Performing Digital Forensics *with Open Source tools*

Dimitris Glynos
{ dimitris *at* census-labs.com }

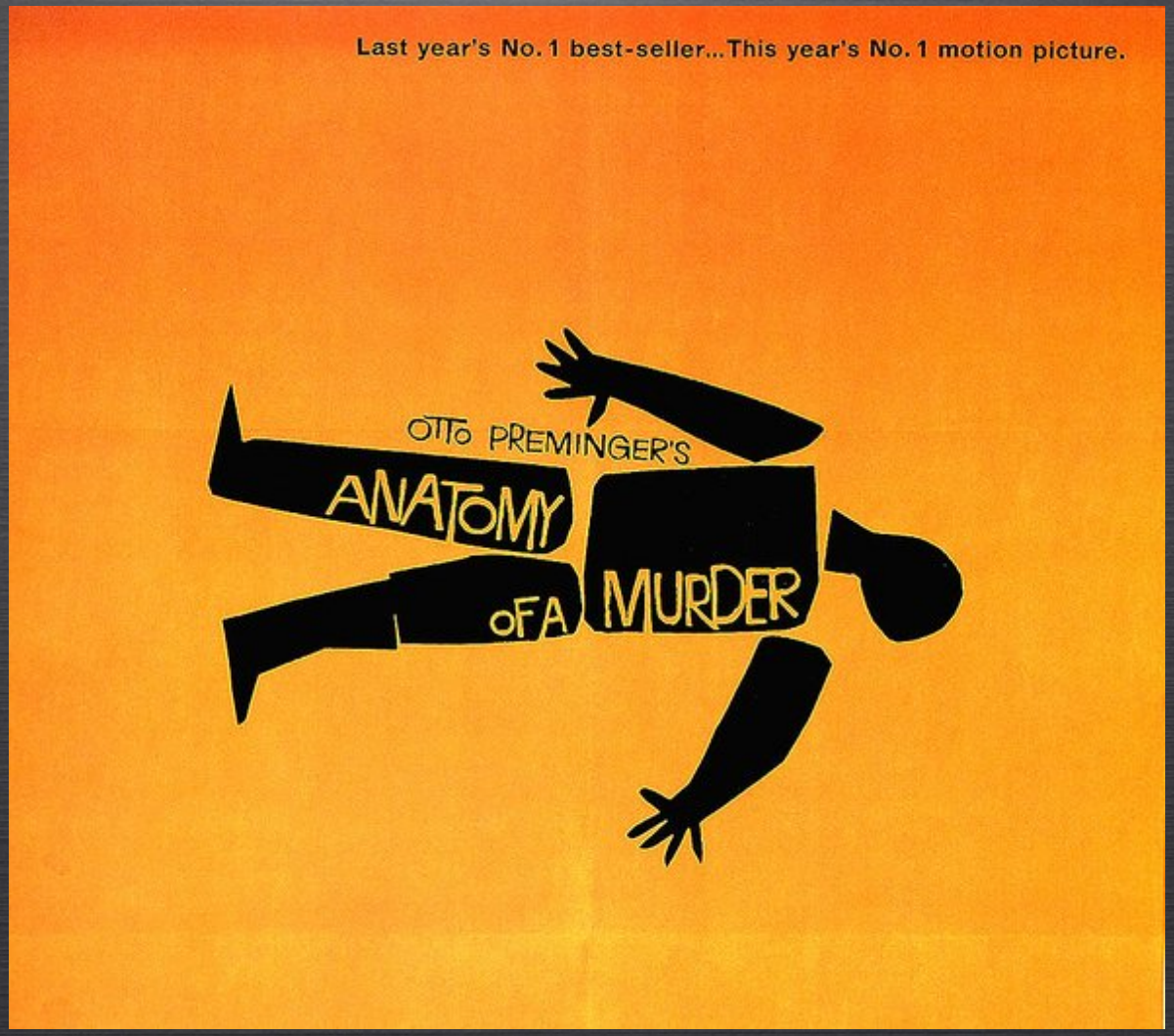
Census, Inc.

FOSSCOMM 2011, Patras

Overview

- Introduction
- Data Acquisition
- Data Examination
- Report Preparation
- Conclusions

Introduction



Digital Forensics

- Electronic transactions leave digital trails
- A Digital Forensics investigator follows these trails searching for evidence
- This evidence may later be used in court to combat crimes such as cyber-attacks, digital fraud, corporate espionage and others

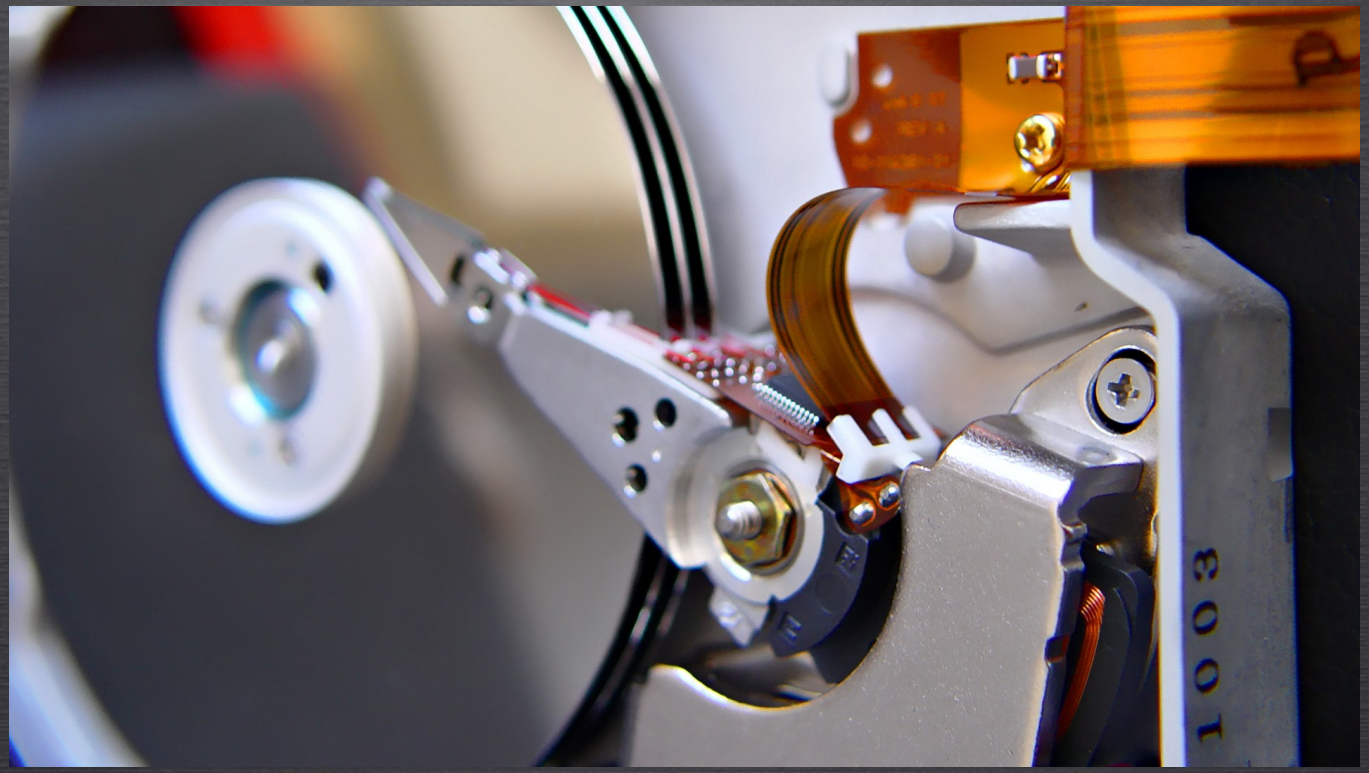
When to perform a Digital Forensics investigation

- A crime has been commited and related evidence must be presented in court
- An incident has occured and the IT department needs more information in order to perform proper service recovery
- Upper management needs inside information on the actions of a rogue employee

Incident Response

- Find out what you will be allowed to examine
- Gather as much volatile information as possible
  - Processes
  - Drivers
  - Sockets
  - Network traffic
- Use statically compiled tools (`busybox?`) and execute these from external media
- Collect disk data
- Look for traces of known malware
- Analyze captured data
- Create a short report to assist service recovery
- Work on longer report

Data Acquisition



The Data Acquisition Process

- Gather information about the host
- Collect volatile data (memory, network dumps, mounted decrypted filesystems)
- Collect disk data
- Gather other related media (logfiles, documents, CDROMs, images of flash drives etc.)
- Acquired data are hashed
- Fill in *Chain of Evidence* document

Acquiring volatile data

- Dump the RAM
  - Through Firewire
  - Windows
    - No OSS solution available that works for a good set of Windows releases.
    - Lots of freeware alternatives.
  - Linux
    - No more /dev/mem, /dev/kmem
    - Dump RAM using a kernel module (`fmem`)
- Capture network traffic (*pcap* format)
  - `tcpdump`
  - `wireshark`
  - `ettercap`

Acquiring disk data

- The Linux kernel supports a large number of disk controllers
- Boot from Linux CD but don't mount anything!
- Create HDD images using a known good version of `dcfldd`
  - An enhanced version of `dd`
  - Developed at Dept. of Defense Comp. Forensics Lab
  - Hashes data while copying them from the input device
- If you encounter a faulty drive use `ddrescue`
- Watch out for Host Protected Areas (HPA) and Device Configuration Overlays (DCO)
- You will need RAID support to capture RAID volumes

Data Examination

Forensic analysis software

- First there was TCT (The Coroner's Toolkit)
- Then came the Sleuthkit
- Autopsy provided a web front-end for Sleuthkit
- Now there's a plethora of new software around, with `pyflag` being perhaps the most promising one
  - supports AFF format
  - stores computed/extracted metadata in database allowing for faster queries
  - performs log analysis
  - supports network forensic analysis
  - supports memory forensic analysis

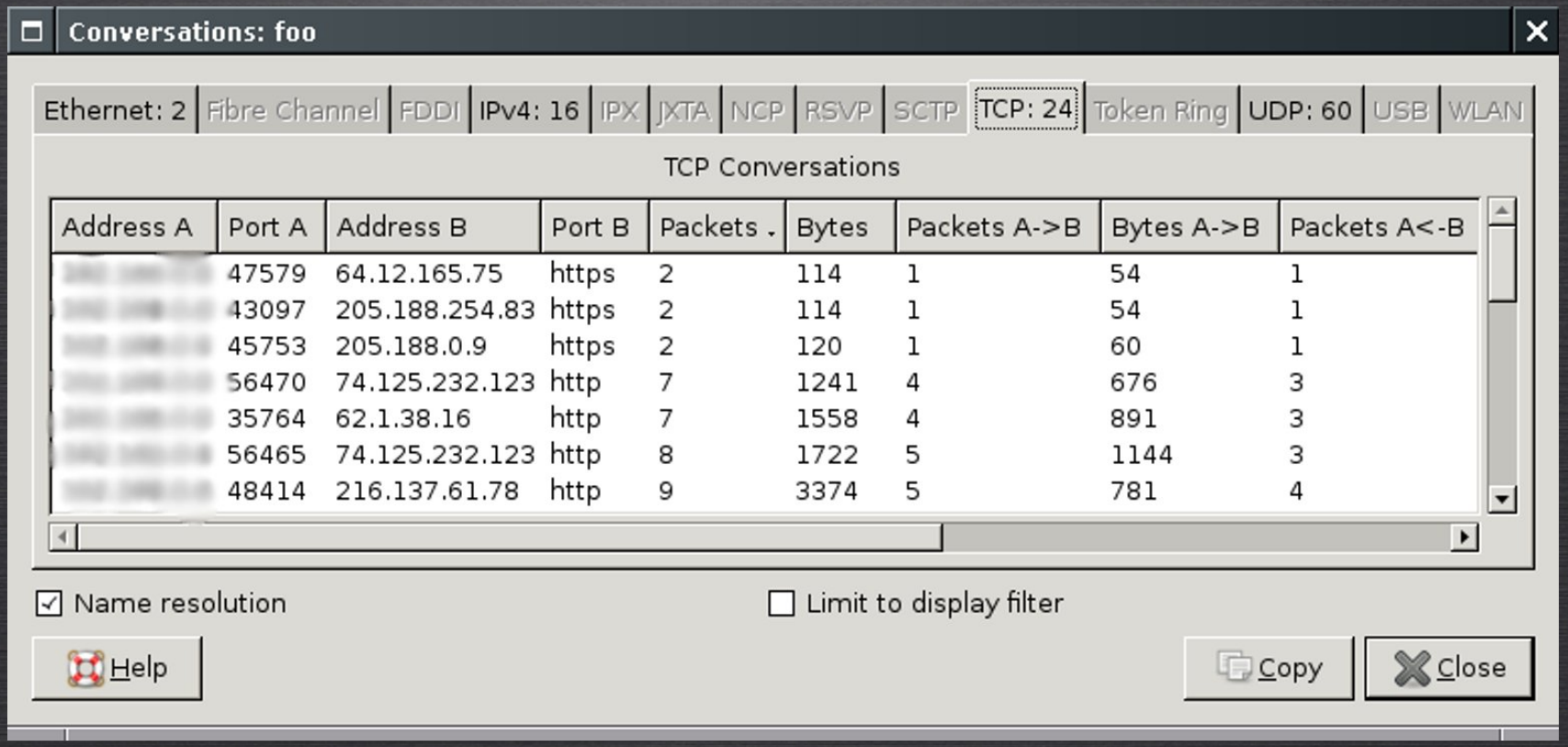
Memory dump analysis

- The `Volatility` framework analyzes memory dumps from Windows XP SP2/SP3 and some GNU/Linux (beta) systems
- Identifies running processes
- Identifies open sockets and connections
- Performs process memory space analysis (memory maps, loaded libraries, list of open files)

```
# python2.6 volatility connections -f /tmp/xp-NIST-sample
Local Address              Remote Address             Pid
127.0.0.1:1056             127.0.0.1:1055             2160
127.0.0.1:1055             127.0.0.1:1056             2160
192.168.2.7:1077           64.62.243.144:80           2392
192.168.2.7:1082           205.161.7.134:80           2392
192.168.2.7:1066           199.239.137.200:80         2392
```

Network traffic analysis

- `Wireshark` is your friend!
- Identify talking hosts
- Identify abnormal traffic



Address A	Port A	Address B	Port B	Packets	Bytes	Packets A->B	Bytes A->B	Packets A<-B
	47579	64.12.165.75	https	2	114	1	54	1
	43097	205.188.254.83	https	2	114	1	54	1
	45753	205.188.0.9	https	2	120	1	60	1
	56470	74.125.232.123	http	7	1241	4	676	3
	35764	62.1.38.16	http	7	1558	4	891	3
	56465	74.125.232.123	http	8	1722	5	1144	3
	48414	216.137.61.78	http	9	3374	5	781	4

Image analysis and file recovery

DEMO

Looking for data

- The forensic equivalent of grep on a file

fosscomm:host1:vol2

File Analysis | Keyword Search | File Type | Image Details | Meta Data | Data Unit | Help | Close

Searching for ASCII: Done
Saving: Done
2 hits- link to results

Searching for Unicode: Done
Saving: Done
0 hits

New Search

2 occurrences of `Accepted password for` were found
Search Options:
ASCII
Case Sensitive

Fragment 113467 (Hex - Ascii)
1: 1666 (11]: Accepted password for glyn)
2: 2584 (14]: Accepted password for glyn)

`Accepted password for` was not found
Search Options:
Unicode
Case Sensitive

Previous | Next
Export Contents | Add Note

ASCII (display - report) * Hex (display - report) * ASCII Strings (display - report)
File Type: ASCII text
Fragment: 113467
Status: Allocated
Group: 3
Find Meta Data Address

```
1664  3a204163 63657074 65642070 61737377   : Ac cept ed p assw
1680  6f726420 666f7220 676c796e 6f732066   ord  for  glyn os f
1696  726f6d20 3137322e 32302e30 2e312070   rom  172. 20.0 .1 p
1712  6f727420 33383132 39207373 68320a4d   ort  3812 9 ss h2.M
1728  61792020 32203130 3a34353a 33382064   ay   2 10 :45: 38 d
1744  65626961 6e207373 68645b34 3431315d   ebia n ss hd[4 411]
1760  3a207061 6d5f656e 76287373 68643a73   : pa m_en v(ss hd:s
1776  65746372 6564293a 20556e61 626c6520   etcr ed):  Una ble
1792  746f206f 70656e20 656e7620 66696c65   to o pen  env  file
1808  3a202f65 74632f64 65666175 6c742f6c   : /e tc/d efau lt/l
1824  6f63616c 653a204e 6f207375 63682066   ocal e: N o su ch f
1840  696c6520 6f722064 69726563 746f7279   ile  or d irec tory
1856  0a4d6179 20203220 31303a34 353a3338   .May   2  10:4 5:38
1872  20646562 69616e20 73736864 5b343431    deb ian  sshd [441
1888  315d3a20 70616d5f 756e6978 28737368   1]:  pam_ unix (ssh
1904  643a7365 7373696f 6e293a20 73657373   d:se ssio n):  sess
1920  696f6e20 6f70656e 65642066 6f722075   ion  open ed f or u
1936  73657220 676c796e 6f732062 79202875   ser  glyn os b y (u
1952  69643d30 290a4d61 79202032 2031303a   id=0 ).Ma y  2  10:
1968  34353a33 38206465 6269616e 20737368   45:3 8 de bian  ssh
```

Find: | Previous | Next | Highlight all | Match case

Linux log recovery

- Most logs in /var/log are text based
- Syslog appends a time prefix to each log entry
- You can search for a time prefix that matches log entries that have been deleted!
  - `Jan 12.*servername`
- Locate the longest version of a log excerpt (you may encounter more than one!)
- Join together the log excerpts found on different disk locations
- ...great fun! (sic)

Building a timeline from Filesystem events

- Gather file activity events from structures of existing and deleted files and encode in *mactime* format
  - Use Sleuthkit's `fls` tool
- Create a timeline by sorting the events in chronological order
  - Use Sleuthkit's `mactime` tool

Filesystem	**m**	**a**	**c**	**b**
Ext2/3	Modified	Accessed	Changed	N/A
FAT	Written	Accessed	N/A	Created
NTFS	File Modified	Accessed	MFT Modified	Created

Quiz #1: What do you see here?

```
Mon May 02 2011 13:45:35 .a.. /etc/protocols
                         .a.. /etc/hosts.allow
                         .a.. /etc/hosts.deny
                         .a.. /etc/ssh/moduli
Mon May 02 2011 13:45:37 .a.. /etc/pam.d/sshd
Mon May 02 2011 13:45:38 .a.. /etc/shadow
Mon May 02 2011 13:45:39 .a.. /lib/terminfo/x/xterm
Mon May 02 2011 13:46:25 mac. /var/log/lastlog
Mon May 02 2011 13:46:29 .a.. /home/john
Mon May 02 2011 13:48:04 .a.. /etc/pam.d/su
Mon May 02 2011 13:50:27 m.c. /etc/passwd
```

Quiz #2: What do you see here?

```
15:13:29 .a.. /tmp/...
15:13:40 .a.. /etc/wgetrc
         .a.. /usr/bin/wget
15:14:02 ..c. /tmp/.../la.c
15:14:40 .a.. /tmp/.../la.c
         .a.. /usr/include/stdio.h
         .a.. /usr/lib/gcc/i486-linux-gnu/4.3/cc1
15:14:41 .a.. /usr/include/pcap/pcap.h
15:14:42 .a.. /usr/bin/as
         .a.. /usr/lib/crt1.o
15:14:43 m.c. /tmp/.../t
15:14:48 .a.. /tmp/.../t
```

Quiz #3: What do you see here?

```
10:04:01 macb C:/Documents and Settings/john/
              Local Settings/Temporary
              Internet Files/Content.IE5/XXXXXXXX/
              ABCDE8FG
10:04:05 .a.. C:/Program Files/Adobe/Acrobat 9.0/
              Acrobat/plug_ins/PfuSsPCapPI/
              PfuSsPCapPI.api
10:04:12 m.c. C:/Documents and Settings/john/
              Local Settings/Temporary
              Internet Files/Content.IE5/XXXXXXXX/
              sexy.pdf
10:05:00 .a.. C:/Documents and Settings/john/
              Local Settings/Temp/foo.bat
```

Windows registry timeline

- Windows keeps an MTIME record for each registry key
- We can browse Windows registry files with `reglookup`
- ..and sort them in chronological order with `reglookup-timeline`

```
# reglookup-timeline /mnt/WINDOWS/system32/config/system
MTIME,FILE,PATH
2010-09-23 06:55:20,system,/WPA/MediaCenter
2010-09-23 07:07:44,system,/WPA/SigningHash-XXXXXXXXXXXXX
2010-09-23 07:07:49,system,/WPA/Key-YYYYYYYYYYYYYYYYY
...
```

File identification

Check

- with databases of known file hashes
- with databases of known file patterns
- information entropy
- contents manually

NSRL DB

- NIST's National Software Reference Library
- Hash values of known files
  - md5 & sha1
  - file origin information (filename, system)
- 7.4GB as of June 2010 (updated every 3 months)
- They are admissible as evidence by US courts
  - All data is traceable to its origin
  - NIST keeps copies at secure facility
- Sleuthkit's `hfind` searches an indexed NSRL DB

```
$ hfind NSRLFile.txt 5f7eaaf5d10e2a715d5e305ac992b2a7
5f7eaaf5d10e2a715d5e305ac992b2a7 CHKDSK.EXE
5f7eaaf5d10e2a715d5e305ac992b2a7 chkdsk.exe
### time: real 0m0.003s, user 0m0.004s, sys 0m0.000s
```

The File utility

- The *magic* database associates data with a file type, based on known patterns, e.g.

```
0 string MZ
>0x18 leshort <0x40 MS-DOS executable
```

- The `file` utility consults the *magic* database and reports the type of a file

```
$ file /tmp/obj
/tmp/obj: PE32 executable for MS Windows (GUI)
          Intel 80386 32-bit
```

Antivirus check

- Antiviruses use signatures (content hashes and pattern-matching) to identify malicious software
- ClamAV is an Open Source Antivirus Engine
  - It detects Trojans, Viruses, Malware and other (possibly) unwanted applications irregardless of their filename

```
# freshclam
ClamAV update process started at Wed Apr 27 ...
bytecode.cld updated (version: 143, sigs: 40, ...)
Database updated (952543 signatures) from
  db.local.clamav.net
$ clamscan --detect-pua /tmp/obj2
/tmp/obj2: PUA.Script.PDF.EmbeddedJS FOUND
```

Sorting files

- File sorting allows the investigator:
  - to filter out files that are known and good
  - to focus the investigation on files of a certain type (e.g. Microsoft Word documents)
- Sleuthkit's `sorter` sorts allocated and unallocated files according to both NSRL-type and *magic*-type databases
  - It also identifies files that have an extension mismatch!

Sorting files

- `sorter` example on a tiny ext2 image with 2 present and 1 deleted files

```
$ sorter -d . -s /tmp/img
$ tree
.
|-- documents
|   `-- mpi-12.pdf
|-- documents.txt
|-- images
|   |-- mpi-13.jpg
|   `-- mpi-14
|-- images.txt
`-- sorter.sum
```

Sorting files

```
$ cat images.txt
name.jpg
  JPEG image data, EXIF standard
  Image: /tmp/mpi  Inode: 13
  Saved to: images/mpi-13.jpg

$OrphanFiles/OrphanFile-14
  JPEG image data, JFIF standard 1.01
  Image: /tmp/mpi  Inode: 14
  Saved to: images/mpi-14
```

Checking file metadata

- Look at a file's internal metadata to obtain information about the environment it was created in
  - `exifprobe`
  - `pdfinfo`
  - ...
- Do you suspect that steganography is taking place?
  - Check with tools like `stegdetect`
  - Check your sample data against various steganography decoding tools

Information entropy

- Measuring the information entropy of a file may give us a hint as to whether a file contains:
  - compressed data
  - random data
  - encrypted data (well, not always)
- `ent` to the rescue!
  - measures entropy
  - performs x^2 test
  - calculates arithmetic mean
  - calculates monte carlo value for π
  - measures serial correlation coefficient

Information entropy

	Ent.	Comp.	x^2	exceed
urandom	7.996433	0%	256.63	50%
calc.exe	6.003569	24%	1661018.85	0.01
calc.zip	7.992996	0%	487.11	0.01
calc.gpg	7.996440	0%	257.08	50%

	Mean	MC	MC error	Serial Cor.
urandom	127.2937	3.102924246	1.23	-0.005558
calc.exe	102.2017	3.080255310	1.95	0.379018
calc.zip	128.2233	3.114373668	0.87	-0.005195
calc.gpg	127.3222	3.142988717	0.04	-0.002486

- AES256 encrypted data (`calc.gpg`) look very much like random data!

Manual file inspection

- Use a hex editor to inspect the file structure
  - `hd`
- Extract any strings available
  - `strings file`
    - extracts ASCII strings
  - `strings -e l file`
    - extracts UTF-16 little endian strings

Reverse engineering

- static / runtime analysis in protected environment (e.g. in qemu guest)
- for Windows binaries
  - pefile / peid
  - ndisasm
  - winedbg / zerowine
  - metasm / radare
- for Linux binaries
  - readelf
  - objdump
  - strace / ltrace
  - metasm / radare / elfsh

File carving

- Use signatures to locate files within raw data
  - Search for a particular file
  - Search for a particular file type
- Structural information is useful in determining the exact length of a file
- `foremost` is a file carver
  - supports a wide variety of file types
  - the user can add more types through the configuration file

```
$ foremost -v -t jpg -i image -o outdir
Num     Name (bs=512)        Size       File Offset
0:      00000134.jpg         33 KB           68608
1:      00000204.jpg         28 KB          104448
```

Windows log recovery

- Windows logs are stored in a record-based binary format (!)
- Part of the textual description of each entry lies within DLL files (!?)
- `grokevt` can parse Windows (evt) logs and turn them into their textual counterparts
  - It resolves the textual descriptions from the corresponding DLL's for logs of known type
- It can also locate Windows log entries within raw disk images (carving!)

```
15367,Error,2011-02-02 10:00:08, Symantec AntiVirus, HOST,
Security Risk Found! Bloodhound.SONAR.1 in File: c:\nc.exe
by: TruScan scan. Action: Leave Alone succeeded.
```

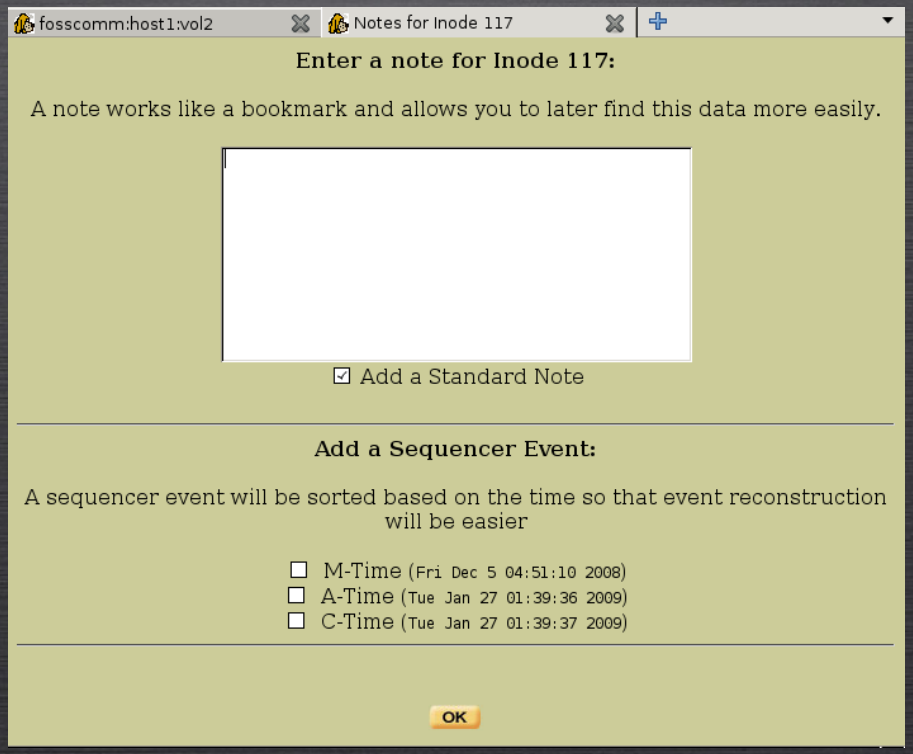
Evidence correlation

- How do you know if a piece of information is trustworthy evidence?
  - Was it found on a tamper-proof medium?
  - Was it produced by a trusted source?
  - Do other evidence also support this?
- Always look for related events
  - A remote login event (a log entry?) may also be supported by Access Time changes to the user's files.
- Combine the evidence under a single timeline
  - Use `log2timeline` to join different types of logs
  - Watch for clock skew between hosts
  - Watch for logs that keep time in UTC or other formats
  - A wall clock reference (time of acquisition?) is always useful!

Report Preparation

Keeping notes

- Document all steps of the investigation process
- Independent investigators must be able to follow all of your steps (and reach the same conclusions!)
- Many GUI forensic analysis tools provide a notes-keeping functionality



Preparing the report

- What usually happens
  - First draft of report goes to client and legal representative
  - Investigator collects feedback (detached notes)
  - Revised copy is sent to client
- The client doesn't edit the report directly, so the investigator is free to use the editing suite of his choice!
  - OpenOffice / LibreOffice
  - XeLaTeX
  - ...
- Tool output is presented in the Appendix
  - You can *pretty-print* this using scripts + XSLT.

Example of an application-generated report

```
                 Autopsy hex Fragment Report

-----------------------------------------------------------------------
                    GENERAL INFORMATION

Fragment: 100360
Fragment Size: 4096

Pointed to by Inode: 49161
Pointed to by files:
  /tmp
  /tmp/.
MD5 of raw Fragment: 6c90e8c78091650a8b19d1043c2c8722  -
MD5 of hex output: c1056d2fc85cb772bfce9cde6b022cf1  -

Image: '/var/lib/autopsy/fosscomm/host1/images/debian.img'
Offset: 63 to 1310399
File System Type: ext

Date Generated: Mon May  2 17:02:49 2011
Investigator: unknown
-----------------------------------------------------------------------
                        CONTENT

0       09c00000 0c000102 2e000000 02000000     .... .... .... ....
16      f40f0202 2e2e0000 00000000 0c000102     .... .... .... ....
32      74636c65 00000000 0c000302 2e2e2e34     tcle .... .... ...4
48      00000000 d00f0601 2e636c65 616e786c     .... .... .cle anxl
64      2e731801 00000000 bc0f0a01 63633670     .s.. .... .... cc6p
80      7752466f 2e6f2e33 00000000 14000a01     wRFo .o.3 .... ....
96      6363574e 6d305343 2e63682d 00000000     ccWN m0SC .ch- ....
112     940f0a01 63636337 45733935 2e6f3435     .... ccc7 Es95 .o45
128     00000000 14000b01 63637552 69486b7a     .... .... ccuR iHkz
144     2e6c642d 00000000 6c0f0b01 63634b6f     .ld- .... l... ccKo
160     71797532 2e6c652e 32353136 33000000     qyu2 .le. 2516 3...
```

Conclusions

Conclusions

- Open Source Landscape: A growing arsenal of forensic tools!
- Many of the tools were created
  - in an "as-needed" basis (by professionals / others)
  - as part of calls in conferences (by the academia)
  - as part of a certification process (by investigators)
- Some of them have been recognized as the "de facto" standard (e.g. `dcfldd`)
- You might find that the tool development process and related research is much more exciting than the actual investigation process itself... :-)

And some rants…

- Need for better coordination between filesystem community and forensic community
  - e.g. once a new filesystem is released, both filesystem and forensic tools should have access to its internal data structures through a common library.
- We've lost a lot (of evidence) in the race towards efficiency
  - Administrators should have the option to switch a filesystem (or logging mechanism) to a more "forensic-friendly" mode.

Questions?

Image courtesy of South Park Studios.