5X-05

仮想計算機モニタを用いた外部記憶装置の監視分析システム ～ ブロックハッシュを用いた分析機能の試作と評価

吉田　光輝†　　高瀬　誉†　平野　学†

独立行政法人国立高等専門学校機構

豊田工業高等専門学校　情報工学科†

１．はじめに

コンピュータに関係する様々な事件や紛争が増え，それらの証拠となる電子データの取り扱いが重要な課題となっている．それらの紛争解決の手段としてデジタル・フォレンジックという用語が用いられ，「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」として定義されている[1]．

外部記憶装置に対するデジタル・フォレンジックは，証拠となるＨＤＤを回収，またはデータを物理コピーしたのち，そのデータを対象に解析をおこなう．外部記憶装置の解析手法には大きく分けて二通りの方法があり，ひとつは EnCase のようなフォレンジック・ツールで採用されているファイルシステムのメタデータを解釈して解析する方法である．もうひとつはカービング（Carving）と呼ばれる手法で，ファイルシステムのメタデータを使わずに，セクタに含まれるデータのみで解析する手法である[3]．本研究では後者のカービングによる解析手法に着目する．前者はファイルシステムの情報を活用できる利点があるが，木構造を読み込むため並列処理が難しいという欠点がある．後者はファイルシステムの情報を使わないため，各セクタに含まれるデータの意味づけが難しいという欠点があるが，ファイルシステムに依存しない処理が可能で，並列処理が容易である利点がある．二つの方法は相互に補完できる関係にある．

一般的なフォレンジック・ツールでは，National Software Reference Library (NSRL)の Reference Data Set (RDS) のような既知の問題のないファイルのハッシュ値のリストを使って，解析対象のファイルを絞ることで効率化を図る．Garfinkel らは，この伝統的なファイルのハッシュ値を用いた検出方法の代わりに，セクタのハッシュ値を計算して，目的ファイルを検出する方法を提案した [2][3]．彼らのシステムは，解析対象となるディスクの各セクタのハッシュ値を計算していき，あらかじめ作っておいたセクタのハッシュ値が含まれるインデックスに対して，そのハッシュ値を照合することでファイルを検出する．ファイルの一部が削除された場合，伝統的なファイルのハッシュ値を用いた手法では該当ファイルを検出できないが，セクタのハッシュ値を使う方法であれば，ディスクに一部のセクタが残っていれば検出できる．

セクタのハッシュ値を用いたファイルの検出は，「オリジナルのファイル以外には含まれないセクタ」の存在が鍵となる．Garfinkel らはこの特性を持つセクタを distinct block と名付けた（今後，本稿では「セクタ」という用語はドライブイメージから得られたデータを指し，「ブロック」はファイルシステムまたはファイルから得られたデータとして区別する）．彼らの実験によると，既知のデータセットである nps-2009-domexusers (Windows XP SP2 のディスクイメージ) の distinct sectors は 56% であった[2]（システムには重複ファイルが多く存在しているが，Govdocs のような人間が作成した文書を集めたデータセットでは 98.93% が distinct blocks であった[3]）本稿では，最新の Windows 8.1 (NTFS), MacOS 10.9 (HFS+), CentOS 6.5 Linux (Ext4) に対して distinct sectors の割合を調査した結果を報告する．この調査のためハードディスクの raw イメージファイルを作成し，Hadoop で並列処理した．本稿では上記の実験結果からセクタのハッシュ値を用いたファイル検出手法の有用性を検討する．

２．仮想計算機モニタでの外部記憶装置の監視

本稿で検証するセクタのハッシュ値を用いた方式は，図１に示す仮想計算機モニタを用いた外部記憶装置の監視分析システムで得られた履歴データに対して適用することを想定している．我々は図１に示すような仮想計算機モニタ Xen（準仮想化）の Blktap 機構を利用して，外部記憶装置への書き込みをブロック単位で時刻情報とともに保全する仮想デバイスドライバと，解析時に任意時刻のディスクを復元できる仮想デバイスドライバを

Implementation and Evaluation of a Block-hash Based Analysis System for Surveillance Mechanisms of Virtualized Storage Devices

† Koki Yoshida, Hayate Takase, and Manabu Hirano, Department of Information and Computer Engineering, National Institute of Technology, Toyota College

開発した[4]．この仮想デバイスドライバで得られた履歴データは，実績のあるフォレンジック・ソフトウェアで解析できる．しかしながら，どの時刻のディスクを復元すればよいのかを探し出す機能や，時系列で問題となるファイルを絞り込んでいくといった機能は，時系列での履歴保全を前提としていない既存ツールの守備範囲外である．そこで，本稿で検証しているセクタ単位のハッシュ値によるファイル検出方式を採用することで，この監視システムから得られたデータから，特定ファイル（漏えいしたファイル等）の書き込み時刻を発見し，その後は既存の実績のあるツールで解析する枠組みを検討する．

３．実験方法

本稿ではディスクイメージの中で distinct であるセクタが占める割合を調査した．調査はゼロ消去したハードディスクに Windows 8.1 (x64) と Microsoft Office 2013 をインストールしたディスクイメージ，MacOS 10.9 と Microsoft Office 2011 をインストールしたディスクイメージ，CentOS 6.5 (x64)を Basic Server でインストールしたディスクイメージの３種類に対して実施した．まず，文献[2]と同様の方法で，各ディスクイメージからすべて 0x00 で埋められているセクタ，0xFF で埋められているセクタ，その他の同一バイトが連続して 512 回出現するセクタ（全部で 256 通り）を除外する．その後，残ったセクタのハッシュ値を計算して，他のセクタと比較をおこなうことで，distinct なセクタの割合を求めた．計算には Hadoop クラスタを利用した．セクタ毎のバイナリデータは SequenceFile 形式に変換してから，（セクタ番号，セクタに含まれるデータ）のキーバリュー・ストアとして処理した．

４．実験結果

三種類のディスクイメージに対してセクタの distinctness を調べた結果を表１に示す．Total sectors はディスクイメージ全体のセクタ数，Removed sectors は 512 バイトが全て 0x00 や 0xFF で埋められているセクタの数である．Data sectors は残りの解析対象として意味のあるデータが含まれるセクタの数である．これにはファイルシステムやパーティションに関するメタデータも含まれる．Singleton（一人っ子）は一回だけしか出現しないセクタの数である．同様に Twin（双子）は２回，Triplet（三つ子）は３回同じ内容が出現するセクタの数である．

５．考察とまとめ

Singleton から Triplet までを合わせたセクタ数のデータセクタ全体に対する割合は Windows 8.1 で 92.4%，MacOS 10.9 で 95.0%，CentOS 6.5 で 93.5% であった．この結果は全データセクタの９割以上が，せいぜい３回程度の重複に収まる程度の distinctness を持つセクタから構成されていることを示している．MacOS では Singleton が 88.5%と他の OS と比べて多い結果を得た．Windows 8.1 と CentOS 6.5 で Singleton のセクタが少ないのは同一ファイルのコピーが存在しているからと考えられるが，この点については追加の調査が必要である．本稿ではセクタのハッシュ値の distinctness を調査したが，実際に検索で使うためには更なる有用性の検証が必要である．

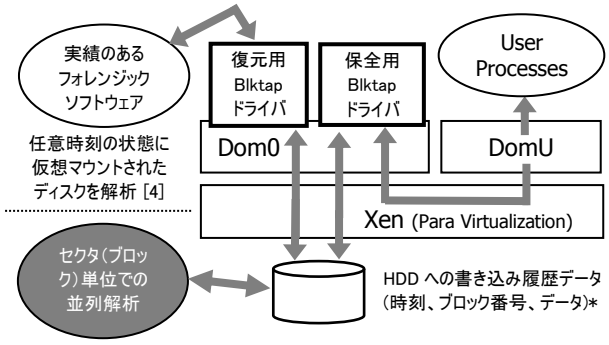

図１　外部記憶装置の書き込み監視分析システム

表１　セクタ単位での distinctness の調査結果

No. of sectors	Windows 8.1 (NTFS)	MacOS 10.9 (HFS+)	CentOS 6.5 Linux (Ext4)
Total sectors	195,371,568	156,301,488	195,371,568
Removed sectors	170,738,500	136,997,813	169,699,709
Data sectors	24,633,068	19,303,675	25,671,859
Singleton	12,562,375 (51.0%)	17,090,144 (88.5%)	15,718,575 (61.2%)
Twin	8,067,594 (32.7%)	1,141,746 (5.9%)	6,767,062 (26.3%)
Triplet	2,160,450 (8.7%)	121,257 (0.6%)	1,542,480 (6.0%)

参考文献

[1] 辻井重男監修，佐々木良一他著，デジタル・フォレンジック事典，日科技連出版社，2006.
[2] Simson Garfinkel, Alex Nelson, Douglas White, Vassil Roussev, Using purpose-built functions and block hashes to enable small block and sub-file forensics, Digital Investigation, Vol. 7, pp.S13-S23, 2010.
[3] Young, J., Foster, K., Garfinkel, S. and Fairbanks, K., Distinct Sector Hashes for Target File Detection, IEEE Computer, vol.45, no.12, pp.28-35, 2012.
[4] 小川拡，平野学，仮想計算機モニタを利用したコンピュータフォレンジックスのための補助記憶装置のデータの保全と回復のシステム，研究報告コンピュータセキュリティ，2013-CSEC-60(1), 1-6, 2013.