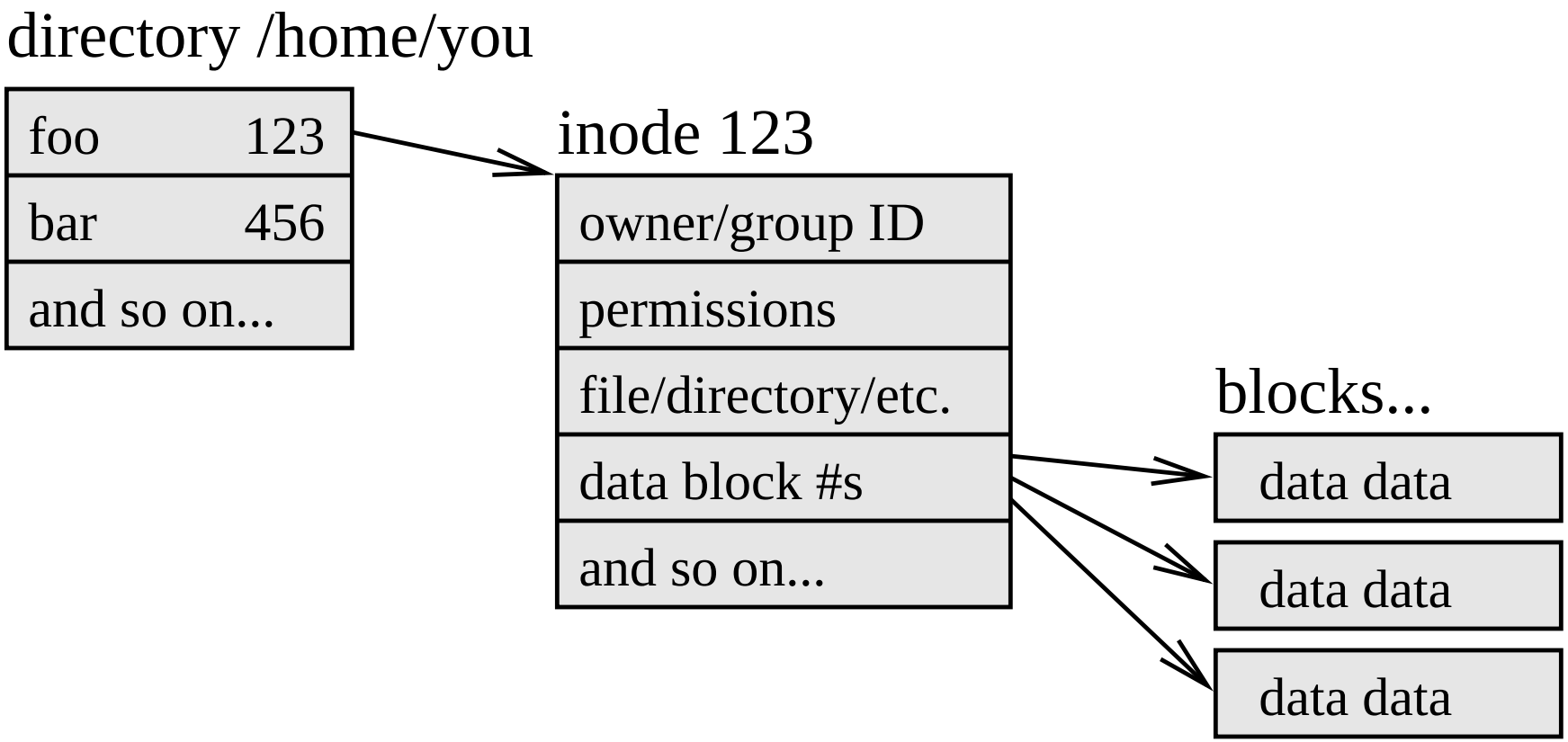
The UNIX file system

A gentle introduction

UNIX file system basics



UNIX file types

- Regular file (most files).
- Directory (a file, nevertheless).
- Symbolic link (alias for other file).
- Device (e.g., terminal, disk, memory).
- Inter-process communication: named pipe, socket.

Unusual file system properties

- Everything is placed in one logical tree. No A:, B:, etc. Even devices are accessible through the file system.
- Directories are files (except users can’t write to them; some remote file systems may disallow reading as well).
- File names can contain anything but / and null (output from "find" may be harmful to programs).
- Files may contain ‘‘holes’’ (where no data is written; ‘‘holes’’ read back as all-zero blocks).

Unusual file system properties

- Multiple references: a file can appear in multiple places (even in places owned by different users).
- Zero references: a file can still exist after it is removed (zero-link file is deleted when closed).
- No built-in undelete provision like DOS.
- Typically, only 0.5 kbytes of wasted space at the end of a file.

UNIX file/directory/etc. attributes

- Ownership: numeric user and group ID.
- Permissions: read/write/execute for owner, group, other.
- Type: file, directory, symlink, device, etc.
- Reference count (0, 1, 2 etc.).
- File size in bytes.
- Time stamps (MAC times):
  - last file Modification time.
  - last file Access time.
  - last status Change (e.g., owner, permissions, refcount).

UNIX file system

Basic forensics

TCP Wrapper-style alert

- Activity at some unlikely hour:

```
Feb  9 03:56:17 wilma in.rlogind[2271]:
                        connect from joe@betty
```

- Email inquiry: Joe was not working at 3 AM.

- An intruder has compromised Joe’s account
  . . . and possibly more.

Sign of trouble - no login record

```
% last | more
sergey    ttyp1    barney       Wed Feb  8 16:44 - 16:45  (00:01)
rob       ttyp1    freddy       Wed Feb  8 09:08 - 09:08  (00:00)
joe       console               Tue Jan 31 17:13   still logged in
erwin     ttyp1    wilma        Fri Jan 27 11:17 - 11:17  (00:00)
nico      ttyp0    betty        Tue Jan 24 08:55 - 08:56  (00:01)
etc.
```

- Either no login record was written (entrance via backdoor)

  . . . or the record was wiped out.

.

Process output looks normal

```
% ps -aux
USER       PID %CPU %MEM   SZ  RSS TT STAT START  TIME COMMAND
you      13048 23.1  3.0  216  428 p3 R    09:12   0:00 ps -aux
root         1  0.0  0.0   52    0 ?  IW   Jan 31  0:00 /sbin/init -
root         2  0.0  0.0    0    0 ?  D    Jan 31  0:02 pagedaemon
root        75  0.0  0.0   16    0 ?  I    Jan 31  0:00  (biod)
root        55  0.0  0.0   68    0 ?  IW   Jan 31  0:00 portmap
joe        183  0.0  1.3 1500  188 co S    Jan 31 11:28 X :0 -auth /ho...
root        60  0.0  1.1  128  156 ?  S    Jan 31  0:05 ypserv
joe        130  0.0  0.0   56    0 co IW   Jan 31  0:00 -csh (csh)
bin         62  0.0  0.0   36    0 ?  IW   Jan 31  0:00 ypbind
root       111  0.0  0.0   12    8 ?  IW   Jan 31 26:24 update
root        76  0.0  0.0   16    0 ?  I    Jan 31  0:00  (biod)
root        77  0.0  0.0   16    0 ?  I    Jan 31  0:00  (biod)
root        78  0.0  0.0   16    0 ?  I    Jan 31  0:00  (biod)
root        89  0.0  0.0   60    0 ?  IW   Jan 31  0:03 syslogd
root       107  0.0  0.2   16   28 ?  S    Jan 31  8:48 /usr/bin/scree...
root       114  0.0  0.0   56    0 ?  IW   Jan 31  0:00 cron
root         0  0.0  0.0    0    0 ?  D    Jan 31  0:40 swapper
joe        182  0.0  0.0   44    0 co IW   Jan 31  0:00 xinit /home/jo...
joe        184  0.0  0.0   28    0 co IW   Jan 31  0:00 sh /home/joe/....
etcetera...
```

.

Some traces in file access times

Output from "ls -lautR /", massaged and sorted by time

```
03:45:43   81920 -rwxr-xr-x root /usr/ucb/ftp
03:46:37   24576 -rwxr-xr-x root /usr/ucb/compress
           24576 -rwxr-xr-x root /usr/ucb/uncompress
           24576 -rwxr-xr-x root /usr/ucb/zcat
03:47:49   65536 -rwsr-x--x root /usr/ucb/rdist
03:50:34    3416 -rwxr-xr-x root /usr/bin/id
03:50:53    1422 -r--r--r-- root /usr/include/setjmp.h
             960 -r--r--r-- root /usr/include/sun4c/setjmp.h
            1089 -r--r--r-- root /usr/include/netinet/icmp_var.h
            1364 -r--r--r-- root /usr/include/netinet/in_pcb.h
             722 -r--r--r-- root /usr/include/netinet/tcp_debug.h
            2060 -r--r--r-- root /usr/include/netinet/tcp_fsm.h
            1117 -r--r--r-- root /usr/include/netinet/tcp_seq.h
            4692 -r--r--r-- root /usr/include/netinet/tcp_timer.h
            6331 -r--r--r-- root /usr/include/netinet/tcp_var.h
             984 -r--r--r-- root /usr/include/netinet/tcpip.h
            1558 -r--r--r-- root /usr/include/netinet/in_var.h
           16716 -r--r--r-- root /usr/include/sys/stream.h
```

.

File access times, continued

(several dozen other /usr/include files omitted)

```
            6970 -r--r--r-- root /usr/include/sys/ttold.h
            1299 -r--r--r-- root /usr/include/sys/ttychars.h
            3756 -r--r--r-- root /usr/include/sys/ttycom.h
             755 -r--r--r-- root /usr/include/sys/ttydev.h
            3053 -r--r--r-- root /usr/include/sys/types.h
03:53:26  344586 -r-xr-xr-x root /usr/lib/ccom
03:53:27  204800 -r-xr-xr-x root /usr/lib/iropt
03:53:32  147456 -r-xr-xr-x root /usr/lib/cg
03:53:33  221215 -r-xr-xr-x root /usr/bin/as
03:53:36  303617 -rwxr-xr-x root /usr/bin/ld
           98304 -rwxr-xr-x root /usr/lib/compile
            1132 -rw-r--r-- root /usr/lib/crt0.o
            7996 -rw-r--r-- root /usr/lib/libc.sa.1.9
03:53:46   10744 -rwxr-xr-x root /usr/bin/install
           16384 -rwxr-xr-x root /usr/bin/strip
03:53:48  115472 -rwxr-xr-x root /usr/bin/make
03:59:06   32768 -rwxr-Sr-x root /usr/kvm/w
```

.

Results so far - not a whole lot

- Wrapper alert: Joe’s account compromised.
- "ls" access times: intruder built/installed software.
- "ls" modification times: no new or modified files.
- "ps" output: no new or unusual processes.

→ Time for more drastic measures.

Finding a good baseline

- Purpose: to find out what has changed relative to some "known to be good" state.
- Questions for different levels of desperation:
  - How useful are the backups?
  - Is a similarly-configured machine available?
  - Are system/application installation media available?

Unmistakable rootkit signature

"find / -type f -print | xargs md5 >file" finds trojan versions of:

- du (hide sniffer, logs, and configuration files)
- ifconfig (hide sniffer activity)
- login (backdoor)
- ls (hide sniffer, logs, and configuration files)
- netstat (hide intruder network connections)
- ps (hide sniffer process)

Plus what turns out to be configuration files, programs, and a network sniffer logfile with login/password information.

Epilog

- Tools can make even unsophisticated intruders smart: smart enough to exploit a vulnerability, wipe their login record, and to almost invisibly install trojan horses.

- Tools do not stop intruders from making unsophisticated mistakes, such as leaving a trail of file access time stamps or stumbling over tripwires.

.

Raw versus cooked

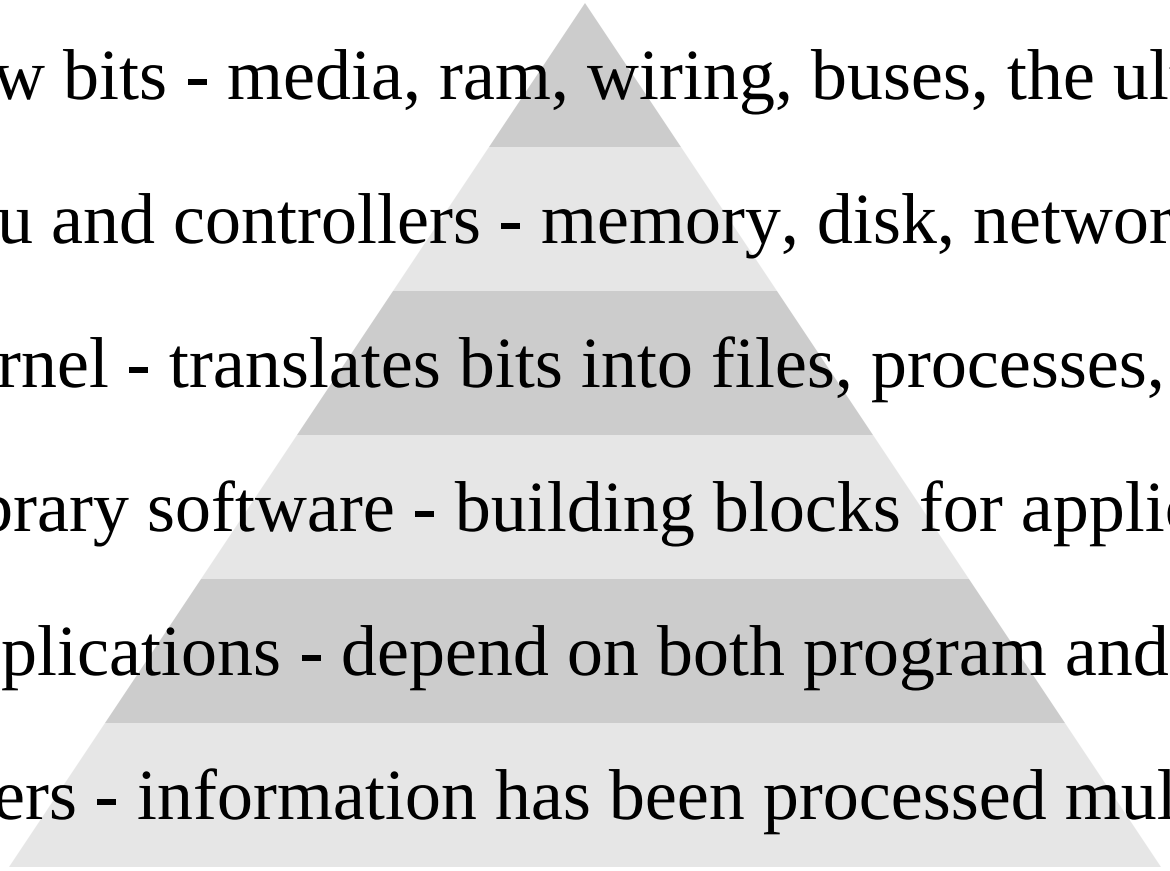
The triangle of truth

UNIX system - logical view

- Users: flesh and blood.
- Processes: agents on behalf of users.
- Connections: process-to-process.
- Files: target of manipulation.

Issue: humans have no direct channels into their brains.

Multiple layers of processing

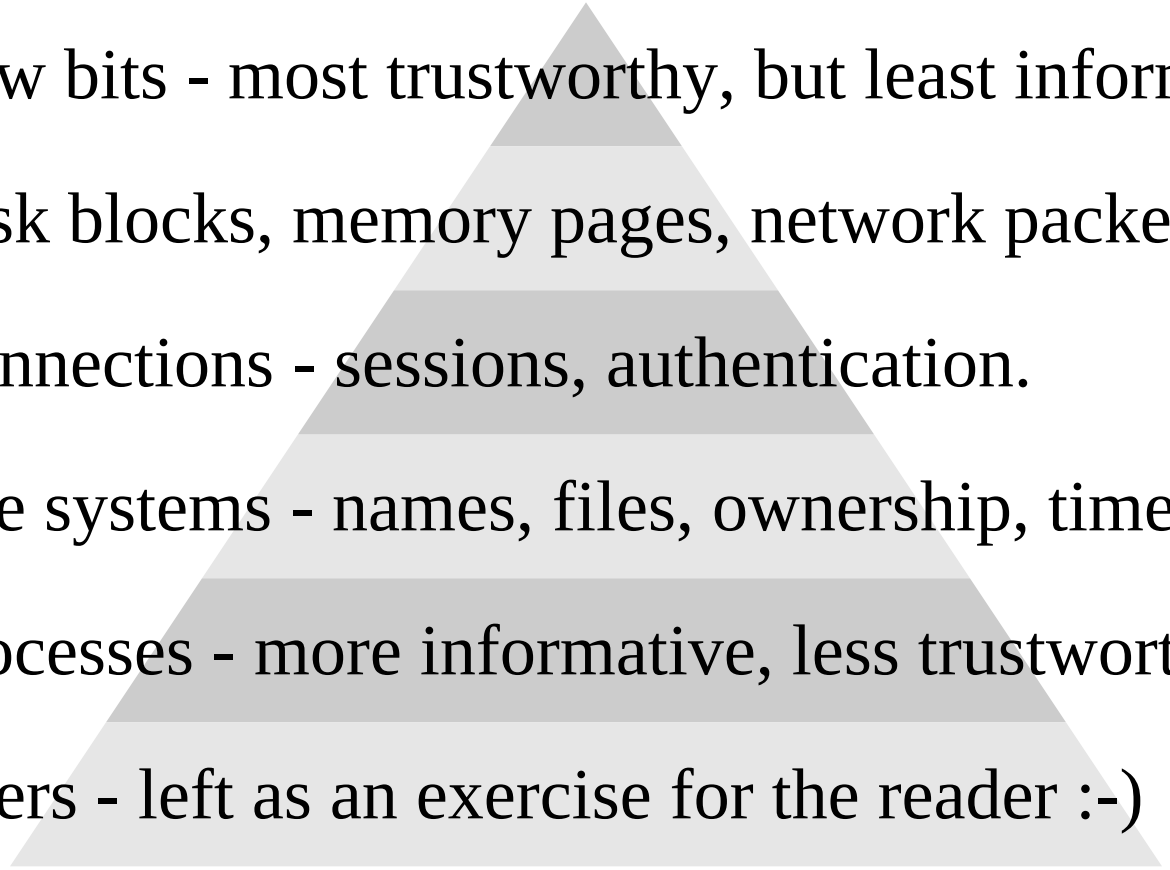


- Raw bits - media, ram, wiring, buses, the ultimate truth.
- Cpu and controllers - memory, disk, network, terminal.
- Kernel - translates bits into files, processes, connections.
- Library software - building blocks for applications.
- Applications - depend on both program and data files.
- Users - information has been processed multiple times.

Opportunities for tampering

- Media - stash data in slack space, bad blocks.
- Firmware - cpu, bios, pal, disk, network controllers.
- Kernel - loadable modules, on-the-fly memory patches.
- Library software - execute trojan, open good file, backdoors.
- Applications - rootkit trojan horse system utilities.
- Processes - on-the-fly memory patches.

Raw versus cooked, and trust



- Raw bits - most trustworthy, but least informative.
- Disk blocks, memory pages, network packets.
- Connections - sessions, authentication.
- File systems - names, files, ownership, time stamps.
- Processes - more informative, less trustworthy.
- Users - left as an exercise for the reader :-)

Limitations

- Truth versus understanding.
- Reverse Turing problem: can an adversary control a system such that it always gives the "right" answer?
- To what extent can procedures be automated?

The triangle of truth

