Windows Server Virtualization & The Windows Hypervisor

Brandon Baker
Lead Security Engineer

Windows Kernel Team
Microsoft Corporation

Black Hat 2007
Agenda - Windows Server Virtualization (WSV)

- Why a hypervisor?
- Quick Background & Architecture
  - For more details, see presentation on conference CD
- Security Characteristics
- Deployment Considerations
- Future Directions
Why a hypervisor?

- Thin, low level microkernel
- Eliminates ring compression
- Runs guest operating systems w/o modification
- Adds defense in depth
- Leverage current & future hardware
- Scalability
Agenda - Windows Server Virtualization (WSV)

- Why a hypervisor?
- Quick Background & Architecture
  - For more details, see presentation on conference CD
- Security Characteristics
- Deployment Considerations
- Future Directions
Windows Server Virtualization Background

- Project code name Viridian
- Full machine virtualization for guest operating systems
- Component of Windows Server 2008
- Final version available within 180 days of Windows Server 2008 RTM
- Installs as a role on Server Core
Windows Server Virtualization Background

Has three major components:
- Hypervisor
- Virtualization Stack
- Virtual Devices

Hypervisor Based
- Takes advantage of (and requires) processor virtualization extensions
- Supported on x64 hardware only, 32/64bit guest support
The Old Way
Virtual Server Architecture

Host

Virtual Server WebApp
IIS
Virtual Server Service

Guests

Ring 3: User Mode
Guest Applications

Ring 1: Guest Kernel Mode
VM Additions
Windows (NT4, 2000, 2003)

Windows Server 2003 or Windows XP
Kernel
Device Drivers

Provided by:
Windows
Virtual Server
ISV

Server Hardware
The New Way
WSV Architecture

Root Partition
- Virtualization Stack
  - WMI Provider
  - VM Service
  - VM Worker Processes
- Server Core
  - Windows Kernel
  - Device Drivers

Guest Partitions
- Guest Applications
- Virtualization Service Clients (VSCs)
- VMBus
- Enlightenments
- OS Kernel
- Ring 0: Kernel Mode

Provided by:
- Windows
- Viridian
- ISV

Ring 3: User Mode

Black Hat 2007
Virtualization Attacks

Root Partition

- Virtualization Stack
  - WMI Provider
  - VM Service
  - VM Worker Processes

- Server Core
  - Windows Kernel
  - Device Drivers

- Virtualization Service Providers (VSPs)

- VMBus

- Virtualization Service Clients (VSCs)

- OS Kernel

- Enlightenments

Guest Partitions

- Guest Applications

- Ring 0: Kernel Mode

- Ring 3: User Mode

Provided by:
- Windows
- Viridian
- ISV
- Hackers

Windows hypervisor

Server Hardware

Black Hat 2007
Hypervisor

- Partitioning Kernel
  - Partition is isolation boundary
  - Few virtualization functions; relies on virtualization stack
- Very thin layer of software
  - Microkernel
  - Highly reliable
- No device drivers
  - Two versions, one for Intel and one for AMD
  - Drivers run in the root
  - Leverage the large base of Windows drivers
- Well-defined interface
  - Allow others to create support for their OSes as guests
Virtualization Stack

- Runs within the root partition
- Portion of traditional hypervisor that has been pushed up and out to make a micro-hypervisor
- Manages guest partitions
- Handles intercepts
- Emulates devices
Agenda - Windows Server Virtualization (WSV)

- Why a hypervisor?
- Quick Background & Architecture
  - For more details, see presentation on conference CD
- Security Characteristics
- Deployment Considerations
- Future Directions
Security Assumptions

- Guests are untrusted
- Root must be trusted by hypervisor; parent must be trusted by children.
- Code will run in all available processor modes, rings, and segments
- Hypercall interface will be well documented and widely available to attackers.
- All hypercalls can be attempted by guests
- Can detect you are running on a hypervisor
- We’ll even give you the version
- The internal design of the hypervisor will be well understood
Security Goals

- Strong isolation between partitions
- Protect confidentiality and integrity of guest data

Separation
- Unique hypervisor resource pools per guest
- Separate worker processes per guest
- Guest-to-parent communications over unique channels

Non-interference
- Guests cannot affect the contents of other guests, parent, hypervisor
- Guest computations protected from other guests
- Guest-to-guest communications not allowed through VM interfaces
Security Non-Goals

Things we don’t do in Windows Server Virtualization*

- Mitigate hardware bleed-through (inference attacks)
- Mitigate covert channels
- Guarantee availability
- Protect guests from the root
- Protect the hypervisor from the root
- Utilize trusted hardware
  - TPM, Device Assignment, DMA protection, Secure Launch

*at least, not yet
WSV Security Hardening (1/2)

- Hypervisor has separate address space
  - Guest addresses != Hypervisor addresses
- No 3rd party code in the Hypervisor
- Limited number of channels from guests to hypervisor
  - No “IOCTL”-like things
- Guest to guest communication through hypervisor is prohibited
- No shared memory mapped between guests
- Guests never touch real hardware i/o
WSV Security Hardening (2/2)

- Hypervisor built with
  - Stack guard cookies (/GS)
  - Hardware No eXecute (NX)
  - Code pages marked read only
  - Memory guard pages
  - Limited exception handling
  - Hypervisor binary is signed

- Hypervisor and Root going through SDL
  - Threat modeling
  - Static Analysis
  - Fuzz testing
  - Penetration testing
Hypervisor Security Model

Memory
- Physical Address to Partition map maintained by Hv
- Parent/Child ownership model on memory
- Can supersede access rights in guest page tables (R, W, X)

CPU
- Hardware guarantees cache & register isolation, TLB flushing, instruction interception

I/O
- Hypervisor enforces Parent policy for all guest access to I/O ports
- WSV v1 policy is guests have no access to real hardware

Hypervisor Interface
- Partition privilege model
- Guests access to hypercalls, instructions, MSRs with security impact enforced based on Parent policy
- WSV v1 policy is guests have no access to privileged instructions
WSV Security Model

- Uses Authorization Manager (AzMan)
  - Fine grained authorization and access control
  - Department and role based
  - Segregate who can manage groups of VMs
- Define specific functions for individuals or roles
  - Start, stop, create, add hardware, change drive image
- VM administrators don’t have to be Server 2008 administrators
- Guest resources are controlled by per VM configuration files
- Shared resources are protected
  - Read-only (CD ISO file)
  - Copy on write (differencing disks)
Time Virtualization
Three types of time

- **Calendar time**
  - Affected by Daylight Savings changes
  - Source is parent-created virtual RTC device

- **Machine time**
  - Unaffected by Daylight Savings changes
  - 5 seconds in the future, etc.
  - Sources
    - Per-VP virtualized APIC timer (periodic or single-shot)
    - Four per-VP SynIC timers (periodic or single-shot)
    - Per-partition constant-rate monotonically-increasing reference counter

- **Scheduling time**
  - How long has this processor been scheduled
Time Virtualization
Design Choice

How to handle RDTSC?

- When a Virtual Processor (VP) is intercepted, a single instruction can appear to take a long time – namely, the time it takes to enter the hypervisor, perform actions, and return to a guest.

- TSC is recorded and can be modified in guest control structure (VMCS/VMCB).

“Allow it to advance naturally”
- Just leave it alone
  But...
- A VP can be rescheduled on a different LP, whose TSC could be smaller
- Can’t allow TSCs to jump backwards in time

“Modify it to appear unchanged”
- On entry into the Hv, record guest TSC.
- On return to guest, reload original TSC value minus some amount
  But...
- Never know how long the return instruction will take (caches!)
- Still observable at a certain granularity

Some software depends on knowing cycle counts between instruction blocks (video/audio codecs)
So, we allow it to advance naturally, with a guarantee that it will never appear to go backwards on a given VP.
Agenda - Windows Server Virtualization (WSV)

- Why a hypervisor?
- Quick Background & Architecture
  - For more details, see presentation on conference CD
- Security Characteristics
- Deployment Considerations
- Future Directions
Deployment Considerations (1/2)

- Patching the hypervisor
  - Windows Update
- Managing lots of virtual machines
  - System Center – Virtual Machine Manager
- Minimize risk to the Root Partition
  - Utilize Server Core
    - Don’t run arbitrary apps, no web surfing
    - Run your apps and services in guests
  - Connect to back-end management network
    - Only expose guests to internet traffic
- Enable NX and virtualization in BIOS
Deployment Considerations (2/2)

- Two virtual machines can’t have the same degree of isolation as two physical machines:
  - Inference Attacks
  - Covert Channels

- Not recommended to host two VMs of vastly differing trust levels on the same system
  - e.g. a front-end web server and a certificate server
Agenda - Windows Server Virtualization (WSV)

- Why a hypervisor?
- Quick Background & Architecture
  - For more details, see presentation on conference CD
- Security Characteristics
- Deployment Considerations
- Future Directions
Future Security Benefits

- Many types of virtualization (app, OS, machine) each with increasing levels of isolation (and overhead)
- Powerful tool for virus isolation and analysis
- Improved forensic capability for compromised operating systems
- Investments in OS hardening through hypervisor features
- Potential for greater intra-OS isolation (e.g. Ring 0 separation of drivers)
- VMs can be leveraged for hosting security appliances
Security Challenges

- VM to VM network monitoring
- Managing VM OS patch levels
- Leakage of information between partitions due to shared hardware
- Larger attack surface than air-gapped machines
- High availability – SLA attacks
- Threat of malicious, unauthorized hypervisors (hypervisor-mode rootkits)
Future Security Work

Secure Launch
- Intel TXT\textsuperscript{tm} (senter) and AMD SVM\textsuperscript{tm} (skinit)
- Gives machine owner ability to control what code can use ring -1
- Policy enforcement in hardware to block launch of unauthorized hypervisors
- Allows hypervisor to protect itself against tampering

DMA Remapping
- Intel VT-d and AMD IOMMU
- Gives guests gated access to real hardware
- Allows hypervisor to protect self against DMA attack
Conclusion

- Hypervisors kick ass.
- Beta available with Server 2008 RTM
- We want your feedback
  
  
  brandon.baker@microsoft.com