Clouds, Virtualization and Security

or

Look Out Below

Lee Badger
Hardware Virtualization
(Box View)

Terminology
Guest OS: runs only on VMM
Host OS: runs only on HW
Domain: virtual machine on VMM
Hypervisor: virtual machine monitor
Making x86 Virtualizable
Using Binary Translation

1. Identify the “next” block by scanning instructions for a jump/call/etc (that ends a basic block).
2. Copy a newly-encountered basic block to the cache.
3. Binary translate any prohibited instruction into a sequence that emulates it “safely.”
4. Run/rerun translated block at full speed.

Technique used by VMware, in 1999.
Making x86 Virtualizable
Using Extra Hardware

Intel 64

Intel version of x86-64 contains ~595 instructions.

Hardware extensions make the instruction set virtualizable.

### Floating Point
- Data: 17
- Arithmetic: 26
- Compare: 14
- Transcendental: 8
- Constants: 7
- Control: 20
- State management: 2
- Total: 94

### SIMD
- MMX: 47
- SSE: 62
- SSE2: 69
- SSE3: 13
- SSSE3: 32
- SSE4: 54
- Total: 277

### General Purpose
- Data transfer: 32
- Arithmetic: 18
- Logical: 4
- Shift/rotate: 9
- Bit/byte: 23
- Control transfer: 31
- String: 18
- I/O: 8
- Enter/leave: 2
- Flag control: 11
- Segment register: 5
- Misc: 6
- Total: 167

VT-x Extensions: 12
Safe mode: 1
System: 34
64-bit mode: 10
Intel Virtual Machine Extensions (VMX)

Legacy software runs in the expected rings, hopefully unaware.

“There is no software-visible bit…indicates…VMX non-root operation”, Intel 64 manual.

Deprivileged (very configurable).

• Many instructions cause fault-like VM exits:
  – interrupts
  – I/O events
  – page table management
  – privileged instructions, etc.

• VMM handles faults
• VM exit rate determines performance
• Address translation is complex
How Complex is Virtualization?

VMM code counts generated using David A. Wheeler's "SLOCCount" tool.
Windows estimate from Bruce Schneier
Linux estimates from Gonzalez-Barahona et al., and David Wheeler

Source Lines Of Code

55,000,000
35,000,000
20,000,000
15,000,000
3,000,000
1,000,000
60,000

1990 2000 2008

Operating system
Virtualization system

Legend

- Debian Linux
- Windows 2k
- Windows NT
- Red Hat Linux
- Windows 95
- Windows 3.1
- Bochs
- Kaffe
- Xen
- Qemu
- VirtualBox
VMM Implementation Quality Should Not be Assumed

In 2007, Tavis Ormandy subjected 6 virtualization systems to guided random testing of their invalid instruction handling and I/O emulation.

- Bochs: 178k SLOC
- QEMU: 373k SLOC
- VMWare: 910k SLOC
- Xen
- Anonymous 1
- Anonymous 2

All of the systems failed the tests, most with “arbitrary execution” failures.

Device emulation was a particular area of vulnerability.

For details, see: taviso.decsystem.org/virtsec.pdf

Code counts generated using David A. Wheeler’s “SLOCCount” tool.
Potential Security Advantages

- An extra layer for defense in depth.
- Stronger encapsulation of errors-or-attacks within a VM.
- More intrusive intrusion detection via introspection.
- More limited exposures of buggy/weak software.
- More flexible discovery/eviction of kernel rootkits.
- Snapshots enable rollback for fault/intrusion tolerant computing.
- Security policy regulating VMs may be simpler than policy regulating processes.
Potential Security Disadvantages

- VM layer is complex too: composite system is complex.
- VM layer configuration is security relevant.
- Mapping VM storage onto host files may cause overlap.
- Trusted Platform Module (TPM) hard to virtualize.
- Remote attestation may not work.
- Covert channels not well understood.
- VM escape.
- VM detection.
- VM-VM interference.
- V networking configuration errors.
- Malicious virtualization risk.

Shared Resources increase risk:
- networks
- clipboards
- clocks
- printers
- desktop management
- folders
NIST Guide to Platform Virtualization Security

• Practical and operational guide
• Security challenges and benefits
• Attributes and properties
• Platform virtualization architecture
• Threat model
• Security recommendations
VMM Formal Requirements
(summary of Popek and Goldberg, 1974 CACM)

For machines having: 1) user/supervisor modes, 2) location-bounds register, and 3) a trapping mechanism.

Sensitive Instructions (change or depend on memory map or mode) ⊆ Privileged instructions (trap iff user mode)

If then a Virtual Machine Monitor (VMM) can be built having 3 properties:

Efficiency: most instructions run directly.

Resource Control: the VMM allocates all resources.

Equivalence: the user program mostly believes it runs on the hardware.