ディジタルフォレンジックのための脆弱な経路における信用できるログ転送方式

友野敬大† 上原稔† 島田裕次†

近年、米国企業に留まらず、日本企業の不祥事が多発し、内部統制の必要性が急速に高まっている。中でも、ログ監査は重要であり、これが行われていない内部統制システムは脆弱なものになる。しかし、ログ監査のためのシステム導入や管理・運用のコストは決して安価ではなく、ログのデータ量に比例してコストがかかる。そこで我々は、VLSD を用いて低コストかつ半永久的にログを保存するシステムを開発した。このシステムではログの転送を一部分でしか保証しておらず、ディジタルフォレンジックは考慮されていない。本研究では、ディジタルフォレンジックに対応するべく、中継サーバにおける悪意あるユーザによるログの改ざん、転送時のログの欠損を検出し、信頼できるログ転送方式を提案する。

Transferring Trusted Logs across Vulnerable Paths for Digital Forensics

AKIHIRO TOMONO† MINORU UEHARA†
YUJI SHIMADA†

In recent years, many scandals of company's accounting are not reported in only United States but also in Japanese. The need of Internal Control is growing rapidly. In particular, auditing logs is important for Internal Control. Internal Control lacking audit is incomplete. However, the cost of information system is depended on the amount of data. Especially, the amount of log data is very large. We overcame this issue by developing VLSD, which is a toolkit for constructing large-scale disks. Using VLSD, we have realized a log storage with the low and the lifetime long. However, this log system does not guarantee to transfer trusted log in vulnerable transferring path. It is not sufficient for digital forensic. In this paper, we propose a trusted log transferring method suited for digital forensic. In this method, we find whether intruders and mal-administrators modify logs in each router node or not.

1. はじめに

近年，米国企業に留まらず，日本企業でも不祥事が多発し，内部統制の必要性が急速に高まっている．内部統制とは，COSO(Committee of Sponsoring Organizations of the Treadway Commission)によれば，『(1) 業務の有効性・効率性，(2) 財務報告の信頼性，(3) 関連する法規の遵守の 3 つの目的の達成に関して，合理的な保証を提供することを意図した，会社の取締役会，経営者およびその他の従業員によって遂行されるプロセスであり，相互に関連する要素，すなわち，統制環境，リスクの評価，統制活動，情報と伝達，モニタリングから構成される』と定義されている．本論では，これを実現するための一連の仕組みを内部統制システムという．

日本 IBM やサン・マイクロシステムズなどのような大手企業では，独自の内部統制システムを構築しているケースが多々ある．システム構築を外注すると，大企業であればあるほどシステム規模が大きくなり，その分のコストが発生することになる．

企業では，コンプライアンス確保のために，内部統制を実現するためのツールを選択し，活用する必要がある．しかし，現在の普及品はどれも高価であり，導入コストだけではなくデータ量に応じた運用コストについても考慮しなければならない．実際のログ（以下，生ログとする）は，何も加工されていないため，その証拠能力は比較的高いものと考えられるが，ログはシステムの規模に比例して，確実に保存領域を圧迫していく．我々は OS の機能を利用し，半永久的にログを保存可能なシステムを開発した．VLSD を用いて容量を確保するので，NAS や SAN などといったストレージシステムを導入しなくても，低コストで大容量のストレージを利用できる．

このシステムでは、一部分においてログを保証しているが、それだけではディジタルフォレンジックを考えると十分ではないと言える。転送中の経路において、改ざんやパケットの欠損があった場合、それを検出できなければ、そのログは証拠性に欠けると言っても過言ではない。そこでディジタルフォレンジックに対応するべく、中継サーバにおける悪意あるユーザによるログの改ざん、転送時のログの欠損を検出し、信頼できるログ転送方式を提案する。サーバ管理者などの権限を持つユーザが悪意を持つ場合、それを事前に知るのは難しく、ログを改ざんされた後では証拠も残らない。ログに対して、ハッシュをその都度作成することで改ざんされにくいログを作成し、このログとハッシュの管理方法について言及する。

2 章で本研究の関連研究、3 章で転送方式の提案、4 章でシステムの評価、5 章でまとめと今後の課題について言及する。

† 東洋大学
Toyo University

2. 関連研究

2.1 ディジタルフォレンジック

ディジタルフォレンジック[1]とは、ディジタルフォレンジック研究会によれば、「インシデント・レスポンスや法的紛争・訴訟に対し、電磁的記録の証拠保全及び調査・分析を行うとともに、電磁的記録の改ざん・毀損等についての分析・情報収集等を行う一連の科学的調査手法・技術」と定義されている。

ディジタルデータの法的証拠能力という問題を考える。日本の法律では、いまだに物的証拠のみ証拠能力を持つとされている。そのため、コンピュータを証拠として採用するときには、保存されているデータではなく、データを格納している媒体すなわち HDD などそのものが必要になる。ディジタルデータのような電磁的記録単体では証拠として認められていない。そのため、電磁的記録を出力した書類等を法廷に提出しなければならないが、書類等を作成するまでの過程で改変される可能性があるので、その真実性に疑問がもたれることになる。実際には、ディジタルデータと関連する物的証拠や証言を組み合わせて、法廷で用いる。通常フォレンジック調査では、対象 HDD から 100%物理コピーをしたのち、ハッシュ値を残すなどしてその証拠性を保つ必要がある。

不正なアクセスやその他のシステムトラブルに備えるべく、通常、サーバではログを収集している。システムのトラブルやユーザの行動のログ情報は、不正侵入などの重要な問題解決の手がかりになるためである。具体的には、UNIX 系 OS の Syslog、Windows のイベントログ、Apache のアクセスログ、IIS のアクティブログなどがある。加えて、ただログを保存するのではなく、改ざんされていないことを証明する仕組みが重要になる。

ログは、それ単体からでは情報を取り出すのが難しい。そのため、種類の違うログファイルや、同じものでも、スパンを長く見て情報を抽出することが多々ある。しかし、違う種類のログはよいとしても、時間軸で考えた場合、過去のログが残されていないと見えてこない情報がある可能性も否めない。現在の法律では記録を義務付けられているが、長期保存の義務はない。「犯罪捜査のための通信傍受に関する法律」第 7 条第 1 項では、ログの保存を 30 日間としており、また、「サイバー犯罪条約」第 16 条第 2 項では 90 日間の保存としているが、これによりプロバイダなどの通信業者は、コスト面や容量の面で過大な負担を課されるため、義務にはなっていない。その一方で、KDDI や日産自動車ではログを永久保存するとしている。前者は 2001 年、P2P ソフトを介して、社員情報および顧客情報の漏えい事故を起こしたことから、個人情報保護を最重要課題とし、2006 年に実施を始めた。後者は 2006 年、旧顧客データベースから顧客情報 200 万件が流出している恐れがあるとして、2007 年に実施を開始した。

ログの永久保存は費用対効果を考えると、非効率的なものになる。不正アクセス禁止法および電子計算機使用詐欺罪や電磁的記録不正作出および供用罪などのコンピュータ犯罪に関する刑法の時効は、およそ 3~5 年とされている。そのため、裁判証拠としてログを用いる場合、その期間分は保証されるのが望ましい。

KDDI や日産自動車のようにログを永久保存する際は、ログに記録されている情報についてのプライバシの問題に気をつけなければならない。送信元など個人を特定できる情報の取り扱いはセンシティブである。個人情報であるそれらを長期間保存するので、流出するのは防がねばならない。

2.2 VLSD

通常、大規模ストレージを構築する際、最初に検討されるのは分散ファイルシステムで、空き容量をマウントする方法だろう。しかしながら、NFS を始めとするファイルシステムレベルの分散ストレージは、空き容量を連結して 1 つのストレージにすることはできない。ディスクレベルの分散ストレージが必要になる。

VLSD とは、Java によるソフトウェア RAID と NBD の実装を含む大規模ストレージ構築のためのツールキットである[2][3]。VLSD は 100% pure Java であり、Java が動作するプラットフォームの上なら VLSD も動作する。そのため Windows や Linux が混在する環境に適している。VLSD を用いると OS に制約されることなく NBD デバイスと RAID を自由に組み合わせることができる。

PC 教室や会社の遊休資源（HDD の空き容量）を連携し、1 つの大きな仮想ストレージを構築して用いる。VLSD ツールキットは多様なクラスを組み合わせることで 8EB までの大容量ストレージを実現する。これを用いて、512 台の PC からなる PC 教室のために、70TB のストレージの試作に成功した。ただし、ストレージは大きくなるほど、信頼性が低くなるため RAID を用いて信頼性を上げる必要がある。試作システムでは、512 台のディスク（1 ディスク=170GB）を 32 グループにして RAID66 を構築する。

また、NBD を用いることで、ファイルシステムに依存しないディスクレベル分散ストレージの実現が可能となる。これにより、高価なストレージの必要性がなくなる。このような分散型ストレージは、HDD 代のみとなるので集中型の費用と比較すると、そのコストを大幅にカットすることができる。このシステムは 64 ビットファイルサーバとディスクサーバがある。ディスクサーバの Linux や Windows などの OS からなる仮想ディスクは、ディスクの読み込みや書き込みを Java の RMI で機能を提供する。ファイルサーバの方は用意されたディスクに接続して RAID66 を構築する。

2.3 VLSD のセキュリティ

ログの保存容量の問題を解決する VLSD だが、実際の運用となると、内部統制を実現するために、そのセキュリティが問題となる[4]。具体的には次のような問題が挙げ

られる。
・PC における盗聴および改ざん
・通信路における盗聴および改ざん
・なりすまし
これらの問題を解決するセキュリティ技法について考える必要がある。VLSD のセキュリティは、AAA アーキテクチャに基づく。それぞれ、認証（Authentication）、認可（Authorization）、監査（Audit）の頭文字を取ったものである。

VLSD による大規模ストレージは、多くのクライアントマシンの容量の一部を収集して構築されている。そのため、サーバ OS がアクセス制限をかけても、クライアント上から直接不正なアクセスを許してしまう。その原因として、リモートディスクへのアクセスを許すことと、その内容を読み取れてしまうことが挙げられる。アクセス権の問題は、ディスクを提供している以上、管理者である可能性は大いにあるので、解決するのは困難である。しかし、後者の原因を解決すれば、内容は読み取られないので、データの保護をすることができる。その方法としてディスクの暗号化を挙げる。VLSD は、暗号化を実装するクラスを持つ。暗号化することで、クライアントマシンからは内容を読み取ることができなくなる。

また、改ざんを防ぐ方法として、書き込みを禁止する方法が挙げられる。書き込みが必要な場合には、別のディスクとして書き込み、その差分を検証すればよい。VLSD は ReadOnlyDisk というラッパークラスを持つ。これをラップすることで実現できる。暗号化のオーバーヘッドは、通常のディスクアクセスなら無視できないものだが、ログを参照する機会はそう多くないと考え、高速性よりも暗号化による安全性を選択すべきだろう。

2.4 ログ管理システム

ログ管理にはいくつかのレイヤがある。生ログを収集・保存するレイヤ、データベースを用いて管理するレイヤ、そこから情報を取り出し、グラフ化や関連付けをするレイヤである。図 1 にログ管理のレイヤを示す。

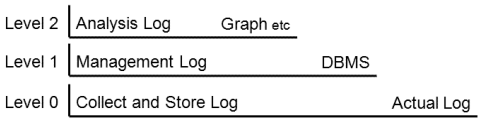

図 1　ログ管理のレイヤ
Figure 1 Layers of Log Management

今日では、ログをデータベース化して管理および解析するのが通例であるが、このシステム[5]では、何も手の加えてない生ログに着目し、保存・管理することでログの信頼性をより高められると考える。システムが排出したそのままの生ログは、証拠能力が最も高いものだと考えられるからである。データベース化することは、ユーザの使いやすさの向上につながるが、その処理を行うことで何らかの情報が損失する可能性も大いに考えられる。また、厳密に言えば、データベース化はログの加工であり、その間に不正な加工や欠損がないとは言い切れない。

今日ではログをデータベース化することが多く、またそのような製品も多く提供されている。逆に言えば、生ログを保存・管理するシステムはほとんど普及しておらず、また通常はそのような管理方法を採用しないだろう。その大きな理由として、容量の問題が挙げられる。単体としては大きくなりにくい（と考えられがちな）ログファイルだが、収集していけば確実にディスクを圧迫することになる。そこで、我々は VLSD を開発し、それを用いて生ログの保存・管理を行う。コンピュータの遊休資源を用いるので、新たな設備を追加する必要はない。そして、将来的には生ログを保存した上で、随時 DBMS と連携する。具体的には、Level0 と Level1 の間にログの正規化を行う必要がある。これにより、機能性とログの証拠性を両立できると考える。システム構成図を図 2 に示す。

このシステムでは、OS にあらかじめ備わっている Syslog や Logrotate といった機能を利用してシステムを開発する。従来では、Syslog が排出したログは上書きされ、古いログは参照できなかった。Logrotate のデフォルトの設定が月に 4 回ログをローテーションし、古いログを上書きしていくため、最高でも 4 週間前のログしか残らないことになる。ローテーションする回数を増やすことで、ある程度は解消できるが、それよりも古いログをニアラインストレージにコピーして管理する方が有益である。ニアラインストレージは、本大学の Web サーバやファイルサーバなどの常時稼働しているサーバ 25 台から遊休資源を収集して、構築する。その構成を図 3 に示す。

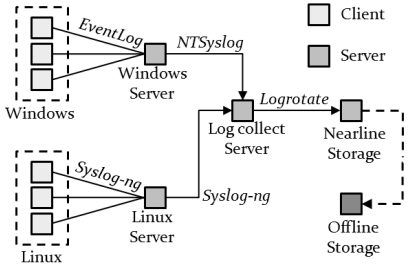

図 2　システム構成図
Figure 2 A Log Management System

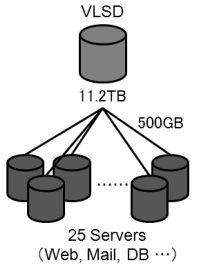

図 3　サーバファーム
Figure 3 Sever Farm

3. ネットワークにおけるログの保証

3.1 概要

ディジタルフォレンジックにおいて重要なのは、ログの整合性である。ログを転送する際、あるいは、一時的にログを保管するサーバで改ざんが行われてはならない。万が一、これがあった場合に、それを検出できる用意をしておく必要がある。ディジタルフォレンジックにおけるログは、その証拠性が問われる。実際にログが法廷において有効であるかどうかを判断できなければならないからである。継続的に行われるようであれば、ネットワーク自体を再構築して中継サーバを切り離す必要がある。

関連研究で挙げた[5]のシステムでは、一部分においてこれを実装しているが、転送するたびに検証する必要があるので改良する。いくつかのモデル例を挙げ、その実用性について考察する。ただし、今回のログ転送において着目するのは、中継サーバにおける悪意ある管理者や第三者を想定するものである。クライアントから排出されるログは改ざんやパケット損失していないものとして考える。

また、ログに対するハッシュに関しては、ファイル単位ではなく、レコード単位に作成する。レコード単位のハッシュは、ファイル単位のものと比較すると精度を増すと考えられるからである。よりディジタルフォレンジックに対応する形となる。次章で詳しく言及する。

モデル図中において、Client はクライアントマシン、LogRouter1 および LogRouter2 は中継サーバをそれぞれ指し、Server は VLSD を用いて作成したニアラインストレージを指すものとする。

3.2 モデル

(a) Server のみへ転送

まず、基本的な形としてクライアントマシンからログが発生した段階で、そのログファイルのハッシュを作成する。これをニアラインストレージでログファイルと一緒に保管することで、途中通過する中継サーバで行われるログの改ざんや欠損を検出することができる。図 4 に示す。

この方法では中継するサーバが単一の場合のみ、その改ざんを検出することができる。しかし、中継サーバが複数台ある場合、経路中のどのマシンで改ざんがされたのかを知ることはできない。Client から Server へと転送されたログが改ざんを受けていた場合、改ざんをされた場所が LogRouter1 なのか LogRouter2 なのか判断し、ネットワークから切り離さなければならない。具体的にどのサーバで改ざんが行われたか検出できる用意をする必要がある。

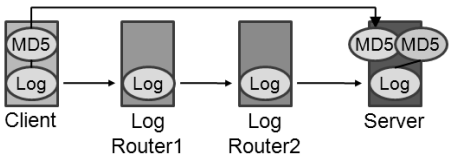

図 4　ハッシュファイルの流れ(a)
Figure 6 Flow of the hash (a)

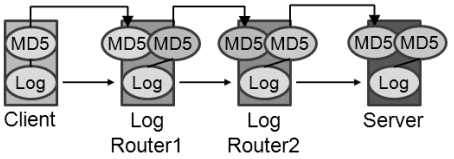

図 5　ハッシュファイルの流れ(b)
Figure 5 Flow of the hash (b)

(b) 隣接するルータへの転送

次に、それぞれの中継サーバでログファイルからハッシュを作成する場合を考える。図 5 に示す。この方法は、各々のマシン間での通信のみを保証する。すなわち、改ざんを検出するためには、LogRouter1 や LogRouter2 でログに対するハッシュをその都度比較しなければならない。一見すべてのマシン間でログが保証されるように見えるが、もし LogRouter1 でログの改ざんが行われた場合、改ざんされたログとハッシュが転送されると、LogRouter2 以降ではこれを知るすべがない。

(c) 上流へのブロードキャスト

モデル(a)、(b)ともに重大な欠陥がある。これらを補う方法を考えなければならない。Client からのログが正しいものだと仮定すると、このハッシュを基準にするのが確実だと言える。途中で経過する LogRouter1 および LogRouter2 にも、それぞれハッシュファイルを保管することを考える。図 6 に示す。

中継サーバおよび Server で、Client のハッシュを元に比較することで、改ざんを検出する。例えば、LogRoutet1 から LogRouter2 へ改ざんされたログとハッシュが転送されたとする。ここで、LogRouter2 で作成したハッシュと Client のハッシュを比較すると、LogRouter1 で改ざんが行われたことを検出できる。

しかし、この方法では Client のハッシュを取り、これと比較できるようにしておけばよく、中継サーバごとにハッシュを保管するのは若干無駄があると考えられる。LogRouter1 および LogRouter2 は VLSD を用いたものではないので、多くのログを保存することが出来ない。

(d) 中間ハッシュを含めて Server へ転送

モデル(c)では、Client のハッシュが中継サーバを含むすべてのサーバに保管されているので、どの場所でもハッシュを比較することができる。しかし、LogRouter1 および LogRouter2 は、ログを保存することを考えていないので、情報価値の観点から得策ではないと考えられる。

LogRouter1 および LogRouter2 におけるハッシュを減らしたものが(d)である。Client のハッシュと各々を比較すればよいのだから、最終的に 1 つの場所にまとめればよい。容量問題を解決した VLSD に保存していくのが適切である。ログファイルと一緒にハ

ッシュも半永久的に保存できる。図 7 に示す。

まず、Client のハッシュと Server に転送されたログのハッシュを比較する。一致した場合は、経路途中で改ざんがなかったことを表す。すなわち、これはディジタルフォレンジックに対応するログであると言える。これが一致しない場合は、Client と LogRouter1、そして LogRouter2 とを順番に比較して、改ざんされた場所を特定する。

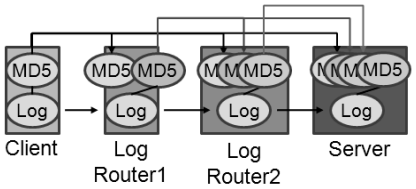

図 6　ハッシュファイルの流れ(c)
Figure 6 Flow of the hash (c)

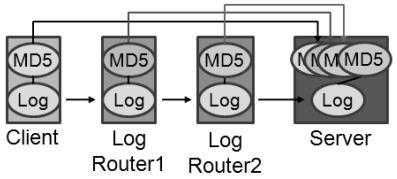

図 7　ハッシュファイルの流れ(d)
Figure 7 Flow of the hash (d)

4. 評価

4.1 ログの収集とハッシュの生成

本研究の実験中に排出されたログの総量は、30 日間・非圧縮時で 6,605KB となり、これを gzip で圧縮すると 1,581KB とおよそ 1/4 になった。また、生ログの 1 日平均は 220KB で、ログファイルの最大サイズは 2,239KB となった。ここで、例えば 512 台の PC 教室で、最悪のケースを考えた場合、年間でおよそ 413GB 必要な計算になる。5 年リースを考えると、2.1TB 必要になる。また、本大学の 4 キャンパスの PC 教室を考えると、年間でおよそ 1.65TB 必要になる。

この最大のログに対して、レコードごとに MD5 によるハッシュを作成した。このログは、レコード数が 25.652 行ある。その容量は 847KB となった。ログファイル自体と比較すると、およそ 4 割程度の大きさになる。すなわち、ログとハッシュを一緒に保管するには、およそ 1.4 倍の保存領域が必要になる。

先ほどと同様に、最大のログに対するハッシュの保存容量を計算すると、年間で 156GB 必要になる。5 年リースを考えれば、781GB 必要になる。生ログと合計すると、2.9TB もの保存容量が必要になる。VLSD ツールキットを用いて、大容量ストレージの試作に成功しているので、保存容量については問題にはならない。

4.2 ログ用ストレージの構築

VLSD ツールキットを用いて、25 台のサーバから 500GB ずつ集め RAID6 を構成し、およそ 11.2TB の仮想ストレージを作成した。ファイルシステムに XFS を採用したので理論上、8EB まで実現可能である。これを Windows からネットワークドライブとして割り当てる。図 8 に実行結果を示す。

ストレージに VLSD を用いた場合のコストについて考える。PC 教室などの環境では、数百台の PC から構成されており、大規模ストレージが必要になる。通常、信頼性や管理の容易さから数 TB の高価なファイルサーバを導入することが多い。

ここで、それぞれのサーバに 500GB の HDD を増設したと考え、その費用を計算する。HDD 単価を 1 万円と仮定すると、25 台で 25 万円になる。24 時間稼働したサーバの 1 ヶ月間の電気使用量はおよそ 149kWh で、電気代は 3,580 円になる。先の例のサーバ 25 台では、年間およそ 108 万円となる。ただし、VLSD を実装する以前から動作させていると考えれば、電気使用量のコストについては無視できると考えられる。本大学でファイルサーバを見積もる場合、60TB で 2.5 億円にもなり、容量差はあるが、これまでコスト差があると、分散型のものも十分に使用価値があると考えられる。

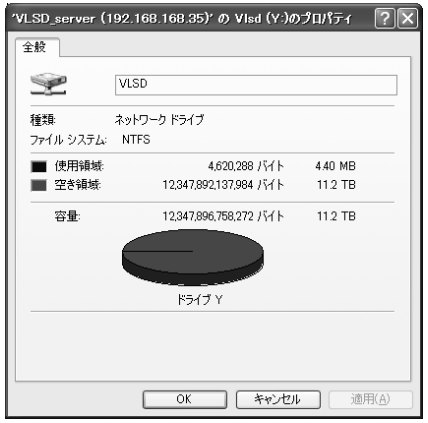

図 8　11.2TB のストレージ
Figure 8 Storage of 11.2 TB

4.3 ログの保証

本論第 3 章における考察を行う。モデル(a)およびモデル(b)は、改ざんや欠損を検出できるが十分ではない。どのマシンによって改ざんがされたのかを知るには、すべてのマシンにおけるハッシュを得る必要がある。モデル(c)およびモデル(d)では、Server においてそれを実現する。モデル(c)とモデル(d)は、中継サーバにハッシュを残すかどうかの違いだけである。前述の通り、ログは保存領域を圧迫していく。そのため、

LogRouter1 や LogRouter2 などの VLSD を用いない一般的なサーバでは、すぐに容量の限界に達することが考えられる。このことから、モデル(d)を元にした管理方法を実現するのが有効であると考える。

ログに対して MD5 によるハッシュを作成すると 128 ビットすなわち 16 文字の英数字の羅列が出力される。Syslog が出力するログの 1 つのレコードがおよそ 100 バイトとするとかなり少ない。ここで、レコード毎にハッシュを取った場合を考える。1 つのログファイルに対して実行した場合とレコード毎に実行した場合とでは大きく違う。記録されたログのレコードの数だけハッシュを計算する必要がある。レコード毎にハッシュを取れば、ファイル毎のものに比べ、精度が増す。一部分のハッシュのみが変わることから、改ざんされたログのレコードだけ、あるいはその複数行だけ検出することができる。図 9 に示す。これによりディジタルフォレンジックに耐えうる形となる。

Oct 21 12:25:24 localhost syslogd 1.4.1: restart.	b51c5f4d6034cc0dd4ed24cffe182ba0
Oct 21 12:34:09 localhost dhclient: DHCPREQUEST on et	22fab933b7cb6c1704ff1d142c274349
Oct 21 12:34:09 localhost dhclient: DHCPACK from 192.	03c34b234f7864adfa8976ac7ca7c577
Oct 21 12:34:09 localhost dhclient: bound to 192.168.	9f41aae7d635a796544dc4ef1a1775e6
Oct 21 12:48:12 localhost dhclient: DHCPREQUEST on et	ff4aad3fa7c1182dfe03e294852bacbb
Oct 21 12:48:12 localhost dhclient: DHCPACK from 192.	4825f770887b80b8fe6cdc8e19c2e0fd
Oct 21 12:48:12 localhost dhclient: bound to 192.168.	9188c6dca457f71a1e152d2318c94f36
Oct 21 12:58:21 localhost scim-bridge: The lockfile	5e5ec741434b375791f807ad877f3f3b
Oct 21 12:58:21 localhost scim-bridge: Cleanup, done.	cdc948e446a63d59e85254b331cce52a
Oct 21 12:58:22 localhost gconfd (root-2552): 終了し	686e634643b652f91cf00fc41ff544de
Oct 21 12:58:23 localhost gdm[2351]: Master halting..	be31c187f54db65dc3eb8f209f88a549
Oct 21 12:58:25 localhost shutdown[2351]: shutting d	df41878738e9b7f0a286551a7864fb9a
Oct 21 12:58:28 localhost smartd[2340]: smartd recei	704088a5217c57c80bb41bc17d6517a9
Oct 21 12:58:28 localhost smartd[2340]: smartd is ex	6a05880c8d109fb188e4db34f011267f

図 9　レコード毎のハッシュ
Figure 9 the Hash created each Record

MD5 によるハッシュは、その脆弱性が発見されている。異なるドキュメントから同一のハッシュ値が求められるというものである。すなわち、ハッシュの一意性が失われたことになる。しかし、ログとしての意味を持ち、かつハッシュ値を同じにするのは難しいと考える。例えば、ある特定のログのレコードを削除する。その分、意味を持ち、正当なレコードを追加する必要があるからである。また、ユーザ名や実行時間などレコードの一部分のみの改変は、前述と同様に難しい。ログのレコードごとのハッシュ値を求めれば、改ざんしにくくなるログが得られる。

クライアントマシンからのログを転送する際、コネクション数にも注意するべきである。コネクション数はクライアントマシンの数に比例する。もしこれが膨大で転送に支障があり、ログが正常に転送されていないと判断されると、改ざんと同様に有効性はなくなると考えられる。

例えば、1Mbps の回線でログを転送した場合、1 日平均のログを 220KB とすると、1.76 秒かかる。1 秒間に 1 万トランザクション処理できると仮定すると、およそ 5,600 クライアントに対応できる計算になる。本大学の PC 教室のクライアントマシンは、500 台なので、大学の PC 教室全体や中小規模の企業でも十分に間に合うと考えられる。

5. まとめ

本論文では、我々が開発したログ管理システムにおけるネットワーク間のログの保証方法の提案をした。ネットワーク間のログの保証は、ディジタルフォレンジックにおいて重要な要素である。法廷では限りなく 100%に近い信用を得られなければ、そのログは有効であるとは言えない。前述のとおり、ハッシュを用いれば、ネットワーク間のログの信頼性は保証される。これにより、ディジタルフォレンジックに対応するログを保存することが可能になることを示した。

謝辞

本研究は科研費基盤(C)「PC グリッドによる高信頼・高効率な分散仮想ストレージの研究(19500066)」により援助されています。

参考文献

[1]辻井 重男 他, “ディジタルフォレンジック事典”, 日科技連, pp.33-36, 2006 年 12 月
[2]上原 稔　“教育環境における仮想大規模ストレージのためのツールキット”, マルチメディア通信と分散処理ワークショップ, pp.205-210, 2006 年 11 月
[3]チャイ エリアント, 上原 稔, 森 秀樹 ”PC 教室のための仮想的大規模ストレージの構築”, マルチメディア、分散、協調とモバイル（DICOMO2007）シンポジウム論文集, pp.617-622, 2007 年 7 月
[4]上原 稔　“仮想大規模ストレージにおけるセキュリティ”, 情報処理学会研究報告書, pp.61-66, 2007 年 11 月
[5]Akihiro.T, Minoru.U, : ”A Log Management System for Internal Control”, NBiS2009, (to be appeared)