

JPNIC・JPCERT/CC Security Seminar 2004

不正侵入の発見

株式会社アイアイジェイテクノロジー
技術開発部 セキュリティコンサルティンググループ
加藤雅彦 < masa@ijj-tech.co.jp >

- 本セッションは「ログ管理・解析」の続きとなりますので、そちらとあわせてご覧ください
- 資料中のコマンド等はLinuxをベースとしています。他のOSでも大きくは変わらないですが、実際にコマンドを使用する場合はそれぞれのOSに付属しているmanualを参照してください
- 使用するソフトウェアはオープンソースを想定しています

- ディスク解析の前提知識
- ディスク解析に必要な準備
- 解析ツール
- HDDデータ復旧の可能性
- 総合的な分析ツール
- 不正か否かの判断
- リスクの発見
- 最後に

不正侵入をどうやって
特定するか

不正発見の考え方

ディスク解析の前提知識

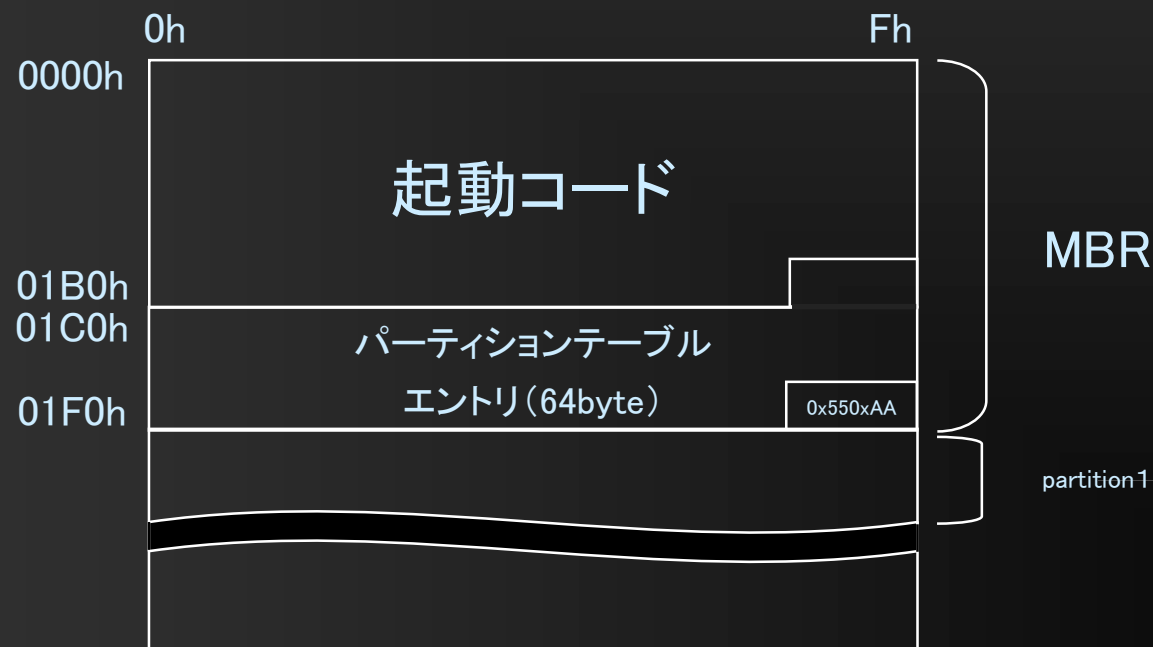
■ 簡単に前提知識のおさらい

- 不正侵入を受けたディスクの解析を行うにあたって知っておいた方がよいこと
 - パーティションの基本
 - UNIXファイルシステムの基本
 - mac time
 - ファイルが消されるとどうなるか
 - ファイル情報保存のためのベカラズ集
 - 作業の流れ

■ パーティションの基本

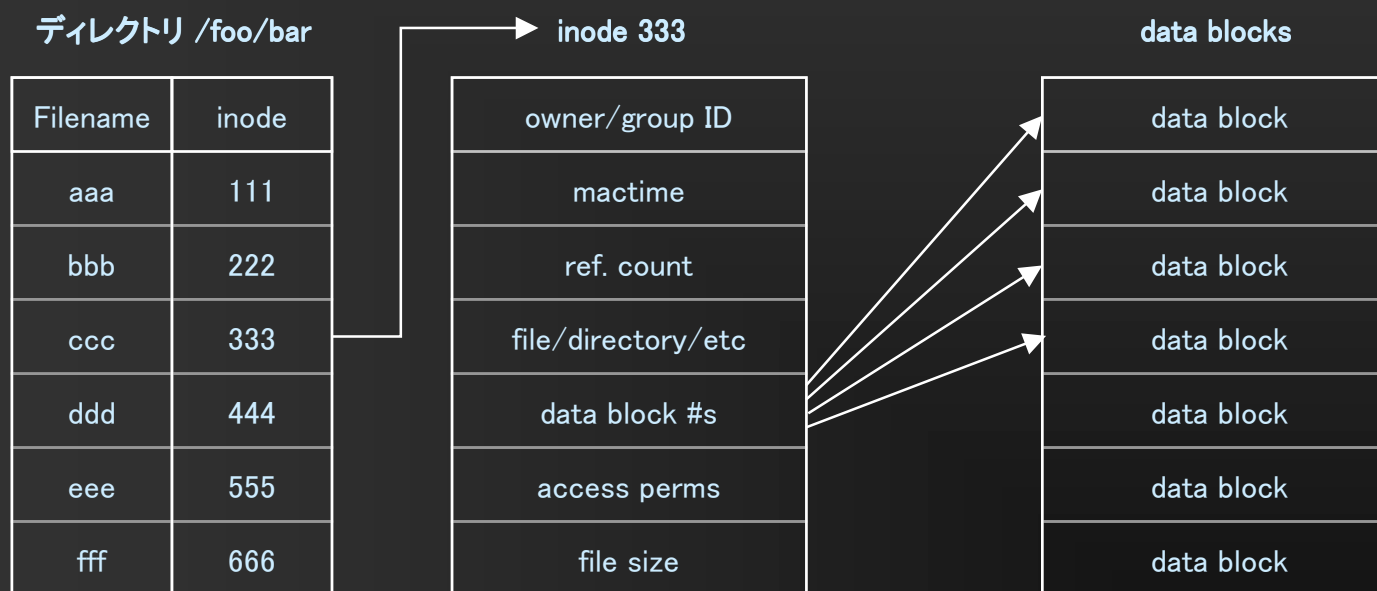
－ パーティション

- 1本のディスクを論理的に分割して見せる
- 特定の領域にデータを集中させ、管理の容易さやアクセス速度の向上を図る
- 起動時に読み込まれる領域(MBR)の中に定義されている
- プライマリが4領域確保可能なのは1パーティションあたり16byte必要で、エントリが64byte分しかないため。それ以上は拡張パーティションとなる
- FD等はこの領域にデータがかかれていない
- ディスクを複数パーティションまとめてバックアップした場合、ファイルシステムとして読み出すにはパーティション情報が必要(後述)



■ UNIXファイルシステムの基本

— inodeを使った参照方式の概念図



※サイズの大きなファイルは
間接参照される

■ mactime

- ファイルに関する操作が行われた場合に変更される時刻情報
 - mtime(time of last modification)
 - 書き込み、ディレクトリエントリの作成、削除等
 - atime(time of last access)
 - 読み込み、実行、ディレクトリエントリのルックアップ等
 - ctime(time of last status change)
 - ファイルオーナー、パーミッション変更等

- 残念ながら、必ずしも信用できるものではない
 - mactimeは上書きは可能
 - 最後に操作された時間しか残らない
 - つまり、履歴が残る訳ではない
 - 時間がたてば上書きされて古い時刻情報は消えゆく

■ ファイルが消されるとどうなるか

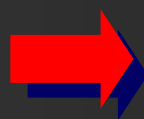
※Linuxの場合

- ディレクトリ情報
 - unallocated マークがつく
 - ファイル名とinode番号はそのまま
- ファイルの属性情報
 - unallocatedマークがつく
 - ctimeに削除時刻が書かれる
 - 参照回数が0になる
 - オーナー、パーミッション、サイズ等の情報はそのまま
- データそのもの
 - unallocatedされてそのまま残る

■ ファイルが消された直後であれば、多くの情報は残ったままとなる

■ ファイル情報保存のためのベカラズ集

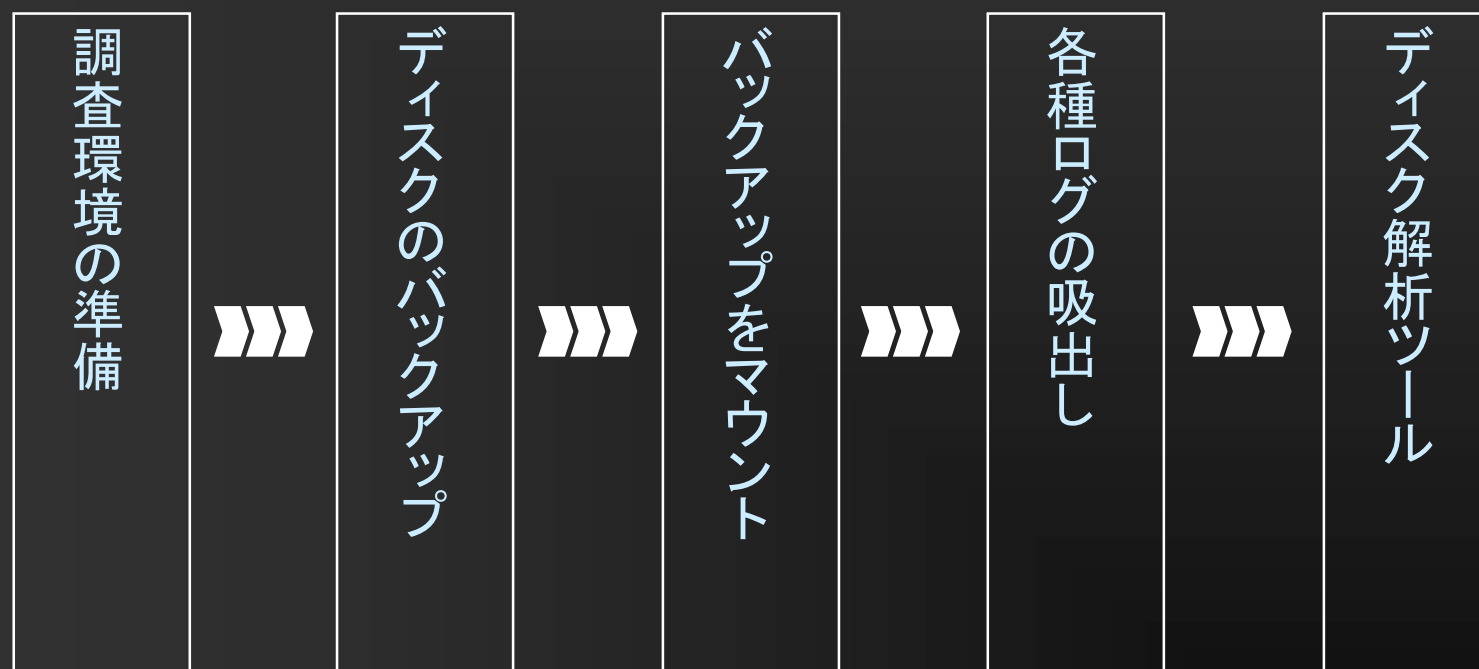
- コマンドを実行しない
 - コマンドが不正なプログラムに置き換えられている可能性あり
- リブートしない
 - 再起動時に何が起きるかわからない
- 再インストールしない
 - 侵入の痕跡がわからなくなる(特に上書されるとアウト)
- 電源を抜かない
 - 揮発情報が完全に消える



**発見した状態を可能な限り
保存することを心がける**

※被害拡大防止の観点で見ると、そうではないときも多々ある

■ 作業のおおまかな流れ



ディスク解析に必要な準備

■ ディスクの解析でわかるかもしれないこと

- プログラムやファイルの改ざんの有無
- 不正なデータやプログラムの有無
- 削除されたファイルの復旧
- 時系列でのファイル操作履歴
- 特定のキーワードや図形の有無
- 一部プログラムの操作履歴

■ これ以上のことを知りたいのであれば、他の方法を併用する

- ネットワークの記録
- 聞き込み ...等

■ 調査作業用の環境

- 調査するためのPCは安全でなければならない
 - OSのハードニング
 - 最新パッチの適用
 - rootkitのチェック
- ディスクの吸出しから調査まで孤立したネットワーク内で行えると吉
- CD-ROMから起動するForensic専用Linuxなどを使う方法もある
 - Helix (おすすめ)
 - F.I.R.E.
 - Penguin Sleuth Kit
- 調査作業用PCはハイスペックなものが望ましい
 - データ量は2倍、3倍に膨れる

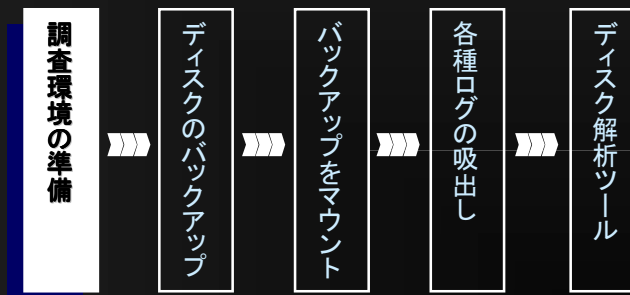


■ チェックサム

- システムファイルが改ざんされていないかのチェックに使用する
- 以下のようなチェック方法を使う
 - インストール後のMD5を保存しておく
 - 例: `find / ¥(-fstype devpts -o -fstype iso9660 ... ¥) -prune -o -type f -exec md5sum {} ¥;`
 - RDS(Reference Data Set)等を利用する
 - iso形式なのでマウントしてファイル NSRFile.txtを読めるようにしておく
 - RedHat等であればrpmコマンドを使用する
 - `rpm -checksig packagefile`



NSRLのWeb
(RDSがダウンロードできる)



■ 作業用ユーティリティの導入

- ネットワーク経由のデータ送受信に使用

■ netcat

- ネットワーク経由のデータ読み書きツール

- コマンドの汚染チェックに使用

■ chkrootkit / rkhunter

- rootkitの検出ツール

- ディスクデータの分類に使用

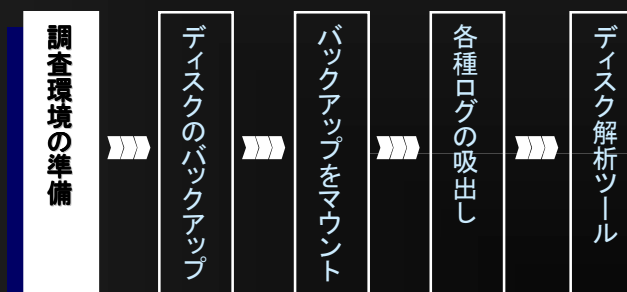
■ TCT (The Coroner's Toolkit) / sleuthkit(おすすめ)

- ディスクイメージからファイルデータの分類ツール

- 解析操作性向上のために使用

■ autopsy

- フォレンジックブラウザ(おすすめ)



■ ディスクイメージの取得

- 被害を受けたディスクをそのまま操作してしまうと、情報が変更されてしまうため、複製したディスクを解析する

- 複製方法

■ 専用ハードウェアを使用して複製を行う

■ ソフトウェアを使用して複製を行う

- dd

- ほとんどのUNIXに標準でインストールされている。(おすすめ)

- sdd

- 多機能版dd。フォレンジックツールでサポートされる

- dcfldd

- US DoD拡張版dd

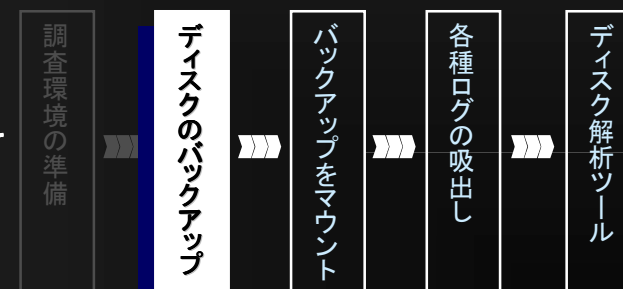
※DCFL = DoD Computer Forensic Laboratory

■ ディスク全体をバックアップ

```
dd if=/dev/hda of=image.dd conv=noerror
```

■ 必要なパーティションのみをバックアップ

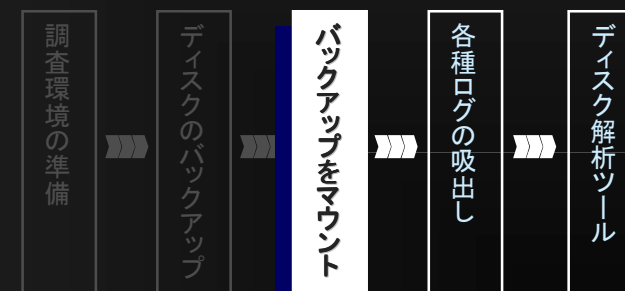
```
dd if=/dev/hdaX of=imageX.dd conv=noerror
```



■ ディスクのマウント

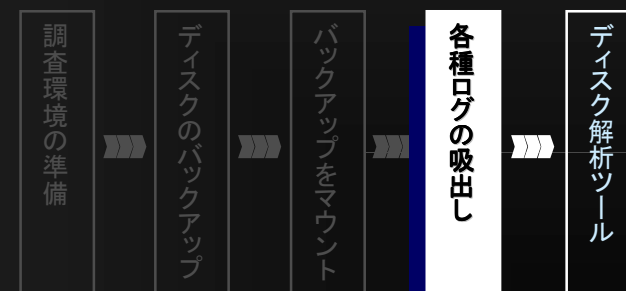
- ddで取得したイメージはそのままではファイルシステムとして読み出せない。
ddイメージをローカルにマウントすることでアクセス可能となる
- 例:EXT3のディスクに含まれる1パーティションをddでimage.ddというファイル名でダンプし、それを/mntにマウントする場合
 - `mount -t ext3 -o loop,ro /tmp/image.dd /mnt`
- 複数のパーティションをまとめてddした場合はlosetupしてからmountする
 - `losetup -o CylinderNo /dev/loopX image.dd`
 - `mount -t ext3 -o ro,nodev,noexec /dev/loopX /mnt`
 - パーティションが始まるシリンダー番号が必要
 - `fdisk -lu image.dd`等で調べる
 - インストール時に記録しておく

※ autopsyを使用する(後述)場合は
パーティションごとにファイルが別に
見える必要がある



■ 各種ログの吸出し

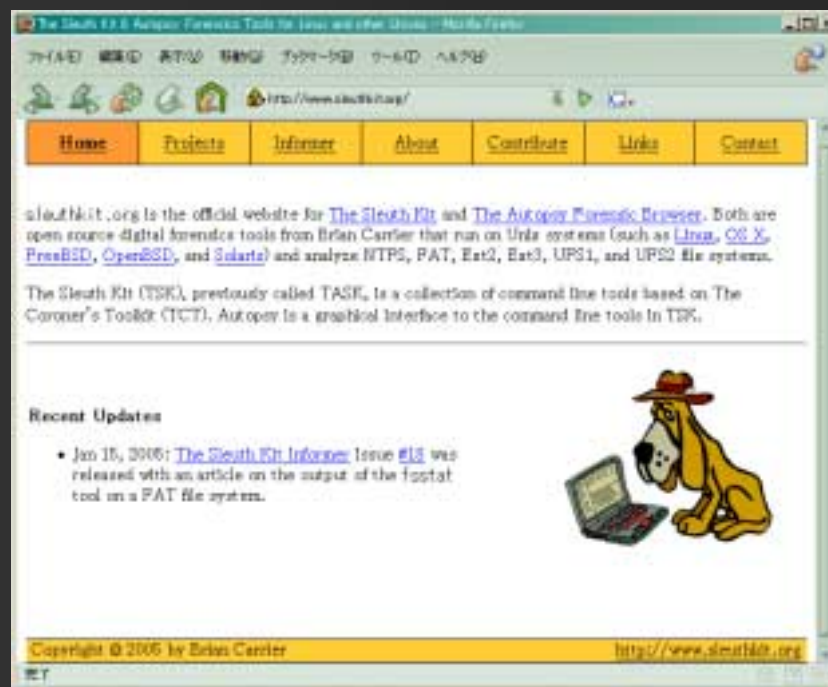
- 前セッション「ログ管理・解析」をご覧ください :-)



解析ツール

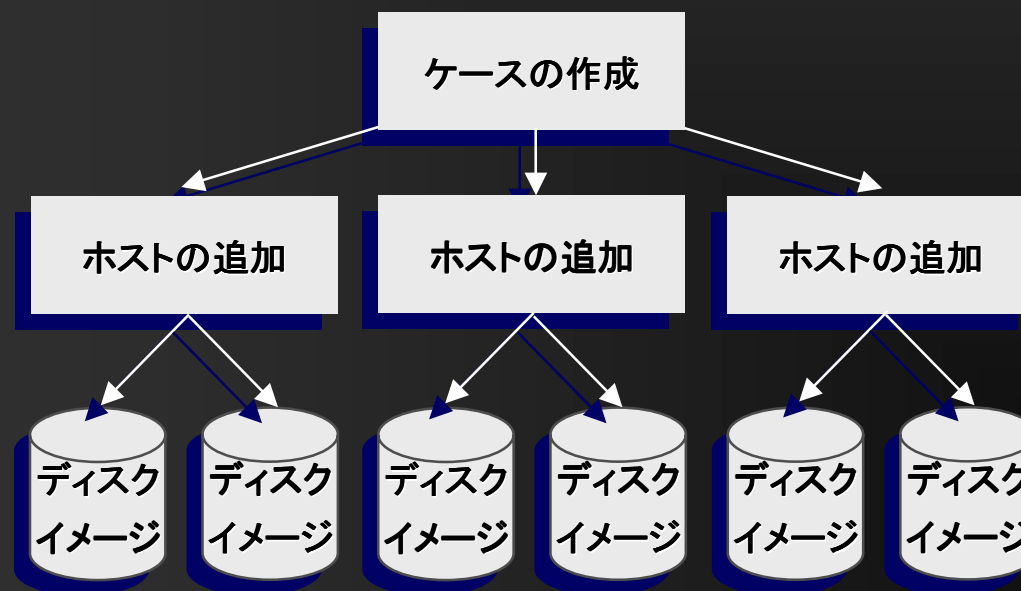
■ sleuthkit

- ディスク解析を効率的に行うためのソフト
- TCT(The Coroner's Toolkit)をベースとしている
- ディスクのダンプファイルを解析するためのコマンド群
- その他の解析ツールのベースとなることが多い



■ autopsy

- ディスクの解析ツール。sleuthkitのフロントエンド
- autopsy操作手順
 1. 調査ケースの作成
 2. 調査ケースへ対象ホストの追加
 3. 対象ホストへ調査ディスクイメージを追加
 4. タイムラインの作成とファイル解析



■ autopsy

－ インストール

```

root@localhost:~/autopsy-2.03
File Edit View Terminal Go Help
[root@localhost autopsy-2.03]# make

Autopsy Forensic Browser Installation

perl found: /usr/bin/perl

-----

Autopsy uses the grep utility from your local system.
grep found: /bin/grep

-----

Autopsy uses forensic tools from The Sleuth Kit.
http://www.sleuthkit.org/sleuthkit/

Enter the directory where you installed it:
/usr/local/sleuthkit
  
```

Sleuthkitがインストール
されているpathを入力

NSRLが展開されている
Pathを入力

```

root@localhost:~/autopsy-2.03
File Edit View Terminal Go Help
Have you purchased or downloaded a copy of the NSRL (y/n) [n]
y
Enter the directory where you installed it:
/root/NSRL/
NSRL database was found (NSRLFile.txt)
NSRL Index file not found, do you want it created? (y/n) [n]:
y

----- begin hfind output -----
Extracting Data from Database (/root/NSRL//NSRLFile.txt)
Valid Database Entries: 8367293
Invalid Database Entries (headers or errors): 1
Index File Entries (optimized): 2267183
Sorting Index (/root/NSRL//NSRLFile.txt-md5.idx)
----- end hfind output -----

-----

Autopsy saves configuration files, audit logs, and output to the
Evidence Locker directory.

Enter the directory that you want to use for the Evidence Locker:
  
```

■ autopsy

- インターフェースはブラウザを使用
- ランダムな数字列はautopsyを起動する毎に変わる

autopsyが起動するポート アクセス許可するアドレス

```

root@localhost: ~/autopsy-2.03
File Edit View Terminal Go Help
[root@localhost autopsy-2.03]# ./autopsy -p 10000 192.168.1.3

=====

Autopsy Forensic Browser
http://www.sleuthkit.org/autopsy/
ver 2.03

=====

Evidence Locker: /root/evidence
Start Time: Wed Nov 3 20:04:34 2004
Remote Host: 192.168.1.3
Local Port: 10000

Open an HTML browser on the remote host and paste this URL in it:

    http://localhost.localdomain:10000/30612486474223986621/autopsy

Keep this process running and use <ctrl-c> to exit
  
```

このURLにアクセス

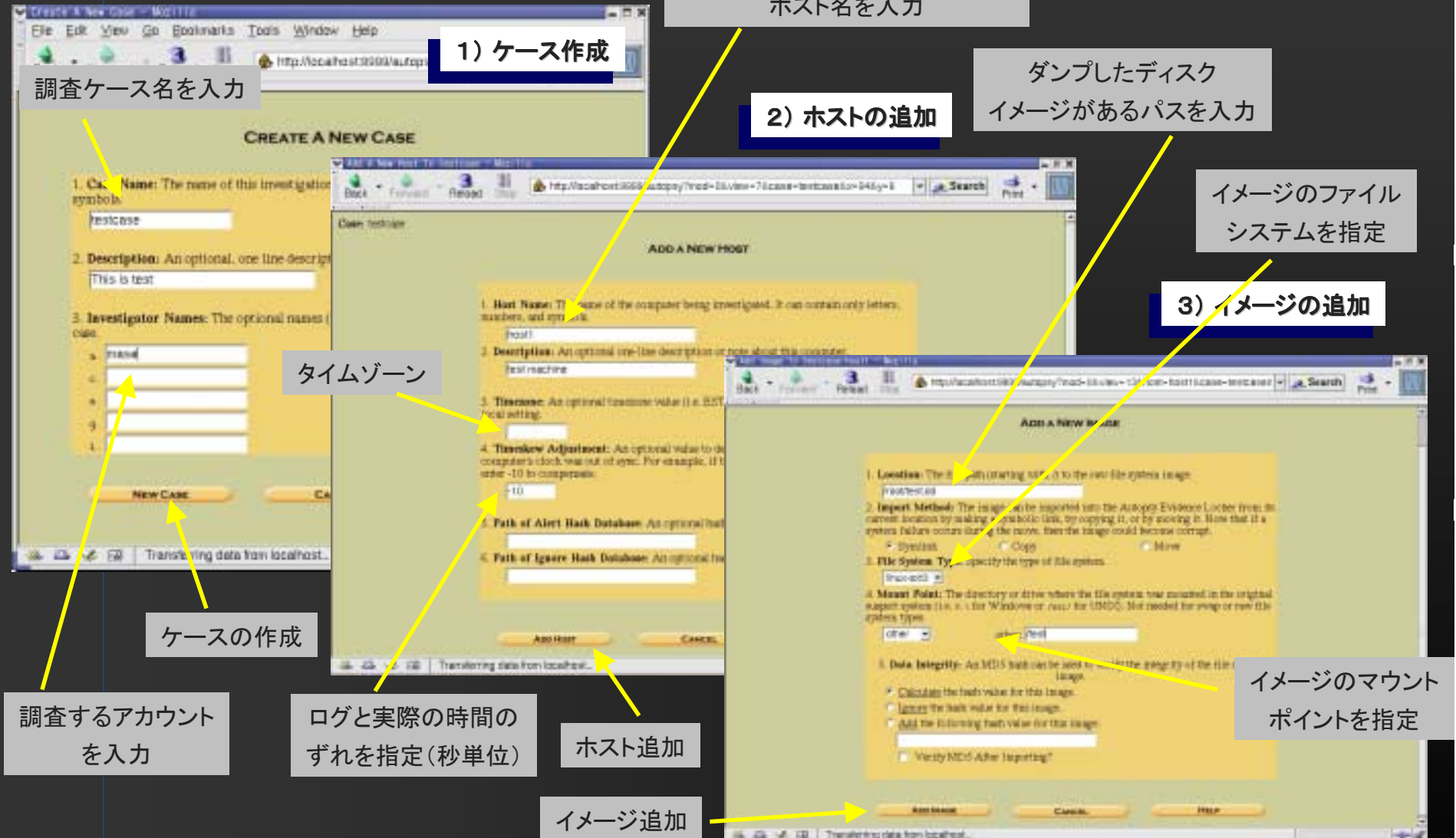
■ autopsy

— autopsy起動画面

ここからスタート



autopsyへの登録



1) ケース作成

調査ケース名を入力

読み込むディスクデータのあった
ホスト名を入力

タイムゾーン

ケースの作成

調査するアカウント
を入力

ログと実際の時間の
ずれを指定(秒単位)

2) ホストの追加

タイムゾーン

ホスト追加

イメージ追加

ダンプしたディスク
イメージがあるパスを入力

イメージのファイル
システムを指定

3) イメージの追加

イメージのマウント
ポイントを指定

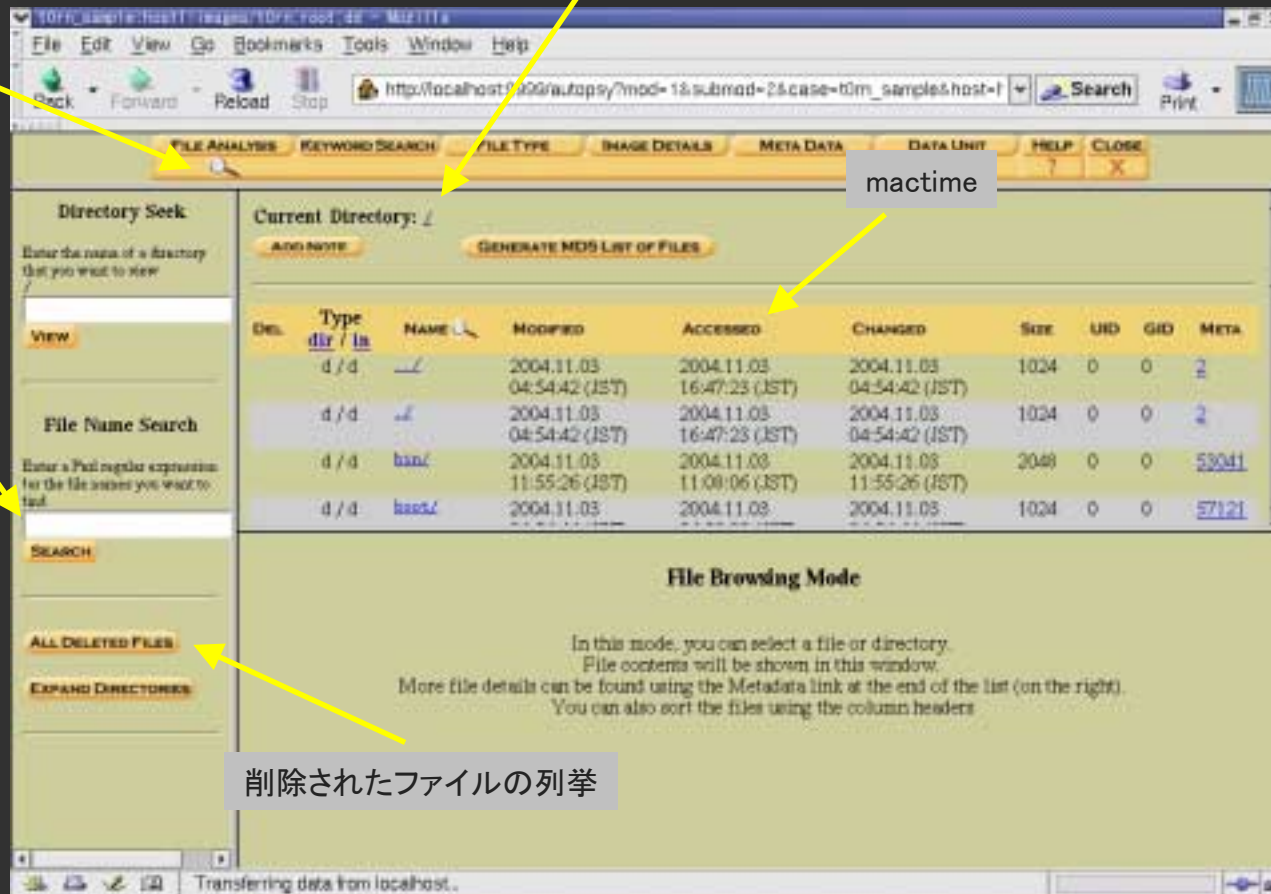
■ autopsy

— ディレクトリの復元

イメージをマウントしたディレクトリ

現在の作業

ファイル名検索



The screenshot shows the Autopsy web interface. The browser address bar displays `http://localhost:8080/autopsy/?mod=1&submod=2&case=t0m_sample&host=1`. The interface includes a menu bar (File, Edit, View, Go, Bookmarks, Tools, Window, Help) and a toolbar with buttons for FILE ANALYSIS, KEYWORD SEARCH, FILE TYPE, IMAGE DETAILS, META DATA, DATA UNIT, HELP, and CLOSE. The main content area is divided into several sections:

- Directory Seek:** Includes a text input field for entering a directory path and a VIEW button.
- File Name Search:** Includes a text input field for a regular expression and a SEARCH button.
- Current Directory:** Shows the current directory as `/` and includes buttons for ADD NOTE and GENERATE MDS LIST OF FILES.
- File Listing Table:** A table with columns: Del., Type, NAME, Modified, Accessed, Changed, SIZE, UID, GID, META. It lists several files and directories.
- File Browsing Mode:** A section with instructions on how to use the interface.
- Navigation Buttons:** ALL DELETED FILES and EXPAND DIRECTORIES.

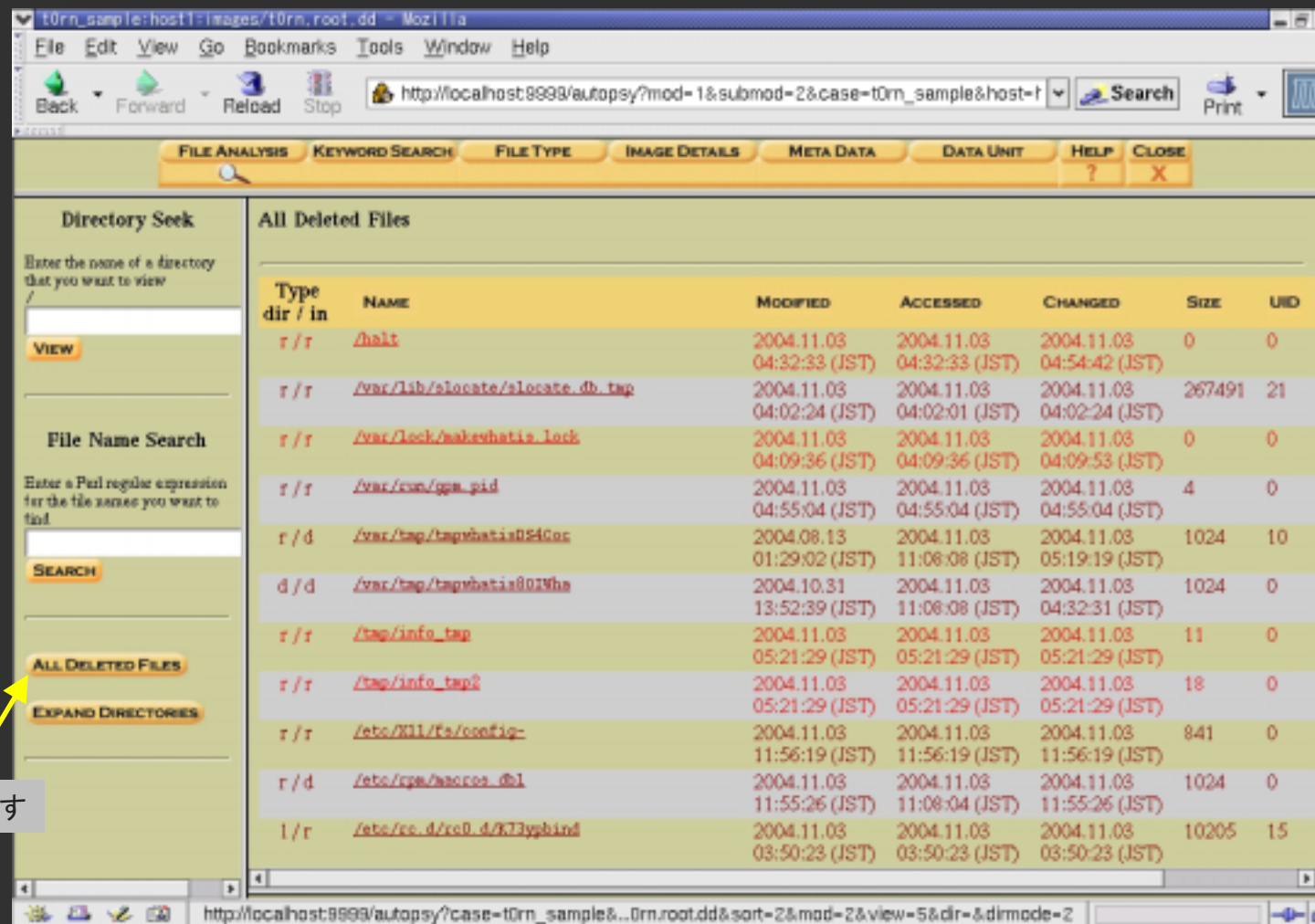
Yellow arrows point from external text boxes to specific elements in the interface:

- From "現在の作業" to the FILE ANALYSIS button.
- From "イメージをマウントしたディレクトリ" to the "Current Directory: /" field.
- From "mactime" to the "Accessed" column header in the file listing table.
- From "ファイル名検索" to the File Name Search input field.
- From "削除されたファイルの列挙" to the ALL DELETED FILES button.

Del.	Type	NAME	Modified	Accessed	Changed	SIZE	UID	GID	META
d/d	dir	/	2004.11.03 04:54:42 (JST)	2004.11.03 16:47:23 (JST)	2004.11.03 04:54:42 (JST)	1024	0	0	2
d/d	dir	/	2004.11.03 04:54:42 (JST)	2004.11.03 16:47:23 (JST)	2004.11.03 04:54:42 (JST)	1024	0	0	2
d/d	file	hanc	2004.11.03 11:55:26 (JST)	2004.11.03 11:09:06 (JST)	2004.11.03 11:55:26 (JST)	2048	0	0	53041
d/d	file	hanc/	2004.11.03	2004.11.03	2004.11.03	1024	0	0	57121

autopsy

削除されたファイル一覧



The screenshot shows the Autopsy web interface in a Mozilla browser window. The URL is `http://localhost:8998/autopsy?mod=1&submod=2&case=t0rn_sample&host=t0rn_sample:localhost:images/t0rn.root.dd`. The interface has several tabs: FILE ANALYSIS, KEYWORD SEARCH, FILE TYPE, IMAGE DETAILS, META DATA, DATA UNIT, HELP, and CLOSE. The 'FILE ANALYSIS' tab is active.

On the left sidebar, there are sections for 'Directory Seek' and 'File Name Search'. Below these, there are buttons for 'VIEW', 'SEARCH', 'ALL DELETED FILES', and 'EXPAND DIRECTORIES'. A yellow arrow points to the 'ALL DELETED FILES' button, with a callout box containing the text 'ここを押す' (Press here).

The main content area displays a table titled 'All Deleted Files' with the following columns: Type, dir / in, NAME, MODIFIED, ACCESSED, CHANGED, SIZE, and UID. The table contains 15 rows of file information.

Type	dir / in	NAME	MODIFIED	ACCESSED	CHANGED	SIZE	UID
r / r	/	/halt	2004.11.03 04:32:33 (JST)	2004.11.03 04:32:33 (JST)	2004.11.03 04:54:42 (JST)	0	0
r / r	/var/lib/slocate/slocate.db	/var/lib/slocate/slocate.db.tmp	2004.11.03 04:02:24 (JST)	2004.11.03 04:02:01 (JST)	2004.11.03 04:02:24 (JST)	267491	21
r / r	/var/lock	/var/lock/nakvhatia.lock	2004.11.03 04:09:36 (JST)	2004.11.03 04:09:36 (JST)	2004.11.03 04:09:53 (JST)	0	0
r / r	/var/run	/var/run/gpm.pid	2004.11.03 04:55:04 (JST)	2004.11.03 04:55:04 (JST)	2004.11.03 04:55:04 (JST)	4	0
r / d	/var/tmp	/var/tmp/tapvhatia024Goc	2004.08.13 01:29:02 (JST)	2004.11.03 11:08:08 (JST)	2004.11.03 05:19:19 (JST)	1024	10
d / d	/var/tmp	/var/tmp/tapvhatia001Vha	2004.10.31 13:52:39 (JST)	2004.11.03 11:08:08 (JST)	2004.11.03 04:32:31 (JST)	1024	0
r / r	/tmp	/tmp/info.tmp	2004.11.03 05:21:29 (JST)	2004.11.03 05:21:29 (JST)	2004.11.03 05:21:29 (JST)	11	0
r / r	/tmp	/tmp/info.tmp2	2004.11.03 05:21:29 (JST)	2004.11.03 05:21:29 (JST)	2004.11.03 05:21:29 (JST)	18	0
r / r	/etc/kill/ks/ksconfig-	/etc/kill/ks/ksconfig-	2004.11.03 11:56:19 (JST)	2004.11.03 11:56:19 (JST)	2004.11.03 11:56:19 (JST)	841	0
r / d	/etc/cpa	/etc/cpa/waocra.db1	2004.11.03 11:55:26 (JST)	2004.11.03 11:08:04 (JST)	2004.11.03 11:55:26 (JST)	1024	0
l / r	/etc/cc_d	/etc/cc_d/rt0_d/K73yphind	2004.11.03 03:50:23 (JST)	2004.11.03 03:50:23 (JST)	2004.11.03 03:50:23 (JST)	10205	15

ここを押す

■ autopsy

— キーワード検索

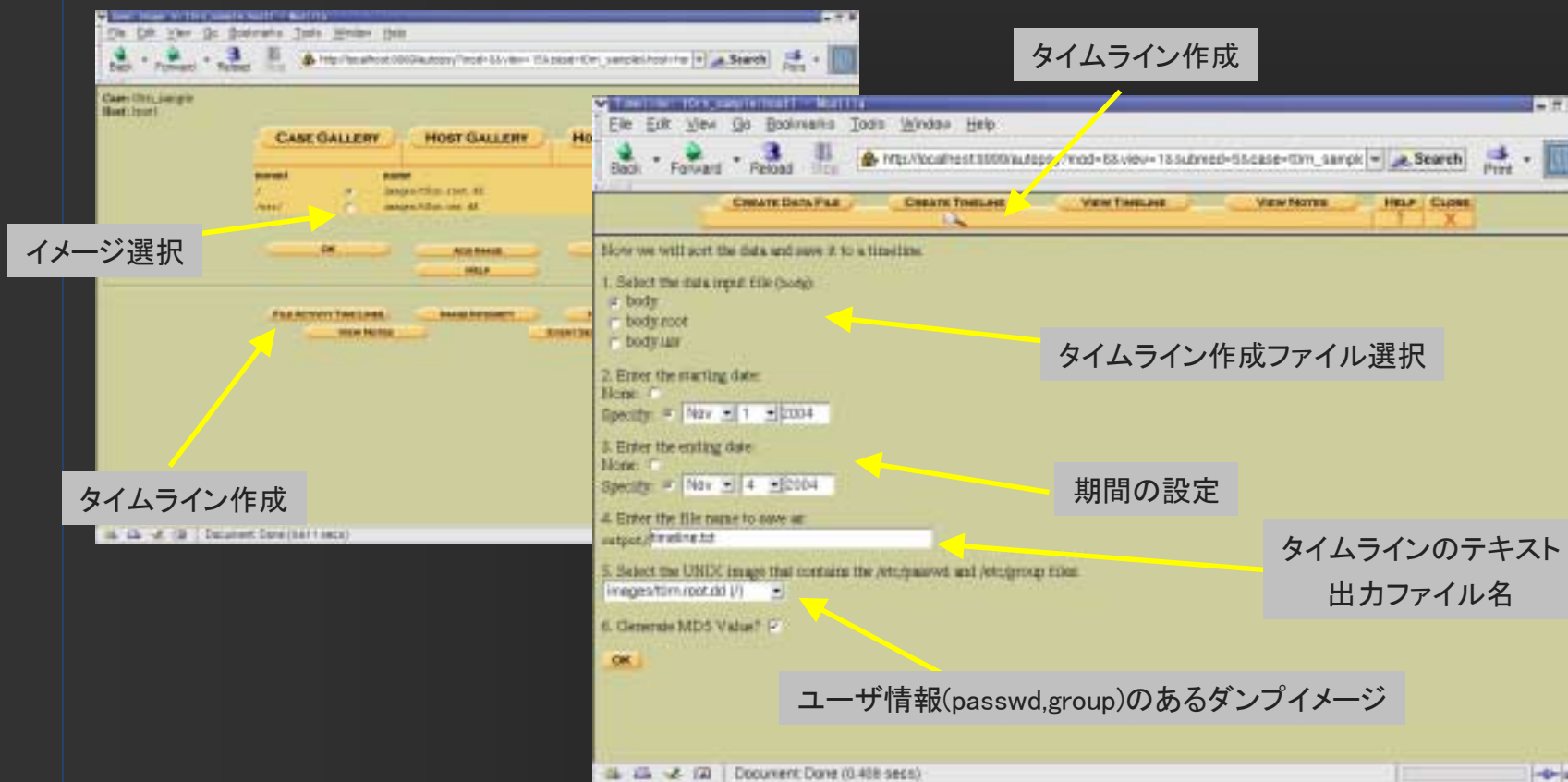
■ キーワード検索は意外と困難

- キーワードを知っている量がそのまま調査の質に結びつく
- 1文字変更してあるだけで検出できなくなる可能性
- ディスク容量が大きくなるほど検索時間が膨大になる
 - 30分、1時間はすぐ過ぎる
- 力技であり、あくまでも補助的な手段



■ autopsy

- タイムライン(時系列情報)の作成
- これがメインの作業
- 時系列の変化を追うことで不審な行動がないかどうかを調べる



タイムライン作成

タイムライン作成

タイムライン作成ファイル選択

期間の設定

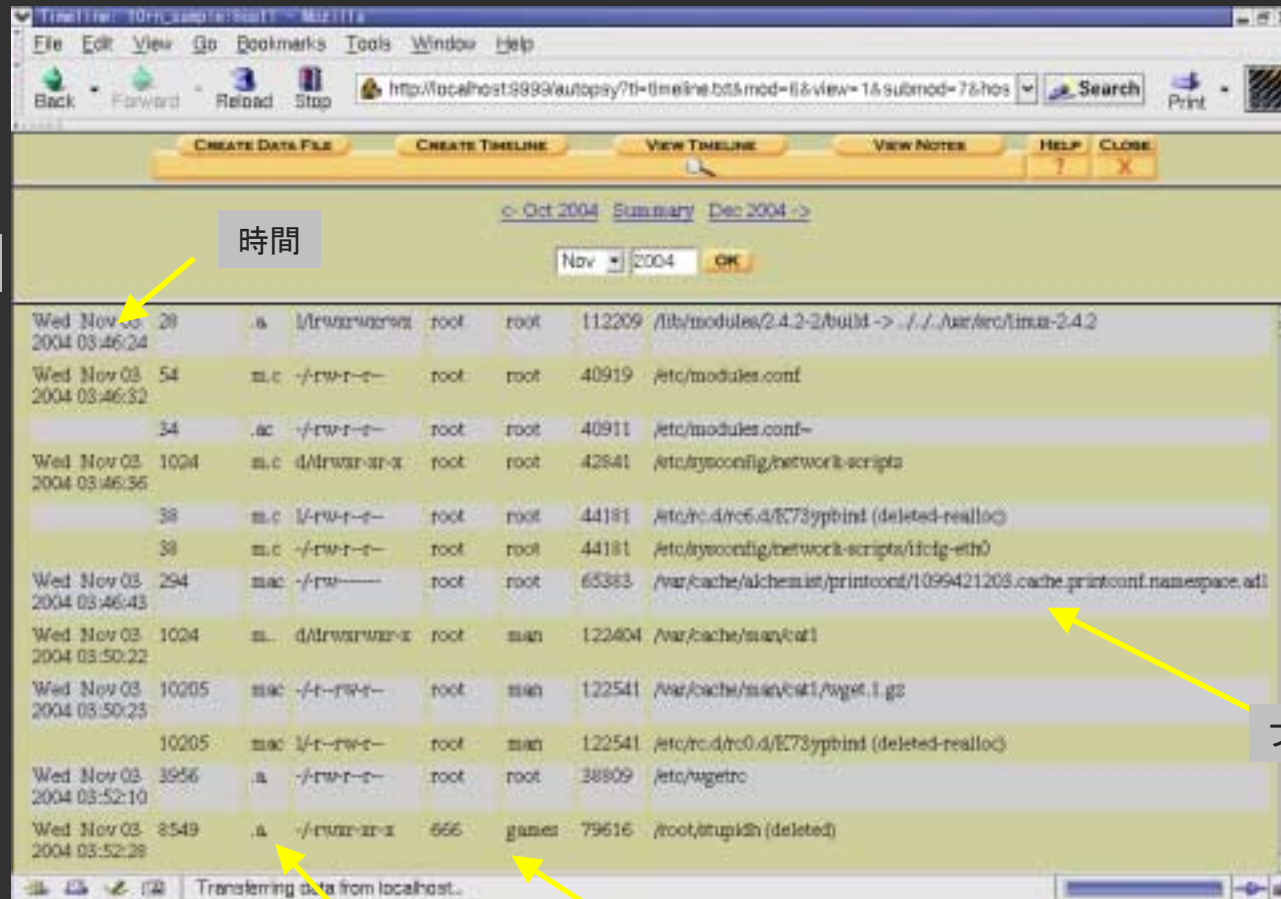
タイムラインのテキスト出力ファイル名

ユーザ情報(passwd,group)のあるダンプイメージ

イメージ選択

■ autopsy

– タイムラインを読む



時間	PID	コマンド	mactime	group	owner	ファイル名
Wed Nov 03 2004 03:46:24	28	./lib/modules/2.4.2-2/build -> ./././var/ro/tima-2.4.2	112209	lib/modules/2.4.2-2/build	root	./././var/ro/tima-2.4.2
Wed Nov 03 2004 03:46:32	54	m.c ./rwr-r-	40919	etc/modules.conf	root	etc/modules.conf
	34	.ac ./rwr-r-	40911	etc/modules.conf-	root	etc/modules.conf-
Wed Nov 03 2004 03:46:36	1024	m.c d/rwr-r-x	42841	etc/sysconfig/network-scripts	root	etc/sysconfig/network-scripts
	38	m.c ./rwr-r-	44181	etc/rc.d/rc6.d/E73ypbind (deleted-realloc)	root	etc/rc.d/rc6.d/E73ypbind (deleted-realloc)
	38	m.c ./rwr-r-	44181	etc/sysconfig/network-scripts/ifcfg-eth0	root	etc/sysconfig/network-scripts/ifcfg-eth0
Wed Nov 03 2004 03:46:43	294	mac ./rw-	65383	/var/cache/alchemy/printconf/1099421203.cache.printconf.namespace.adl	root	/var/cache/alchemy/printconf/1099421203.cache.printconf.namespace.adl
Wed Nov 03 2004 03:50:22	1024	m. d/rwrwr-x	122404	/var/cache/man/cat1	man	/var/cache/man/cat1
Wed Nov 03 2004 03:50:23	10205	mac ./r-rwr-	122541	/var/cache/man/cat1/wget.l.gz	man	/var/cache/man/cat1/wget.l.gz
	10205	mac ./r-rwr-	122541	etc/rc.d/rc0.d/E73ypbind (deleted-realloc)	man	etc/rc.d/rc0.d/E73ypbind (deleted-realloc)
Wed Nov 03 2004 03:52:10	3956	a ./rwr-r-	38809	etc/wgetrc	root	etc/wgetrc
Wed Nov 03 2004 03:52:28	8549	a ./rwr-r-x	79616	/root/stupidh (deleted)	games	/root/stupidh (deleted)

時系列

時間

全部で4日間分

ファイル名

mactime

group/owner

■ 今後の検討課題

— データ量の問題

- 現在のファイル容量では、シーケンシャルにデータを処理する方式は限界
- 検査を行うためのPCに対する要求スペックが高すぎる。現実に入手できるPCでは処理が遅くて時間が非常にかかる

— 時刻の精度の問題

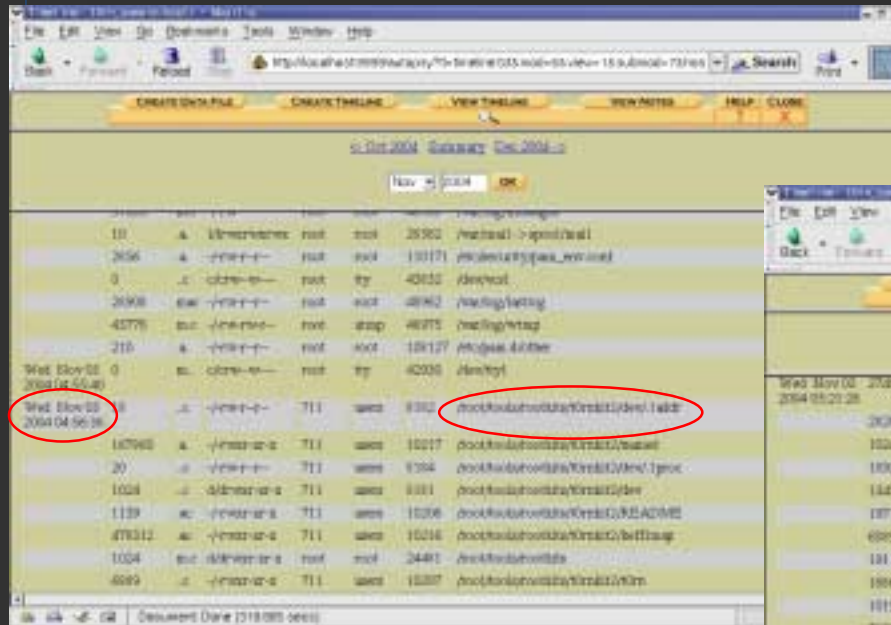
- DOS/V機のクロックの精度が悪い。せめて月に数秒ずれくらいにして欲しい

— シグネチャ

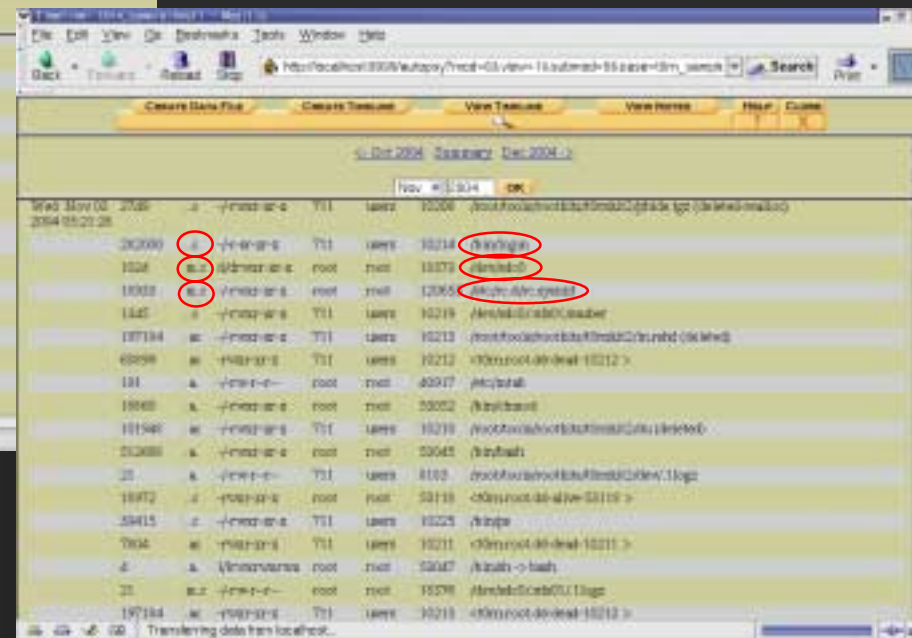
- このコマンドを実行すると、このタイミングでこのファイルがこう変更されるはず、というようなまとまったデータが必要
- ネットワークやクライアントPCでは実現できている (IDSとかウィルススキャン) ので、そう難しくはないはず
- ここをインテリジェントに処理してくれれば、作業量は相当減るはず

■ autopsy

— 運良く発見できた場合



File Name	Size	Owner	Permissions	Path
10	4	Administrator	root	/usr/sbin
2056	4	Administrator	root	/usr/sbin
0	0	Administrator	root	/usr/sbin
2098	64	Administrator	root	/usr/sbin
4575	64	Administrator	root	/usr/sbin
210	4	Administrator	root	/usr/sbin
Wed Nov 02 2004 04:56:36	0	Administrator	root	/usr/sbin
16760	4	Administrator	root	/usr/sbin
20	4	Administrator	root	/usr/sbin
1024	4	Administrator	root	/usr/sbin
1139	4	Administrator	root	/usr/sbin
478312	4	Administrator	root	/usr/sbin
1024	4	Administrator	root	/usr/sbin
6092	4	Administrator	root	/usr/sbin



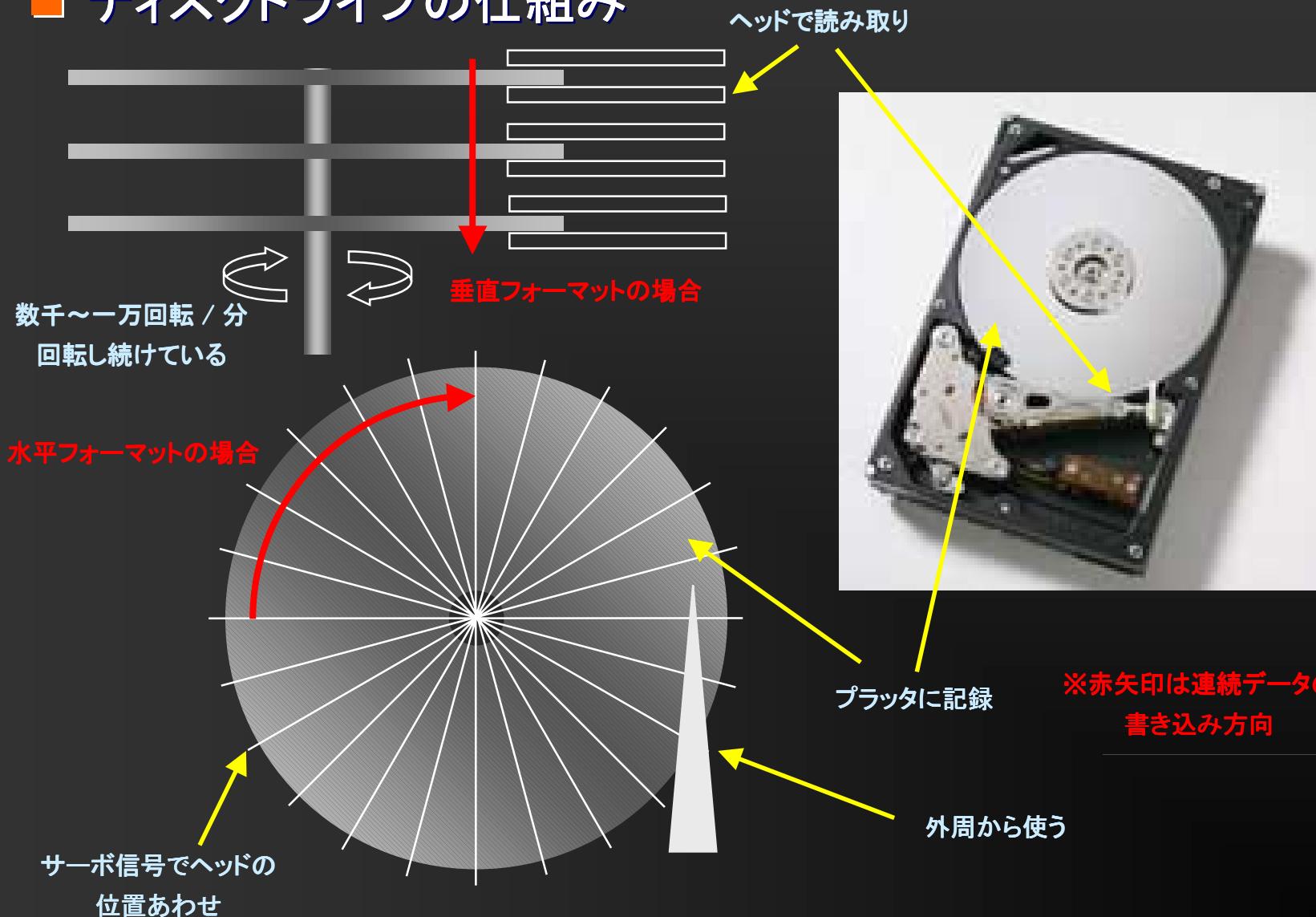
File Name	Size	Owner	Permissions	Path
20200	4	Administrator	root	/usr/sbin
1024	4	Administrator	root	/usr/sbin
1002	4	Administrator	root	/usr/sbin
1442	4	Administrator	root	/usr/sbin
187184	4	Administrator	root	/usr/sbin
60998	4	Administrator	root	/usr/sbin
181	4	Administrator	root	/usr/sbin
1866	4	Administrator	root	/usr/sbin
181948	4	Administrator	root	/usr/sbin
212688	4	Administrator	root	/usr/sbin
20	4	Administrator	root	/usr/sbin
18872	4	Administrator	root	/usr/sbin
28415	4	Administrator	root	/usr/sbin
7802	4	Administrator	root	/usr/sbin
4	4	Administrator	root	/usr/sbin
21	4	Administrator	root	/usr/sbin
197184	4	Administrator	root	/usr/sbin

これで何もわからなかったらどうする???

HDDデータ復旧の可能性

- ddでダンプしたディスクを調べても有効なデータは出てこない場合、それで終わりと言えるか？
- ディスクは1回消しただけでは「完全に」消えてないと言われているが本当？
- ダンプ以外にハードディスクからデータを読み取る方法はないのか？

■ ディスクドライブの仕組み



■ ATA

- AT Attachmentの略
- ハードディスクのインターフェース仕様
- CD-ROM等を接続するためにATAPI(ATA Packet Interface)が制定
- ネットワークと同じようにホスト(CPU)とデバイス(HDD)間で決まったプロトコル(コマンド)を使い、データのやり取りをする

■ ディスクの読み書き場所の指定について

- 現在はLBA方式(Logical Block Addressing)
- 論理的な番号で指定し、どこのヘッド、シリンダ、セクタに読み書きするかはディスクがエラーセクタを避けて、自動的に計算して割り出す(なので、単純な掛け算ではない)
- エラーセクタの情報は工場出荷時に書かれている。動作後に出てくるエラーセクタは随時記録される
- ディスクの回転誤差は0.02%とか0.03%くらいでトラックの幅はミクロン単位。恐るべき精度での制御が行われている

※余談

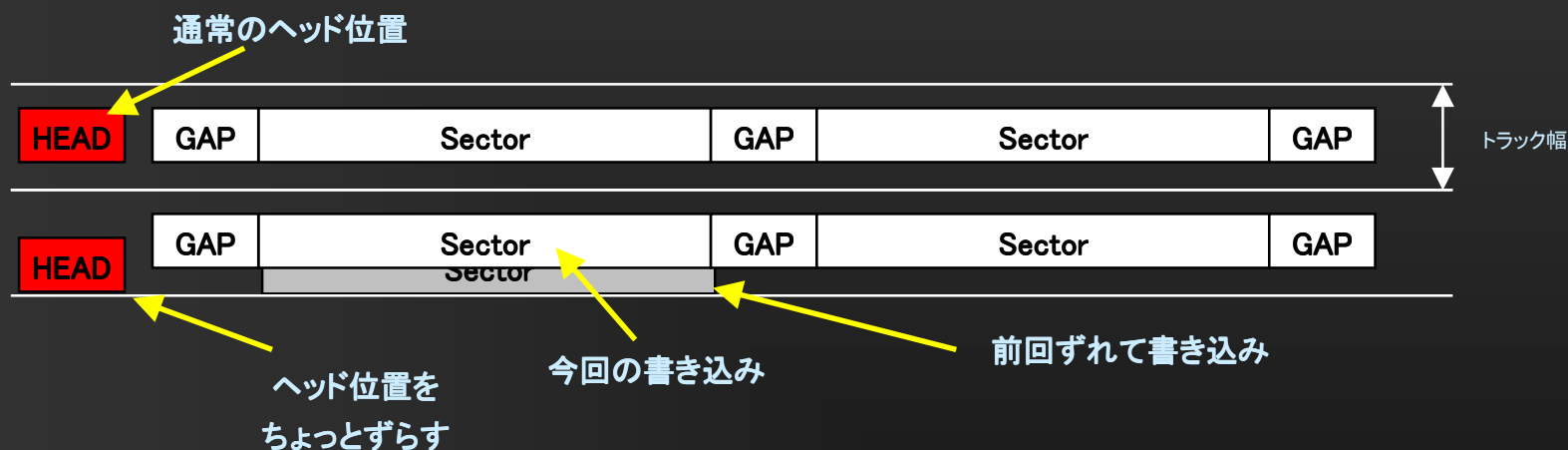
HDDパスワードは意外と(失礼)強い! ?

利用できるなら利用しておくとい

■ データの残る可能性

－ オフセットリード・ライト

- 書き込み誤差でちょっとだけ、位置がずれたものを読み出す



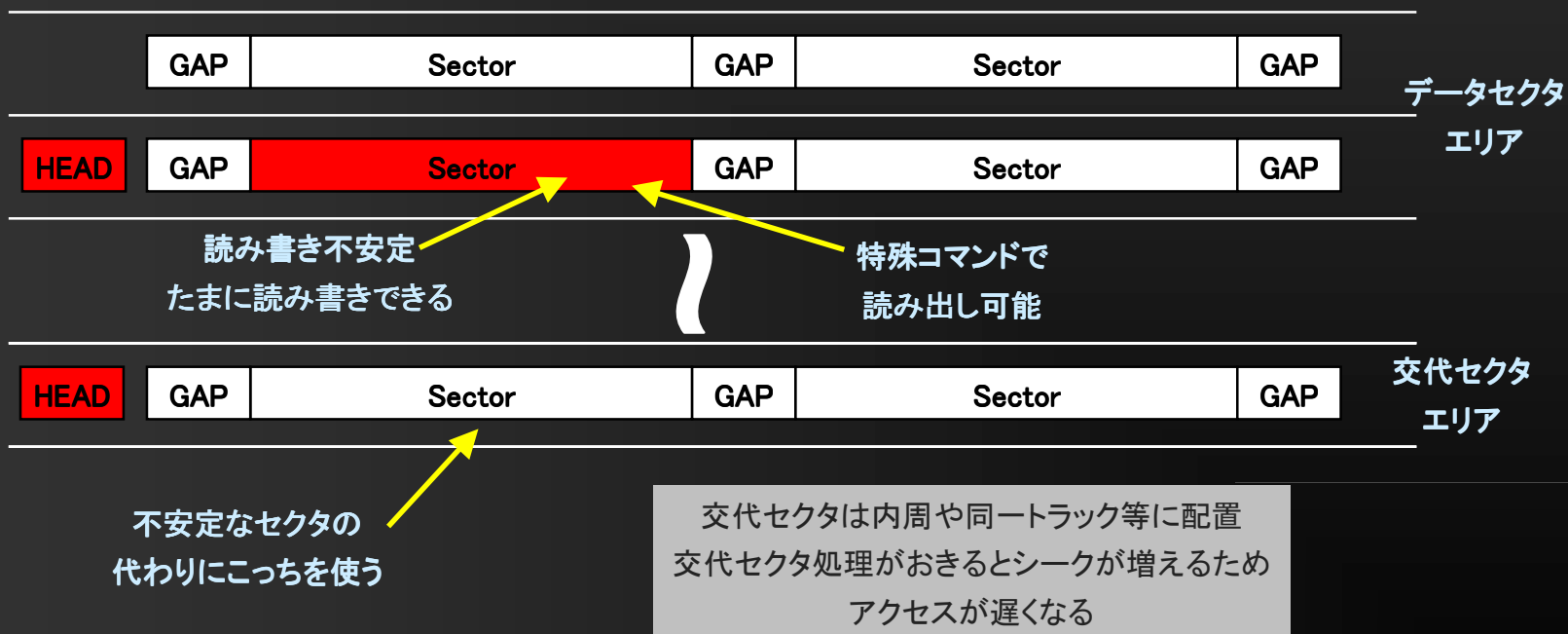
－ ライト不良

- 以前に書き込みがあり、再度書き込みが発生したが、何らかの原因により書き込みできなければ以前の書き込みが残るので読み出し可能

■ データの残る可能性

－ 交代セクタ

- 読み込みや書き込みにおいてリトライしてなんとか読み書きできるようなセクタはスキップし、別の専用エリア(交代セクタエリア)を使ってデータを読み書きする。そのため、旧セクタのデータは残ったままになる。残ったエリアは特殊コマンドで読み出しが可能



- 強制的にデータを読み出すことはどこまで可能か
 - コントローラを直接制御
 - 現在のATA仕様では決まったプロトコル内でのやり取りになっており、コントローラを直接制御してディスクを操るのは無理
 - Vendor Specificなコマンドを使用すればできる可能性もあるが、当然ながら専用の設備や企業秘密を知らなければできない
 - プラッタを入れ替え
 - 垂直フォーマットでは連続したデータが拾えない
 - そもそもディスクを分解した時点でアウト。クリーンルームのあるオフィスがあれば別かもしれないが。
 - 位相ずれを起こせば元のデータと全く異なるものしか読めない
 - そもそも、出荷時に書かれた制御情報(不良クラスタ等)がないので読み様がない
 - 顕微鏡とかで見る
 - 200GBとか300GBを？
 - 太平洋の中から石ころ1個探すようなもの
 - 微視的に見ることができても巨視的に有効かどうかはわからない
- 膨大な時間とお金と労力をかけて、読み出せるデータはほんのちょっとだけ。実質的には難しい
- 他の方法を考えたほうがよい場合がほとんど

■ データ量問題

- ハードディスクの容量は順調に増加している
 - 500GBも登場
- まだまだ容量は増えつづけると考えられる

■ データが破壊されたときのインパクトは増大

- バックアップは必須となる

■ しかし、バックアップが困難

- HDDのバックアップはHDDで取るしかない

■ データ量が増えれば情報処理時間も増える

 **今後解決が必要な課題**

総合的な分析ツール

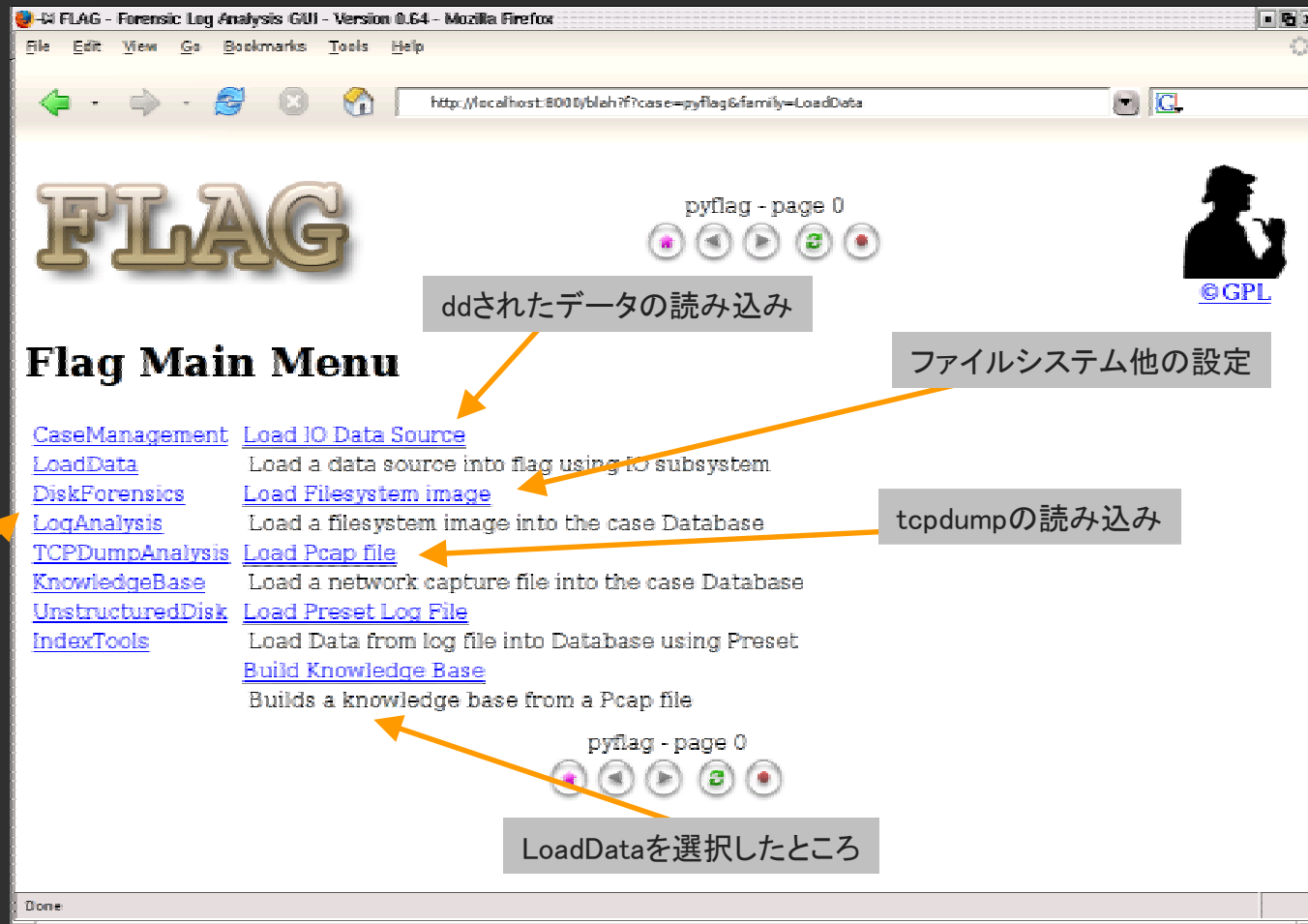
■ FLAG

- ディスクダンプ、ネットワークダンプ、ログをまとめて解析するツール
- データを効率的に処理するため、データベース(mysql)を使用
- 様々なログフォーマットの読み出しが可能
- pcap方式のネットワークダンプの分析が可能
- ddで出力したディスクイメージの読み込みが可能



■ FLAG

－ メインメニュー



The screenshot shows the FLAG web interface in a Mozilla Firefox browser window. The URL is `http://localhost:8000/blah/?case=pyflag&family=LoadData`. The page title is "FLAG" and it includes navigation controls and a "pyflag - page 0" indicator. The main menu is titled "Flag Main Menu" and lists several options:

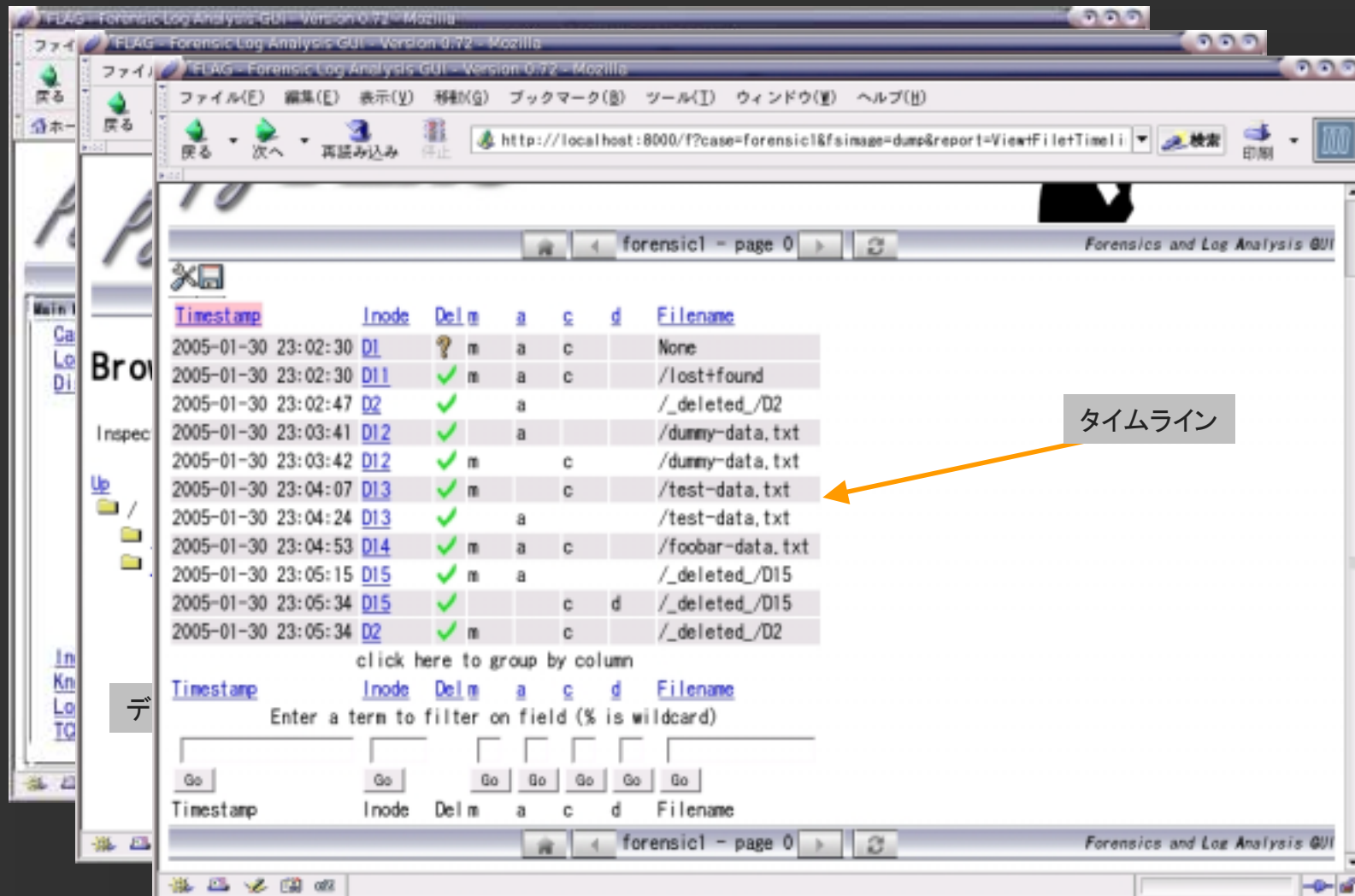
- [CaseManagement](#)
- [LoadData](#): Load a data source into flag using io subsystem
- [DiskForensics](#)
- [LogAnalysis](#): Load a filesystem image into the case Database
- [TCPDumpAnalysis](#): [Load Pcap file](#): tcpdumpの読み込み
- [KnowledgeBase](#): Load a network capture file into the case Database
- [UnstructuredDisk](#): [Load Preset Log File](#)
- [IndexTools](#): Load Data from log file into Database using Preset
- [Build Knowledge Base](#): Builds a knowledge base from a Pcap file

Annotations on the screenshot include:

- "メインメニュー" (Main Menu) pointing to the left sidebar.
- "ddされたデータの読み込み" (Loading dd data) pointing to the "Load IO Data Source" option.
- "ファイルシステム他の設定" (File system other settings) pointing to the "Load Filesystem image" option.
- "tcpdumpの読み込み" (Loading tcpdump) pointing to the "Load Pcap file" option.
- "LoadDataを選択したところ" (Where LoadData was selected) pointing to the "LoadData" option.

FLAG

– Disk Forensics



The screenshot displays the FLAG Forensic Log Analysis GUI. The main window shows a table of file operations with columns for Timestamp, Inode, Delm, a, c, d, and Filename. An orange arrow points to the 'a' column, labeled 'タイムライン' (Timeline).

Timestamp	Inode	Delm	a	c	d	Filename
2005-01-30 23:02:30	D1	?	m	a	c	None
2005-01-30 23:02:30	D11	✓	m	a	c	/lost+found
2005-01-30 23:02:47	D2	✓		a		/_deleted_/D2
2005-01-30 23:03:41	D12	✓		a		/dummy-data.txt
2005-01-30 23:03:42	D12	✓	m		c	/dummy-data.txt
2005-01-30 23:04:07	D13	✓	m		c	/test-data.txt
2005-01-30 23:04:24	D13	✓		a		/test-data.txt
2005-01-30 23:04:53	D14	✓	m	a	c	/foobar-data.txt
2005-01-30 23:05:15	D15	✓	m	a		/_deleted_/D15
2005-01-30 23:05:34	D15	✓			c d	/_deleted_/D15
2005-01-30 23:05:34	D2	✓	m		c	/_deleted_/D2

click here to group by column

Timestamp Inode Delm a c d Filename

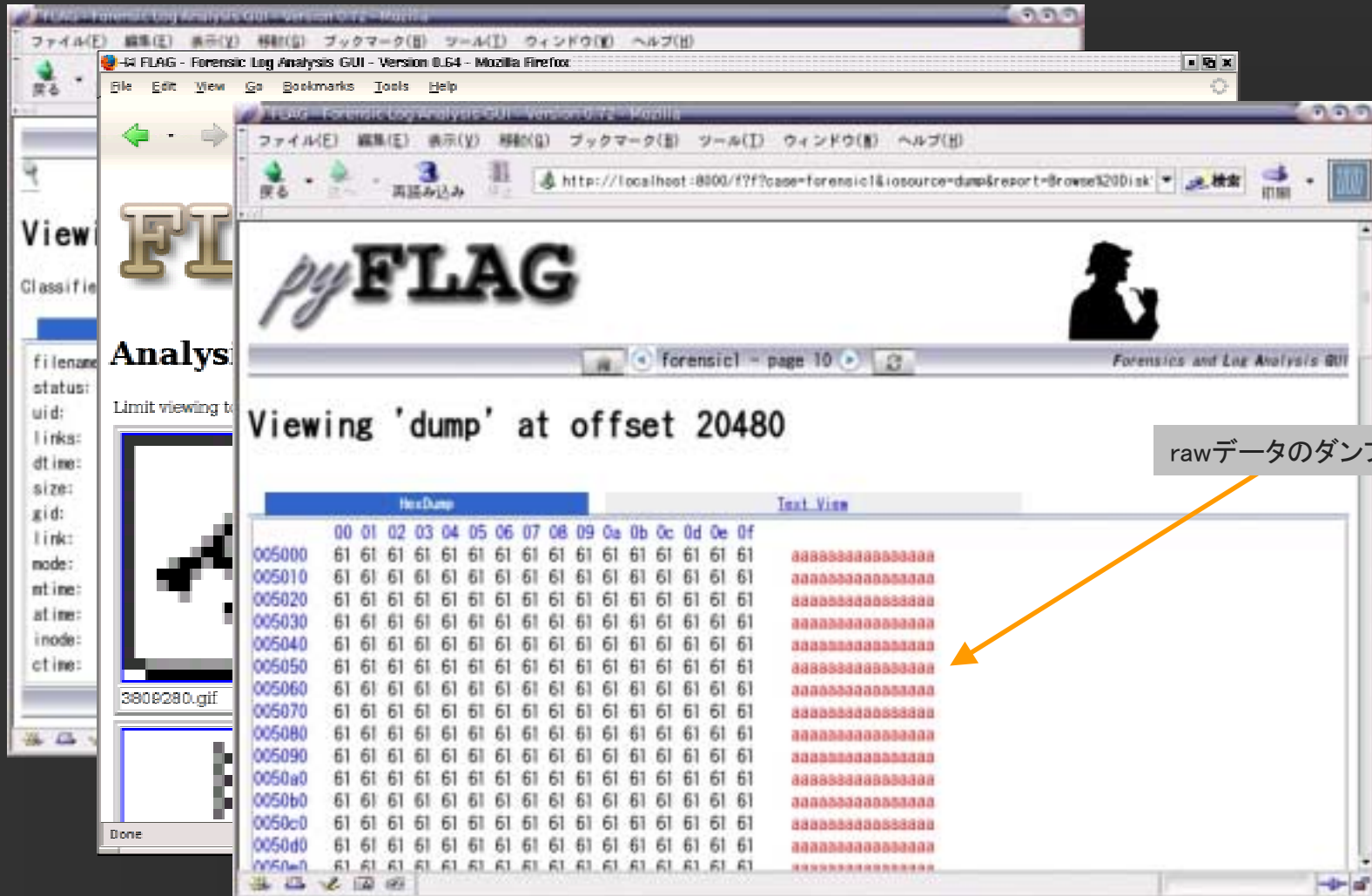
Enter a term to filter on field (% is wildcard)

Go Go Go Go Go Go Go

Timestamp Inode Delm a c d Filename

FLAG

- Disk Forensics

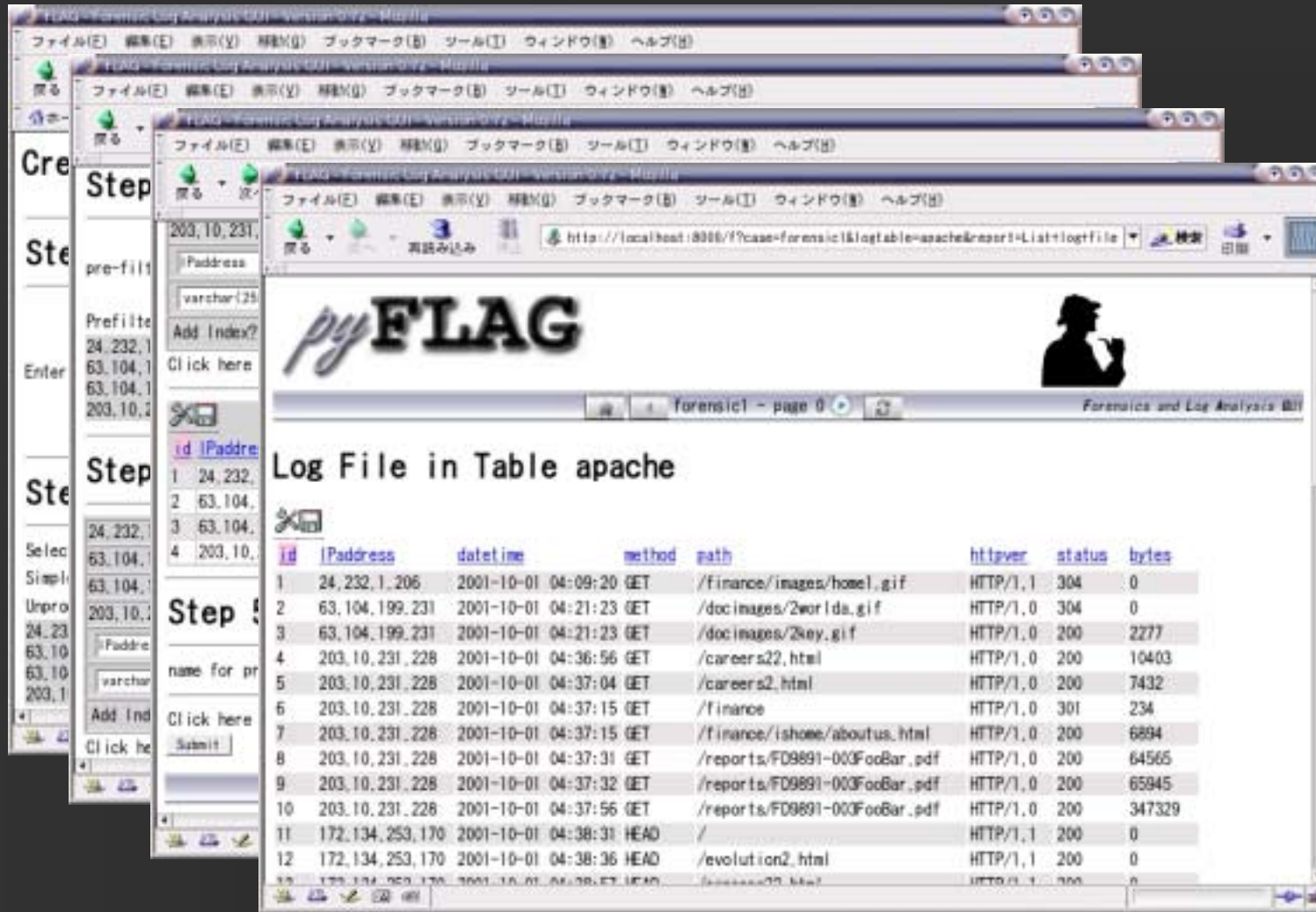


Viewing 'dump' at offset 20480

	Hex Dump	Text View
005000	61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaaaa
005010	61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaaaa
005020	61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaaaa
005030	61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaaaa
005040	61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaaaa
005050	61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaaaa
005060	61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaaaa
005070	61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaaaa
005080	61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaaaa
005090	61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaaaa
0050a0	61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaaaa
0050b0	61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaaaa
0050c0	61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaaaa
0050d0	61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaaaa
0050e0	61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaaaa

rawデータのダンプ

FLAG

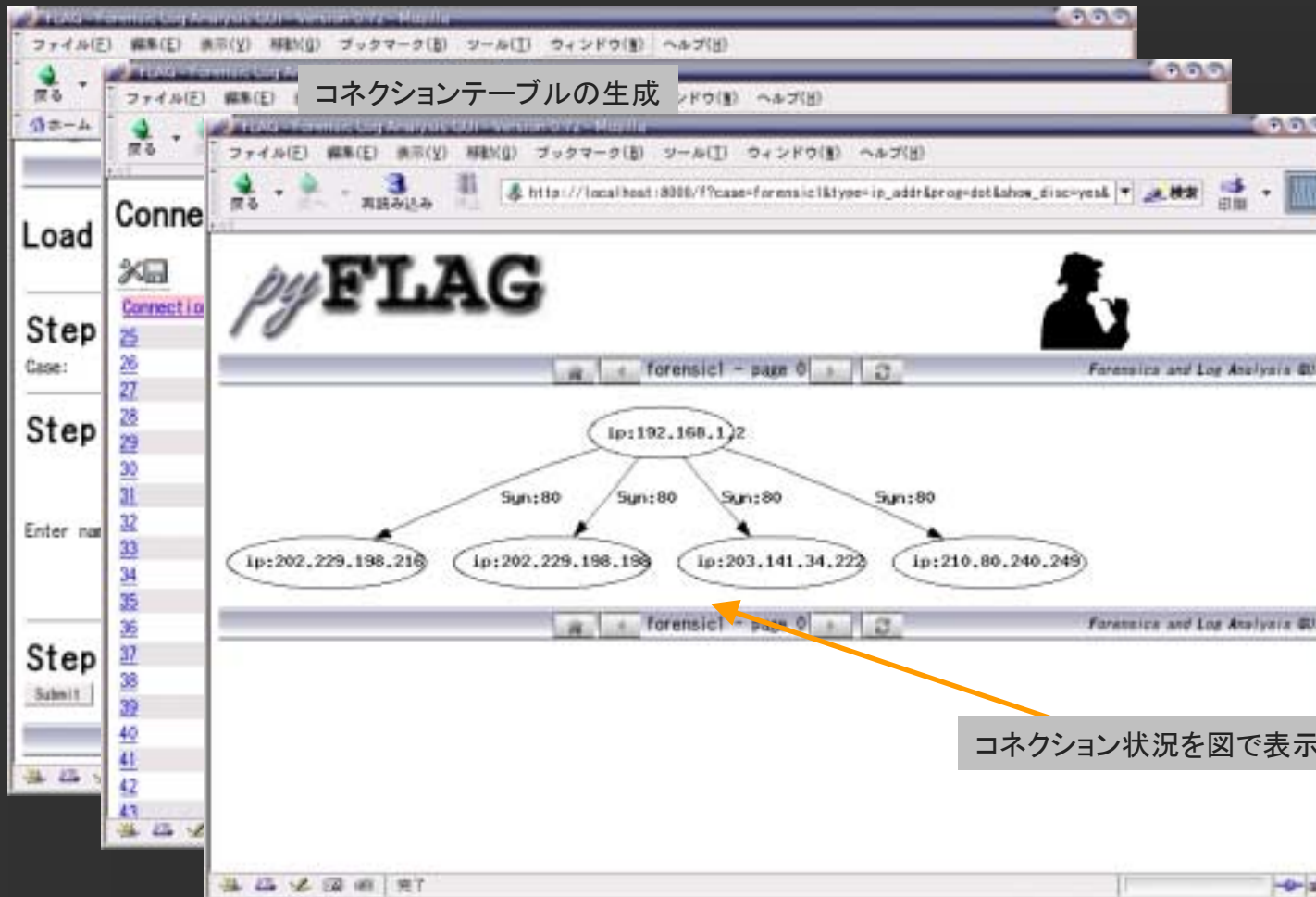


The screenshot displays the FLAG web application interface. The main content area shows a table titled "Log File in Table apache" with the following data:

id	IPaddress	datetime	method	path	httpver	status	bytes
1	24,232,1,206	2001-10-01 04:09:20	GET	/finance/images/home1.gif	HTTP/1.1	304	0
2	63,104,199,231	2001-10-01 04:21:23	GET	/docimages/2worlda.gif	HTTP/1.0	304	0
3	63,104,199,231	2001-10-01 04:21:23	GET	/docimages/2key.gif	HTTP/1.0	200	2277
4	203,10,231,228	2001-10-01 04:36:56	GET	/careers2.html	HTTP/1.0	200	10403
5	203,10,231,228	2001-10-01 04:37:04	GET	/careers2.html	HTTP/1.0	200	7432
6	203,10,231,228	2001-10-01 04:37:15	GET	/finance	HTTP/1.0	301	234
7	203,10,231,228	2001-10-01 04:37:15	GET	/finance/ishome/aboutus.html	HTTP/1.0	200	6894
8	203,10,231,228	2001-10-01 04:37:31	GET	/reports/FD9891-003FooBar.pdf	HTTP/1.0	200	64565
9	203,10,231,228	2001-10-01 04:37:32	GET	/reports/FD9891-003FooBar.pdf	HTTP/1.0	200	65945
10	203,10,231,228	2001-10-01 04:37:56	GET	/reports/FD9891-003FooBar.pdf	HTTP/1.0	200	347329
11	172,134,253,170	2001-10-01 04:38:31	HEAD	/	HTTP/1.1	200	0
12	172,134,253,170	2001-10-01 04:38:36	HEAD	/evolution2.html	HTTP/1.1	200	0

■ FLAG

— Tcpdump Analysis



コネクションテーブルの生成

pyFLAG

forensic1 - page 0

Forensics and Log Analysis GUI

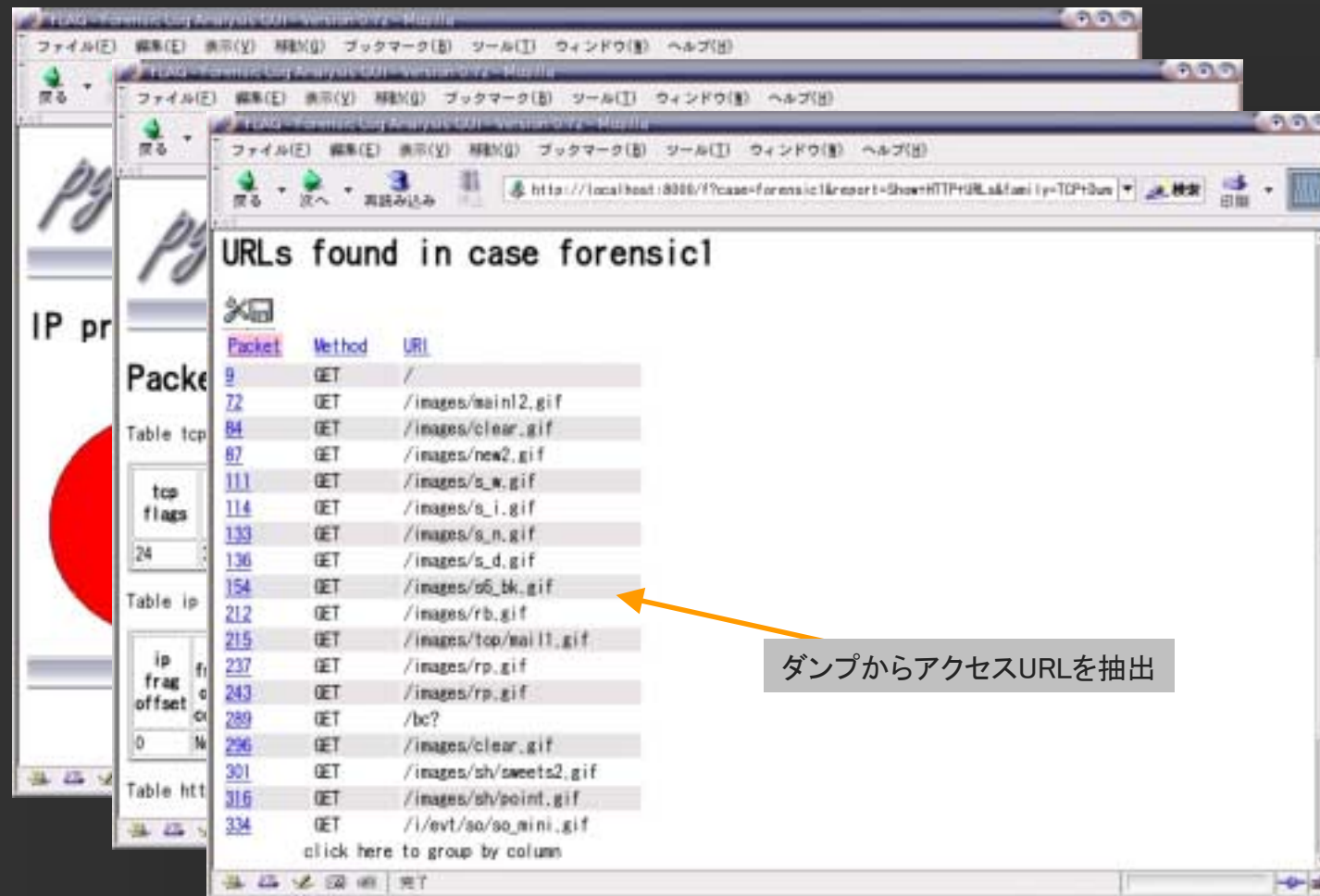
```

graph TD
    Root([ip:192.168.1.2]) -- Syn:80 --> Node1([ip:202.229.198.216])
    Root -- Syn:80 --> Node2([ip:202.229.198.198])
    Root -- Syn:80 --> Node3([ip:203.141.34.222])
    Root -- Syn:80 --> Node4([ip:210.80.240.249])
  
```

コネクション状況を図で表示

FLAG

Tcpdump Analysis



URLs found in case forensic1

Packet	Method	URI
9	GET	/
72	GET	/images/main12.gif
84	GET	/images/clear.gif
87	GET	/images/new2.gif
111	GET	/images/s_w.gif
114	GET	/images/s_i.gif
133	GET	/images/s_n.gif
136	GET	/images/s_d.gif
154	GET	/images/s6_bk.gif
212	GET	/images/rb.gif
215	GET	/images/top/mail1.gif
237	GET	/images/rp.gif
243	GET	/images/rp.gif
289	GET	/bc?
296	GET	/images/clear.gif
301	GET	/images/sh/sweets2.gif
316	GET	/images/sh/point.gif
334	GET	/i/evt/so/so_mini.gif

click here to group by column

ダンプからアクセスURLを抽出

■ FLAG

- 現時点での注意点
 - Debian推奨 (RedHat系でもOK)
 - mysql入りバイナリパッケージの使用を推奨 (pyflag_0.74_bin_mysql)
 - バイナリパッケージではいくつかのpluginがdisableになっている (例えばUnstructuredDiskメニュー) ので、使用する場合は若干手直しが必要
 - 例: plugins/UnstructuredDisk.pycを削除
 - UnstructuredDisk.pyの中のActive=FalseをTrueに変更
 - 複数の別ソフトウェアのインストールが必要
 - Network解析用にetherreal
 - グラフ表示用にgraphviz
 - アンチウイルス機能でclamav
 - まだVer 0.7
 - DBエラーとかが時々出る

不正か否かの判断

- ありえそうな例え話を一つ

- 勝手にカードが使われていた！
 - ある日カードの明細を見ると、自分で買った記憶が無いものが購入されていた
 - カード管理はきちり行っている。情報が漏れるはずはない
 - 自分は絶対使っていないので、これはスキミングされたか、システムの不具合

- でも、システム担当者から見ると。。。
 - ログを調べたが本人のアカウントで普通に使用されている
 - カードシステムに不正侵入された形跡は無い
 - スキミングされたのかもしれないけれど、本当にそうかどうかはわからない
 - 利用者のPCが不正侵入されてるのでは？
 - 実は使ったのを忘れていただけでは？
 - 本当は使ってるのに、うそをついているのでは？

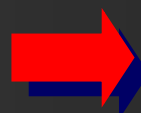
情報の不足による判断の限界
不正かどうかはシステムからは判断ができない

■ どこまで調べれば判断できるか

- ログの調査が全て、ではない
- ディスクの調査が全て、ではない
- ネットワークの調査が全て、ではない
- 1台の機器を調べれば終わり、ではない
- 一人の人間に状況を聞けば終わり、ではない
- 例えば、アカウントを詐称されていたら？
- 例えば、有権限者による不正操作だったら？

■ 何が正規のアクセスで何が不正なアクセスか？

- 機械の記録だけでは判断を行うのに十分ではないのは自明

 **単純な情報収集による
問題解決の限界**

■ 現実のforensic現場はどうやっている???

■ real forensicの世界

— First Responderの役割

- 現場を確保する
- 人身の安全を確保する。すでに現場にいる人々だけでなく、これから到着する人々についても
- 必要に応じてけが人の手当てをする
- 目撃者、被害者、容疑者を、それぞれ別々の場所に確保する
- 見張りを置き、活動記録をとりはじめる
- すべての資料を確保する**

ログ
ネットワークダンプ
ディスクダンプ

■ 「完全科学捜査マニュアル」より

— ここまでは単なる情報収集

■ 確保された情報はただの「点」

- 「点」をつなぐ作業が重要
 - 「点」と「点」をつないで「線」に、「線」と「線」をつないで「面」に。しかし、これでもまだ不正かどうかはわからない
- 不正かどうかはそこにどのような「意図」があるかによる
- 情報に意図が加わることで有益な判断材料となる

■ Intelligence Cycle

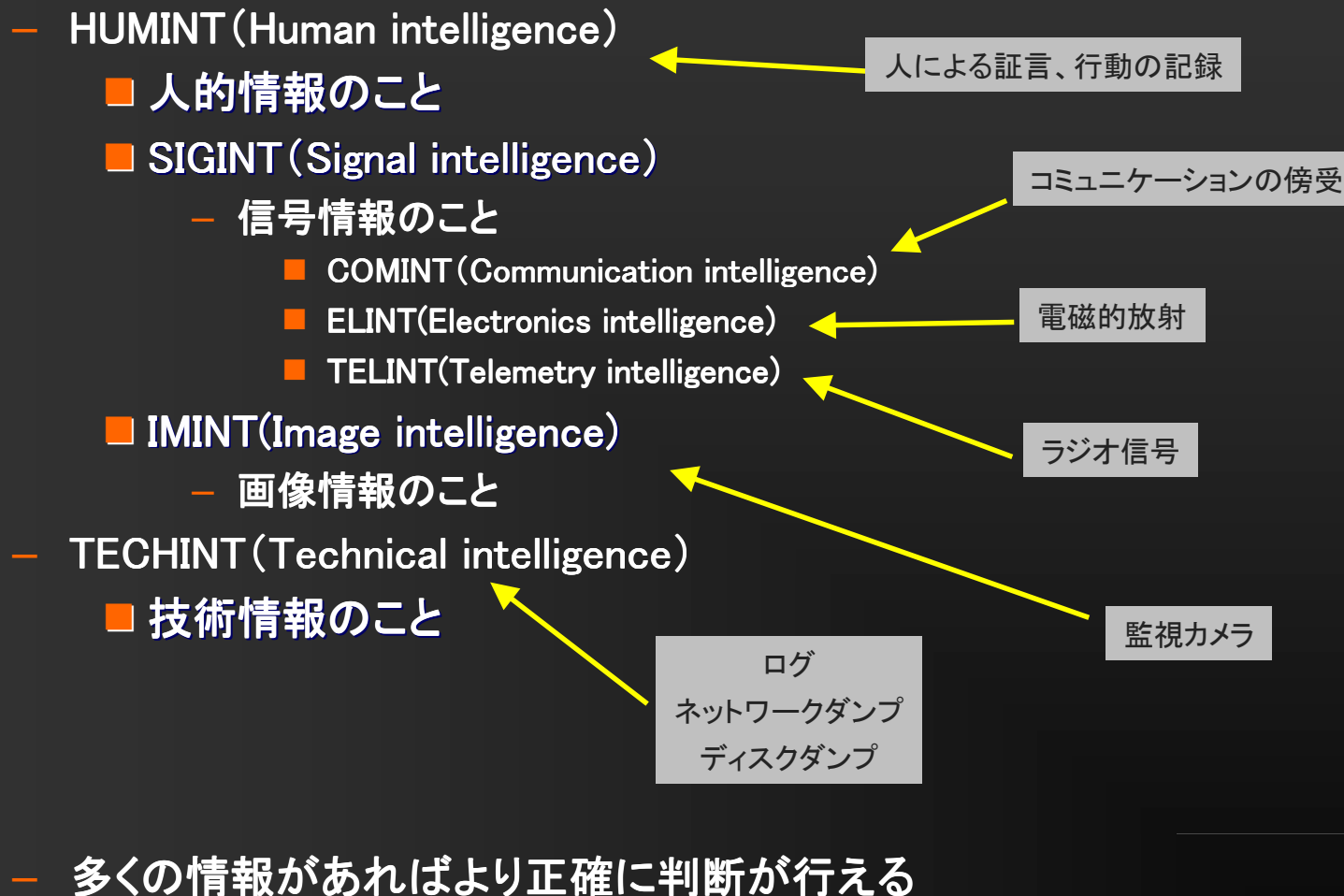
- 単なる「情報」を目的に添った形の「知識」にするプロセス
- 以下、CIAモデルの例
 - Planning and Direction
 - 計画と指示(要求事項)
 - Collection
 - 情報の収集
 - Processing
 - 情報の加工
 - All-Source Analysis and Production
 - 情報の統合、分析、評価、解釈
 - Dissemination
 - 情報の配布



- The Intelligence Cycle is the process of developing raw information into finished intelligence for policymakers to use in decisionmaking and action.

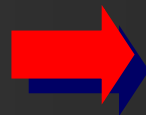
※ CIA Webより (http://www.cia.gov/cia/publications/facttell/intelligence_cycle.html)

■ Intelligence Cycleにおける情報の種類



■ 不正か否かの判断をするには

- 要求事項を明確にし
 - いつからいつまで、何に対して、何のために
- 要求事項にしたがって情報収集し
 - 人的情報と技術情報の収集
- 情報を見やすい形に加工する
 - ツール等と使用
- 時間を軸に情報を整理、判断
- 要求事項に沿ったoutputを行う

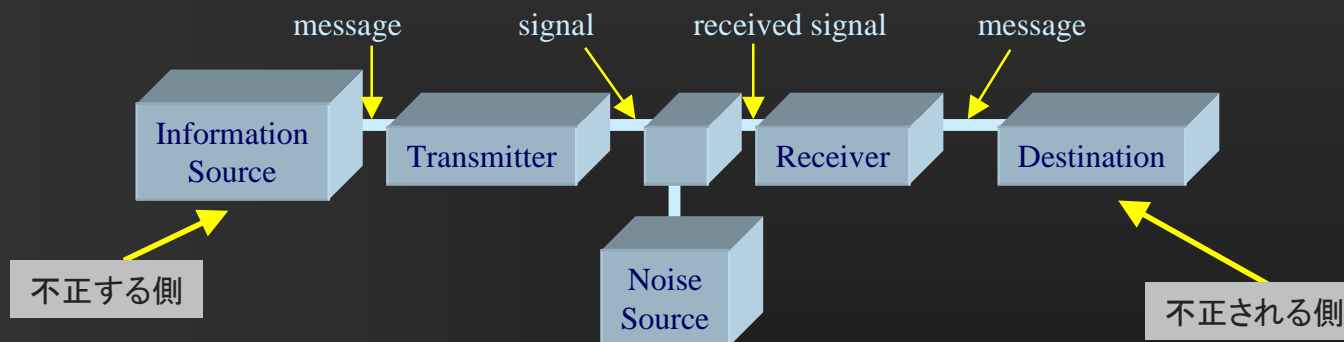


これらを繰り返し
outputの精度を上げる

リスクの発見

■ 情報とは

- 情報の定義
 - 情報って何？
- 情報の流れ



■ セキュリティとは

- 誰にとっての安全か

■ 情報セキュリティって何？

- 情報セキュリティも「情報」であることに変わりはない
- 情報理論から逸脱することはない

- 情報セキュリティの「リスク」とは何かについて考えてみる
 - リスクが高い状態とは
 - 安全が脅かされる可能性が高い
 - $\text{risk} = \text{脅威の発生頻度} \times \text{被害の大きさ}$ (コートニイ理論)
 - $\text{risk} = \text{threat} \times \text{vulnerability}$ (SANS Institute)
 - $\text{risk} = \text{資産価値} \times \text{脅威} \times \text{脆弱性}$ (BS7799-GAP方式)
 - 脅威や脆弱性が増加すれば安全でなくなる
 - 安全ではない＝不利益をこうむる確率が高い
 - 脅威や脆弱性は不利益を起こす確率をあげる要因
 - 安全＝安定としてみる
 - 安定状態＝変化の少ない状態
 - 変化の少ない状態＝変化する確率が低い状態
 - ここで情報エントロピー
 - 不確定度を表す(確率)
 - 「情報」を得ることで不確定度を減らす
 - 安定＝不確定度が低い
 - 不確定度が低い＝情報エントロピーが低い
 - 脆弱性
 - 内部的な情報エントロピーの増加
 - 脅威
 - 外部的な情報エントロピーの増加

■ リスクが高い状態

- 情報エントロピーが最大
- 不確定なことがめちゃくちゃたくさんある
- こうなると何が起きかわからない＝不安定＝安全ではない
- リスクを低減するには、情報エントロピーを減らす。つまり、情報を与える

■ 変化を見極める

- 「巨視的状态」と「微視的状态」
 - 10秒以内に訪れる危機と10万年以内に訪れる危機はどちらがリスクが高いか
 - 10行に1行不正アクセスがあるのと10万行に1行不正アクセスがあるのは同じか
 - 同じ事象でも巨視的か微視的かで異なってくる
 - 不正侵入というのは、おきないはずのことがおきている状態。つまり、確率の低いことが発生している。確率の低いことほど情報量は大きい
- 「量」と「時間」を考える
 - 情報が多ければ情報エントロピーは減少する
 - 情報エントロピーが減少している状態を作るには、母数を広げるか時間をかける
 - つまり、巨視的に見て情報エントロピーの発生が少ない状態を維持すれば、確率の低いことが発生した場合の影響が大きく、気がつきやすい
- 定常状態
 - ある目的を達成する場合に定常状態を保って行えば、エントロピーの発生は最小となる

■ 要するに・・・

- 安定した状態の維持があって初めて不正を正確に把握できる、ということ
- もっと平たく言うと、安定して運用された環境を作りましょう、ということ

■ 変化をとらえる

- 情報がどう変化しているか？
- その変化量は？
- 変化した意味は？

■ 例えば

- IDSとは何をしているのか
 - 微視的状態の変化を検出
- 定点観測とは何をしているのか
 - 巨視的状態の変化を検出

※IDSや定点観測はもと評価されてもよいはず。
使う側がうまくつかえていないだけ。

- IDSを定点観測することで精度高く変化をつかむことはできる？
- 変化がおきたとき、つまり、ネットワークが定常状態から非定常状態へ遷移したとき、安全な状態ではなくなる
- その安全ではない状態に対して、不正をとらえるかどうか、何らかの「意図」が加わることで初めてIntelligenceとして情報の活用が可能となる

■ 結局、不正侵入を発見するには

- 不正侵入: 意図の把握(人的情報)
- 発見: 情報エントロピー変化量の検出(技術情報)

- ...の把握につけるのではないか

- IDSを置いていけばよいというものではない
- 特定のキーワードを検出すればよいというものではない
- 閾値を超えればアラートあげればよいというものでもない

最後に

■ まとめ

- 最後のほうがぜんぜん具体的じゃないのですが…
 - 結局、不正侵入の発見でやるべきことは
 - まず、安定したシステムにすること
 - 継続的に観察しつづけること
 - 情報の収集を心がけること

安定した運用基盤
情報が収集できる仕組み

■ 謝辞

- 「HDDデータ復旧の可能性」について、株式会社 日立グローバルストレージテクノロジーズの三枝様、高師様、渡部様、上村様に多大なるご協力をいただきました。この場をお借りしてお礼を申し上げます。

■ 参考文献・URL

- Linux Filesystems HOWTO
- Linux Ext2fs Undeletion mini-HOWTO
- FIRST Conference 2002 “Forensic Discovery & Hawaii” dan farmer & Wietse Venema
- 「Linuxのブートプロセスをみる」白崎博生 著 アスキー
- Sleuthkit <http://www.sleuthkit.org/sleuthkit/index.php>
- Autopsy <http://www.sleuthkit.org/autopsy/index.php>
- HELIX <http://www.e-fense.com/helix/>
- 「セキュリティウォリア」サイレス・パイカリ、アントン・チュバキン 著 オライリージャパン
- 「不正アクセス調査ガイド」渡辺勝弘、伊原秀明 著 オライリージャパン
- 「ATA(IDE)/ATAPIの徹底研究」CQ出版社
- pyFLAG <http://pyflag.sourceforge.net/>
- 「情報セキュリティ技術大全」ロス・アンダーソン 著 日経BP社
- 「完全科学捜査マニュアル」N.E.ゲンジ 著 河出書房新社
- 「鑑識捜査三十五年」岩田政義 著 中公文庫
- 「なぜ、正しく伝わらないのか」ジョン・ヒューズ＝ウィルソン 著 ビジネス社
- 「インテリジェンス入門」北岡元 著 慶應義塾大学出版会
- CIA Factbook on Intelligence 2002
- 「エントロピーの科学」細野俊夫 著 コロナ社
- 「情報とは」青柳忠克 著 産業図書
- 「時間のなぞ」Newton Press

ご清聴ありがとうございました

ありがとう

