能となる。

以上の方法により、P2P による柔軟なネットワーク構成を可能としながら P2P ネットワークの問題であるトラヒックの増加を抑えた通信を実現することが可能である。

ログ収集解析サーバ

「ログ管理システムの概要」で述べたように単体のログ収集管理サーバに関しては、様々なログの形式への対応を行う必要があり、受信するログの種類の増加に伴う異常検出手段の増強を柔軟に実現可能な手法を構築する必要がある。我々はこの要求を満たすために次のように考えた。

まず、多様なログの収集を実現する最も容易な手段はログ収集対象ノードにログ収集用のエージェントプログラムを導入し、ログ形式の変換を行った上でログ収集解析サーバに送信することである。この方法はログ収集対象が追加された場合にその対象ノードごとにログ収集エージェントを開発すればよく、ログ収集解析サーバの変更がないため効率が良い。しかし、例えばルータのような機器の場合、追加のプログラムを導入できないことが多い。このような条件を考慮すると、ログ収集解析サーバ側でログの形式の拡張に対応する必要がある。これには、ログ収集対象ノードがネットワークを用いて外部にログを送信する機能を持つ場合、そのプロトコル、形式に対応したログ受信インターフェースだけをログ収集解析サーバ側に用意すればよい。必要な機能だけを追加可能であれば開発期間を短縮し、既存機能の影響を少なくできる。

次に、受信可能なログの拡張を行った場合、既存の異常検出機能では有効な検出を行えない可能性が高いため異常検出機能の拡張も必要となる。しかし、対応するログの増加に対して常に全ての異常検出機能を作り直す必要があるとは限らない。

例えば、新たに対応したログ形式にのみ必要な異常検出機能の場合、既存の仕組みを変更する必要はない。また、既存のログデータ形式と新しく収集することにしたログデータ形式の両方を利用する場合でも既存の異常検出機能に依存しないならば独立に機能を追加可能である。既存の異常検出機能に変更が必要な場合でも既存の異常検出機能の全てに変更が必要とは限らない。従って、内部で動作する異常検出機能についてもログ受信インターフェース部分と同様に必要な機能のみを追加変更可能であるべきである。

以上の考えから、ログ収集解析サーバは必要な機能ごとのモジュールの組み合わせによって実現するべきであると考えた。しかし、全体をモジュールとして構成したとしてもそれぞれの入出力が独自に定義される場合、特定のモジュール間に強い依存関係を生じ柔軟な追加変更が不可能になる可能性が高い。そこで我々はログ収集解析サーバを階層に分け、それぞれの階層ごとのインターフェースを統一することで必要な機能の追加を容易に行えると考えた。図 3 はログ収集解析サーバのイメージを示した図である。

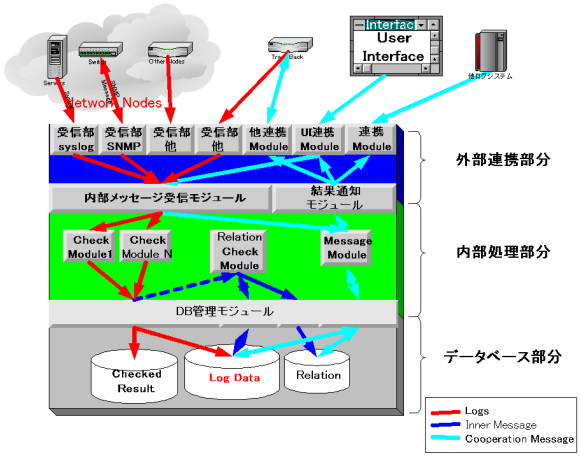

図 3 ログ収集管理サーバ構成図

この図の中央の三層からなる矩形部分がログ収集解析サーバである。各層は上から外部のノードとのインターフェースとなる階層(外部連携部分)、ログ収集解析サーバ内部での処理を行う部分（内部処理部分)、データベースを配置する階層（データベース部分）である。それぞれの階層に配置したオブジェクトはログ収集解析サーバを構成するモジュール群である。また、図中の矢印は受信したログデータがデータベースに蓄積されるまで（Logs)、ログ収集解析サーバ内部での検査などのためのデータ（Inner Message)、他ログ収集解析サーバとの連携のためのデータ（Cooperation Message)、の流れを示す。以下、図を用いてログ収集解析サーバの動作を説明する。

まず､外部連携部分にはログの受信機能のほか、他ログ収集解析サーバやユーザインターフェースとのやり取りを受け持つモジュールを配置する。ログ受信用モジュールはそれぞれ対象とする形式、プロトコルでログを受信し、ログ収集解析サーバ内部で利用可能な形式に変換する。これによってログ収集解析サーバ内部ではデータ形式を意識することなく必要な処理だけを実装できる。

次に、内部処理部分には主にログの解析を行うモジュールなどを配置する。そのほかにも、異常検出時に他のログ収集解析サーバと連携を開始するためのモジュールなど、ログ収集解析サーバ内での処理を行うためのモジュールを配置する。また、これらのモジュールは外部連携部とデータベース部分を流れるデータに対して処理を行うことも（矢印：Logs)、データベース部分に蓄積されたデータを処理することもできる（矢印：Inner Message)。

また、外部連携部分と内部処理部分の間に内部メッセージ受信モジュール、内部処理部分と外部連携部分の間に結果通知モジュールを配置している。これらは設定に応じて外部連携部分のモジュールから内部処理部分のモジュールへのデータの流れを制御する。これにより外部連携部分のモジュールと内部処理部分のモジュールが強く依存しあうことを防ぎ、それぞれが独立に交換可能となるようにする。

最後にログ収集解析サーバ内で使用するデータベース部分も独立した階層として扱う。これは本システムを利用する規模によって使用するデータベースシステムを変更可能とするためである。そして、データベースごとのインターフェースの違いを吸収するためログ収集解析サーバで利用するインターフェースを提供するための階層を用意する。この階層の存在によってデータベース選択の自由度が増すだけではなく、データベースをログ収集解析サーバとは別に配置するような構成も可能となる。

このように各部分の独立性を高めることで機能の追加､更新時に高い柔軟性を実現できる｡また、

規模に応じてモジュールを減らしたり、データベース変更したりすることが可能となるため規模応じた柔軟な構成を実現できる。

現在の進捗状況

我々は前述の構想に基づいてログ集中管理システムの基本設計を行い、プロトタイプの実装を行った。表１に実装した機能の概要を示す。

表 1 実装済み機能一覧

機能分類	実装済み機能
ログの受信	Syslog、SNMP、Mail（SMTP、POP3）
連携機能	P2P を利用した連携機能
単一ログの検査	過去のログの出現確率とカテゴリ分類に基づく方法 ベイズ推定に基づく方法[15]
複数ログの関連性に基づく異常検出	Automaton を用いた手法
データベース管理機能	データベースの隠蔽 上位層へのインターフェースの提供

これらの機能実装でログの収集、異常検出、異常検出後に複数のログ収集解析サーバを連携しシステムとして動作するための一連の機能を試験可能となった。現在、我々は本プロトタイプを使用し、ログの収集管理、異常検出、連携動作が予想通り機能することの確認と性能の検証を行っている。現在までの機能検証において、ログ収集解析サーバ間の連携機能については我々の構想どおり、トポロジーに対して柔軟であり、効果的にトラヒックの削減が可能であることを確認している[16]。ログの異常検出については２種類の統計的なモデルに基づいた手法と、高速に複数のログの関連性に基づいた異常検出を実現するために Automaton を用いた手法の評価を行っている。同時に、複数のログ収集管理サーバの連携動作に関する評価を進めている。

また、上記の処理に要する時間についても測定データからパラメータを定め、規模の拡大に対してどの程度になるかの予測を行っている。

まとめ

以上で述べたように複数のログ収集解析サーバの協調動作によりログの統一管理が可能なシステムを考案した。そして、そのために必要と考える連係動作機能、容易に機能の追加、変更が可能なログ収集解析サーバのプロトタイプの開発を行った。現在我々はこのプロトタイプを用いた予備的な機能検証を行い、その結果から本システムが有効に機能するであろうという感触を得ている。今後、このプロトタイプの拡張、修正を行いながら各機能の検証、性能の改善を進め、本システムが有効に機能することの実証を行う。

謝辞

本研究は、通信・放送機構における委託研究テーマ「大規模ネットワークセキュリティの確保に向けた研究開発」によっている。ここに記して謝意を表す。

（参考文献）

[1]不正アクセス行為の禁止等に関する法律
[2]電子署名方
[3]個人情報保護法
[4]電気通信事業法
[5]プロバイダー責任法
[6]不正競争防止法
[7]電子契約法
[8]刑法（コンピュータ犯罪防止法）
[9]サイバー犯罪条約
[10] 警察庁生活安全局生活安全企画課, ハイテク犯罪等に係る被害状況の調査《報告書》,平成 15 年 3 月
[11]http://www.npa.go.jp/cyber/toukei/html/html18.htm
[12]http://www3.nict.go.jp/ns/s802/seika/71/71yasukawa.pdf
[13]http://www.jnutella.org/docs/gnutellang/gnutella_protocolv4.shtml
[14]L. Stoica, et al.、Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications、http://pdos.lcs.mit.edu/chord/
[15]Graham, Paul.、Better Baysian Filtering、http://www.paulgraham.com/better.html 、2003.1
[16]http://www2.nict.go.jp/ns/s802/seika/h15/seika/71/7103_yasukawa.pdf