Persistence and volatility

a paradox of computing

Wietse Venema
IBM T.J. Watson Research Center
Hawthorne, NY, USA

cover.fig

This talk in a nutshell

The paradox:

- Easy to lose information by accident
- Hard to lose information if you want to

Outline of this presentation:

- MACtimes and mostly volatile information
- Persistence of dead information

What are MACtimes

```
($dev, $inode, $mode, $nlink, $uid, $gid, $rdev, $size,
  $atime, $mtime, $ctime, $blksize, $blocks) = lstat($filename);
```

- lstat() looks up the attributes of a UNIX file
- Most information is also available on Windows NTFS
- Some information is even available on old DOS FAT16

More about MACtimes

mtime last modification
Write/truncate file; create/delete directory entry

atime last access*
Read/execute file; look up directory entry

ctime last status change
Owner, permission, reference count, write access

dtime Linux-only delete time

* Well, almost, grumble. Windows NTFS is weird

Example: login session (SunOS 4)

Time	*Size*	*MAC*	*Permissions*	*Owner*	*Group*	*File name*
19:47:04	49152	.a.	-rwsr-xr-x	root	staff	/usr/bin/login
	32768	.a.	-rwxr-xr-x	root	staff	/usr/etc/in.telnetd
19:47:08	272	.a.	-rw-r--r--	root	staff	/etc/group
	108	.a.	-r--r--r--	root	staff	/etc/motd
	8234	.a.	-rw-r--r--	root	staff	/etc/ttytab
	3636	m.c	-rw-rw-rw-	root	staff	/etc/utmp
	28056	m.c	-rw-r--r--	root	staff	/var/adm/lastlog
	1250496	m.c	-rw-r--r--	root	staff	/var/adm/wtmp
19:47:09	1041	.a.	-rw-r--r--	root	staff	/etc/passwd
19:47:10	147456	.a.	-rwxr-xr-x	root	staff	/bin/csh

login.fig

Recent example (Lance Spitzner)

```
Sep 25 00:44:49 dionysis rpc.statd[335]: gethostbyname error for
^X<F7><FF><BF>^X<F7><FF><BF>^Y<F7><FF><BF>^Y<F7><FF><BF>^Z<F7><FF>
<BF>^[<F7><FF><BF>^[<F7><FF><BF>bffff750 8049710     1b068746567627
[several more lines of RFC non-compliant characters...]

Sep 25 00:45:16 dionysis inetd[473]: extra conf for service telnet/
tcp (skipped)

Sep 25 00:45:28 dionysis in.telnetd[11554]: connect from 209.83.81.7
```

statd.fig

MACtimes after rpc.statd exploit

```
Sep 25 2000 01:45:15
       20452 m.c -rwxr-xr-x /bin/prick
      207600 .a. -rwxr-xr-x /usr/bin/as
       63376 .a. -rwxr-xr-x /usr/bin/egcs
       63376 .a. -rwxr-xr-x /usr/bin/gcc
       63376 .a. -rwxr-xr-x /usr/bin/i386-redhat-linux-gcc
        2315 .a. -rw-r--r-- /usr/include/_G_config.h
        1297 .a. -rw-r--r-- /usr/include/bits/stdio_lim.h
        4680 .a. -rw-r--r-- /usr/include/bits/types.h
        9512 .a. -rw-r--r-- /usr/include/features.h
        1021 .a. -rw-r--r-- /usr/include/gnu/stubs.h
       11673 .a. -rw-r--r-- /usr/include/libio.h
       20926 .a. -rw-r--r-- /usr/include/stdio.h
        4951 .a. -rw-r--r-- /usr/include/sys/cdefs.h
     1440240 .a. -rwxr-xr-x /usr/lib/gcc-lib/[...]/cc1
       45488 .a. -rwxr-xr-x /usr/lib/gcc-lib/[...]/collect2
       87312 .a. -rwxr-xr-x /usr/lib/gcc-lib/[...]/cpp
        5794 .a. -rw-r--r-- /usr/lib/gcc-lib/[...]/include/stdarg.h
        9834 .a. -rw-r--r-- /usr/lib/gcc-lib/[...]/include/stddef.h
        1926 .a. -rw-r--r-- /usr/lib/gcc-lib/[...]/specs
       20452 .a. -rwxr-xr-x <hda8-inode-30199>
         537 ma. -rw-r--r-- <hda8-inode-30207>
```

prick.fig

MACtimes after rpc.statd exploit, continued

```
Sep 25 2000 01:45:16
           0 m.c -rw-r--r-- /etc/hosts.allow
           0 m.c -rw-r--r-- /etc/hosts.deny
        3094 mac -rw-r--r-- /etc/inetd.conf
      205136 .a. -rwxr-xr-x /usr/bin/ld
      176464 .a. -rwxr-xr-x /usr/bin/strip
        3448 m.. -rwxr-xr-x /usr/bin/xstat
        8512 .a. -rw-r--r-- /usr/lib/crt1.o
        1124 .a. -rw-r--r-- /usr/lib/crti.o
         874 .a. -rw-r--r-- /usr/lib/crtn.o
        1892 .a. -rw-r--r-- /usr/lib/gcc-lib/[...]/crtbegin.o
        1424 .a. -rw-r--r-- /usr/lib/gcc-lib/[...]/crtend.o
      769892 .a. -rw-r--r-- /usr/lib/gcc-lib/[...]/libgcc.a
      314936 .a. -rwxr-xr-x /usr/lib/libbfd-2.9.5.0.22.so
         178 .a. -rw-r--r-- /usr/lib/libc.so
       69994 .a. -rw-r--r-- /usr/lib/libc_nonshared.a
           0 mac -rw------- <hda8-inode-22111>
           0 mac -rw------- <hda8-inode-22112>
           0 mac -rw-r--r-- <hda8-inode-22113>
       20452 ..c -rwxr-xr-x <hda8-inode-30199>
         537 ..c -rw-r--r-- <hda8-inode-30207>
       12335 mac -rwxr-xr-x <hda8-inode-30209>
        3448 m.. -rwxr-xr-x <hda8-inode-30210>
```

xstat.fig

Timeline of an incident

00:44:49 Exploit rpc.statd buffer overflow

00:45:15 Save existing login program as /bin/prick

00:45:16 Install backdoor /bin/login + /usr/bin/xstat

00:45:16 Add (redundant) telnet service entry to inetd.conf

00:45:16 Disable TCP Wrapper access control

00:45:28 Test the backdoor with telnet connection

17:31:47 Install floodnet DOS tool, update login backdoor

timeline.fig

Examples of MACtime applications

- Post-mortem analysis of incident

  (reconstruction of past behavior)

- Hardening system security

  (determining the footprint of a system)

- MACtimes can be applied to existing and deleted files

Limitations of MACtimes

- Volatile
  - Quickly erode as result of normal activity
  - Only unusual behavior leaves persistent trail
- Easy to forge
  - utime($new_atime, $new_mtime, $filename);
  - Or simply apply the change to the raw disk

Interesting Windows features

- Time stamps change "after the fact" because of the way Windows implements daylight savings time
- Windows NTFS updates the last access time only if the time stamp would change by more than an hour
  Result: Windows shows the time of FIRST access
- Windows NTFS preserves mtime when copying file
  Result: file appears to be created AFTER modified

The UNIX FAQ on recovering deleted files

For all intents and purposes, when you delete a file with "rm" it is gone. Once you "rm" a file, the system totally forgets which blocks scattered around the disk were part of your file.

Even worse, the blocks from the file you just deleted are going to be the first ones taken and scribbled upon when the system needs more space.

faq.fig

"Brute force" survival of deleted data

Kids, don’t do this at home :-)

- Downloaded Linux rootkit V4
- Compiled, installed and removed rootkit
- Downloaded the Coroner’s toolkit (TCT)
- Compiled and ran the TCT software
- Burst of 460 "deleted" MACtimes at time of "incident"
- 300 of those MACtimes were "modified" Nov. 23, 1998*
- Footprints: TCT 300 files, rootkit about 800 files

*The apparent time that Linux rootkit V4 was packaged

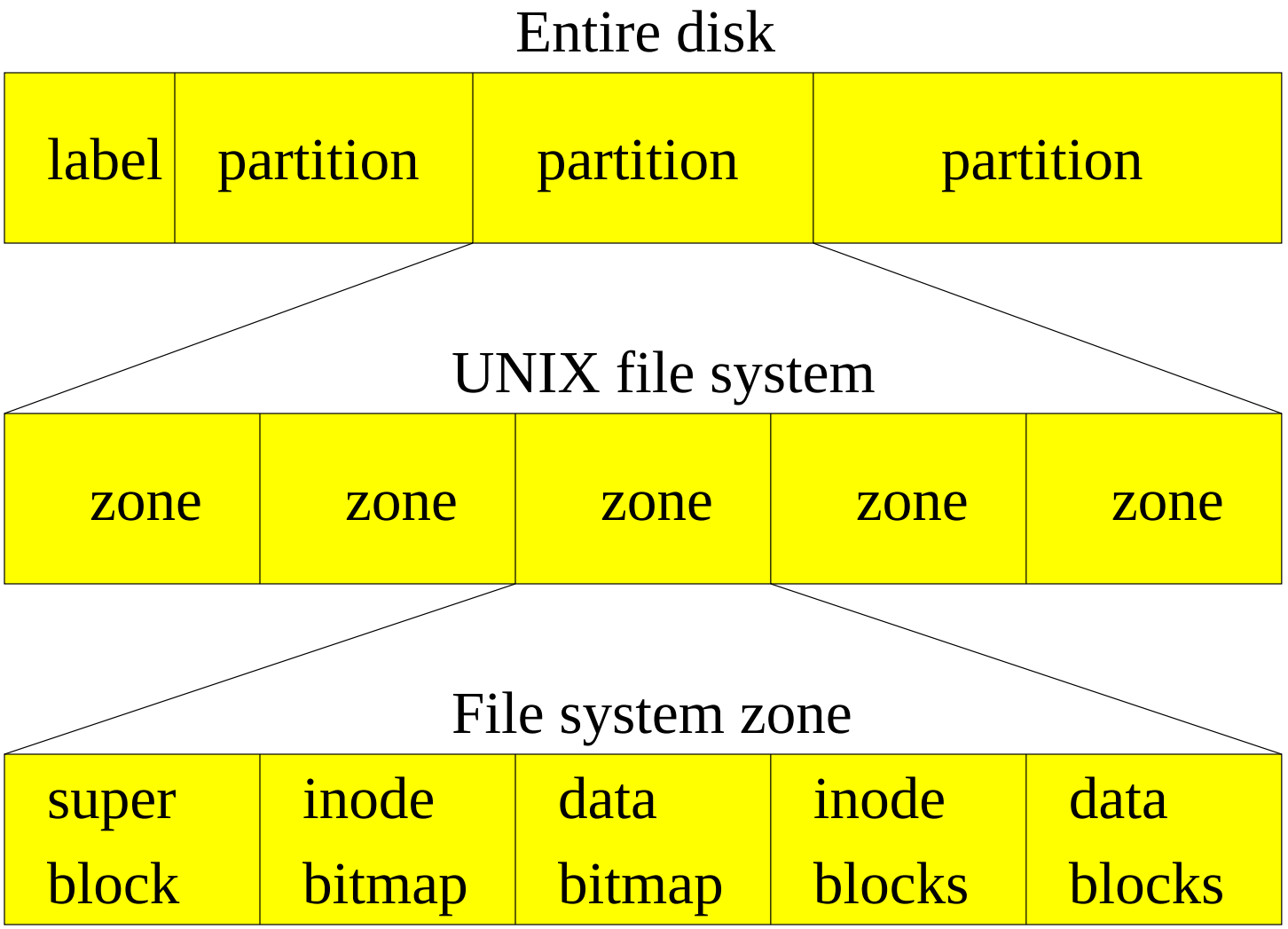
"Long-term" survival of deleted data

Modern UNIX systems do not scatter a file all over the disk

- Less fragmentation gives better read/write performance
- Typically, a file is contained within a file system zone
- Grouping related files together improves access time
- Good locality allows deleted file contents to survive
- Good locality allows deleted file MACtimes to survive

locality.fig

Layout of a typical UNIX/Linux file system



layout.fig

The hello world exploit

Creating and compiling the exploit

```
Aug 04 16:00:14

    85 m.c -rw-r--r-- wietse /home/wietse/hello.c (create source file)

Aug 04 16:00:21

  1024 m.. drwxr-xr-x wietse /home/wietse

  4173 mac -rwxr-xr-x wietse /home/wietse/hello   (create executable)

    85 .a. -rw-r--r-- wietse /home/wietse/hello.c (read source file)
```

The hello world exploit, covert

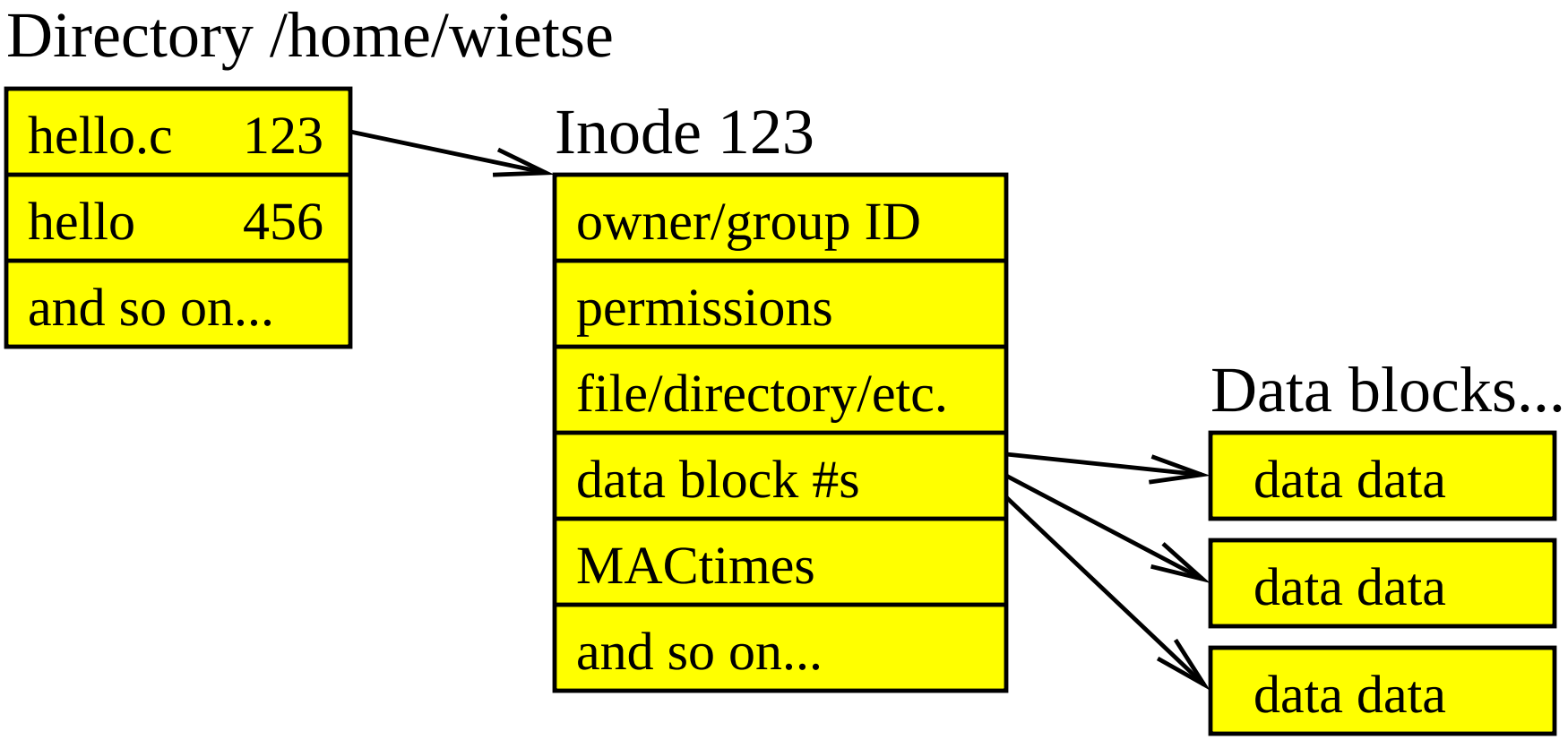
Creating, compiling, running and deleting the exploit

```
Aug 04 16:00:14
    85 m.c -rw-r--r-- wietse /home/wietse/hello.c (create source file)

Aug 04 16:00:21
  1024 m.. drwxr-xr-x wietse /home/wietse
  4173 mac -rwxr-xr-x wietse /home/wietse/hello   (create executable)
    85 .a. -rw-r--r-- wietse /home/wietse/hello.c (read source file)
```

hello-covert.fig

UNIX file system basics



basic-fs.fig

What happens when a UNIX file is deleted?

name (directory entry)	preserved, not linked to inode
attributes (inode block)	
ownership	preserved
MAtime	preserved
Ctime	time of deletion
reference count	zero
file type	Linux: preserved Other UNIX: erased
permissions	
size	
data block locations	
contents (data blocks)	preserved

deleted.fig

The hello world exploit, revealed

```
Aug 04 16:13:08
    85 m.. -rw-r--r-- wietse <hda6-311549> (create source file)
Aug 04 16:13:16
    85 .a. -rw-r--r-- wietse <hda6-311549> (read source file)
  4173 m.. -rwxr-xr-x wietse <hda6-311550> (create executable)
Aug 04 16:13:22
  4173 .a. -rwxr-xr-x wietse <hda6-311550> (run executable)
Aug 04 16:13:28
  1024 m.. drwxr-xr-x wietse /home/wietse
    85 ..c -rw-r--r-- wietse <hda6-311549> (delete source file)
  4173 ..c -rwxr-xr-x wietse <hda6-311550> (delete executable)
```

hello-revealed.fig

Longevity of deleted file MACtimes

Deleted inodes	Time since deletion
1283	day
881	week
2112	month
171	2 months
175	3 months

hades.fig

Longevity of deleted file MACtimes, cont'd

Deleted inodes	Time since deletion
20267	1 month
3226	2 months
10423	3 months
172	4 months
1120	5 months
945	6 months
5107	7 months
262	8 months
1057	9 months
51205	10 months

The paradox

- Visible information is volatile
- Invisible information is persistent
- This paradox repeats at every level of abstraction:
  - File systems
  - Bitmaps, inodes and data blocks
  - Logical disk blocks
  - Magnetic patterns on disk

paradox.fig

Pointers

- The Coroner's toolkit (TCT)
- Doctor Dobb's column on computer forensic analysis
- Full-day class on computer forensic analysis (1999)

  http://www.porcupine.org/forensics/

  http://www.fish.com/forensics/