HTML5 WebStorage 生成物のメインメモリイメージからの取得

松本 晋一†‡　　櫻井 幸一†‡

†公益財団法人 九州先端科学技術研究所
814-0001　福岡県福岡市早良区百道浜 2-1-22
smatsumoto@isit.or.jp

‡九州大学 システム情報科学研究院
819-0395 福岡県福岡市西区元岡 744
sakurai@cscs.kyushu-u.ac.jp

あらまし WebブラウザはHTML5言語を用いて記述されたアプリケーションの実行環境として成長し つつあり，当該技術は各種スマートフォンプラットフォームのフレームワークの中核部分を占めている．この状況においてWebブラウザのHTML5固有の属性に関する生成物を抽出できれば，モバイルフォレンジクスにおいて活用できるものと予想される．本稿では，Windows OS上のWebブラウザを対象として，端末のメモリイメージから各種Webブラ ウザのWebStorage属性の記録内容をエビデンスとして抽出する実験を行った．実験の結果，各種Webブラウザについてエビデンスの抽出に成功し，また記録フォーマットを特定した．

Acquisition of HTML5 WebStorage artifacts from main memory

Shinichi Matsumoto†‡　　Kouichi Sakurai†‡

†Institute of Systems, Information Technologies and Nanotechnologies.
2-1-22, Momochihama, Sawaraku, Fukuoka-shi, Fukuoka 814-0001, JAPAN
smatsumoto@isit.or.jp
‡Department of Informatics, Kyushu University
744 Motooka, Nishi-ku, Fukuoka 81900395, JAPAN
sakurai@inf.kyushu-u.ac.jp

Abstract Nowadays, Web browser is a growing as a platform for applications based on HTML5 technology. Furthermore, variety of smartphone platforms support web technology as a foundation of application execution framework. These situations may lead to a higher importance of forensic investigations on artifacts within the web browser HTML5 specific attributes as evidences in mobile forensics. In this paper, we experimented to acquire the memory image within terminal running Windows OS and extract the webStorage stored data as an evidence of browsing activity. The results of experiment elucidated the formats of evidences left by webStorage attributes.

1　はじめに

1.1　背景

コンピュータやインターネットの普及に伴い，あらゆるデータがデジタル化して流通，記録されるようになっている現代，個人および組織の様々な活動がコンピュータやインターネット上で行われるようになっている．その結果，犯罪行為などもこれらの上で行われるようになり，法的な対処が急がれている．

辻井[1]らは，デジタル・フォレンジックが重要となった背景として，

- デジタル化の進展により，デジタルデータの破壊や改竄，漏洩の影響が増大していること．
- デジタルデータに対する不正や犯罪が増加し，また不正アクセス禁止法や個人情報保護法などといった法律の整備により刑事訴訟や民事訴訟が増大し，また今後も増加が予想されること．

を挙げている．

不正アクセス禁止法や個人情報保護法の制定，また

企業内の不正行為に関する訴訟や，国際的なビジネス上の係争にも，デジタルデータを証拠として採用する事例が存在する．

このような民事訴訟に関しては，起訴を行う側だけでなく起訴される側(起訴される事に備える側)も証拠を整備する必要がある．

1.2　モチベーション

デジタルフォレンジクスをモバイルデバイスに対して実施する事例は増加している．モバイルデバイスのフォレンジクスに対する要求は，市場の移り変わりとともにフィーチャフォンからスマートフォンへと以降している．スマートフォンの第一の特徴は，種々のアプリケーションを端末へインストール可能なことであるが，アプリケーション実効環境はスマートフォンプラットフォームによって異なる．

一方，スマートフォンプラットフォームはアプリケーション実行環境を備える一方で，Web ブラウザフレームワークを備える．HTML の新版である HTML5 が普及することにより，スマートフォンプラットフォームは Web ブラウザフレームワークをプラットフォームとするよう移行するこよが予想される．一方，スマートフォン以外にも Web ブラウザフレームワークをアプリケーション実行環境とする OS が広まりつつあり，この予想を強化している．

このような予想のもと，HTML5 をサポートする Web ブラウザフレームワークの生成物を対象とするデジタルフォレンジクス技術の確立が求められる．

1.3　HTML5 生成物のフォレンジクス

HTML 5 [2]においては，Web アプリケーション開発を助けるために様々な機能が追加されており，次章で述べる種々の拡張により，Web ブラウザ側で，地理情報(Geolocation)[3]などを扱える一方，従来の cookie[4]よりも Web ブラウザ側で大幅に制約の少ない利用法が可能な WebStorage[5]により Web アプリケーション次第で種々の情報を扱うことが可能である．考えられる用途としては，

- メールアプリケーション内での，メール本文および送信元，宛先などのヘッダ情報．
- フォトアルバムアプリケーション内での，写真/画像データや撮影時刻，撮影場所などのメタデータ情報．
- オフィスアプリケーション内での，文書データや編集履歴に関する情報．
- ゲームアプリケーション内でのアカウント情報やプレイ履歴情報．

モバイルデバイスのフォレンジックにおいては，これらのデータを取り扱い可能とすることで，適用可能な領域が大幅に拡張されることが想定される．

1.4　関連研究

Web ブラウザを対象とするデジタルフォレン

ジクス研究[6][7]は，Web ブラウザが生成するログファイルやブラウザ動作時の内部情報を記憶する SQLite DB を調査対象としている．

一方，Donny[8]らは様々な Web ブラウザ(Mozilla Firefox, Google Chrome, Microsoft Internet Explorer, Apple Safari)について，以下に示すプライベート閲覧モードのエビデンスの獲得を試みている．

- Microsoft Internet Explorer: InPrivate Browsing
- Google Chrome: Incognito mode
- Mozilla Firefox: Private Browsing
- Apple Safari: Private Browsing

更に，彼らは以下に示すポータブル Web ブラウザについても実験を行っている．

- Mozilla Firefox Portable
- Google Chrome Portable
- Opera Portable

Mulazzani ら[9]，及び Satvat ら[10]も同様に Web ブラウザのプライベート閲覧モードのエビデンス獲得を試みている．Satvat らはプライベート閲覧モードに対する DNS および主記憶，ファイルタイムスタンプ，Index.dat ファイル，ブラウザ拡張などを含む攻撃手法について検証している．また Aditya ら[11]は Web ブラウザのプライベート閲覧モードの残すエビデンスについてのメモリフォレンジクス実験を行っている．彼らはプライベート閲覧モードの種々の脆弱性について明らかにしている．

彼らは主に主記憶イメージの獲得と解析について実験を行っている．取得に成功したエビデンスはブラウザにより異なるが，以下について成功している．

- プライベート閲覧モードにて閲覧したことを示す情報
- 閲覧履歴
- 画像データ
- ユーザ名/e-mail アカウント

著者ら[12]は Web ブラウザの HTML5 webStorage 記憶内容に対するフォレンジクスについて，仮想マシン内の Windows OS を対象に実験を行っている．

1.5 課題

HTML5 をサポートする Web ブラウザを対象とするデジタルフォレンジクスの実現において，特に HTML5 において新たに追加されたクライアントストレージである WebStorage は，ブラウザ上で実行されるアプリケーションが残すエビデンスを調査するのに有効である．

本研究は，WebStorage の残すエビデンスをメモリフォレンジクスの技法を用いて探索することを目指したものである．現在市場で主要な Web ブラウザは複数種類存在し，それぞれにおいて WebStorage の記憶内容がメインメモリから取得可能か否か，また可能であるならば，どのような形式で収容されているかを明らかにすることを目標としている．

1.6 結果

実験の結果の概略を表 1 に示す．現在市場において主要な Web ブラウザ三種類(Mozilla Firefox, Microsoft IE, Google Chrome)について，WebStorage の内容取得に成功した．ただし一部のブラウザについては WebStorage を構成する要素全ての取得には成功していないという結果になっている．

1.7 既存研究との比較

Donny[8]らの研究は Web ブラウザの残すエビデンスをメモリフォレンジクスの技法を用いて取り出すことに成功している．ただし当該研究は HTML5 固有の情報は対象としていない．Aditya[11]ら，Mulazzani ら[9]，および Satvat[10]らの研究は同様にメモリフォレンジ

表 1 実験結果概要まとめ

ブラウザ	ストレージ	origin	key	value
Firefox	ls	✓	✓	✓
	ss	✓	✓	✓
MS IE	ls	N/A	✓	✓
	ss	N/A	✓	✓
Chrome	ls	N/A	✓	✓
	ss	N/A	✓	✓

クスの技法を用いているが，HTML5 固有情報は対象としていない．

著者らの研究[13]は，HTML5 固有機能の残したエビデンスの取得を目指したものであるが，仮想マシン上でのエミュレーションを行ったものであり，実験としては端緒に過ぎないものであった．本研究はこれをベア環境で行い，エビデンス抽出を目指したものである．一方，著者らによる別研究[12]は，HTML5 固有機能の残すエビデンスの，ファイルシステムからの獲得を行い，またエビデンス調査のための構造化ツールの設計，実装を行ったものである．

2 HTML 5 概要

HTML 5 は現在 W3C にて標準策定中であり，2014 年中の制定が予定されている．

2.1 HTML 5 の特徴的要素

Pieters ら[14]によって述べられている，HTML5 の旧仕様との特徴的な違いとしては，以下がある．

- media (audio, video)
- canvas
- embed
- menu
- command
- keygen

また同文献で述べられている HTML5 のブラウザ機能としては以下がある．

- カスタムスキーマとコンテンツハンドラ
- 編集可能なコンテンツ
- ドラッグ&ドロップ
- 履歴インタフェース

2.2 WebStorage

HTML 5 において追加された特徴的な機能として WebStorage がある．従来 Web ブラウザ内に永続的記憶を設ける機構としては cookie が存在したが，WebStorage はこれに比して以下の特徴を持つ．

- 大容量(cookie の 4kbytes に対し，一般にオリジン[15]毎に 5Mbytes)
- ブラウザ側からサーバ側へのデータ送信は，明示的に指示しない限り行わない．
- 有効期限がない

また，Web ブラウザに対するアドオンの機能を用いるものとして，クライアント側への記憶機能として Flash Cookie を用いる場合がある．Soltani[16]ら，また Ayenson[17]らは Web サイトにおける Flash Cookie の使用状況についてサーベイを行っている．

クライアント側の記憶機能としての WebStorage には，記憶期間の違いにより以下の 2 種類が存在する．

- sessionStorage　ブラウジングセッション内でのみ保持される記憶．例えばブラウザが終了すると，記憶内容は失われる．
- localStorage　ブラウザ内に永続的に保存される．ブラウザ終了後，再度起動してアクセス可能．

両者は記憶期間の定義が異なり，ブラウザ内では異なる実装により管理されていることが予想される．

3 実験概要

3.1 実験目的

メモリフォレンジクスはコンピュータフォレンジクスにおいても特に注目を集めている分野である．ハードディスクドライブやフラッシュメモリといった，コンピュータシステム内の永続的記憶を扱うものとは異なり，メモリフォレンジクスは揮発性の主記憶を調査対象とする．メモリフォレンジクスの重要性については Amari ら[18]が論じている．また Web ブラウザの閲覧履歴に対するメモリフォレンジクスについては Waksman ら[19]の研究がある．

HTML5 では，Web ブラウザをアプリケーションの実効環境となるウプラットフォーム本実験では，HTML 5 において，デジタル・フォレンジックの対象として特に有効と考えられる WebStorage の内容を，メモリイメージから取得する方法を探る．

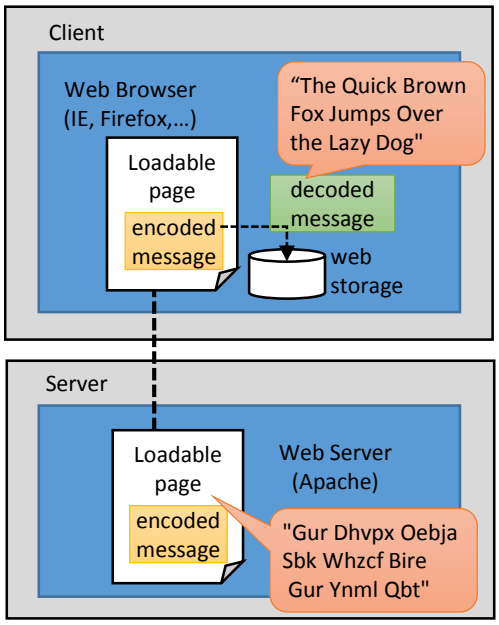

図 1 実験システム概要

3.2 実験環境

本稿における実験に用いた環境を以下に示す. 実験システムの構成は図 1 に示す. 実験においては市場シェア上位を占める三種類の Web ブラウザ Mozilla Firefox, Google Chrome および Microsoft Internet Explorer を対象とした.

- クライアント
  - OS Microsoft Windows 7 Ultimate(64bit)
  - CPU AMD Phenom Ⅱ X4 905e Processor
  - Main memory 4GB(512MBをビデオ RAM に割り当て)
- Web ブラウザ
  - Mozilla Firefox 25.0.1
  - Google Chrome 31.0.1650.63m
  - Microsoft Internet Explorer 10.9.9200.16750
- ツール
  - AccessData FTKImager 3.1.4.6(主記憶イメージ取得用)[20]
  - Memtest86+ 5.01(主記憶クリア用)

3.3 実験手順概要

実験は, 以下の手順に従い実施した.

1) クライアントを Memtest86+を用いて起動し, メモリテストを実施する.
2) クライアントをシャットダウンし, Windows OS で再起動する.
3) Windows OS 起動を確認後, FTK imager を起動する.
4) 試験対象とする Web ブラウザを起動する.
5) Web ブラウザで試験ページをロードする.
6) Web ブラウザを終了する.
7) FTK imager を使用しメモリイメージをメモリイメージファイルに保存する.
8) 外部 HDD を接続し, 前ステップで保存したイメージファイルを退避する.
9) クライアントをシャットダウンする.

今回, 第一項におけるメモリテストは, 前の実験によりメモリに保持され, 消去されていない値を完全に消去することを目的としている.

4 実験結果

本章ではメモリイメージ取得の結果を Web ブラウザ毎, また localStorage 及び session Storage の別毎に示す.

4.1 Mozilla Firefox

4.1.1. localStorage

Firefox ブラウザの localStorage について実験を行った結果, localStorage 内容のエビデンス取り出しに成功した. ダンプ結果の抜粋は以下の通りとなる.

```
012a1ffa0|00 00 00 00 00 00 00 00 00 00 54 01 06 35 2b 63
012a1ffb0|00 00 36 2e 30 2e 38 36 31 2e 32 39 31 2e 3a 68
012a1ffc0|74 74 70 3a 38 30 6c 6f 63 61 6c 53 74 6f 72 61
012a1ffd0|67 65 4b 65 79 54 68 65 20 51 75 69 63 6b 20 42
012a1ffe0|72 6f 77 6e 20 46 6f 78 20 4a 75 6d 70 73 20 4f
012a1fff0|76 65 72 20 54 68 65 20 4c 61 7a 79 20 44 6f 67
```

上記アドレスに対応するキャラクタダンプの結果は以下の通りである.

```
012a1ffa0|..........T..5+c..6.0.861.291.:h
012a1ffc0|ttp:80localStorageKeyThe Quick B
```

012a1ffe0|rown Fox Jumps Over The Lazy Dog

Origin と key および value を表す文字列は連結されている．更に，origin のアドレス部は逆順になっている．当該オリジンを元に key, value を検索可能である．

4.1.2. sessionStorage

Firefox の sessionStorage について実験を行った結果，sessionStorage 内容のエビデンス取り出しに成功した．ダンプ結果の抜粋は以下の通りとなる．

```
10cb411b0|2e 30 2e 36 2f 66 61 76 69 63 6f 6e 2e 69 63 6f
10cb411c0|22 2c 22 69 6e 64 65 78 22 3a 32 2c 22 73 74 6f
10cb411d0|72 61 67 65 22 3a 7b 22 68 74 74 70 3a 2f 2f 31
10cb411e0|39 32 2e 31 36 38 2e 30 2e 36 22 3a 7b 22 73 65
10cb411f0|73 73 69 6f 6e 53 74 6f 72 61 67 65 4b 65 79 22
10cb41200|3a 22 54 68 65 20 51 75 69 63 6b 20 42 72 6f 77
10cb41210|6e 20 46 6f 78 20 4a 75 6d 70 73 20 4f 76 65 72
10cb41220|20 54 68 65 20 4c 61 7a 79 20 44 6f 67 22 7d 7d
10cb41230|7d 5d 2c 22 73 65 6c 65 63 74 65 64 22 3a 31 2c
10cb41240|22 5f 63 6c 6f 73 65 64 54 61 62 73 22 3a 5b 5d
```

上記アドレスに対応するキャラクタダンプの結果は以下の通りである．

```
10cb411b0|.0.6/favicon.ico","index":2,"sto
10cb411d0|rage":{"http://192.168.0.6":{"se
10cb411f0|ssionStorageKey":"The Quick Brow
10cb41210|n Fox Jumps Over The Lazy Dog"}}
10cb41230|}],"selected":1,"_closedTabs":[]
```

Origin, key, value は JSON 形式で収容されていることがわかる．ツリー構造として，key および value が origin の下に存在する．

4.2. Microsoft Internet Explorer

4.2.1. localStorage

Internet Explorer について，locaoStorage の内容をメモリイメージから取り出すことに成功した．以下にメモリイメージの抜粋をキャラクタダンプしたものを示す．

```
03f8a55a0|........<root><item name="localS
03f8a55c0|torageKey" value="The Quick Brow
03f8a55e0|n Fox Jumps Over The Lazy Dog" l
03f8a5600|time="620143472" htime="30355593
03f8a5620|" /></root>..........G.........
```

Key および value が DOM 形式で収容されていることがわかる．時間情報が同一エレメント内に含まれている．

4.2.2. sessionStorage

Microsoft Internet Explorer 実行後のメモリイメージ内から sessionStorage 記憶のキー情報，データ情報の取得に成功した．以下にメモリイメージのキャラクタダンプ内容を示す．

```
06c5a2d30|............1...sessionStorageKe
06c5a2d50|y..........V........T.h.e. .Q.u.
06c5a2d70|i.c.k. .B.r.o.w.n. .F.o.x. .J.u.
06c5a2d90|m.p.s. .O.v.e.r. .T.h.e. .L.a.z.
06c5a2db0|y. .D.o.g..7.namespace-DC3AF5DE_
06c5a2dd0|BF73_458A_AFEB_8ED2F272D18A-....
```

メモリイメージ内容は localStorage に対応するものとは異なる．Value は UTF-16 でエンコードされている．

4.3. Google Chrome

4.3.1. localStorage

Chrome の localStorage 内容をメモリイメージから取得することに成功した．以下にメモリイメージのキャラクタダンプの結果を示す．

```
0370cc710|................x..I..l.o.c.a.l.
0370cc730|S.t.o.r.a.g.e.K.e.y.T.h.e. .Q.u.
0370cc750|i.c.k. .B.r.o.w.n. .F.o.x. .J.u.
0370cc770|m.p.s. .O.v.e.r. .T.h.e. .L.a.z.
0370cc790|y. .D.o.g.d..!..l.s.K.e.y.T.h.e.
0370cc7b0| .Q.u.i.c.k. .B.r.o.w.n. .F.o.x.
0370cc7d0| .J.u.m.p.s. .O.v.e.r. .T.h.e. .
0370cc7f0|L.a.z.y. .D.o.g................
```

Internet Explorer の sessionStorage のメモリダンプイメージ同様に，value が UTF-16 エンコードされていることがわかる．一方，Internet Explorer とは対照的に，key も UTF-16 エンコードされている．Origin をメモリイメージ上に発見することはできなかった．

4.3.2. sessionStorage

Chrome の sessionStorage についての結果は，Internet Explorer の sessionStoage についての結果と同様である．即ち，value の値は UTF-16 でエンコードされ，key の値は ASCII

表 2 実験結果詳細まとめ

Browser	storage type	Format	origin	key	value
Mozilla Firefox	localStorage	concatenated string	plain, but address part is inverted	plain format	plain format
	sessionStorage	JSON format	plain format	plain format	plain format
MS Internet Explorer	localStorage	DOM format	not observed	plain format	plain format
	sessionStorage	concatenated string	not observed	plain format	UTF-16
Google Chrome	localStorage	concatenated string	not observed	UTF-16	UTF-16
	sessionStorage	concatenated string	not observed	plain format	UTF-16

形式で，両者連結した形で収容されている．

5 まとめ

Windows 上の各種 Web ブラウザの WebStorage 使用後のメモリイメージから，記録内容の取り出しを試みた結果の表を表 2 に示す．“not observed”は，発見できなかった要素を表す．その他は発見されたフォーマットを表す．Firefox については Origin, Key, Value 全てを確認し，Internet Explorer については Key, Value について収容されていることを確認した．

これらの観察結果から，Firefox については調査者が Origin についての情報を保持していればメモリイメージ内から Key, Value の値を探索可能であることがわかる．一方 Internet Explorer, Chrome についても，調査者が Key についての情報を保持していれば，Value の値を探索可能であることが分かる．

今後は他の OS を対象とした，また他の種類のブラウザを対象とした同様の実験の実施を課題としている．

謝辞

本発表は JSPS 科研費 26330169 の助成を受けたものです．

参考文献

[1] 辻井 重男監修，特定非営利活動法人 デジタルフォレンジック研究会編集,「デジタル・フォレンジック辞典」株式会社日科技連出版社，2006 年

[2] Berjon, R., Faulkner, S., Leithead, T., Navara, E.D., O’Connor, E., Pfeiffer, S.,Hickson, I.: HTML5 a vocabulary and associated APIs for HTML and XHTML W3C candidate recommendation 31 July 2014 http://www.w3.org/TR/2014/CR-html5-20140731/

[3] Popescu, A.: Geolocation API Sepcification W3C Recommendation 24 October 2013, http://www.w3.org/TR/2013/REC-geolocation-API-20131024/

[4] Barth, A.: HTTP State Management Mechanism (2011) RFC6265.

[5] Hickson, I.: Web Storage W3C Recommendation 30 July 2013. http://www.w3.org/TR/2013/REC-webstorage-20130730/

[6] Tito, M.: Forensic analysis of the firefox 3 internet history and recovery of deleted sqlite records. Digital Investigation: The International Journal of Digital Forensics & Incident Response archive 5 (March 2009) 93–103

[7] Oh, J., Lee, S., Lee, S.: Advanced evidence collection and analysis of web browser activity. Digital Investigation: The International Journal of Digital Forensics & Incident Response archive 8 (August 2011) S62–S70

[8] Ohana, D.J. , Shashidhar, N.: Do private and portable web browsers leave incriminating evidence? a forensic analysis of residual artifacts from private and portable web browsing sessions. Security and Privacy Workshop (SPW) (May 2013) 135–142.
[9] Mulazzani, M.: New challenges in digital forensics: online storage and anonymous communication. PhD thesis, Vienna University of Technology (2014)
[10] Satvat, K., Forshaw, M., Hao, F., Toreini, E.: On the privacy of private browsing - a forensic approach. In: Data Privacy Management and Autonomous Spontaneous Security, Springer Berlin Heidelberg (2014) 380–389
[11] Aditya Mahendrakar, James Irving, S.P. In: Forensic analysis of private browsing artifacts. IEEE (2011) 197–202.
[12] 松本晋一, 櫻井幸一: Web ブラウザにおける HTML 5 固有属性のメモリからの獲得, SCIS2014 暗号と情報セキュリティシンポジウム(2014).
[13] 松本晋一, 鬼塚雄也, 川本淳平, 櫻井幸一: デジタルフォレンジクスの為の Web 閲覧履歴可視化方式の提案, 第 65 回情報処理学会コンピュータセキュリティ研究会(2014).
[14] Pieters, S.: Differences from HTML4 W3C Working Draft 28 May 2013. http://www.w3.org/TR/2013/WD-html5-diff-20130528/
[15] Barth, A.: The web origin concept. http://tools.ietf.org/html/rfc4627 (2011)
[16] Soltani, A., Canty, S., Mayo, Q., Thomas, L., Hoofnagle, C.J.:Flash Cookies and Privacy. http://ssrn.com/abstract=1446862 (2009)
[17] Ayenson, M.D., Wambach, D.J., Soltani, A., Good, N., Hoofnagle, C.J.: Flash Cookies and Privacy II: Now with HTML5 and Etag Respawning. http://ssrn.com/ abstract=1898390 (2011).
[18] Amari, K.: Techniques and tools for recovering and analyzing data from volatile memory. http://www.sans.org/reading-room/whitepapers/forensics/techniques-tools-recovering-analyzing-data-volatile-memory-33049 (2009)
[19] Waksman, A., Sethumadhavan, S.: Silencing hardware backdoors. SP '11 Proceedings of the 2011 IEEE Symposium on Security and Privacy (2011) 49–63.
[20] Computer forensics software for digital investigations, AccessData, : http://www.accessdata.com/products/digital forensics/ftk