HTGR: Jails, VMs, and Sandboxes

Bill Cheswick
AT&T Research
ches@research.att.com

at&t

The Problem

- Applications fail
- Operating systems fail
- We lose control of our machines

Solution: belt and suspenders

- Build a new, completely different container (a sandbox) to make breakout much less likely

Solutions

- Separate computer per function
- Sandbox for the application that limits access to the OS and rest of the machine
- A virtual machine, that contains the entire activity, good or bad, to a machine

Separate computer

- limits the damage to the computer
- power and software maintenance issues
- simplest solution

Sandboxes

- Unix *chroot* functionality
  - limits access at the file system level
    - I keep learning of new ways to break out
- FreeBSD *jail* function
  - Chroot plus network and other limitations

Solutions are old

- Virtual machines were used by IBM in the 1960s
- Extended operating system protections go back to the 1960s (Multics rings)

Sandboxes: procedure containment

- Janus (Wagner *et al*, 1996): SunOS at library level
- Numerous solutions since then
- PeaPod paper at this LISA
- Not solved yet

Example: defaceless read-only web server

- Each web connection fires up a new web server
- Web server runs as not *root*
- Web server in chroot. Has write permission only to the logs
- This can support 20 queries a second, easily

Not solved yet: problems

- Hard to configure
  - Detail all the permitted system calls for *firefox*
- Not available on all *nix systems
- Hard to detail what information to save between instantiations

Virtual machines

- Commercial products like VMWare and Parallels
- Xen for Linux
- All benefit from recent hardware improvements to the x86 architecture

Virtual Machines (VMs)

- simulate real or other machine at the hardware level
- operating systems run at a higher level
- many may run on a piece of hardware
- often may be checkpointed and restarted
  - invaluable in dangerous environments

VMs

- IBM used them in their mainframes since the 1960s
- Haven't been seen much until recently
- VMware, Parallels (Mac), Xen
- Recent x86 mods have made that hardware more suitable

VM dream

- Build once, load many times
  - different machines, different sessions, different days
  - a read-only web server is a program that emits logs
- Save power, administrative manpower

Example: this Mac

- runs Parallels
- Current systems: FC-8 (4 versions), FreeBSD 6.2, Ubuntu, Windows XP

Program containment

- File system name space control (Plan 9 1980s)
- Replace the library (Janus, 1996)
- Limit system calls (systrace, 2003)
- File system level (PeaPods, 2007)

Program containment dream

- configuration files for important clients (firefox, thunderbird, etc.) and network services (samba, apache, etc.)
- simple enough to understand
- available on all *nix systems

Containment problems

- configuration tends to be hard
  - what system calls should be allowed?
    - what dynamic libraries are needed?
- How do you identify and preserve state across instantiations? (e.g. bookmarks for a browser)

Detecting VMs: arms race

- the bad guys want to know if there are suspenders present
  - abandon or attack
  - limits usefulness as a lab tool
- Ultimately, the detectors win this one, I think

HTGR: Jails, VMs, and Sandboxes

Bill Cheswick
AT&T Research
ches@research.att.com

at&t