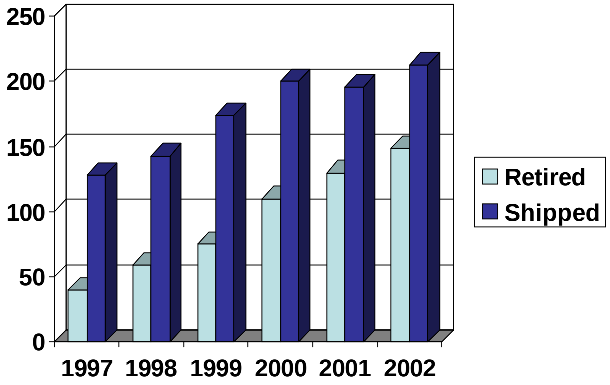
Memory forensics
(well, that’s what the title says)

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 were purchased.
- 129 Drives still worked.
-   51 Drives were “formatted”.
-   12 Drives were 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 laptops after Taliban fall 2001/11

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

http://cryptome.org/nyt-wsj-dod.htm

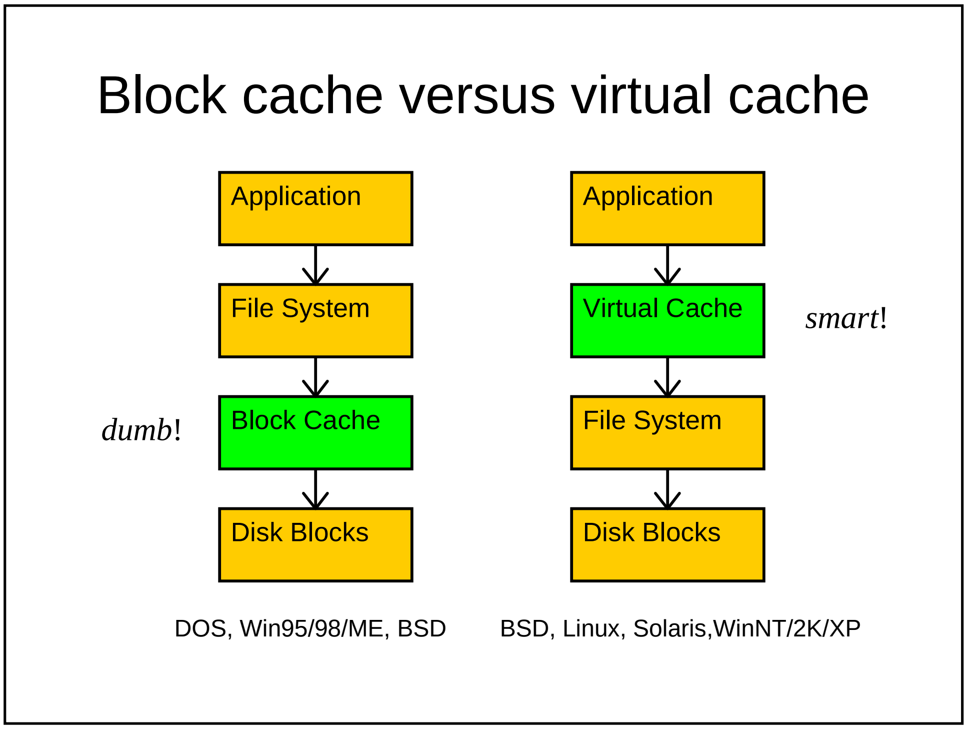
What information can be found in main memory

In this presentation:

- Any file or directory that was accessed recently.
- Running and terminated processes (may also be found in swap files).

Not in this presentation:

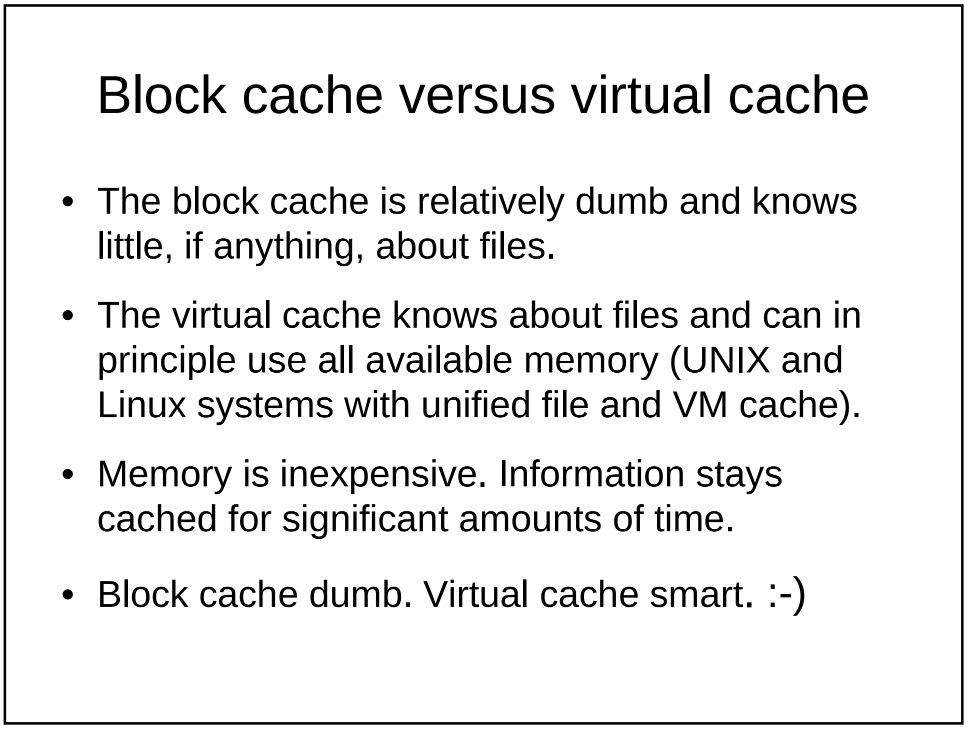
- Operating system, device/network buffers.
- Memory-mapped hardware (not really main memory, but hard to distinguish from it).



Block cache versus virtual cache

DOS, Win95/98/ME, BSD

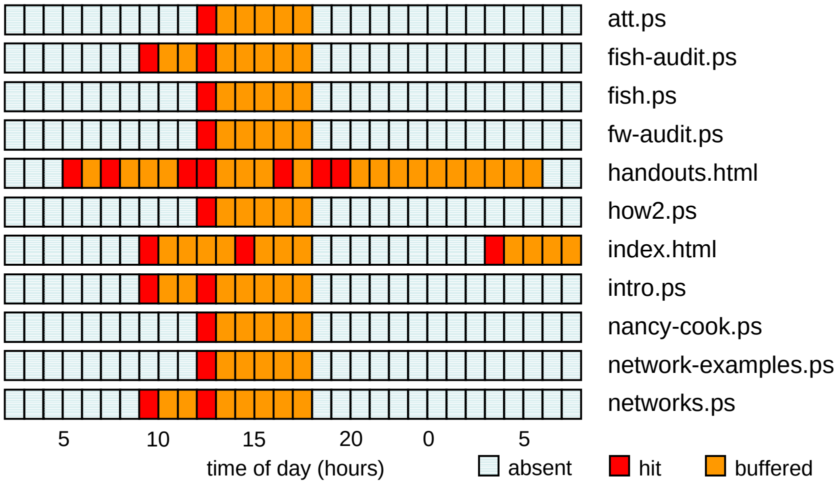
BSD, Linux, Solaris,WinNT/2K/XP



Block cache versus virtual cache

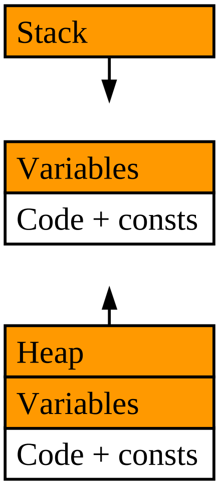
- The block cache is relatively dumb and knows little, if anything, about files.
- The virtual cache knows about files and can in principle use all available memory (UNIX and Linux systems with unified file and VM cache).
- Memory is inexpensive. Information stays cached for significant amounts of time.
- Block cache dumb. Virtual cache smart. :-)

File caching in main memory of rarely accessed web pages



Private process memory

(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.

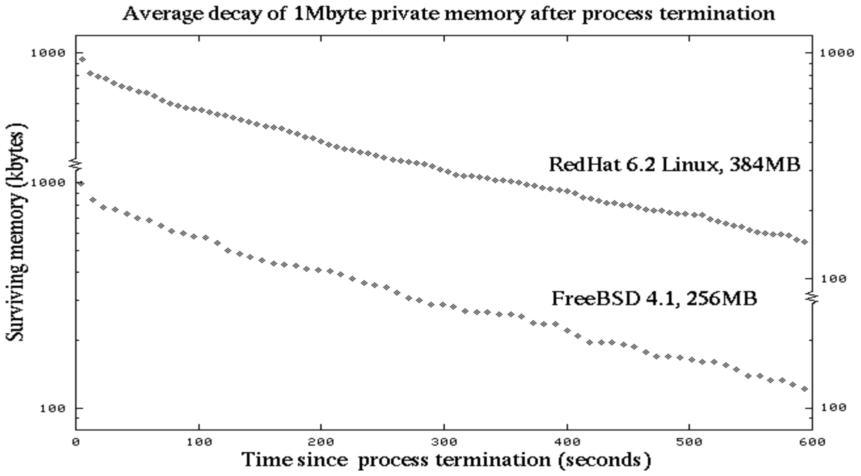
Variables — Private; initialized from executable.

Code + consts — Shared; paged in from executable.

Persistence of anonymous memory
(for UNIX/Linux systems)

- Read-only, executable, memory is normally backed by a specific executable or library file. Content stays intact after process termination, for as long as it is part of the virtual cache.
- Read/write, private, memory is normally not backed by a specific executable or library file. Memory is recycled after a process terminates.
- For the same reason, cached content of deleted file is recycled after the file becomes inactive.

Persistence of private memory



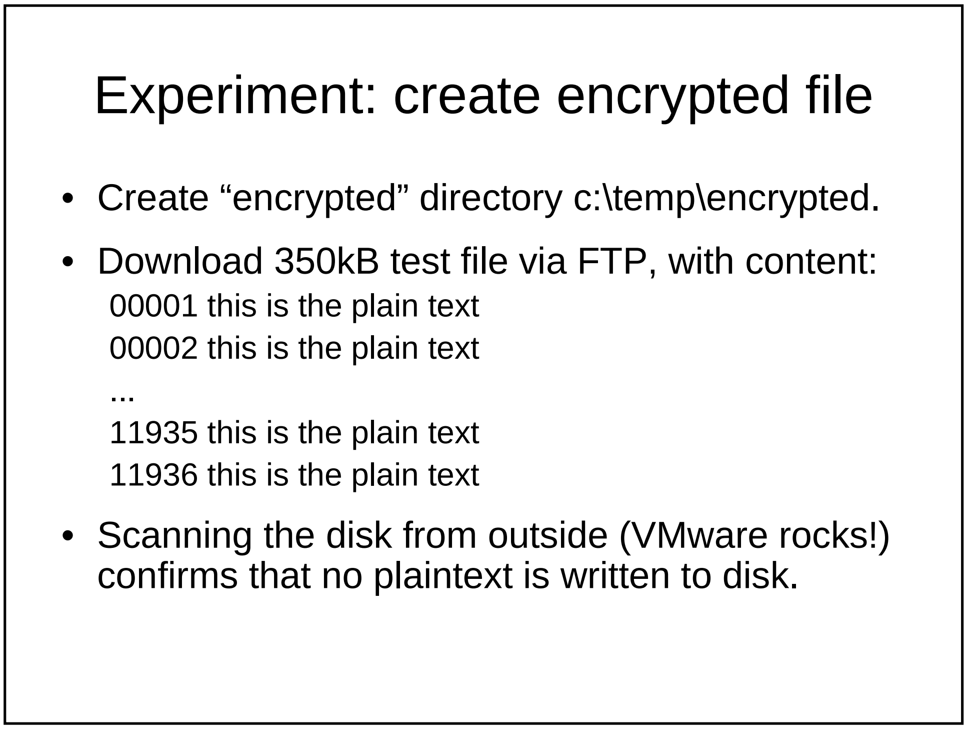
Summary: persistence of main memory

- 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. **Bummer** :-(

Windows/2K/XP encrypted files

(to end on an optimistic note :-)

- EFS 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?

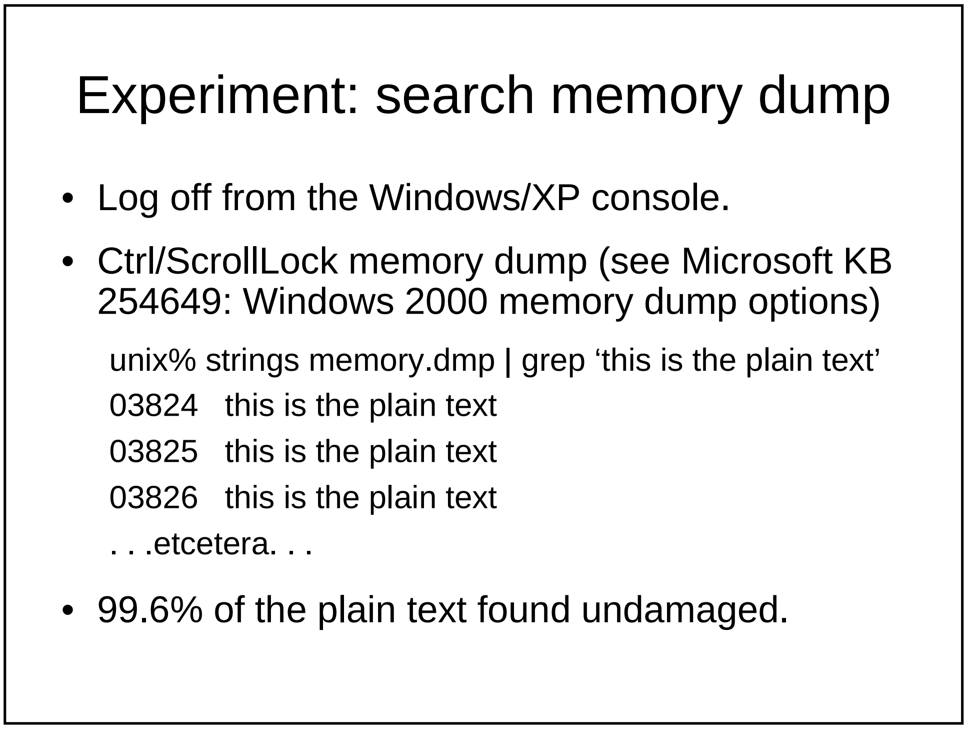


Experiment: create encrypted file

- Create “encrypted” directory c:\temp\encrypted.
- Download 350kB test 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.
- Ctrl/ScrollLock memory dump (see Microsoft KB 254649: Windows 2000 memory dump options)

```
unix% strings memory.dmp | grep ‘this is the plain text’
03824   this is the plain text
03825   this is the plain text
03826   this is the plain text
. . .etcetera. . .
```

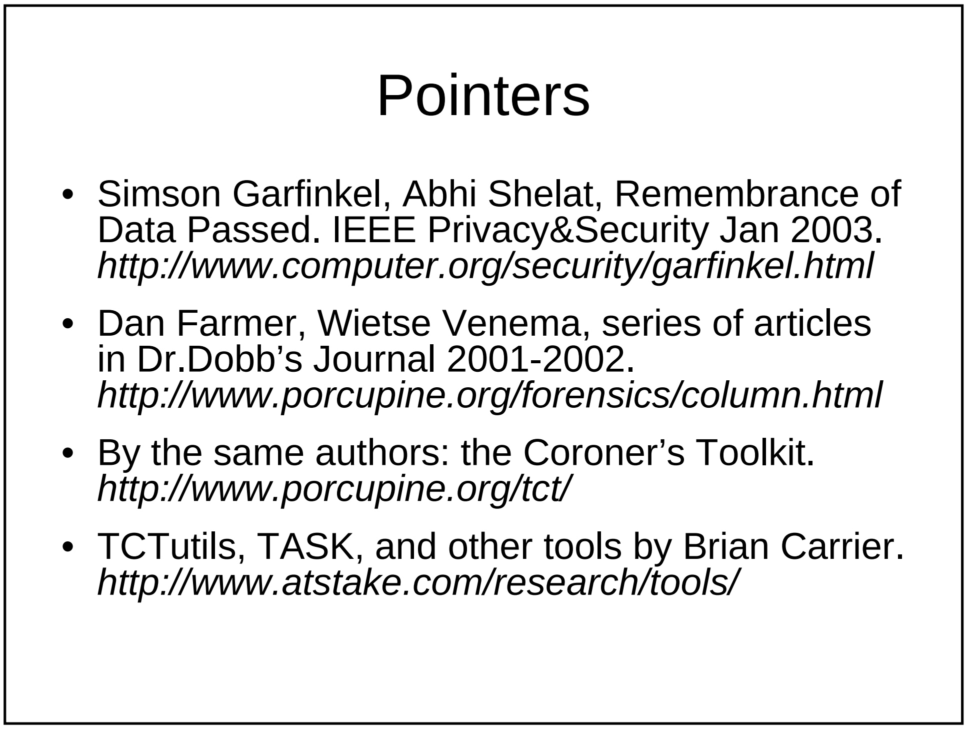
- 99.6% of the plain text found undamaged.

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.

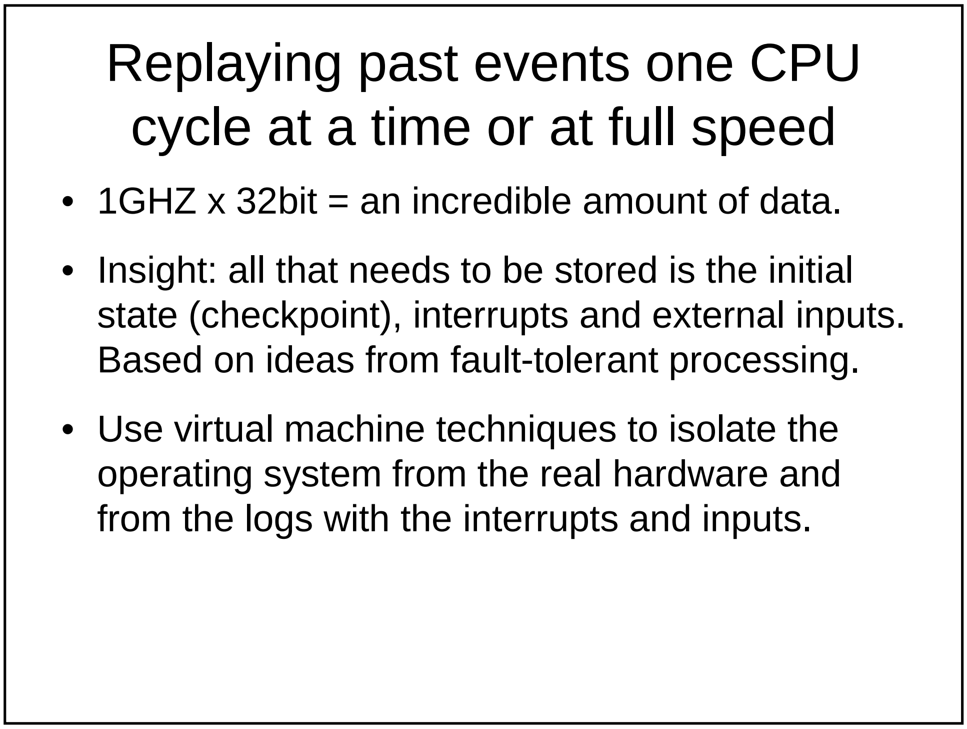
Conclusion

- 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

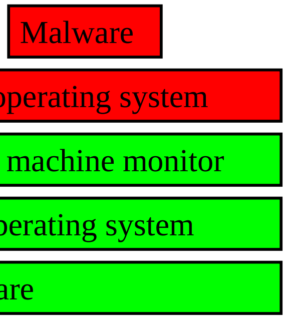
- Simson Garfinkel, Abhi Shelat, Remembrance of Data Passed. IEEE Privacy&Security Jan 2003. *http://www.computer.org/security/garfinkel.html*
- Dan Farmer, Wietse Venema, series of articles in Dr.Dobb’s Journal 2001-2002. *http://www.porcupine.org/forensics/column.html*
- By the same authors: the Coroner’s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



Replaying past events one CPU cycle at a time or at full speed

- 1GHZ x 32bit = an incredible amount of data.
- Insight: all that needs to be stored is the initial state (checkpoint), interrupts and external inputs. Based on ideas from fault-tolerant processing.
- Use virtual machine techniques to isolate the operating system from the real hardware and from the logs with the interrupts and inputs.

Using virtual machine techniques for malware confinement



Applications abound

- Stop replay at an arbitrary point.
- Log into the machine and look around before the evidence was destroyed.
- Go back and resume replay.
- Reduce volume of backups :-)
- Logging rate: 0.2GB/day for workstation.
- OSDI paper by Peter Chen and others: http://www.eecs.umich.edu/CoVirt/papers/revirt.pdf