Advanced Unix System Administration

Lecture 9
March 12, 2008

Steven Luo
<sluo+decal@OCF.Berkeley.EDU>

A Tour of Userspace

- Name Service Switch
  - C library grew functions getpwnam(), etc., to have standard ways of reading /etc/passwd, looking up hostnames, etc.
  - NSS mechanism allows C library to load (via dlopen()) different providers for this information
  - Providers include files, DNS, NIS, NIS+; can be written/installed separately from libc
  - Configured via /etc/nsswitch.conf

A Tour of Userspace

- Pluggable Authentication Modules
  - (Mostly) standardized way of checking users' passwords
  - Allows powerful access controls, multiple authentication providers via dlopen()
  - Available on Linux, Free/NetBSD, Solaris, most commercial Unix
- BSD Authentication
  - Like PAM, but uses program helpers instead of shared libraries; OpenBSD and BSDi

Problem Solving Tools

- Double-check your work!
- Logging/debug output
  - Most daemons and many other programs can be configured to spit out lots of info
  - See the documentation for your program/daemon
- Don't overlook the simple/obvious options before applying more complicated/advanced tools!

Problem Solving Tools

- Program instrumentation
  - Reading the source frequently isn't enough – and that's assuming you have it
  - Even if you have it, source tells you what should be happening, not what is
  - Instrumentation tools can provide a valuable look into what a program is doing
  - Output is slightly difficult to interpret – look for alternatives first!

Problem Solving Tools

- strace(1), truss(1), ktrace(1)
  - Provides a view of the syscalls used by a program
  - Can be run on new processes, follow their children, or be attached to an existing process
  - Output is valuable when process is doing I/O, sleeping, or otherwise talking to the kernel; of no use when purely userspace
  - Can filter out selected syscalls – useful because output is very noisy

Problem Solving Tools

- ltrace(1)
  - Provides a view of the library functions used by a program; can also show syscalls
  - Very similar in use to strace
- gdb, other debuggers
  - Can provide a view of what's happening in the program itself
  - Certain modifications to program behavior also possible
  - Usually requires debugging symbols and code knowledge

Problem Solving Tools

- lsof(1), pfiles(1)
  - Shows you what files a process has open
  - Extremely useful in conjunction with strace
- System instrumentation
  - These tools give you an idea of what the system as a whole is doing
  - Useful when you suspect kernel-level or systemwide problems

Problem Solving Tools

- ps(1)
- top(1)
  - Gives a display of processes that can be sorted by resource usage, and summary of system resource status
- vmstat(1), free(1)
  - Shows amount of free memory
  - vmstat can show other system statistics

Problem Solving Tools

- iostat(1)
  - Shows per-device I/O statistics
  - Various (not widely distributed) tools called iotop show you per-process I/O statistics
- sar(1), sadc(1)
  - sadc collects most of the possible kinds of system activity statistics
  - Can be displayed (sar) or logged somewhere

Problem Solving Tools

- netstat(1)
  - Shows the state of your system's network connections
  - Useful to find out who your system's talking to, finding hanging/excessive network connections, etc.
  - We'll talk plenty more about network debugging tools later

Problem Solving Tools

- DTrace and SystemTap
  - Provide “dynamic tracing” of the kernel's execution
  - Allow you to attach to certain points in the kernel's execution and obtain statistics and information about the execution
  - SystemTap less mature than DTrace, but more powerful (but less safe?)
  - Useful to identify what the kernel's busy doing
  - DTrace also can trace user-space programs