 Nested Virtualization: Hyper-V on KVM

Ladi Prosek

October 26th 2017
AGENDA

1. What
2. Why
3. How
AGENDA

1. **What** is Hyper-V? What is nested virtualization?
2. Why
3. How
AGENDA

1. **What** is Hyper-V? What is nested virtualization?
2. **Why** should we care?
3. **How**
AGENDA

1. **What** is Hyper-V? What is nested virtualization?

2. **Why** should we care?

3. **How** does it work?
Hyper-V

- Microsoft's x86 virtualization solution
- Ships as
  - Microsoft Hyper-V Server – standalone product
  - Hyper-V role – Windows Server, higher editions of client Windows
- Type-1 hypervisor, root/parent partition hosts the management OS
Hyper-V (cont)

Hyper-V High Level Architecture

Root Partition
- VMWPs
- VMMS
- WM1
- VSps
- VID
- WinHv
- VMBus
- I/O Stack
- Drivers

Enlightened Windows Child Partition
- User Applications
- VSCs/ICs
- WinHv
- VMBus
- I/O Stack
- Drivers

Enlightened Linux Child Partition
- User Applications
- Linux VSCs/ICs
- LinuxHv
- I/O Stack
- Drivers
- VMBus

Unenlightened Child Partition
- User Applications
- Kernel

Hypervisor
- Hypercalls
- MSR
- APIC
- Scheduler
- Address Management
- Partition Manager

Processors

Memory
Nested virtualization

- Running a hypervisor in a VM

Windows

Nested VM

Hyper-V

... L2 guest

... L1 guest, L1 hypervisor

Linux / KVM

... L0 hypervisor

Hardware
Why Hyper-V on KVM?

- Testing, development, training, demos, ..., general tinkering
  - May eliminate the need for dedicated HW
  - All the benefits of virtualization when the workload is virtualization

- Virtualization-based security (VBS)
  - New in Windows Server 2016 and Windows 10
  - Hyper-V used under the covers to protect the OS from itself / from malware
  - Praised by security researchers

- Hyper-V on Hyper-V works (Azure supports nested already)
  - Expecting demand for Hyper-V on KVM also
    - And maybe KVM on Hyper-V as well
Virtualization-based security

- Virtual Trust Level (VTL)
  - VTL 0 is normal, VTL 1 is secure
  - SLAT enforced

- Hyper-V no longer trusts the root partition running in VTL 0

- Small amount of code runs in VTL 1
  - Minimal kernel + security related modules
  - User mode trustlets

- **Device Guard** prevents running unsigned/untrusted code
  - SLAT enforced W^X
  - Together with secure boot, IOMMU, TPM, ...

- **Credential Guard** hides cryptographic secrets
  - SLAT enforced !R
void experiment(void)
{
    char ret_instruction = 0xc3;
    void (*func_ptr)(void) = (void (*)(void))&ret_instruction;

    func_ptr();
}
Experiment (cont)

:-(

Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.

40% complete

For more information about this issue and possible fixes, visit https://www.windows.com/stopcode

If you call a support person, give them this info:
Stop code ATTEMPTED EXECUTE OF NOEXECUTE MEMORY
void experiment(void)
{
    char ret_instruction = 0xc3;
    void (*func_ptr)(void) = (void (*)(void))&ret_instruction;

    make_page_executable(&ret_instruction);

    func_ptr();
}
Experiment (enhanced, VBS on)

:::

Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.

20% complete

For more information about this issue and possible fixes, visit https://www.windows.com/stopcode

If you call a support person, give them this info:
Stop code: SYSTEM SERVICE EXCEPTION
How does it work?

- Same as KVM on KVM but the devil is in the detail...

- Issues found so far
  - Missing features (very few)
    - Ex: Descriptor table exits (EXIT_REASON_GDTR_IDTR, EXIT_REASON_LDTR_TR)
  - KVM bugs (many)
    - Ex: Dereferencing CR3 under PAE + EPT
  - Hyper-V bugs (very few)
    - Ex: Assuming the presence of IOAPIC_REG_EOI

- ~20 KVM patches so far, and a tiny bit of QEMU work
Future work

- Performance!
  - Windows boot time currently doubles after enabling VBS :(

- Paravirtualized features as per Hyper-V Top-Level Functional Spec
  - Enlightened VMCS
  - Enlightened MSR bitmap
  - Virtual TLB

- Please test Hyper-V L1 when making nVMX / nSVM changes
  - Windows Server 2016 evaluation available for download
  - Use standard HV enlightenments
    - -cpu ...,hv_relaxed,hv_spinlocks=0x1fff,hv_vapic,hv_time
    - -cpu ...,-hypervisor required for older Hyper-V versions
Debugging tips

- QEMU GDB stub
  - `qemu -gdb tcp::1234`
  - `(gdb) target remote localhost:1234`
  - Hyper-V lives in hvix64.exe (Intel), hvax64.exe (AMD)
    - No public symbols
    - Addresses change due to ASLR – search for patterns

- Windows kernel debugger
  - `bcdedit /hypervisorsettings {serial or 1394 settings}`
  - `bcdedit /set hypervisordebug on`
  - `bcdedit /set hypervisorlaunchtype auto`

  - Run windbg on another Windows VM
Demo!

Welcome to Microsoft Windows . . .

The PC operating environment that lets you use today’s most popular applications -- and tomorrow’s most powerful ones.
Summary

- Hyper-V on KVM works
  - But expect rough edges, especially around performance

- Virtualization-based security uses Hyper-V
  - Marketing names: Device guard, Credential guard

- Installing the Hyper-V role starts an L2
Resources & QA

- Ladi Prosek <lprosek@redhat.com>
- http://ladipro.wordpress.com