Forensic Discovery

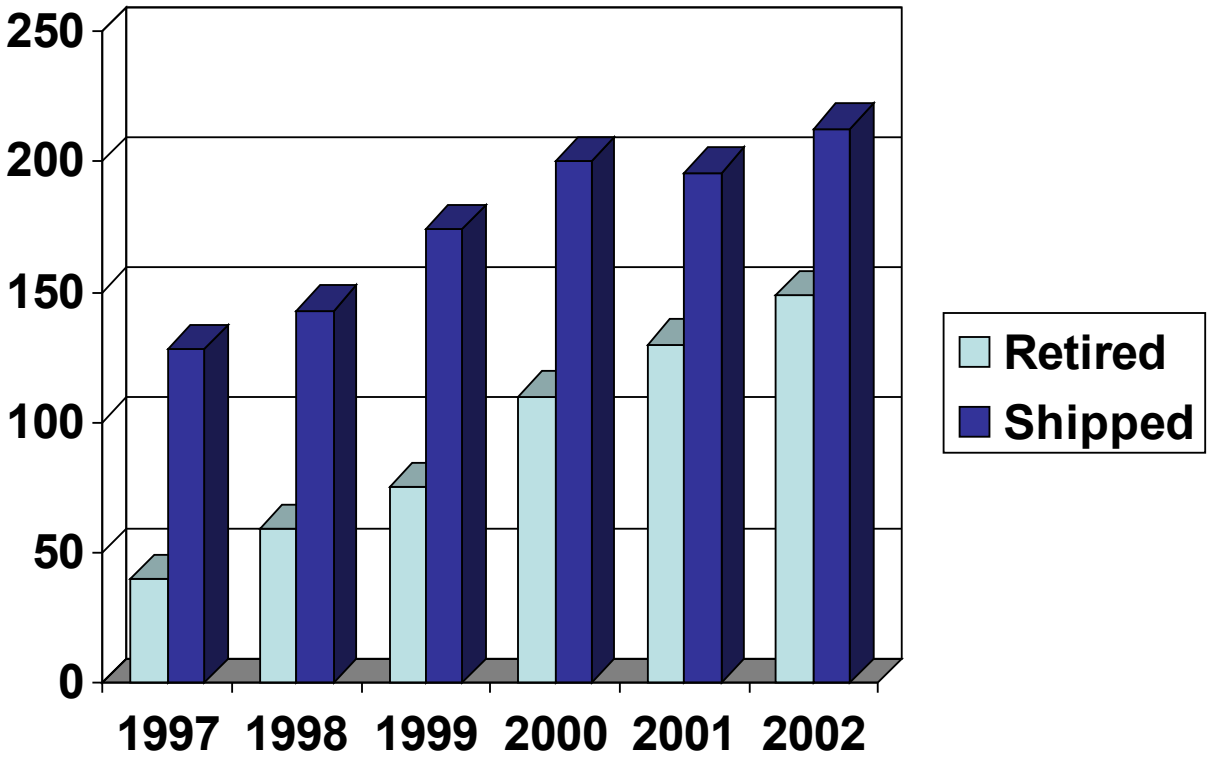
Wietse Venema
wietse@porcupine.org
IBM T.J.Watson Research, USA

Overview

- Information on retired disks.
- Information on overwritten disks.
- Persistence of deleted file information.
- Persistence of information in main memory.
- Recovering Windows/XP files without key.
- Trends in computer system subversion.

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.

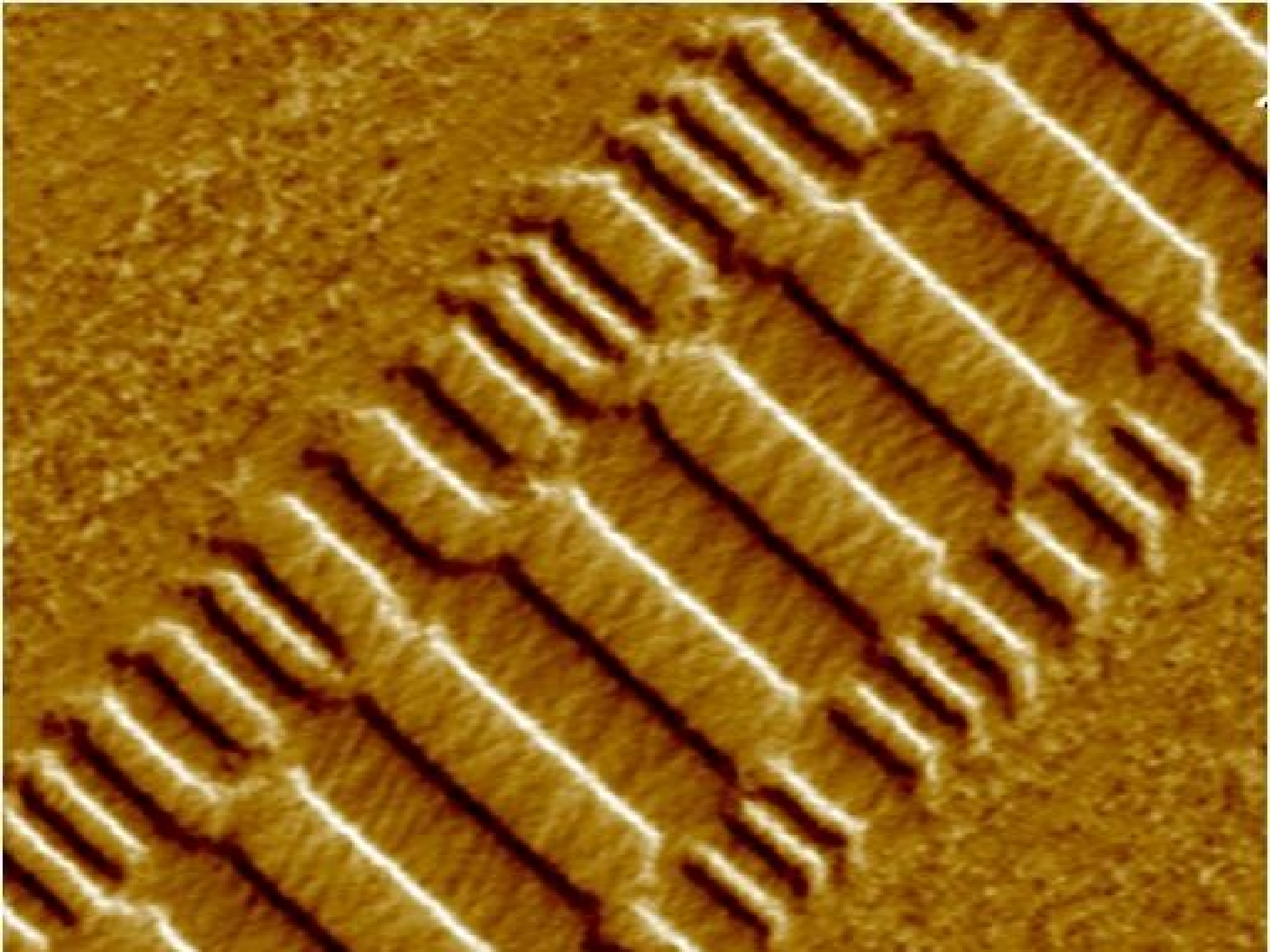
IEEE Privacy & Security January/February 2003,
http://www.computer.org/security/garfinkel.pdf

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.
- Letter to doctor from cancer patient's parent.
- Email (17 drives with more than 100 messages).
- 675 MS Word documents.
- 566 MS Powerpoint presentations.
- 274 MS Excel spreadsheets.

File System Persistence

Deleted file data can be more persistent than existing file data



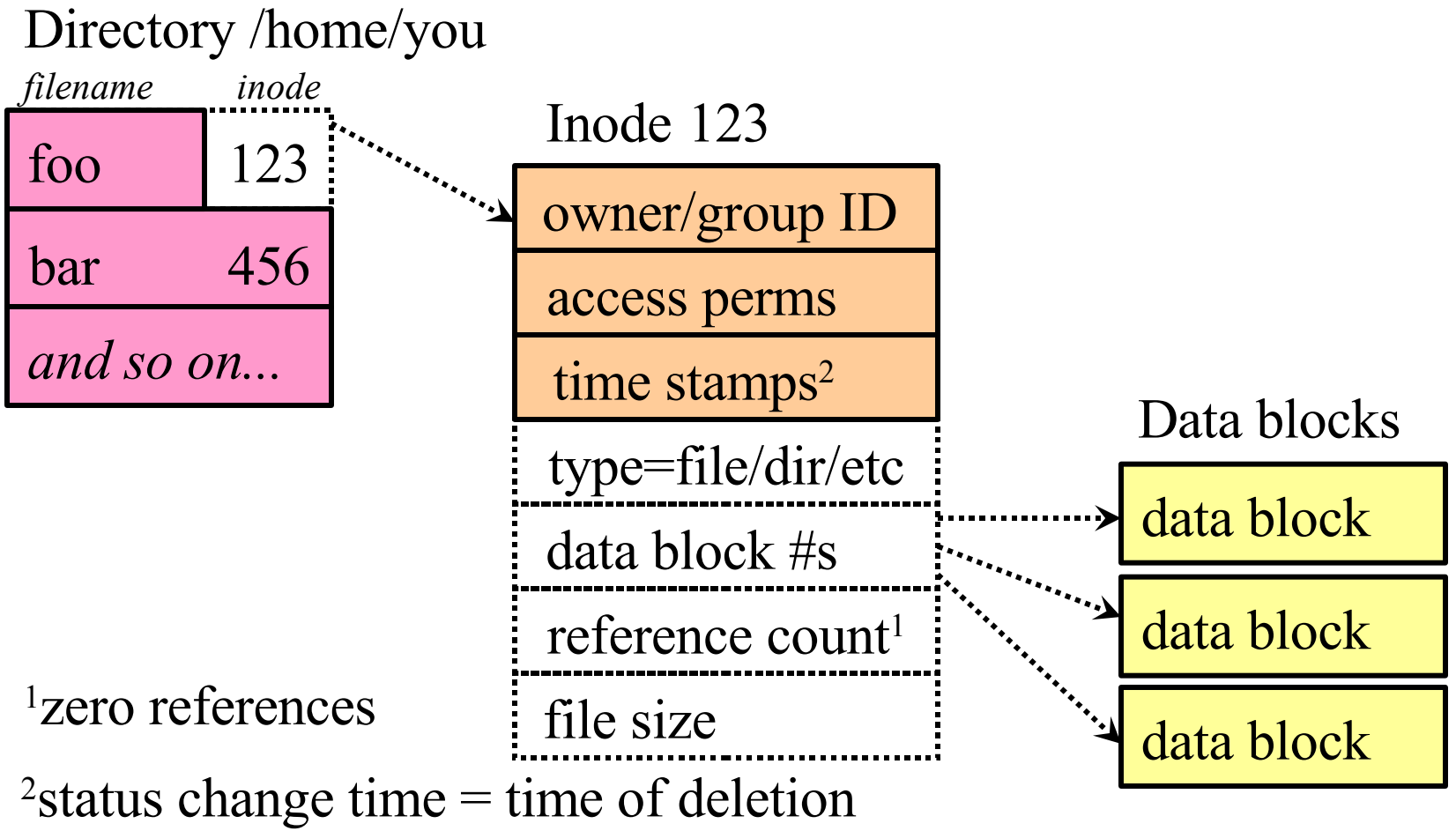
Digital media aren't

- Information is digital, 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).

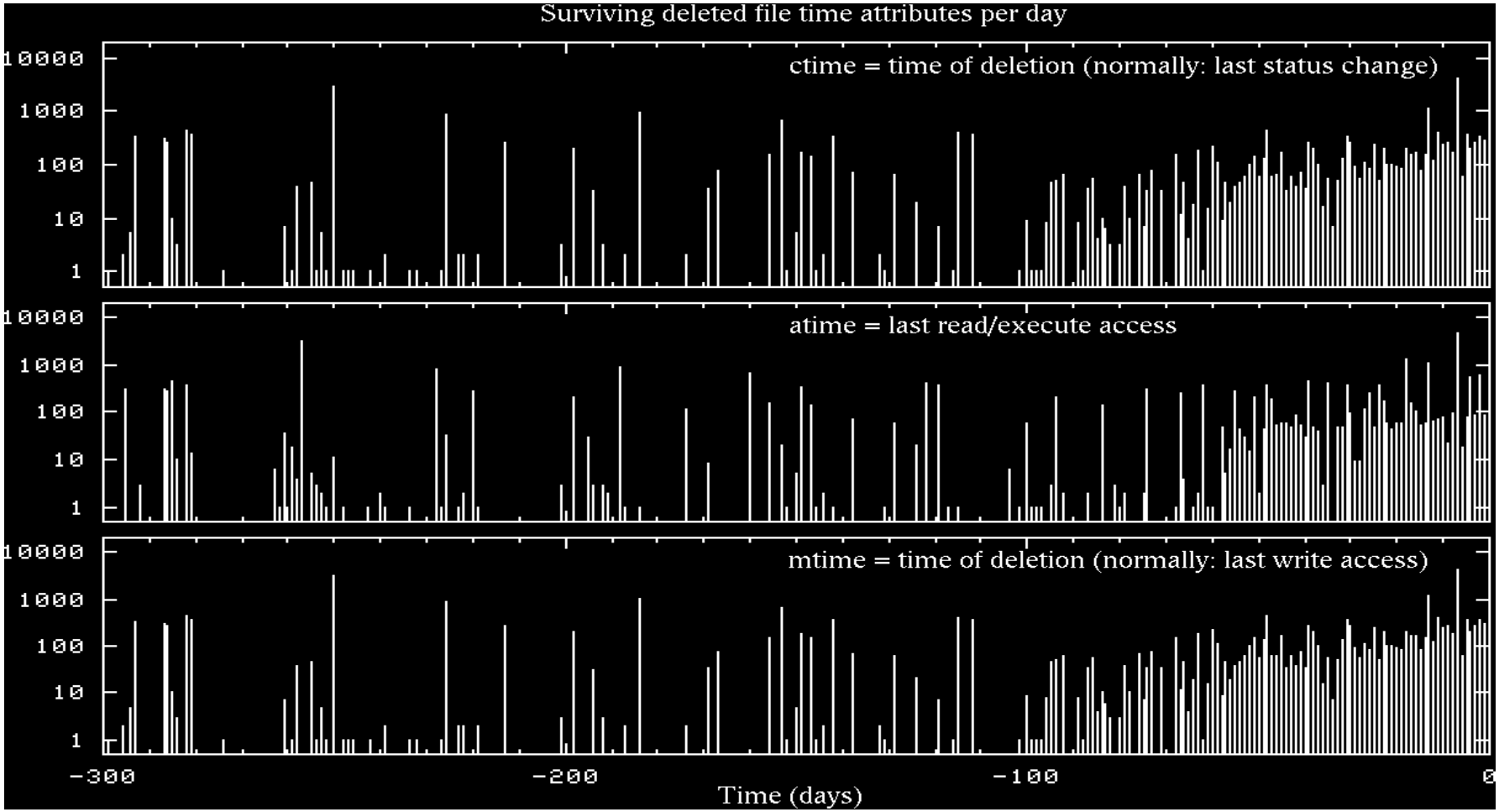
Disk track images: http://www.veeco.com/

Peter Gutmann's papers: http://www.cryptoapps.com/~peter/usenix01.pdf and http://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html

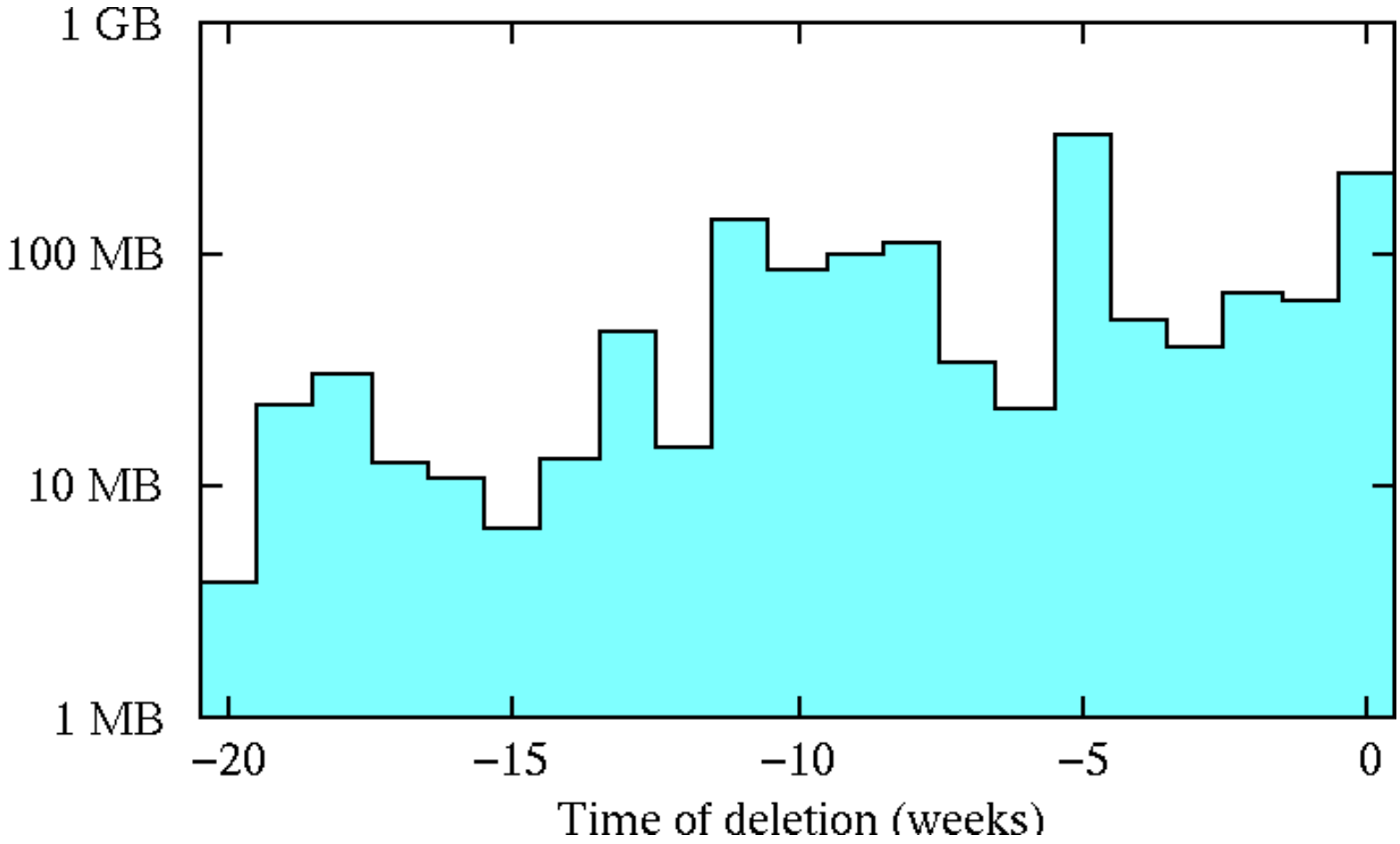
Deleting a file destroys structure not content



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	Filesystem	Half-life
spike.porcupine.org[1]	entire disk	35 days
flyingfish.com[2]	/	17 days
flyingfish.com[2]	/usr	19 days
www.porcupine.org[1]	entire disk	12 days

[1]FreeBSD [2]Linux

Why deleted file data can be more persistent than existing file data

- Existing files are easy to access, and therefore easy to modify. Deleted files are less accessible.
- UFS and Ext*fs file systems are organized into *zones* of 32768 blocks with directories, files, etc. A deleted file in zone X survives writing activity in zone Y. Other file systems have comparable locality properties.
- Information from deleted files becomes a “fossil”. It may be incomplete but it does not change until it is destroyed.

Main Memory Persistence

Recovering Windows/XP files
without knowing the key

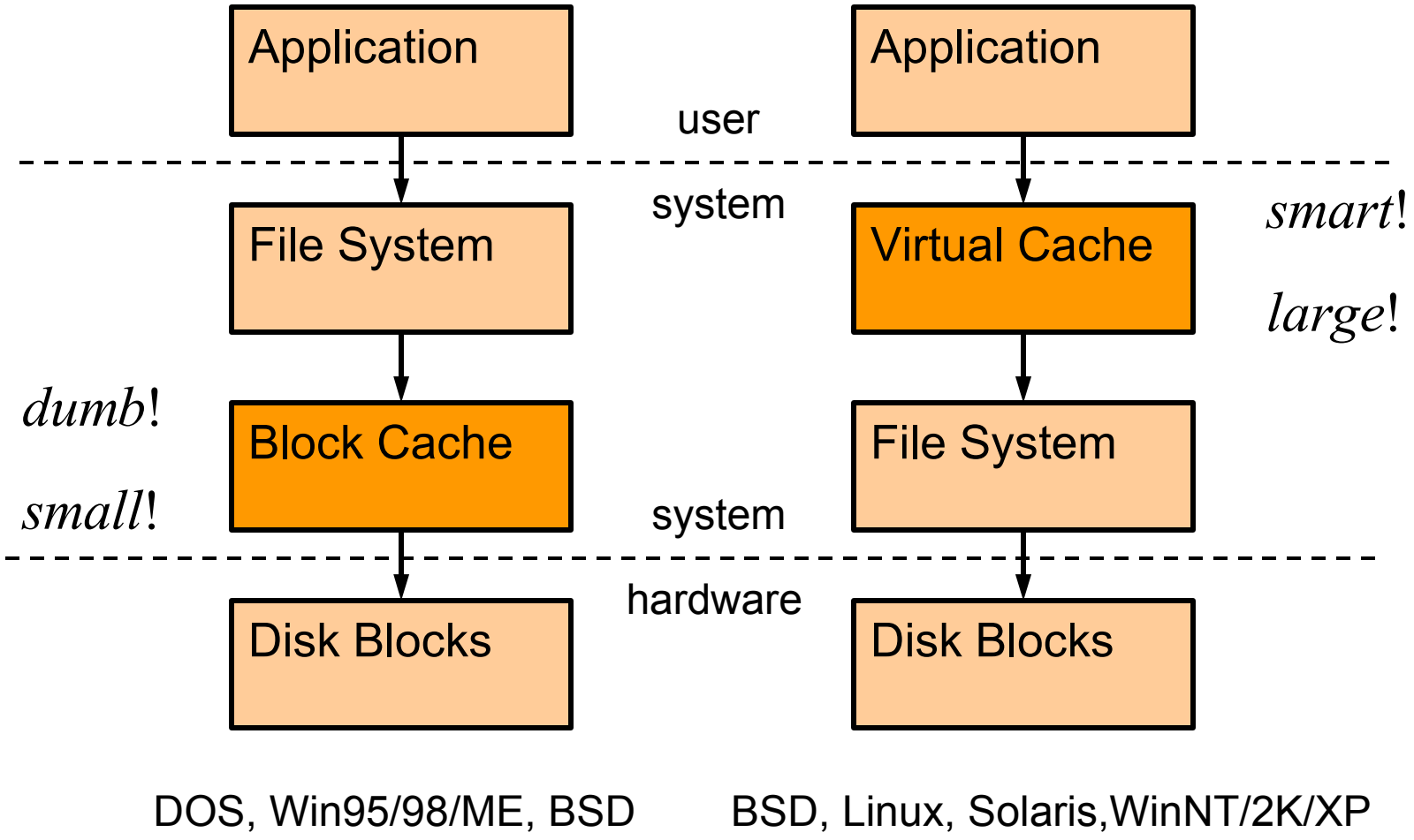
Information in main memory

- Running processes[1].
- Terminated processes[1].
- Kernel memory.
- Recently active files/directories (file cache).
- Deleted files (from process or from cache).
- All have different persistence properties.

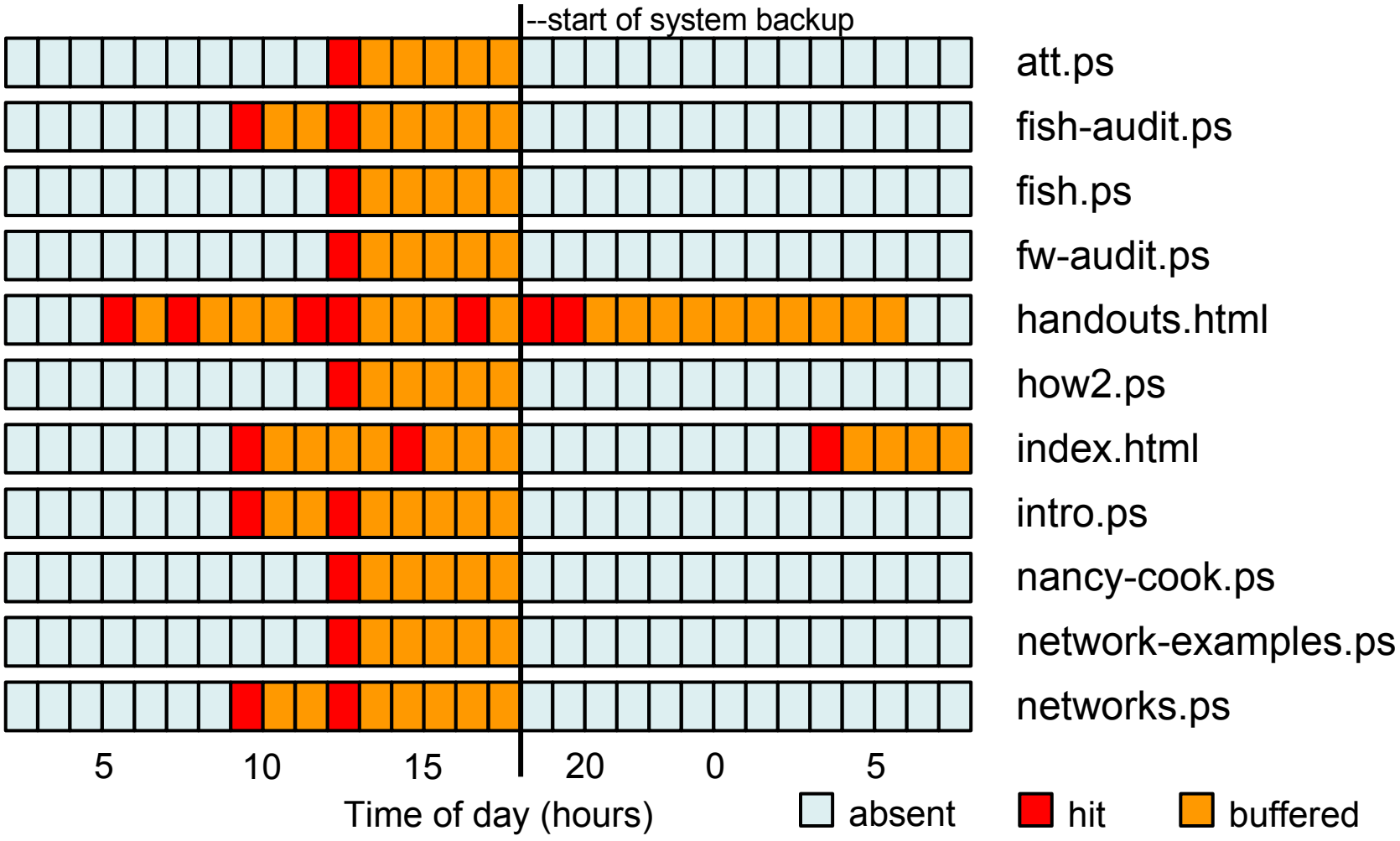
[1]Some information may be found in swap files.

Block cache versus virtual cache

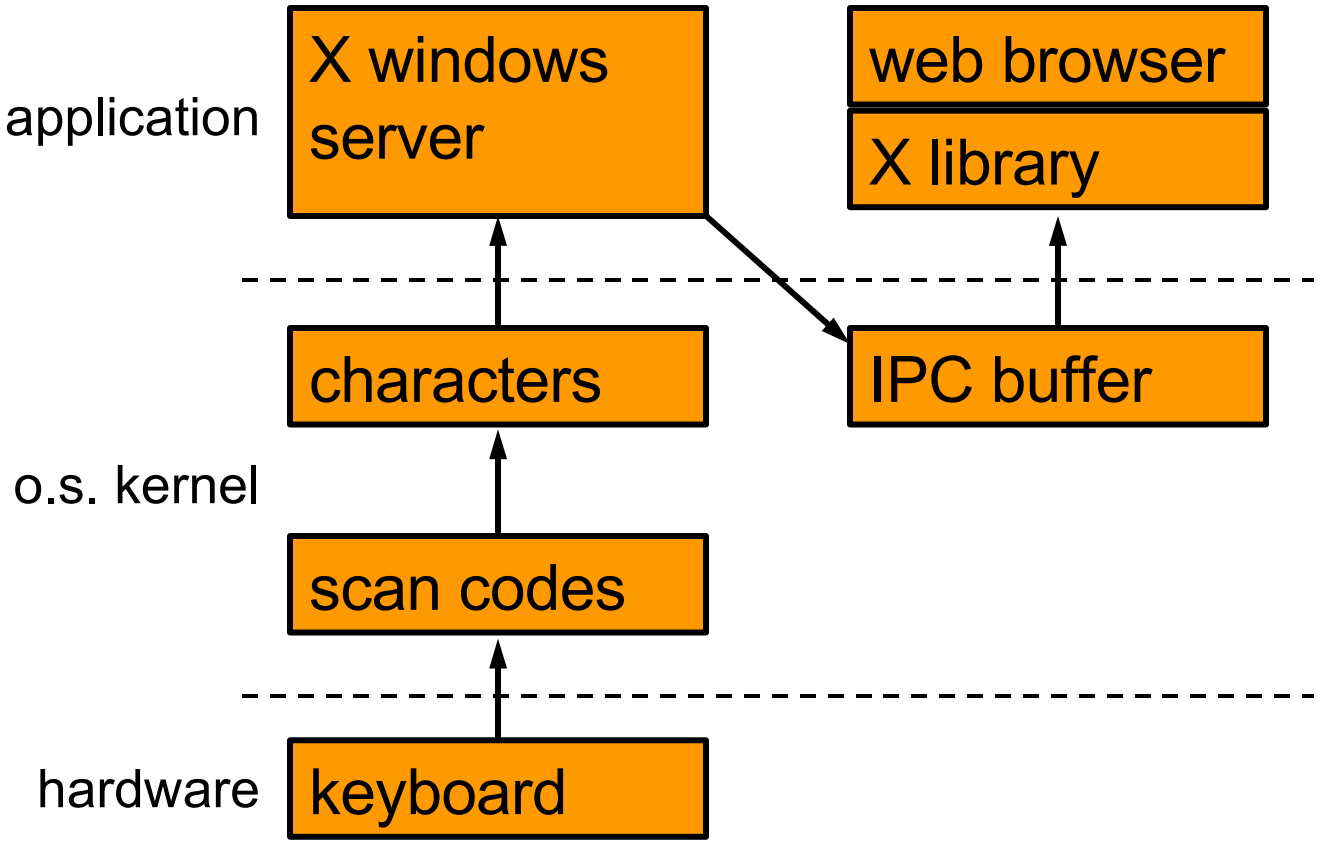
(owned by system, not by applications)



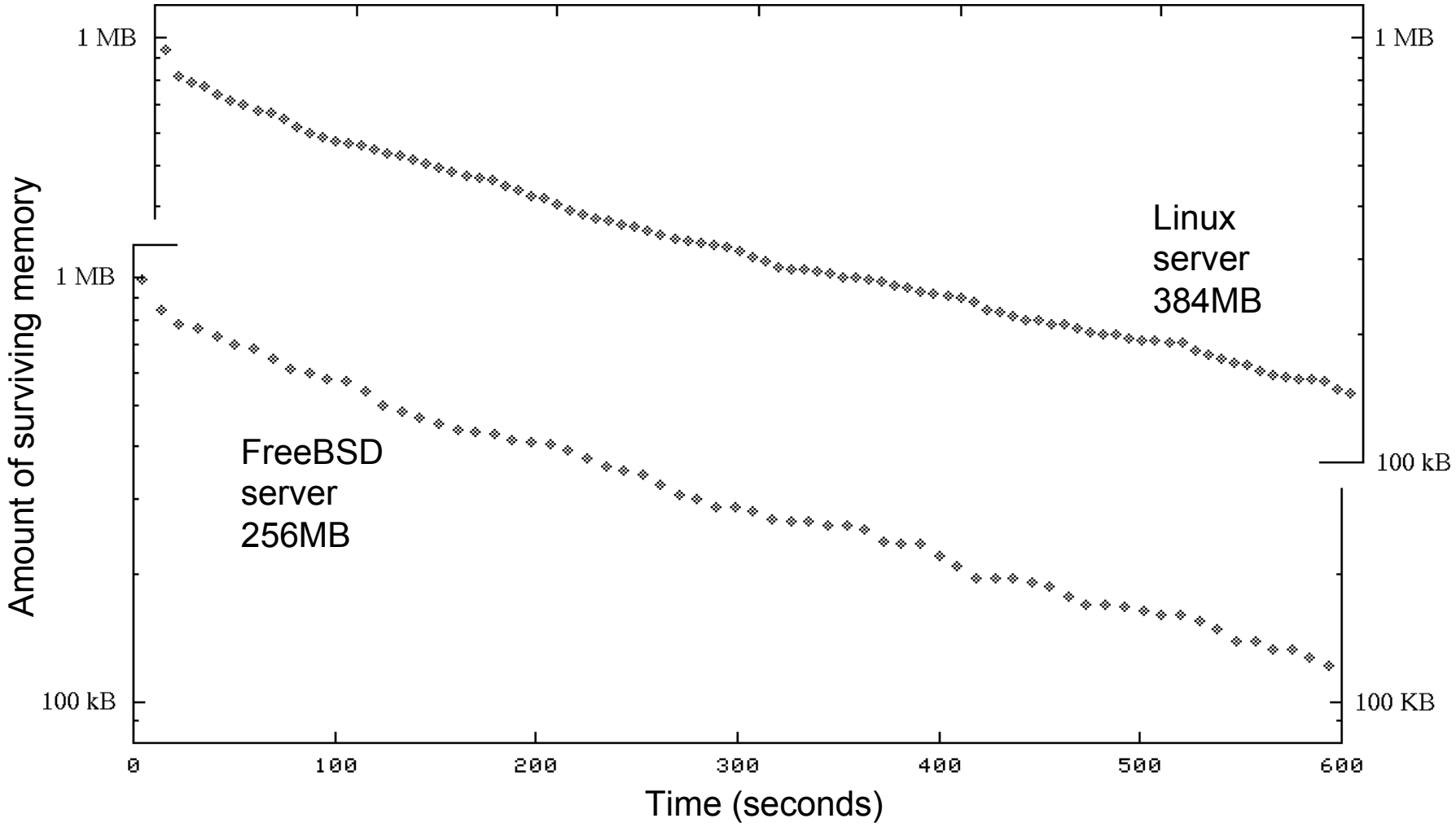
File caching in main memory
(low-traffic web pages, FreeBSD)



Trail of secrets across memory
(after Chow *et al*.)

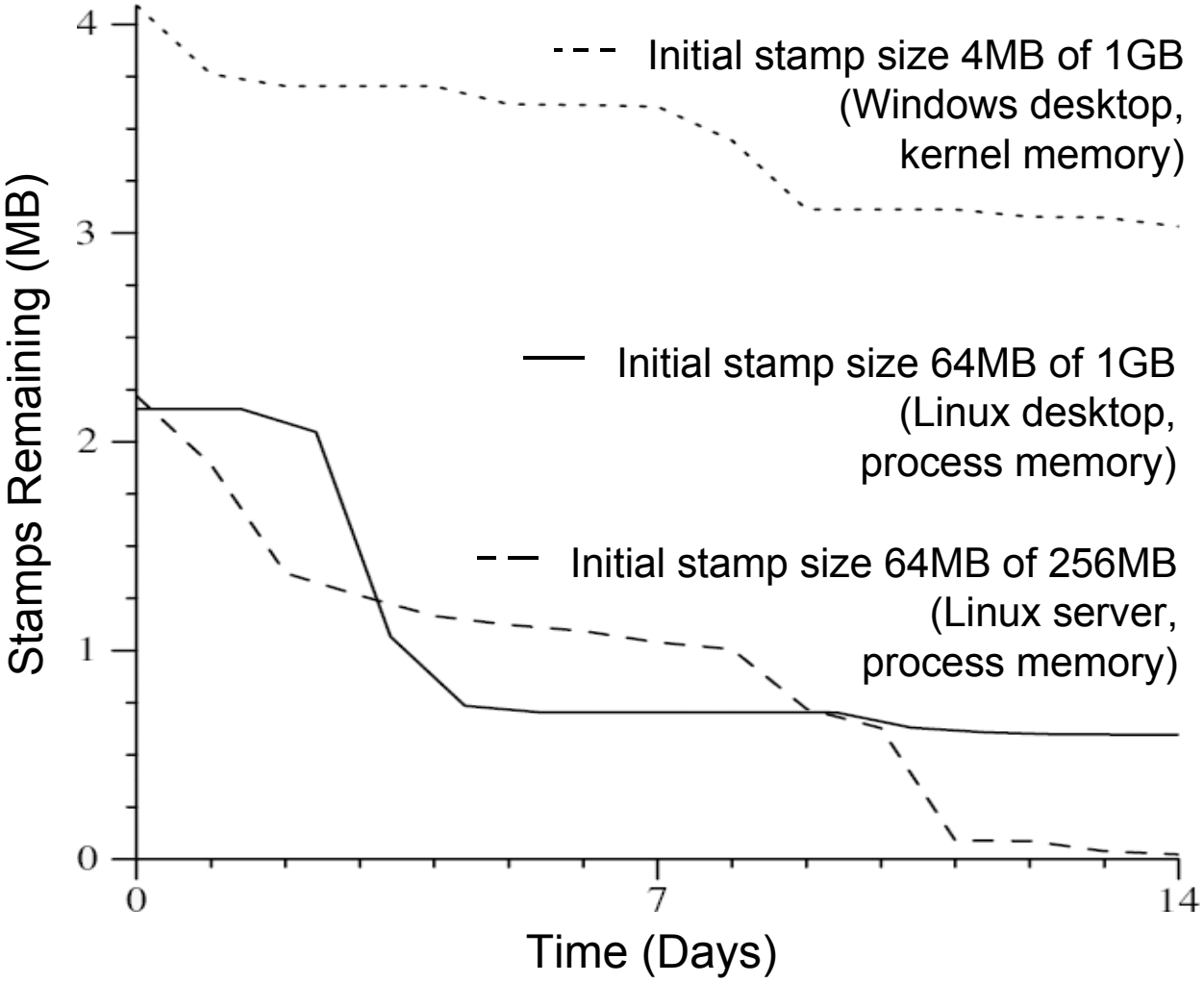


Short-term memory persistence
after process termination (1MB stamp)



Long-term memory persistence

(Chow *et al*., USENIX Security 2005)



Recovering Windows/2K/XP encrypted files without key

- EFS[1] provides encryption *by file* or *by directory*. Encryption is enabled via an Explorer property dialog box or via the equivalent system calls.
- With encryption by directory, files are encrypted *before* they are 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 key

- *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.

Trends in Subversion

Hardware is getting softer as complexity increases

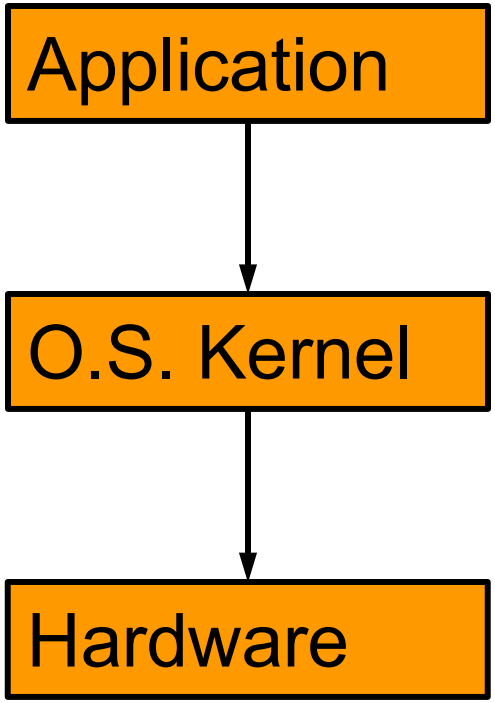
Root kits gen#1 - command level

- Malware (example: ethernet password sniffer).
- Backdoor (example: modified login program).
- Patched *command/library files* to hide malware and backdoor processes/files/connections.
- Sometimes: logfile editors, file checksum fixers.
- Easy to find via inconsistencies (*echo .** <=> *ls*).
- *Easy to find in post-mortem disk images.*

Root kits gen#2 - kernel level

- Malware (distributed denial of service, spam relay, or other remote control).
- Backdoor (example: modified system call or network handling code).
- Patched *running kernel* to hide malware and backdoor processes, files, or connections.
- May show up via inconsistencies (*ps* <=> */proc*).
- *May not show up in post-mortem disk images.*

Progression of subversion
(also known as rootkits)



First generation

Second generation

The future is here?
(focus on the machine itself,
not evil plug-in hardware)

Hardware is not what it used to be

- Nowadays, almost every electronic device has firmware that can be updated.
- Popularity ranking according to Google (8/2005):

+dvd +firmware	1.2M hits
+satellite +firmware	1.0M
+disk +firmware	***930k***
+phone +firmware	910k

- Not all hits are “officially supported”.

Reflashing for fun and profit

(‘lock-in’ versus ‘unlocking the true potential’)

It’s all about business models.

- Time to market: ship it now, fix it later.
- Watch satellite etc. TV without paying.
- Re-enable wireless telephone features.
- Disable DVD player region locks.
- Upgrade camera to more expensive model.

Note, these are all *special-purpose* devices.

What about general-purpose computer systems?

- Pentium CPU instruction set updates require digital signature, and don't survive 'power off'.
- Little variation in system BIOS implementations; some variation in processors or in operating systems as used in disks and other peripherals.
- Enough variation to make worm-like exploitation error-prone (lots of systems become door stops).
- Of course, this won't stop motivated individuals from updating the firmware in *specific* machines.

Conclusion

- Deleted file information can survive for a year or more, even with actively used file systems.
- Main memory becomes a *primary* source of forensic information, especially with infection of running processes or running operating system kernels.
- Hardware is becoming softer all the time, as complexity increases. Do not blindly trust that a hardware device will give you all the information that is stored on it.

Pointers

- Simson Garfinkel, Abhi Shelat: “Remembrance of Data Passed”. IE^{3}Privacy&Security, Jan 2003. http://www.simson.net/pubs.php
- Dan Farmer, Wietse Venema: “Forensic Discovery”, Addison-Wesley, Dec. 2004. http://www.porcupine.org/forensics/ http://www.fish2.com/forensics/
- Jim Chow *et al.*: “Shredding Your Garbage”, USENIX Security 2005; “Understanding Data Lifetime”, USENIX Security 2004.