Forensic Discovery

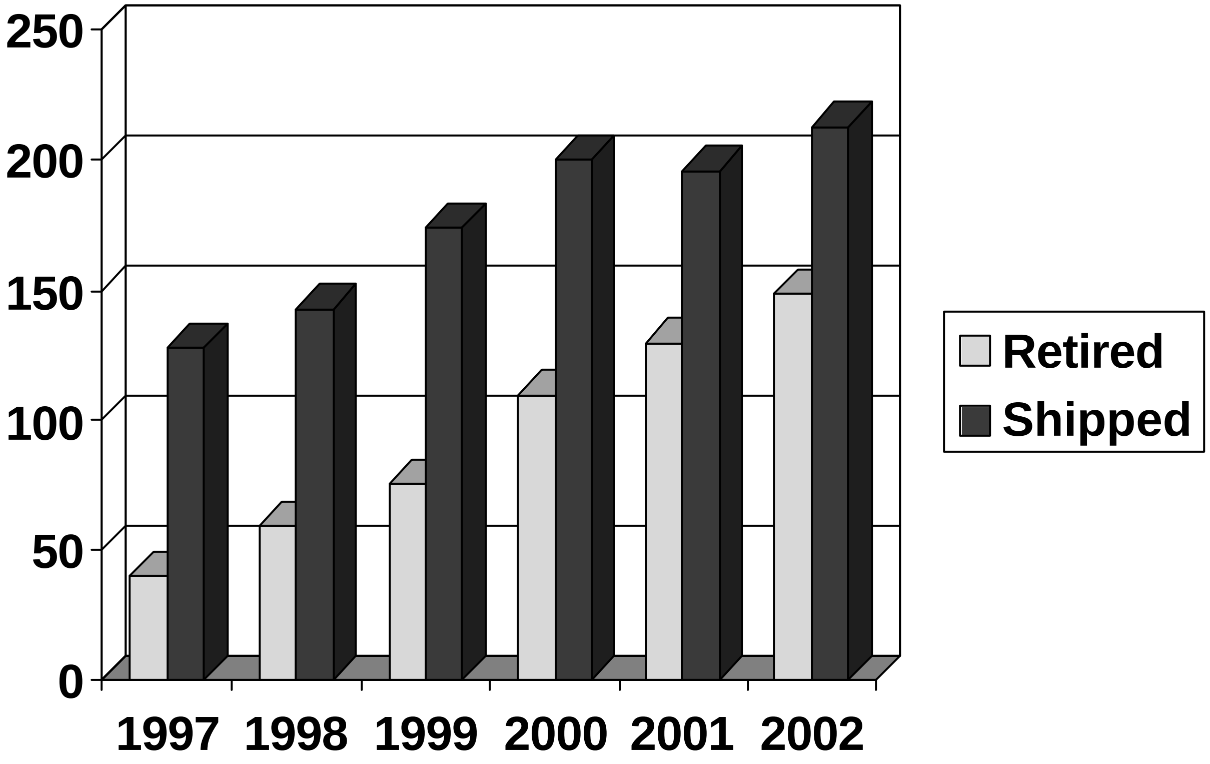
Wietse Venema

wietse@porcupine.org

IBM T.J.Watson Research, USA

Global hard disk market

(Millions of units, source: Dataquest)



Informal survey of retired disks

(Garfinkel & Shelat)

- Experiment: buy used drives, mainly via Ebay.
- Time frame: November 2000 - August 2002.
- 158 Drives purchased.
- 129 Drives still worked.
- 51 Drives “formatted”, leaving most data intact.
- 12 Drives overwritten with fill pattern.
- 75GB of file content was found or recovered.

IEEE Privacy & Security January/February 2003,

http://www.computer.org/security/garfinkel.hmtl

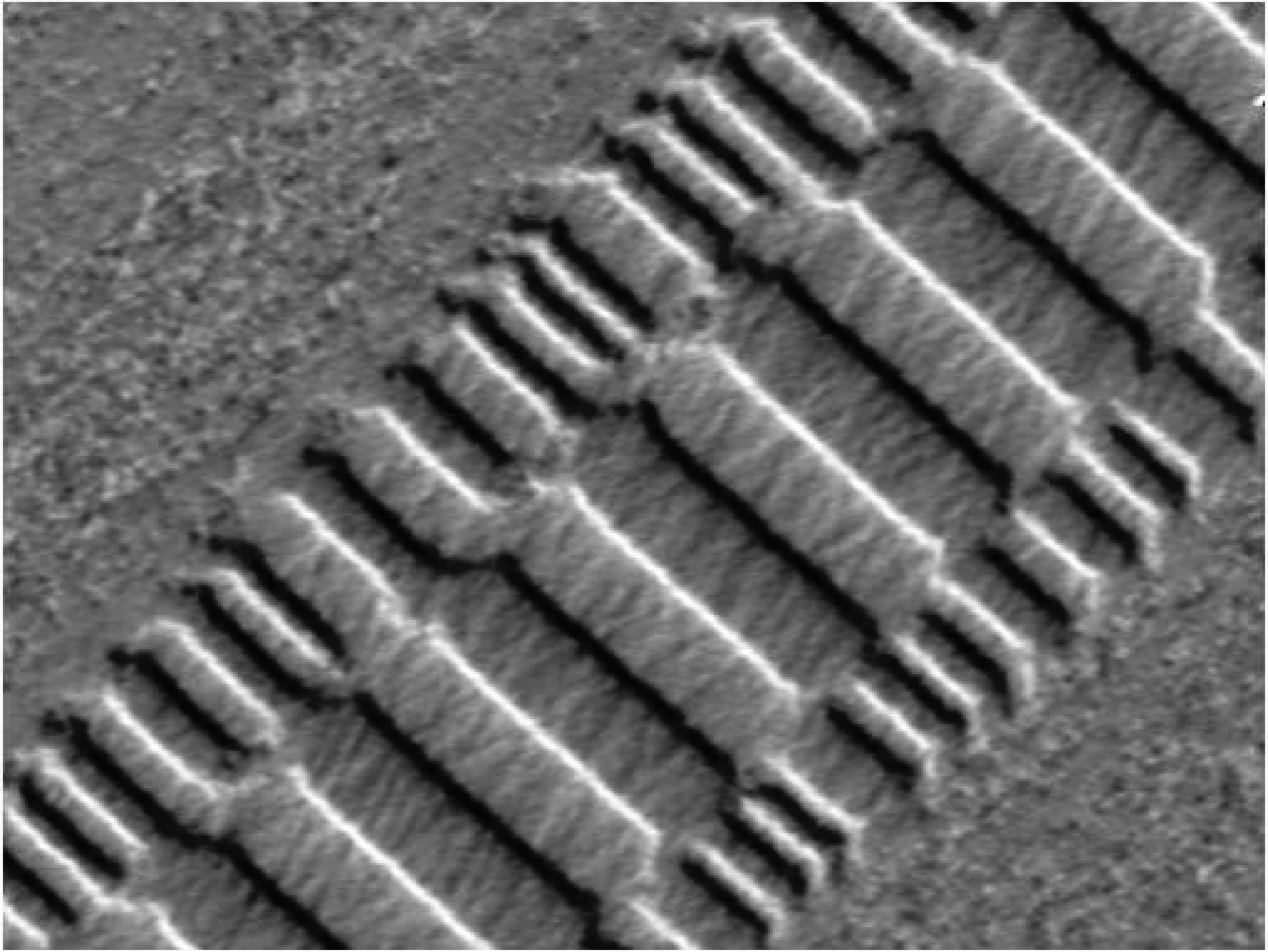
What information can be found on a retired disk

- One drive with 2868 account numbers, access dates, balances, ATM software, but no DES key.
- One drive with 3722 credit card numbers.
- Corporate memoranda about personnel issues.
- Doctor’s letter to cancer patient’s parent.
- Email (17 drives with more than 100 messages).
- 675 MS Word documents.
- 566 MS Powerpoint presentations.
- 274 MS Excel spreadsheets.

WSJ reporter buys two computers after Taliban fall November 2001

- Windows 2000.
- 1750 text and video files.
- Some files protected by “export strength” encryption (40 bit).
- Five-day effort to decrypt one file by brute force.
- Report of scouting trip for terrorist targets (shoe bomber Richard Reid?).

http://cryptome.org/nyt-wsj-dod.htm WSJ=Wall Street Journal



Digital media aren’t

- Information is digital, but storage is analog.
- Information on magnetic disks survives multiple overwrite operations (reportedly, recovery is still possible with 80GB disk drives!).
- Information in semiconductor memory survives “power off” (but you have little time).

Disk track images: nanotheatre at http://www.di.com/

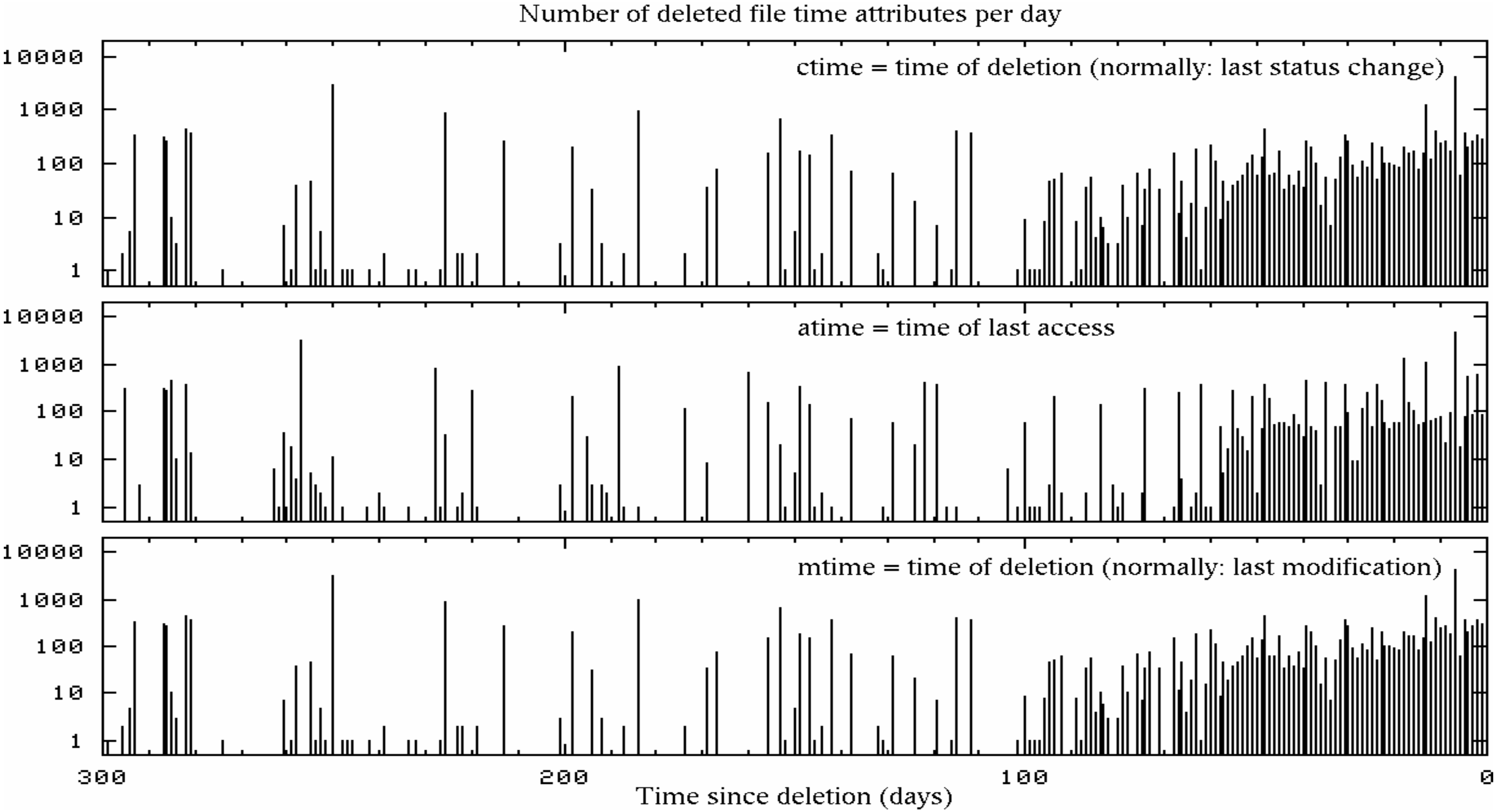
Peter Gutmann’s papers: http://www.cryptoapps.com/~peter/usenix01.pdf

and http://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html

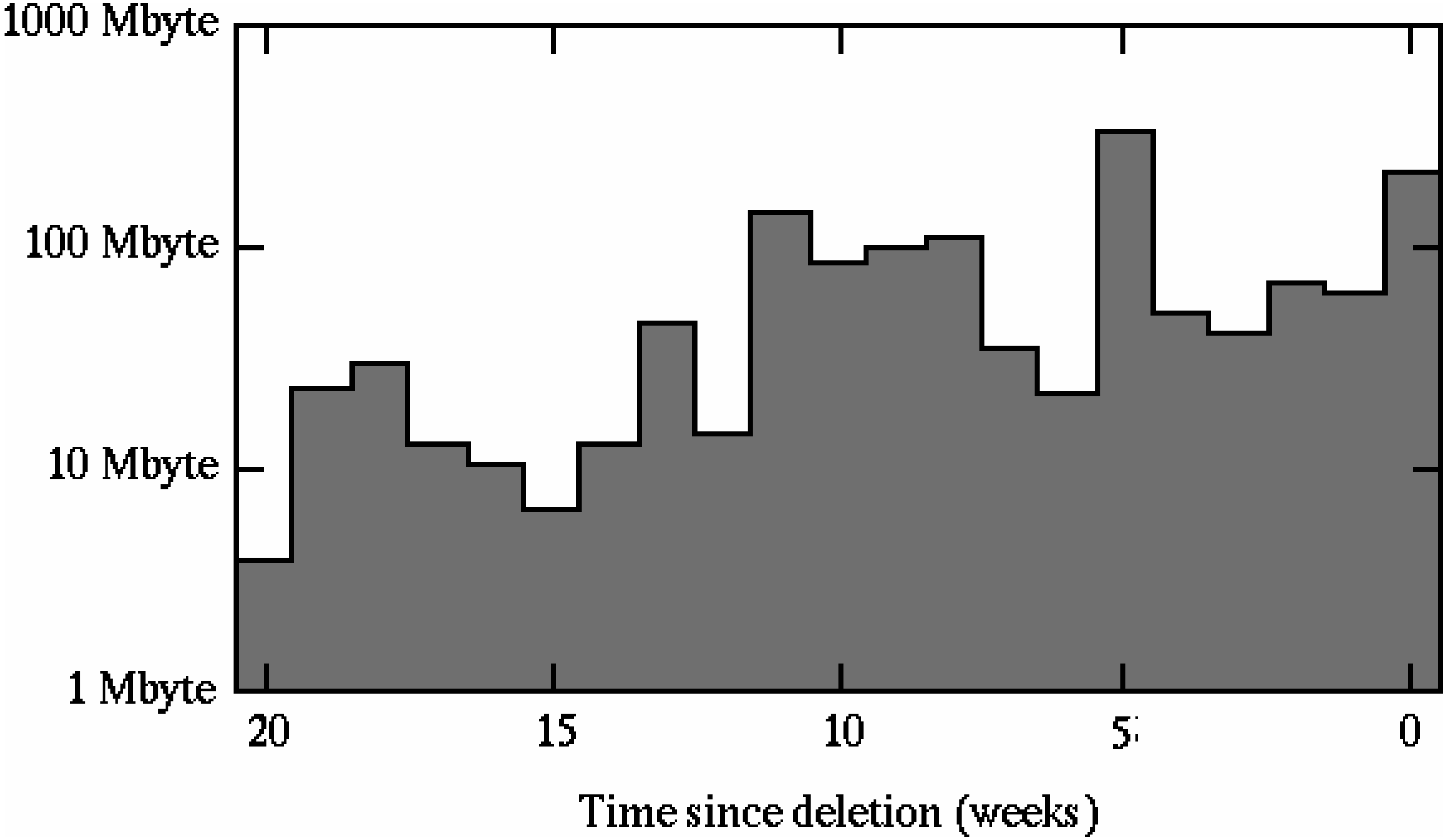
What happens when a file is deleted?

- Structure is lost, information survives.
- Preserved: file names/attributes/content.
- Destroyed: connections between file names/ attributes/content.
- On UNIX/Linux file systems, the result can be a puzzle with many loose pieces.
- On DOS/Windows file systems, many of the connections remain intact.

Persistence of deleted file time attributes - dedicated UNIX server



Persistence of deleted file content - same dedicated UNIX server



Summary: persistence of deleted file content

Machine	File system	Half-life
spike.porcupine.org[1]	entire disk	35 days
flying.fish.com[2]	/	17 days
flying.fish.com[2]	/usr	19 days
www.porcupine.org[1]	entire disk	12 days

[1] FreeBSD [2] Linux

Will file encryption solve the problem?

- Plenty opportunity for information leakage:
  - Swap files (fixed in, e.g., OpenBSD).
  - Unencrypted application temporary files.
  - Main memory (see next section).
- Some files/directories/attributes must not be encrypted (for booting and file system checks).
- Implementors sometimes make bad mistakes.
- Concerns about data recovery after crash.

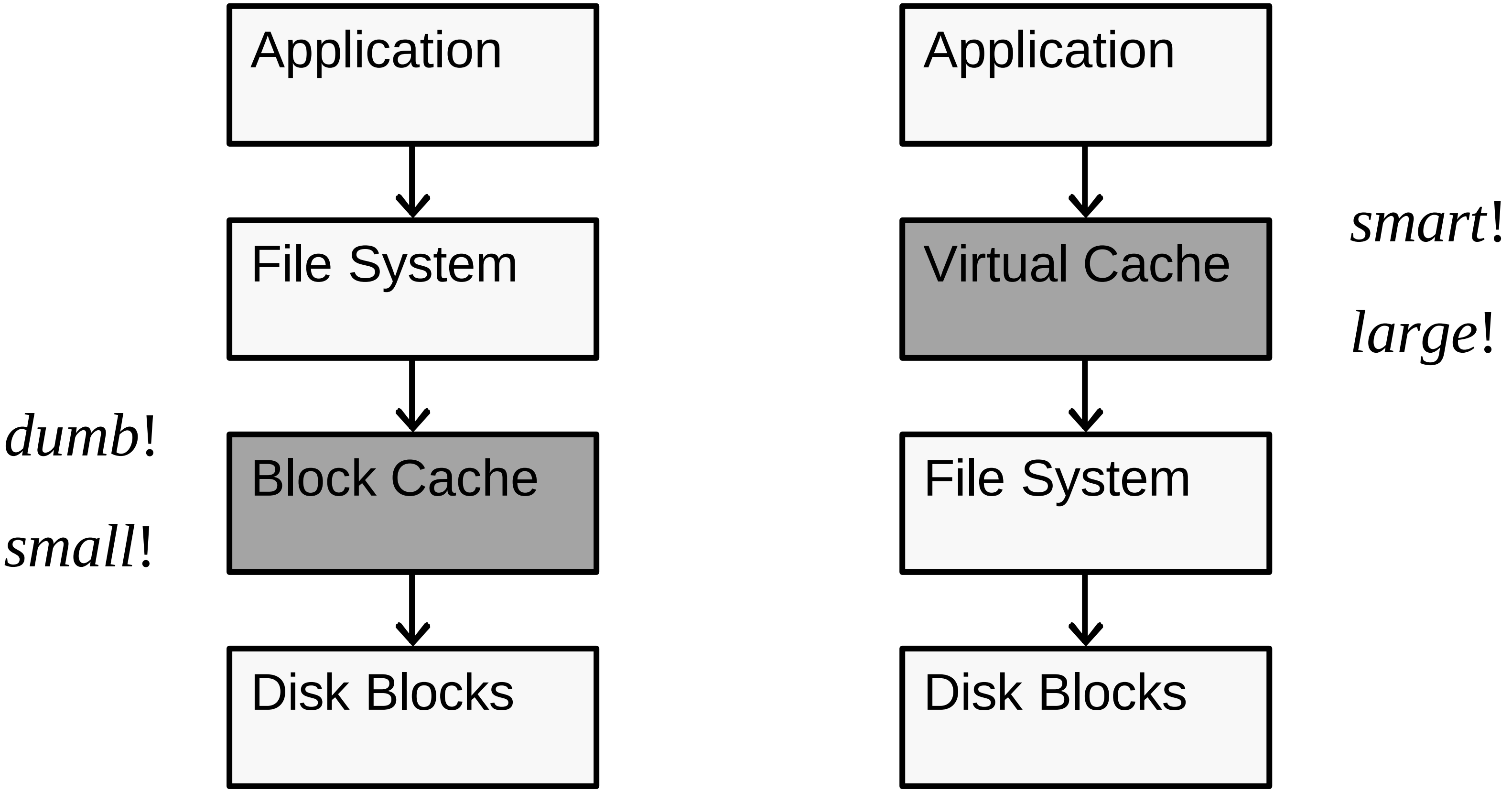
Persistence of information in main memory

Information that may be found in main memory:

- Running processes[1].
- Terminated processes[1].
- Operating system.
- Cached (buffered) copies of recently accessed or executed files and directories.

[1] Some information may be found in swap files.

Block cache versus virtual cache
(owned by system, not by applications)

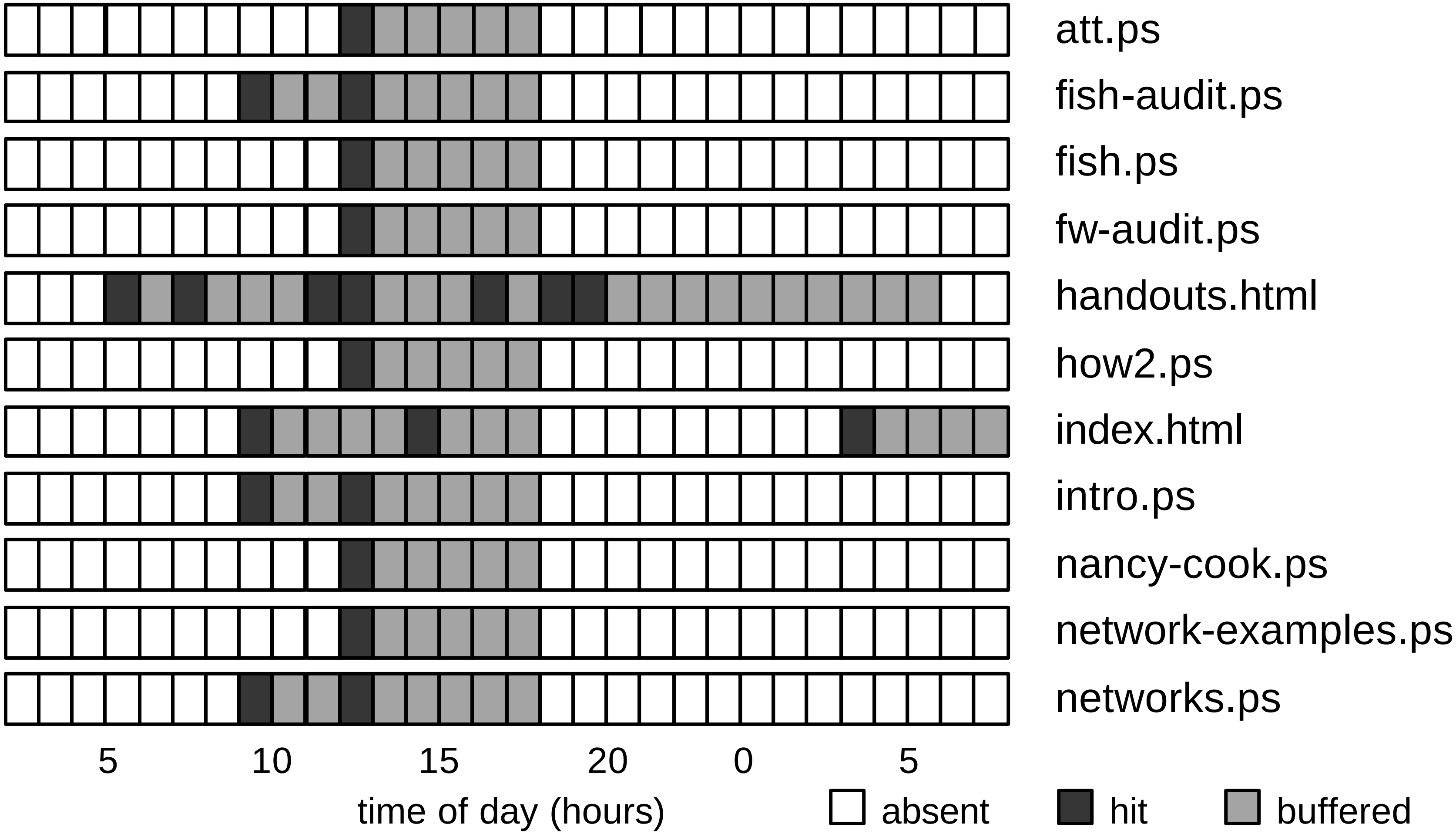


DOS, Win95/98/ME, BSD

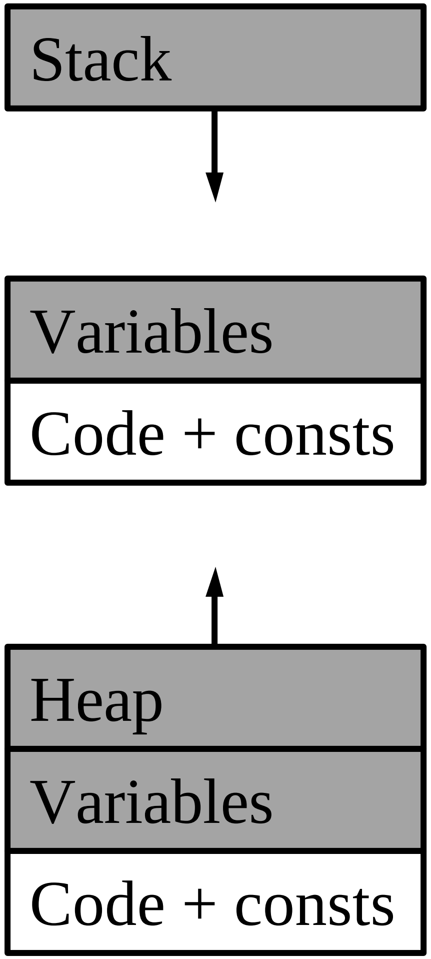
BSD, Linux, Solaris,WinNT/2K/XP

File caching in main memory

(low-traffic web pages, FreeBSD)

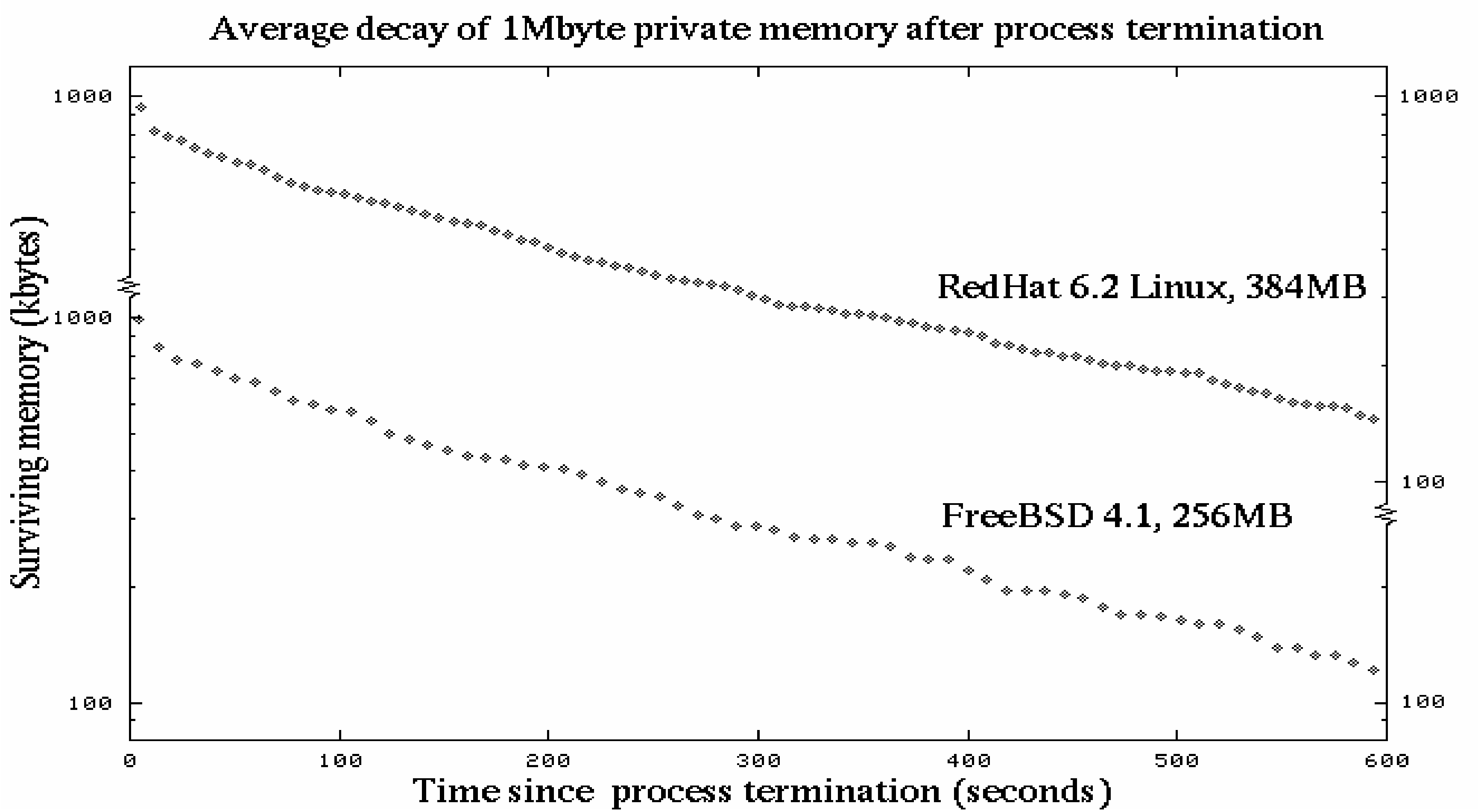


Private process memory - UNIX
(the bits that must be saved when swapping)



Stack	Private; grows on demand.
Variables	Private; initialized from libc.so.
Code + consts	Shared; paged in from libc.so.
Heap	Private; grows on demand.
Variables	Private; initialized from executable.
Code + consts	Shared; paged in from executable.

Persistence of private memory



Summary: persistence of main memory (Linux, FreeBSD)

- Hours-days: cached (buffered) file data. Modern systems have lots of available main memory.
- Minutes: private data after process termination, even on lightly loaded systems.
- Minutes: cached data from deleted files, just like private memory from terminated processes.
- The information of most interest is the first to be destroyed. **Bad luck** :-(

Recovering Windows/2K/XP encrypted files without key

- EFS[1] provides encryption by file or by directory. Encryption is enabled via Explorer property dialog box or via the equivalent system calls.
- With encryption by directory, files are encrypted before being written to disk.
- Is unencrypted content of EFS files cached in main memory?
- If yes, for how long?

[1] EFS=encrypting file system

Experiment: create encrypted file

- Create “encrypted” directory c:\temp\encrypted.
- Download 350kB text file via FTP, with content:

```
00001 this is the plain text
00002 this is the plain text
...
11935 this is the plain text
11936 this is the plain text
```

- Scanning the disk from outside (VMware rocks!) confirms that no plaintext is written to disk.

Experiment: search memory dump

- Log off from the Windows/XP console and press Ctrl/ScrollLock twice for memory dump[1].
- Analyze result with standard UNIX tools:

```
%strings memory.dmp | grep ‘this is the
  plain text’
03824   this is the plain text
03825   this is the plain text
```

. . .etcetera. . .

- 99.6% of the plain text was found undamaged.

[1] Microsoft KB 254649: Windows 2000 memory dump options.

Recovering Windows XP encrypted files without keys

- Good: EFS encryption provides privacy by encrypting file content before it is written to disk.
- Bad: unencrypted content stays cached in main memory even after the user has logged off.
- Similar experiments are needed for other (UNIX) encrypting file systems. Most are expected to have similar plaintext caching behavior.

Conclusion

- Disk “dumpster diving” remains a source of information with great potential.
- Memory dumps reveal clues about recent activity on a computer system, including plaintext of encrypted files.
- Big brother and the arms race between the good and the evil forces.

Pointers

- Simson Garfinkel, Abhi Shelat, Remembrance of Data Passed. IEEE Privacy&Security Jan 2003. *http://www.computer.org/security/garfinkel.html*
- Dan Farmer, Wietse Venema, series of articles in Dr.Dobb’s Journal 2001-2002. *http://www.porcupine.org/forensics/column.html*
- By the same authors: the Coroner’s Toolkit. *http://www.porcupine.org/tct/*
- TCTutils, TASK, and other tools by Brian Carrier. *http://www.atstake.com/research/tools/*