# OpenBSD vmm/vmd Update

Mike Larkin

bhyvecon 2018
09 Mar 2018 – Tokyo, Japan

# Agenda

- Where we were a year ago
- Current status
- Future plans
- Q&A

# One Year Ago ...

- Limited guest VM choices
  - Decent support for OpenBSD i386/amd64
  - Not much else ...
- amd64 and i386 host support
- Early/basic SVM support
- Functional vmctl(8)/vmd(8)
  - A bit unstable at times ...

# This Past Year ...

- Improving core features
- Adding new guest OS support
- Bug fixing / paying down technical debt

# 2017 vmm(4) Improvements

- Main goal was to broaden guest OS support ...
- Added code to support SeaBIOS/UEFI
  - Needed for Linux (and other) guest support
  - Missing PIC/PIT features
  - Missing PCI config space features
  - Missing MC146818 RTC features

# 2017 vmm(4) Improvements (cont'd)

- SeaBIOS delivered via fw_update(1)
  - vmm_firmware package
  - Includes sgabios VGA-to-serial redirector
  - Supports VMX and SVM
    - VMX users need Westmere or later CPU :(

# 2017 vmm(4) Improvements (cont'd)

- Improved platform support
  - Substantially better SVM code
  - AVX/AVX2/AVX512 guest support
  - TSC support in guest
    - Helps avoid too-fast or too-slow time in VM
- … plus many other small changes

# 2017 vmm(4) Improvements (cont'd)

Goal : Support More Guest OSes

# 2017 vmm(4) Improvements (cont'd)

- Linux guest support
  - 32/64 bit
  - No known nonfunctional distributions
  - Latest to be added was CentOS/RHEL
    - Required CD-ROM support
  - Guest still sees virtio devices
  - Graphics can be redirected locally via VNC

# 2017 vmm(4) Improvements (cont'd)

- Other less common guest OSes now work as well:
  - DOS
  - Plan9
  - Android
    - Just really Linux, though ...
  - Solo5/ukvm (Courtesy Adam Steen)
  - Solaris/Illumos/OI
    - Not 100% - graphics related?

# 2017 vmm(4) Improvements (cont'd)

- What about FreeBSD/NetBSD guests?
  - pd@ has these locally working
- Requires instruction emulation
  - bus_space_write_multi(..) used in console I/O
  - turns into a "*rep outsb* from memory" instruction
- We have not needed an instruction emulator until now ...

# 2017 vmd(8) Improvements

- vmd(8) saw improvements as well ...

# 2017 vmd(8) Improvements

- vmd(8) saw improvements as well …
- VirtIO SCSI host-side support for .iso images (CD/DVD images)
  - Implemented by ccardenas@

# 2017 vmd(8) Improvements (cont'd)

- vmd(8) "local networks"
  - Implemented by reyk@
  - Makes configuring NAT networking for VMs much easier:

```
/etc/pf.conf:
pass out on $ext_if from 100.64.0.0/10 to any nat-to $ext_if

/etc/sysctl.conf:
net.inet.ip.forwarding=1

vmctl start -L myvm
```

# 2017 vmd(8) Improvements (cont'd)

- vmd(8) "local networks"
  - vmd has a built-in DHCP/BOOTP server
  - Assigns IP addresses from 100.64.0.0/10 range
    - "Carrier Grade NAT" reserved IP range
    - Can be overridden if desired
  - Assigns corresponding gateway on host side
    - Sends DHCP option to guest to configure gateway

# 2017 vmd(8) Improvements

- VM pause/resume & send/receive (snapshots)
  - vmctl pause ubuntu
  - vmctl unpause ubuntu
  - vmctl send ubuntu > ubuntu.vm
  - vmctl receive ubuntu < ubuntu.vm
- Features implemented initially by team of 4 SJSU MSSE students
  - Committed and maintained by pd@

# 2017 vmd(8) Improvements

- Send / Receive can also be performed over SSH (paused migration):

```
vmctl send openbsd | ssh mlarkin@host vmctl receive
```

- The VM send files can be stored (eg, snapshots), if desired:

```
vmctl send openbsd > /home/mlarkin/vm_backups/openbsd.vm
```

# How Send/Receive Work

- Send/Receive wait until the VM is HLTed
  - Eg, while the OS is in it's idle loop
- Pause the VM
- Serialize device and CPU state to output stream
  - CPUID feature flags
  - Internal legacy device state (PIC state, PIT counter state, etc)

# How Send/Receive Work (cont'd)

- Transfer memory pages to output stream
- Destroy the VM

- On Resume …
  - Read CPUID flags, compare with local host capabilities
    - Abort if incompatible
  - Restore memory pages and device state
  - Resume VM

# How Send/Receive Work (cont’d)

- Ideally, can use switch(4)/switchd(8) to manage connection state across send/receive

# vmctl send/receive Demo

# 2018 Goals

- Isn’t every year the year of “reduce the bug count”?
- Solicit community involvement
  - Glad to have lots of new faces at the vmm table
- Continue pd@’s effort
  - Instruction emulation and memory walker
  - Needed for SMP, proper shadow paging, support for older CPUs, more guest OS support, etc…

# 2018 Goals (cont'd)

- Add support for more modern emulated hardware
  - … 1997 called, they want their PC back
- Did I mention "fix bugs"?

# New Ideas For vmm(4)

- At the t2k17 Toronto Hackathon, a bunch of us were sitting around having beer …

  … oh no, not this again :)

# New Ideas For vmm(4) (cont'd)

- At the t2k17 Toronto Hackathon, a bunch of us were sitting around having beer …
- … talking about how we might be able to use vmm(4) to help secure memory
  - Part of a broader conversation about reducing attack surfaces

# New Ideas For vmm(4) (cont'd)

- Nested Paging (used by vmm currently) can offer execute-only memory on some CPUs
  - Can't read it, can only execute it
- Could we use this to protect code pages from scanning?
  - ROP gadget scans and generally keeping prying eyes away

# New Ideas For vmm(4) (cont'd)

- Idea:
  - Start vmm(4) early
  - Convert existing host into VM
  - Protect code pages as XO
- Note – This idea is not new
  - Concepts first (?) introduced as bluepill in 2006
  - Others have done similar things

# New Ideas For vmm(4) (cont'd)

- Challenges:
  - Legitimate reads
    - ddb(4)
    - Compiler-generated data islands
  - Compatibility with vmd(8)
- ddb(4) is easily handled
  - Hypercall (VMCALL instruction) to exit host-VM
  - Need to make sure that doesn't become a new gadget

# New Ideas For vmm(4) (cont'd)

- Switch/jump tables (data islands) were a problem with gcc
  - … then fixed
  - … then became a non-issue with clang/llvm anyway
- Compatibility with vmd(8) requires at least some nesting
  - Shadow VMCS (or emulation)
  - Exits for VMX instructions
  - Some sort of minimalist VM scheduler in the kernel

# New Ideas For vmm(4) (cont'd)

- Early proof-of-concept:
  - ~1600 line diff
  - .ktext protected
  - No nesting
- Similarly protecting userland code requires more work
  - UVM requires copy-on-read support
  - "Do kernel first, userland later"

# XO Kernel (“Underjack”) Demo

# Questions?

- Any questions?

# Thank You

Mike Larkin
mlarkin@openbsd.org
@mlarkin2012