Memory Forensics with Hyper-V Virtual Machines

By

Wyatt Roersma

Presented At

The Digital Forensic Research Conference

DFRWS 2014 USA Denver, CO (Aug 3rd - 6th)

Memory Forensics with Hyper-V Virtual Machines

By: @wyattroersma

Who Am I? – Wyatt Roersma

- NVINT – Senior Security Engineer
- Mad Security – DFIR challenge creator
- The Hacker Academy – Content Dev
- I research for fun
- Lots of Debug testing for Volatility

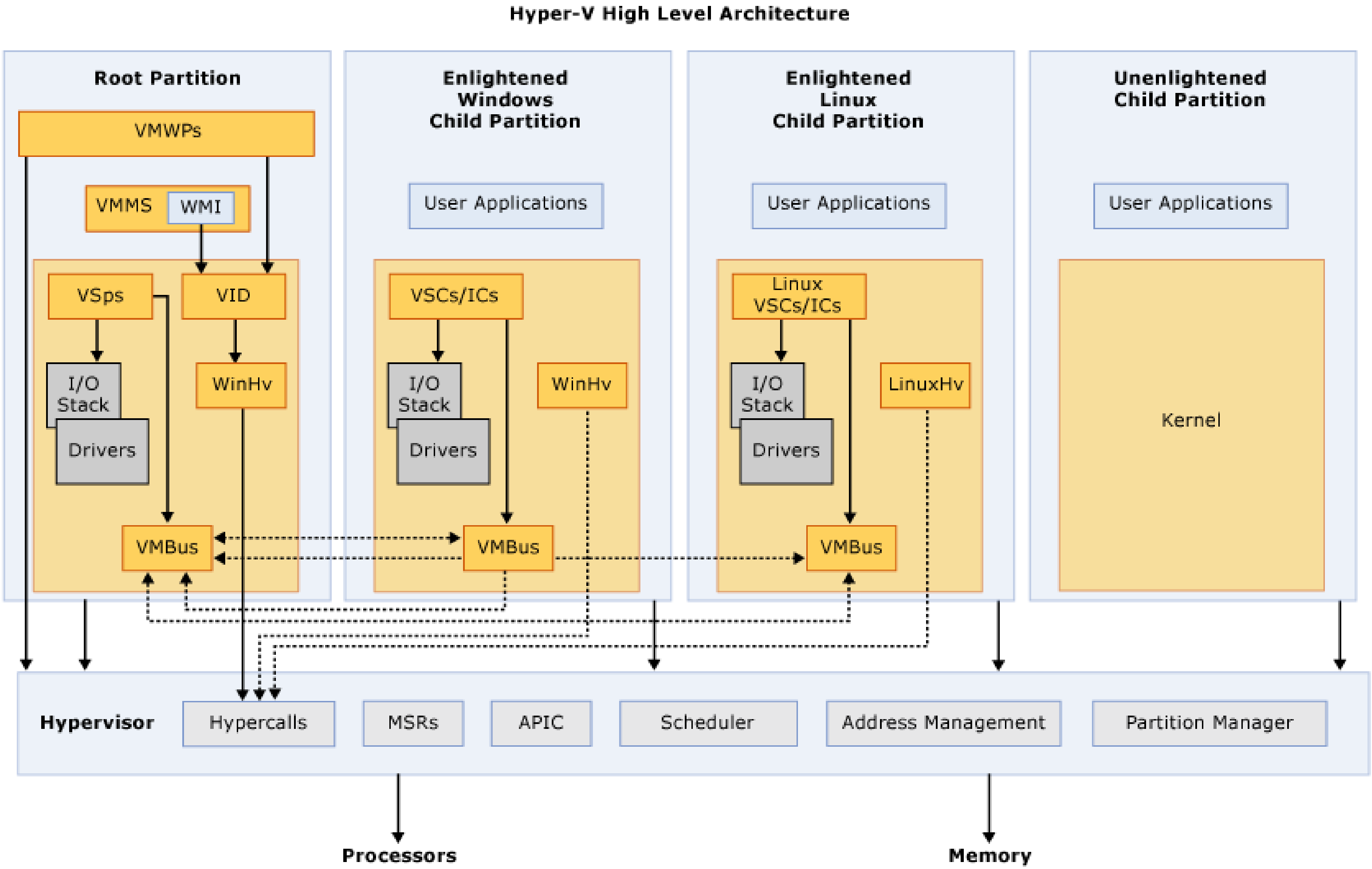
Motivation for this Research

- Better understanding of Hyper-V virtual machines
- Preform in depth host analysis
- I work with lots of Hyper-V servers
- Need for better documentation for Hyper-V forensic analysis
- Hyper-V accounts for %28 of the hypervisor market share [4]

Agenda

- Hyper-V Architecture Overview
- Overview of the Hyper-V Virtual Machine
- Acquisition tools 2008 – 2012 R2
- Hyper-V Host Memory Analysis
- Conclusions/Q&A

[1]


Hyper-V High Level Architecture
Root Partition
VMWPs
VMMS
WMI
VSps
VID
I/O Stack
Drivers
WinHv
VMBus
Enlightened Windows Child Partition
User Applications
VSCs/ICs
I/O Stack
Drivers
WinHv
VMBus
Enlightened Linux Child Partition
User Applications
Linux VSCs/ICs
I/O Stack
Drivers
LinuxHv
VMBus
Unenlightened Child Partition
User Applications
Kernel
Hypervisor
Hypercalls
MSRs
APIC
Scheduler
Address Management
Partition Manager
Processors
Memory

- APIC – Advanced Programmable Interrupt Controller – A device which allows priority levels to be assigned to its interrupt outputs.

- Child Partition – Partition that hosts a guest operating system - All access to physical memory and devices by a child partition is provided via the Virtual Machine Bus (VMBus) or the hypervisor.

- Hypercall – Interface for communication with the hypervisor - The hypercall interface accommodates access to the optimizations provided by the hypervisor.

- Hypervisor – A layer of software that sits between the hardware and one or more operating systems. Its primary job is to provide isolated execution environments called partitions. The hypervisor controls and arbitrates access to the underlying hardware.

- IC – Integration component – Component that allows child partitions to communication with other partitions and the hypervisor.

- I/O stack – Input/output stack

- MSR – Memory Service Routine
- Root Partition – Manages machine-level functions such as device drivers, power management, and device hot addition/removal. The root (or parent) partition is the only partition that has direct access to physical memory and devices.

- VID – Virtualization Infrastructure Driver – Provides partition management services, virtual processor management services, and memory management services for partitions.

- VMBus – Channel-based communication mechanism used for inter-partition communication and device enumeration on systems with multiple active virtualized partitions. The VMBus is installed with Hyper-V Integration Services.

- VMMS – Virtual Machine Management Service – Responsible for managing the state of all virtual machines in child partitions.

- VMWP – Virtual Machine Worker Process – A user mode component of the virtualization stack. The worker process provides virtual machine management services from the Windows Server 2008 instance in the parent partition to the guest operating systems in the child partitions. The Virtual Machine Management Service spawns a separate worker process for each running virtual machine.

- VSC – Virtualization Service Client – A synthetic device instance that resides in a child partition. VSCs utilize hardware resources that are provided by Virtualization Service Providers (VSPs) in the parent partition. They communicate with the corresponding VSPs in the parent partition over the VMBus to satisfy a child partitions device I/O requests.

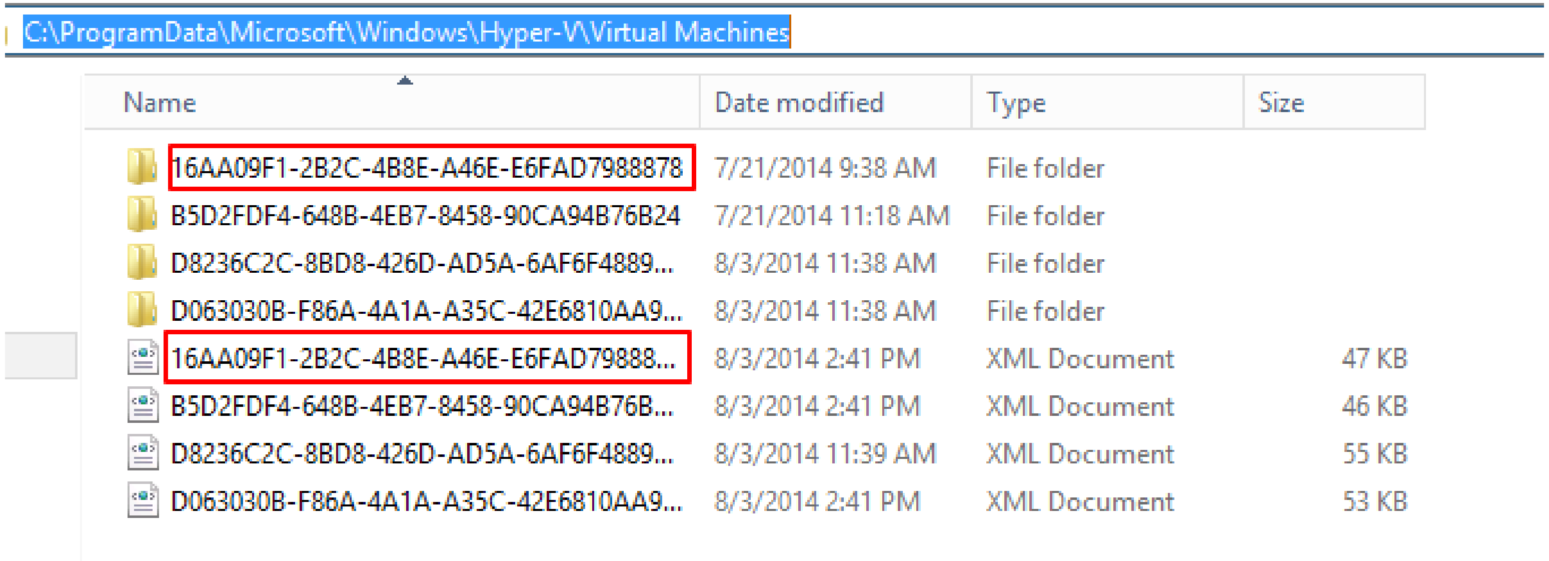
- VSP – Virtualization Service Provider – Resides in the root partition and provide synthetic device support to child partitions over the Virtual Machine Bus (VMBus).

- WinHv – Windows Hypervisor Interface Library - WinHv is essentially a bridge between a partitioned operating system’s drivers and the hypervisor which allows drivers to call the hypervisor using standard Windows calling conventions

- WMI – The Virtual Machine Management Service exposes a set of Windows Management Instrumentation (WMI)-based APIs for managing and controlling virtual machines.

Windows Hyper-V Virtual Machine Basics

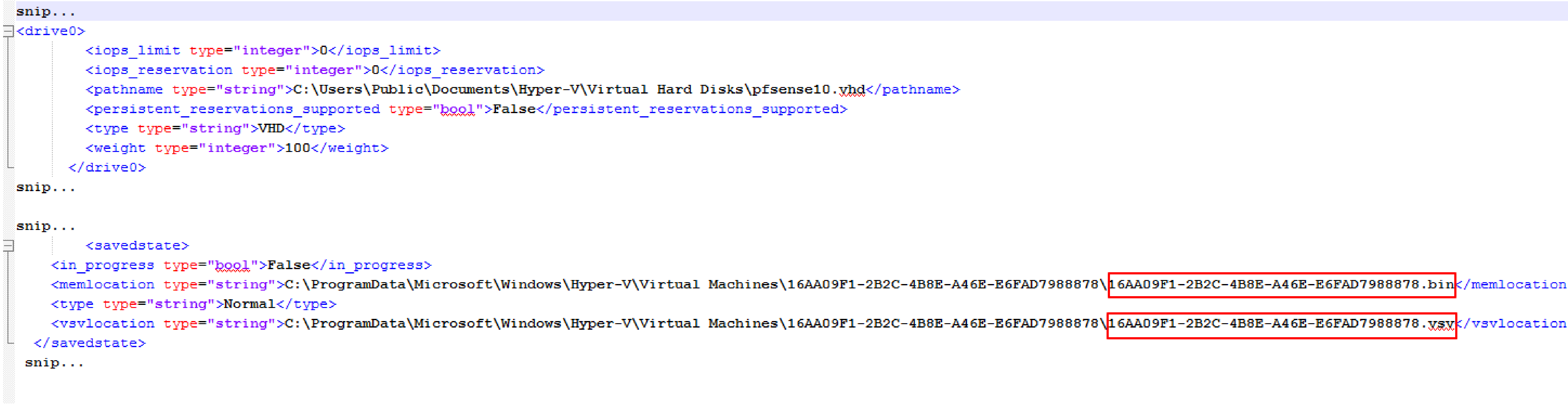
- .bin - Physical Memory Chunks
- .vsv - Memory Metadata
- .xml – Virtual Machine Config



C:\ProgramData\Microsoft\Windows\Hyper-V\Virtual Machines

Name	Date modified	Type	Size
16AA09F1-2B2C-4B8E-A46E-E6FAD7988878	7/21/2014 9:38 AM	File folder	
B5D2FDF4-648B-4EB7-8458-90CA94B76B24	7/21/2014 11:18 AM	File folder	
D8236C2C-8BD8-426D-AD5A-6AF6F4889...	8/3/2014 11:38 AM	File folder	
D063030B-F86A-4A1A-A35C-42E6810AA9...	8/3/2014 11:38 AM	File folder	
16AA09F1-2B2C-4B8E-A46E-E6FAD79888...	8/3/2014 2:41 PM	XML Document	47 KB
B5D2FDF4-648B-4EB7-8458-90CA94B76B...	8/3/2014 2:41 PM	XML Document	46 KB
D8236C2C-8BD8-426D-AD5A-6AF6F4889...	8/3/2014 11:39 AM	XML Document	55 KB
D063030B-F86A-4A1A-A35C-42E6810AA9...	8/3/2014 2:41 PM	XML Document	53 KB

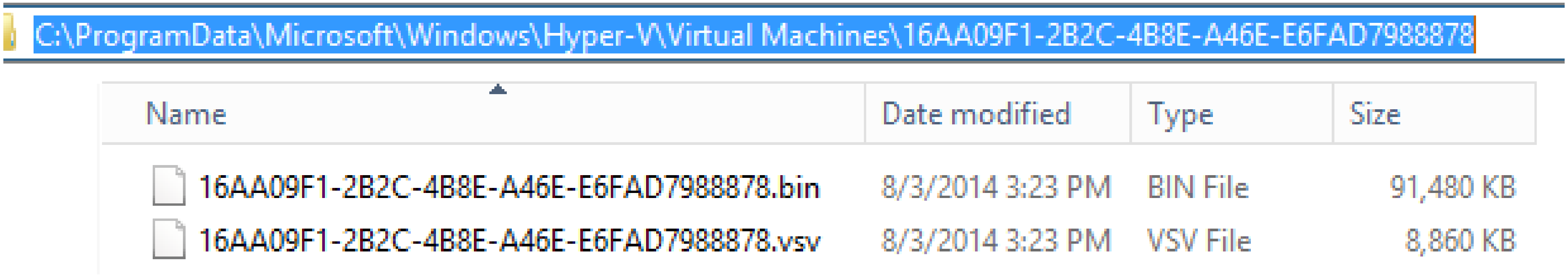
.xml – Hyper-V Virtual Machine Configuration



```
snip...
<drive0>
        <iops_limit type="integer">0</iops_limit>
        <iops_reservation type="integer">0</iops_reservation>
        <pathname type="string">C:\Users\Public\Documents\Hyper-V\Virtual Hard Disks\pfsense10.vhd</pathname>
        <persistent_reservations_supported type="bool">False</persistent_reservations_supported>
        <type type="string">VHD</type>
        <weight type="integer">100</weight>
      </drive0>
snip...

snip...
        <savedstate>
    <in_progress type="bool">False</in_progress>
    <memlocation type="string">C:\ProgramData\Microsoft\Windows\Hyper-V\Virtual Machines\16AA09F1-2B2C-4B8E-A46E-E6FAD7988878\16AA09F1-2B2C-4B8E-A46E-E6FAD7988878.bin</memlocation
    <type type="string">Normal</type>
    <vsvlocation type="string">C:\ProgramData\Microsoft\Windows\Hyper-V\Virtual Machines\16AA09F1-2B2C-4B8E-A46E-E6FAD7988878\16AA09F1-2B2C-4B8E-A46E-E6FAD7988878.vsv</vsvlocation
  </savedstate>
 snip...
```

.bin - Physical Memory Chunks
.vsv - Memory Metadata



C:\ProgramData\Microsoft\Windows\Hyper-V\Virtual Machines\16AA09F1-2B2C-4B8E-A46E-E6FAD7988878

Name	Date modified	Type	Size
16AA09F1-2B2C-4B8E-A46E-E6FAD7988878.bin	8/3/2014 3:23 PM	BIN File	91,480 KB
16AA09F1-2B2C-4B8E-A46E-E6FAD7988878.vsv	8/3/2014 3:23 PM	VSV File	8,860 KB

Acquisition Tools: Windows Server 2008 -2008R2

Vm2dmp.exe

- archive.msdn.microsoft.com/vm2dmp - no longer works
- Used to convert saved state files (.vsv .bin)into a crashdump

Requirements:

- Windows Debugging tools

Problems:

- 4GB RAM or more will cause the VM2DMP “ERROR: Failed to map guest block 4096 to any saved state block! ERROR: Element not found.”Doesn't support saved Linux virtual machines

Examples:

- Create a dump file using virtual machine state files:

```
> vm2dmp.exe -bin C:\dir\ VM-Instance-ID.bin -vsv C:\VM\ VM-Instance-ID.vsv -
dmp C:\dir\crashdump.dmp
```

- Create a dump file from virtual machine and snapshot name:

```
> vm2dmp.exe –vm VMName -dmp C:\VM\crashdump.dmp
```

```
> vm2dmp.exe –vm VMName –snap “vm VMName -snap-SP1” -dmp
C:\VM\crashdump.dmp
```

- Note: If you have a downloaded path of the debugging symbols then you can specify –sym and then the directory of the symbols path.

Acquisition Tools: Windows Server 2012 -2012R2

Livekd.exe

- technet.microsoft.com/en-us/sysinternals/bb897415.aspx
- Converts live, saved, and snap shot files to windows crash dump format

Cons:

- No Linux System Support
- Windows API

LiveDump.exe

- crashdmp.wordpress.com/2014/08/04/livedump-1-0-is-available/

Examples:

- If you want to list the virtual machines on the server just use the –hvl options and it will list GUIDs and names of running Hyper-V VM’s.

>livekd.exe –hvl

- If you want to create a full crash dump of a virtual machine running on the host system you would run

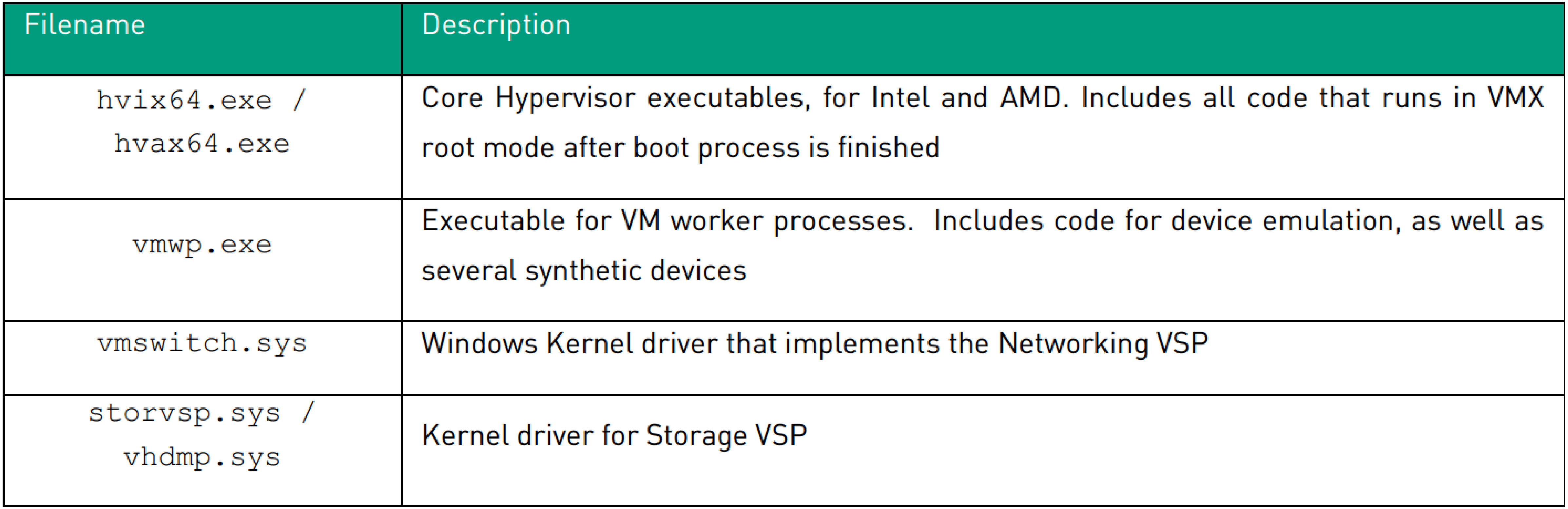
>livekd.exe –hv *(System name or GUID)* ***–p*** *(to pause the system to create a more consistent image)* ***–o*** *(output-file)*

>livekd.exe –hv AD –p –o AD.dmp

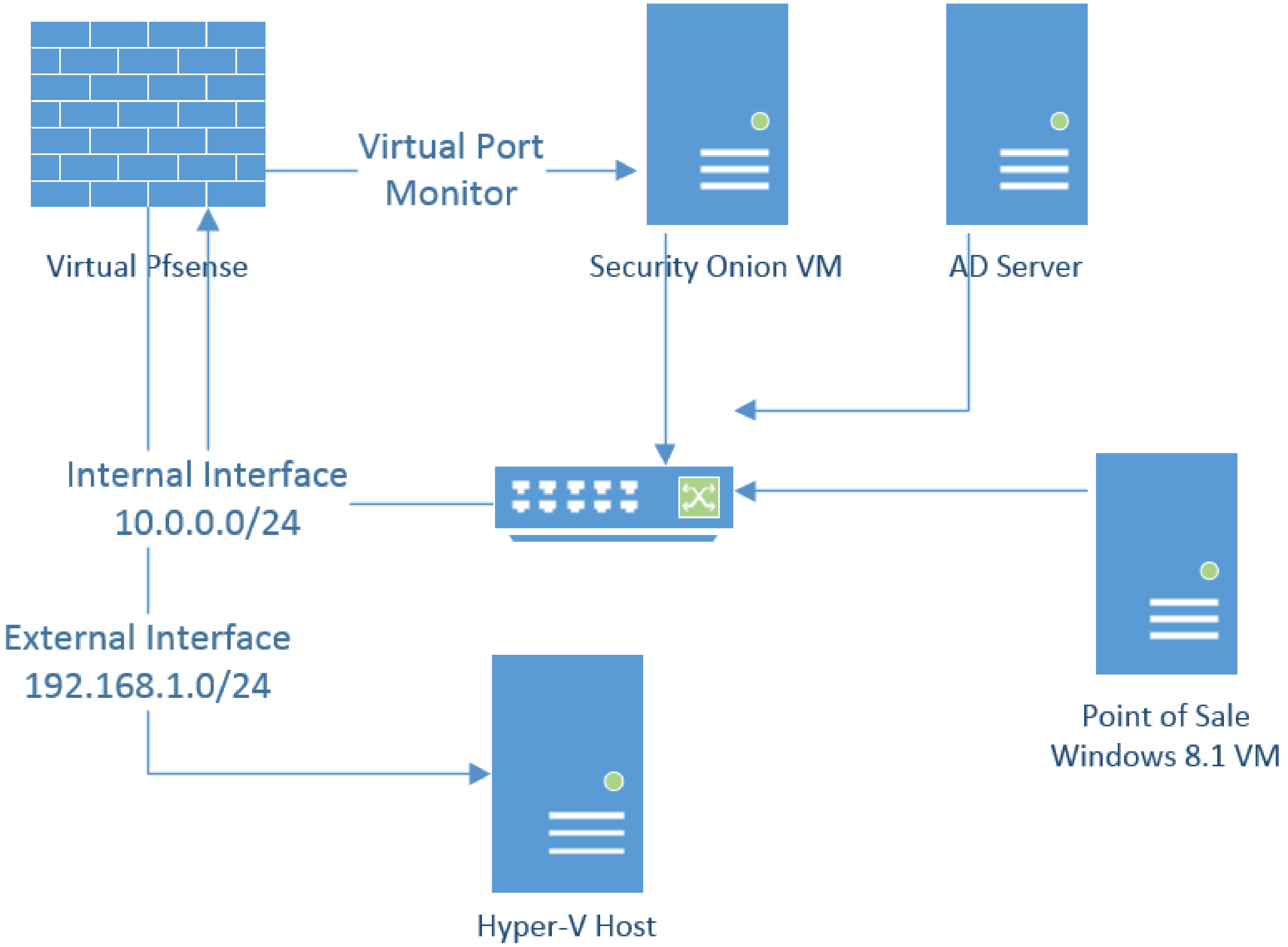
Hyper-V Host Memory Analysis

- Host Memory dumps >148GB
- Checking Host + VM’s for Artifacts
- Virtual Firewall Analysis
- All Traffic Live on the host
- Network Artifacts

[3]



Filename	Description
hvix64.exe / hvax64.exe	Core Hypervisor executables, for Intel and AMD. Includes all code that runs in VMX root mode after boot process is finished
vmwp.exe	Executable for VM worker processes. Includes code for device emulation, as well as several synthetic devices
vmswitch.sys	Windows Kernel driver that implements the Networking VSP
storvsp.sys / vhdmp.sys	Kernel driver for Storage VSP


Virtual Pfsense
Virtual Port Monitor
Security Onion VM
AD Server
Internal Interface
10.0.0.0/24
External Interface
192.168.1.0/24
Point of Sale
Windows 8.1 VM
Hyper-V Host

Virtual Machine Worker Process

```
python vol.py -f HV2012R2.raw --profile=Win2012R2x64 psscan |grep vmwp.exe
Volatility Foundation Volatility Framework 2.4 (Beta)
0x000000000cd7c100 vmwp.exe           3740   1504 0x00000000b73a5000 2014-08-03 21:42:06 UTC+0000
0x000000002223a080 vmwp.exe           2380   1504 0x0000000111eb3000 2014-08-03 20:41:03 UTC+0000
0x00000000fa742900 vmwp.exe           1952   1504 0x00000000bc856000 2014-08-03 21:42:07 UTC+0000
0x00000001cab2a900 vmwp.exe            808   1504 0x00000001f80fc000 2014-08-04 05:21:35 UTC+0000
```

Security Onion VM Artifacts

```
grep -i "sovm-eth1" ps-strings.txt |grep ET
```

```
19851536 [FREE MEMORY] 0 2 network-scan sovm-eth1 {2014-07-22 18:31:19} 3 3697 {ET TOR Known Tor
Relay/Router (Not Exit) Node Traffic group 226} 192.42.116.161 10.0.0.103 6 9001 49987 1 2522450 1913 1918
1918

19851824 [FREE MEMORY] 0 2 network-scan sovm-eth1 {2014-07-22 18:31:19} 3 3700 {ET TOR Known Tor
Relay/Router (Not Exit) Node Traffic group 18} 176.10.100.227 10.0.0.103 6 989 49983 1 2522034 1913 1921 1921

19852256 [FREE MEMORY] 0 2 network-scan sovm-eth1 {2014-07-22 18:31:19} 3 3701 {ET TOR Known Tor Exit
Node Traffic group 18} 176.10.100.227 10.0.0.103 6 989 49983 1 2520034 1913 1922 1922
```

Pfsense FW

```
grep -i "pfsense" ps-strings.txt
snip…
2525873247 [FREE MEMORY] Jul 21 16:03:23 pfSense dnsmasq[19767]: reading /etc/resolv.conf
2525873312 [FREE MEMORY] Jul 21 16:03:23 pfSense dnsmasq[19767]: using nameserver 208.67.220.220#53
2525873387 [FREE MEMORY] Jul 21 16:03:23 pfSense dnsmasq[19767]: using nameserver 208.67.222.222#53
snip…
```

```
2468293085 [FREE MEMORY] Aug  3 20:42:47 pfSense kernel: FreeBSD 8.3-RELEASE-p16 #0: Fri Jun 20 13:19:29 EDT 2014

2468293174 [FREE MEMORY] Aug  3 20:42:47 pfSense kernel:
root@pf2_1_1_amd64.pfsense.org:/usr/obj.amd64/usr/pfSensesrc/src/sys/pfSense_SMP.8 amd64

2468293295 [FREE MEMORY] Aug  3 20:42:47 pfSense kernel: Timecounter "i8254" frequency 1193182 Hz quality 0

2468293378 [FREE MEMORY] Aug  3 20:42:47 pfSense kernel: CPU: Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30GHz (1653.76-MHz
K8-class CPU)

2468293483 [FREE MEMORY] Aug  3 20:42:47 pfSense kernel: Origin = "GenuineIntel"  Id = 0x206d7  Family = 6  Model = 2d
Stepping = 7

Snip…

2468293932 [FREE MEMORY] Aug  3 20:42:47 pfSense kernel: AMD Features2=0x1<LAHF>

2468293988 [FREE MEMORY] Aug  3 20:42:47 pfSense kernel: TSC: P-state invariant

2468294043 [FREE MEMORY] Aug  3 20:42:47 pfSense kernel: real memory  = 1073741824 (1024 MB)

2468294111 [FREE MEMORY] Aug  3 20:42:47 pfSense kernel: avail memory = 1009954816 (963 MB)

2468294178 [FREE MEMORY] Aug  3 20:42:47 pfSense kernel: ACPI APIC Table: <VRTUAL MICROSFT>

2468294245 [FREE MEMORY] Aug  3 20:42:47 pfSense kernel: FreeBSD/SMP: Multiprocessor System Detected: 2 CPUs

2468294329 [FREE MEMORY] Aug  3 20:42:47 pfSense kernel: FreeBSD/SMP: 1 package(s) x 2 core(s)

2468294399 [FREE MEMORY] Aug  3 20:42:47 pfSense kernel: cpu0 (BSP): APIC ID:  0

2468294455 [FREE MEMORY] Aug  3 20:42:47 pfSense kernel: cpu1 (AP): APIC ID:  1

Snip…
```

Conclusion/Q&A

- Firewall, Switch, and Router Memory Analysis?
- Hyper-V recap
- Watch my blog for more information wyattroersma.com
- Useful Plugin? Vmwp?
- Questions?

References:

- [1] http://msdn.microsoft.com/en-us/library/cc768520(v=bts.10).aspx
- [2] http://csis.gmu.edu/VMSec/presentations/Hyper-V_Security_Baker.ppt
- [3] https://www.ernw.de/wp-content/uploads/ERNW_Newsletter_43_HyperV_en.pdf
- [4] http://www.trefis.com/stock/vmw/articles/221206/growing-competition-for-vmware-in-virtualization-market/2014-01-07
- https://media.blackhat.com/bh-dc-11/Suiche/BlackHat_DC_2011_Suiche_Cloud%20Pocket-Slides.pdf
- https://code.google.com/p/volatility/wiki/CommandReference22#imagecopy
- http://www.wyattroersma.com/?p=87
- http://www.wyattroersma.com/?p=77
- http://hypervking.info/hyper-v-tools/
- http://crashdmp.wordpress.com/2014/08/04/livedump-1-0-is-available/